Chapter 6

Federated Learning, IA Generativa e LLMs: Conceitos e Aplicações Práticas em Multimídia e Web

Helio N. Cunha Neto[1], Rafaela C. Brum[2], Paulo Mann[1], Raissa Barcellos[1]

[1]Instituto de Matemática e Estatística – LCC/IME/UERJ
Universidade do Estado do Rio de Janeiro (UERJ)

[2]Faculdade de Engenharia
Universidade do Estado do Rio de Janeiro (UERJ)

{helio.cunha, raissa.barcellos, paulo.mann}@ime.uerj.br,

rafaelabrum@eng.uerj.br

Abstract

The growth of the Internet of Things (IoT) has introduced challenges related to privacy and the efficient processing of data. Federated learning offers a solution by enabling model training directly on devices, preserving data privacy and minimizing communication with central servers. When combined with Large Language Models (LLMs), this approach facilitates the development of interactive and personalized applications, especially in multimedia and web domains. These technologies allow for the creation of systems such as content recommenders and virtual assistants, delivering safer and more efficient user experiences. Moreover, they promote sustainability by reducing reliance on centralized data centers. This work explores the impact of this integration in the context of multimedia and web applications.

Resumo

O crescimento da Internet das Coisas (IoT) trouxe desafios em termos de privacidade e eficiência no processamento de dados. O Aprendizado Federado oferece uma solução ao permitir o treinamento de modelos diretamente nos dispositivos, mantendo os dados privados e minimizando a comunicação com servidores centrais. Quando combinado com Modelos de Linguagem de Grande Escala (LLMs), essa abordagem possibilita o desenvolvimento de aplicações interativas e personalizadas, especialmente nas áreas de multimídia e web. Essas tecnologias permitem criar sistemas como recomendadores de conteúdo e assistentes virtuais, oferecendo experiências mais seguras e eficientes. Além disso, promovem a sustentabilidade ao reduzir a dependência de data centers centralizados. Este trabalho explora o impacto dessa integração no contexto de aplicações de multimídia e web.

6.1. Introdução

A proliferação da Internet das Coisas (*Internet of Things* - IoT) tem impulsionado um crescimento exponencial no número de dispositivos conectados à Internet, estimado atualmente em mais de 20 bilhões [45]. Esses dispositivos, equipados com sensores, processadores e sistemas de comunicação, são capazes de coletar e processar dados de alta qualidade em diversos cenários. Tais dados são fundamentais para o desenvolvimento de aplicações inteligentes, particularmente através do treinamento de modelos de Aprendizado de Máquina, como o Aprendizado Profundo (*Deep Learning*).

Na abordagem tradicional, os dados coletados por esses dispositivos são enviados para servidores ou *data centers* centralizados na nuvem para processamento e análise [44]. No entanto, essa centralização enfrenta desafios significativos, especialmente com políticas de proteção de dados cada vez mais rigorosas que limitam a transferência e armazenamento de dados privados. Além disso, o custo e a eficiência do envio de grandes volumes de dados para a nuvem podem ser comprometidos por conexões de rede lentas ou instáveis.

O Aprendizado Federado (FL, do inglês) surge como uma solução inovadora para o aprendizado colaborativo, focando na preservação da privacidade e na eficiência de comunicação [8]. Este paradigma permite: (i) o treinamento de modelos diretamente nos dispositivos móveis usando dados reais, o que é mais vantajoso do que usar dados artificiais; e (ii) a utilização de dados sensíveis de maneira que esses dados não sejam expostos ou centralizados, mantendo a privacidade dos usuários.

Integrar o Aprendizado Federado com Modelos de Linguagem de Grande Escala (*Large Language Models* - LLMs) e IA Generativa abre novas possibilidades, especialmente no contexto de multimídia e web. LLMs e IA Generativa têm o potencial de transformar a maneira como o conteúdo é criado, personalizado e distribuído, oferecendo experiências altamente interativas e adaptativas aos usuários [56]. Entretanto, o treinamento desses modelos em um ambiente centralizado não só exacerba os problemas de privacidade e segurança, mas também impõe altos custos computacionais e limita a inovação a grandes entidades com vastos recursos.

O Aprendizado Federado oferece uma abordagem descentralizada para o treinamento de LLMs e modelos de IA Generativa, utilizando a capacidade de processamento distribuída dos dispositivos de borda. Isso não apenas democratiza o acesso ao desenvolvimento desses modelos avançados, mas também reduz o impacto ambiental associado ao uso intensivo de grandes data centers [56]. No contexto de aplicações de multimídia e web, essa integração permite a criação de sistemas mais responsivos e personalizados, como recomendadores de conteúdo, assistentes virtuais, e sistemas de tradução em tempo real, tudo isso enquanto mantém a privacidade e a segurança dos dados dos usuários [48].

Esta combinação promete revolucionar a forma como interagimos com tecnologias de multimídia e web, proporcionando uma experiência mais segura, eficiente e personalizada, e ao mesmo tempo, promovendo uma abordagem mais sustentável e colaborativa no desenvolvimento de inteligência artificial avançada.

6.2. Fundamentos do Aprendizado de Máquina Distribuído

Nesta seção, são descritos os conceitos básicos de Aprendizado de Máquina, Aprendizado de Máquina profundo e uma introdução ao Aprendizado de Máquina colaborativo, além do porquê de as estratégias de aprendizado distribuído tradicionais não funcionarem em um ambiente de dispositivos móveis. O Aprendizado Federado pode ser considerado um Aprendizado de Máquina colaborativo que preserva a privacidade dos participantes [68]. Portanto, está intimamente relacionado ao Aprendizado de Máquina multipartidário. Com os recentes avanços no Aprendizado Profundo, técnicas de redes neurais que preservam a privacidade dos usuários também estão recebendo muito interesse em pesquisas [7]. Em Aprendizado Federado, são utilizados modelos baseados em algoritmos de Aprendizado Profundo, treinados através de variações da técnica de gradiente descendente estocástico.

6.2.1. Aprendizado de Máquina

De acordo com [57], o Aprendizado de Máquina se consolidou como uma das áreas mais relevantes da computação moderna. Pesquisas extensivas têm sido conduzidas para aprimorar a inteligência das máquinas, refletindo o aprendizado, um comportamento inerente ao ser humano, agora essencial também para as máquinas. Algoritmos tradicionais de Aprendizado de Máquina têm encontrado aplicação em diversas áreas, evidenciando a importância prática e teórica dessa tecnologia. O primeiro programa de inteligência artificial a incorporar capacidades de aprendizado, desenvolvido por Anthony Oettinger em 1951, foi denominado "programa de aprendizado por resposta" e "programa de compras" [57, 2]. Este último simulava o comportamento de uma criança em um shopping, representando um dos primeiros esforços significativos para criar máquinas com capacidade de aprendizado. Em 1955, Arthur Samuel avançou nessa direção ao adicionar aprendizado ao seu algoritmo de Damas, o que resultou no primeiro framework de Aprendizado de Máquina a receber reconhecimento público. O programa de Damas foi descrito por adversários humanos como "astuto, mas vencível" [57, 2].

Décadas de pesquisa na área de Aprendizado de Máquina resultaram no desenvolvimento de diversos algoritmos amplamente utilizados, como o classificador linear, regressão logística, naïve bayes, redes bayesianas, máquinas de vetores de suporte, árvores de decisão, florestas aleatórias, adaBoost, agregação por bootstrap, k-vizinhos mais próximos e redes neurais artificiais [57]. Atualmente, uma vasta gama de *frameworks* de Aprendizado de Máquina de código aberto está disponível no mercado, oferecendo aos desenvolvedores ferramentas para criar, implementar e manter sistemas avançados, além de gerar novos projetos e desenvolver sistemas inovadores com impacto significativo [57]. Entre os frameworks disponíveis, destacam-se *Apache Singa*, *Shogun*, *Apache Mahout*, *Apache Spark MLib*, *TensorFlow*, *Oryx 2*, *Accord.NET* e *Amazon Machine Learning*, os quais possibilitam a implementação de praticamente qualquer aplicação de Aprendizado de Máquina [57].

A área de Aprendizado de Máquina tem sido palco de avanços significativos, especialmente no desenvolvimento de algoritmos mais sofisticados [38]. Um dos principais marcos foi a evolução das redes neurais artificiais para arquiteturas mais profundas e complexas, conhecidas como Aprendizado Profundo, que aprimoram consideravelmente as capacidades de aprendizado das máquinas. Em determinadas aplicações, o Aprendizado Profundo já demonstrou desempenho que ultrapassa as capacidades humanas, marcando

um progresso notável na área [38].

6.2.2. Aprendizado Profundo

O Aprendizado Profundo revitalizou a pesquisa em redes neurais no início dos anos 2000 ao introduzir elementos que facilitaram o treinamento de redes mais profundas [6]. A emergência das GPUs e a disponibilidade de grandes conjuntos de dados foram fatores-chave para o avanço do Aprendizado Profundo. Além disso, o desenvolvimento de plataformas de software flexíveis e de código aberto, com diferenciação automática, como *Theano*, *Torch*, *Caffe*, *TensorFlow* e *PyTorch*, desempenhou um papel crucial [6] no avanço da área. Essas aplicações tornaram o treinamento de redes profundas complexas mais acessível e possibilitaram o reaproveitamento dos modelos mais recentes e de seus componentes [6].

O Aprendizado Profundo se destaca particularmente em domínios que envolvem grandes volumes de dados e dados de alta dimensionalidade. Isso explica por que redes neurais profundas frequentemente superam algoritmos rasos em aplicações que requerem processamento de texto, imagens, vídeos, fala e áudio [38]. Técnicas convencionais de Aprendizado de Máquina são limitadas em sua capacidade de processar dados naturais em sua forma bruta [43]. Durante décadas, a construção de um sistema de reconhecimento de padrões ou de Aprendizado de Máquina exigia engenharia cuidadosa e considerável *expertise* no domínio para projetar um extrator de características que transformasse os dados brutos — como os valores de *pixels* de uma imagem — em uma representação interna ou vetor de características adequado, a partir do qual o subsistema de aprendizado, muitas vezes um classificador, pudesse detectar ou classificar padrões na entrada [43].

Tendo em vista que o aprendizado de representações é um conjunto de métodos que permite que uma máquina seja alimentada com dados brutos e descubra automaticamente as representações necessárias para detecção ou classificação [43]; os algoritmos de Aprendizado Profundo são métodos de aprendizado com múltiplos níveis de representação, obtidos por meio da composição de módulos simples, mas não lineares, que transformam cada representação em um nível — começando com a entrada bruta — em uma representação em um nível superior, ligeiramente mais abstrato. Com a composição de transformações suficientes, funções muito complexas podem ser aprendidas [43].

Para tarefas de classificação, camadas superiores de representação amplificam aspectos da entrada que são importantes para a discriminação e suprimem variações irrelevantes. Uma imagem, por exemplo, é representada na forma de uma matriz de valores de *pixels*, e as características aprendidas na primeira camada de representação geralmente representam a presença ou ausência de bordas em orientações e locais específicos na imagem [43]. A segunda camada tipicamente detecta padrões ao identificar arranjos particulares de bordas, independentemente de pequenas variações nas posições das bordas. A terceira camada pode montar esses padrões em combinações maiores que correspondem a partes de objetos familiares, e camadas subsequentes detectariam objetos como combinações dessas partes [43]. O aspecto fundamental do Aprendizado Profundo é que essas camadas de características não são projetadas por engenheiros humanos: elas são aprendidas a partir dos dados usando um procedimento de aprendizado de propósito geral [43].

6.2.3. Comparação entre paradigmas

6.2.3.1. Aprendizado Distribuído

De acordo com Chen *et al* [16], nos últimos anos, a área de Aprendizado de Máquina testemunhou uma mudança de paradigma significativa, passando do chamado paradigma de "*big data*", no qual grandes volumes de dados são coletados e processados em uma nuvem central, para um paradigma de "*small data*", em que um conjunto de agentes ou dispositivos distribuídos deve processar seus dados localmente, na borda de um sistema sem fio ou de computação. Essa mudança de paradigma significa que as abordagens clássicas de Aprendizado de Máquina centralizado — que exigem grandes conjuntos de dados de treinamento para realizar tarefas de inferência de forma eficaz — já não são mais aplicáveis [16].

Em contraste, há uma necessidade crescente por novas soluções de aprendizado distribuído que possam colaborar para realizar inferências e aprendizado sem a necessidade de trocar conjuntos de dados locais. Tais soluções de aprendizado distribuído devem, essencialmente, estar cientes da natureza multiagente e distribuída das novas aplicações e sistemas baseados em *small data* [16]. O uso real dessa mudança de paradigma em direção ao aprendizado distribuído pode ser exemplificado no contexto da Internet das Coisas e da autonomia conectada — por exemplo, veículos ou drones conectados [16]. Em tais sistemas, cada dispositivo coleta seu próprio conjunto de dados individualizado, que muitas vezes é privado, e, coletivamente, os dispositivos devem ser capazes de treinar um modelo superando a escassez de dados locais. Nesses cenários, a troca de dados brutos é frequentemente indesejável — devido a questões de privacidade — ou, em alguns casos, até mesmo inviável — devido a restrições de comunicação e computação [16].

O aprendizado distribuído é visto como a base das redes inteligentes de próxima geração, onde agentes inteligentes, como dispositivos móveis, robôs e sensores, trocam informações entre si ou com um servidor de parâmetros, a fim de treinar modelos de Aprendizado de Máquina de forma colaborativa, sem a necessidade de enviar dados brutos para uma entidade central para processamento centralizado [12]. Um servidor de parâmetros pode ser entendido como um *framework* capaz de gerenciar e compartilhar os parâmetros de um modelo de Aprendizado de Máquina entre os agentes [12]. Ao utilizar a capacidade de computação e comunicação dos agentes individuais, o paradigma de aprendizado distribuído pode aliviar a carga nos processadores centrais e ajudar a preservar a privacidade dos dados dos usuários.

Apesar de suas aplicações promissoras, uma desvantagem do aprendizado distribuído é a necessidade de troca iterativa de informações por canais sem fio, o que pode resultar em uma sobrecarga de comunicação elevada, inviável em muitos sistemas práticos com recursos de rádio limitados, como energia e largura de banda [12]. Os esforços de pesquisa direcionados a resolver desvantagens no uso do aprendizado distribuído levaram ao surgimento de muitos *frameworks* importantes de aprendizado distribuído nos últimos anos [16]. Entre eles, destaca-se o popular Aprendizado Federado, que permite a um grupo de agentes executar colaborativamente uma tarefa de aprendizado comum, trocando apenas os parâmetros do modelo, em vez de seus dados brutos [16].

6.2.3.2. Aprendizado Federado

Na área da inteligência artificial, os dados são a base fundamental, e o treinamento de modelos não pode ser realizado sem eles. Contudo, os dados frequentemente existem na forma de "ilhas de dados", e a solução direta para esse problema é processar os dados de maneira centralizada. O método popular de processamento de dados envolve a coleta centralizada, processamento unificado, limpeza e modelagem [73]. Entretanto, na maioria dos casos, ocorrem vazamentos de dados durante a coleta e o processamento. Com o aprimoramento das regulamentações, a informação privada dos usuários está mais protegida, mas isso torna cada vez mais difícil coletar dados para treinar modelos. Como resolver legalmente o problema das ilhas de dados tem atraído muita atenção e reflexão na inteligência artificial. Para resolver o dilema das ilhas de dados, os métodos tradicionais de estatística estão se mostrando insuficientes frente às diversas regulamentações. O Aprendizado Federado direciona o foco da pesquisa para o problema das ilhas de dados [73].

O Aprendizado de Máquina tradicionalmente utiliza o método centralizado para treinar os modelos, o que exige que os dados de treinamento sejam concentrados em um mesmo servidor [73, 1]. Na realidade, devido às leis e regulamentações de proteção à privacidade dos dados, o método de treinamento centralizado, que pode levar ao vazamento de dados e à invasão da privacidade dos proprietários dos dados, está se tornando cada vez mais difícil de ser implementado [73]. No contexto de treinamento centralizado, se os usuários de dispositivos móveis quiserem treinar modelos de Aprendizado de Máquina com seus próprios dados, é evidente que a quantidade de dados disponível é insuficiente. Assim, antes do Aprendizado Federado, os usuários precisavam enviar os dados de seus próprios celulares para um servidor central, que treinava os modelos de Aprendizado de Máquina com os dados integrados dos usuários [73, 1]. Em comparação com o método de treinamento centralizado, o Aprendizado Federado, que pertence ao método de treinamento distribuído, permite que usuários individuais em diferentes localizações colaborem com outros usuários para treinar modelos de Aprendizado de Máquina, mantendo todos os dados pessoais que podem conter informações sensíveis no próprio dispositivo [73]. Com a ajuda do Aprendizado Federado, os usuários podem se beneficiar ao obter um modelo de Aprendizado de Máquina bem treinado sem precisar enviar seus dados pessoais sensíveis a um servidor central [73, 1].

Zhang *et al.* salientam que o Aprendizado Federado abre novas direções de pesquisa para a inteligência artificial. A tecnologia oferece um método de treinamento inovador para construir modelos personalizados sem violar a privacidade dos usuários. Com o advento dos *chipsets* de inteligência artificial, os recursos de computação dos dispositivos dos clientes se tornaram mais poderosos [12]. O treinamento de modelos de inteligência artificial também está gradualmente migrando do servidor central para os dispositivos terminais. O Aprendizado Federado proporciona um mecanismo de proteção de privacidade que pode utilizar de maneira eficaz os recursos de computação dos dispositivos terminais para treinar modelos, evitando que informações privadas sejam vazadas durante a transmissão de dados. Considerando que o número de dispositivos móveis e de dispositivos em outras áreas é incontável, há uma grande quantidade de recursos de conjuntos de dados valiosos, e o Aprendizado Federado pode fazer pleno uso desses recursos [12].

É importante enfatizar que o conceito de Aprendizado Federado é distinto do con-

ceito de computação distribuída [66]. A diferença mais significativa reside nas suposições feitas sobre os conjuntos de dados. No aprendizado distribuído, assume-se que as partes do conjunto de dados são independentes e identicamente distribuídas (i.i.d.), o que significa que elas são geradas a partir do mesmo processo estocástico sem memória. No entanto, nenhuma suposição desse tipo é feita no contexto do Aprendizado Federado [66]. Em vez disso, os conjuntos de dados podem ser heterogêneos. Por exemplo, um modelo de Aprendizado de Máquina projetado para reconhecer criminosos em um bairro pode depender de imagens de câmeras coletadas por um grupo diverso de usuários. É evidente que não se pode esperar razoavelmente que as imagens coletadas entre dois usuários sejam i.i.d [66].

Segundo Xia *et al.* [66], o ambiente especialmente adequado para a aplicação de *frameworks* de Aprendizado Federado é o ambiente de computação em borda. Isso ocorre porque o Aprendizado Federado pode aproveitar o poder computacional dos servidores de borda e os dados coletados por dispositivos de borda amplamente distribuídos [66]. Em um sistema de Aprendizado Federado implementado em um contexto de computação em borda, é possível realizar o treinamento de modelos de forma colaborativa e eficiente, sem a necessidade de centralizar os dados em um único servidor. Esse ambiente distribuído e próximo à fonte dos dados permite uma integração mais eficaz das capacidades computacionais locais e a utilização dos dados coletados em diferentes locais, proporcionando vantagens significativas em termos de desempenho e privacidade [66].

6.2.3.3. Computação em Borda

Com o rápido desenvolvimento da Internet das Coisas, o número de dispositivos inteligentes conectados à rede tem aumentado, resultando em grandes volumes de dados [11]. Isso tem causado problemas como sobrecarga de largura de banda, lentidão na resposta, baixa segurança e privacidade inadequada nos modelos tradicionais de computação em nuvem. Dada a crescente diversidade das necessidades de processamento de dados na sociedade inteligente atual, a computação em borda surgiu como uma solução [11]. Esse paradigma de computação realiza cálculos na borda da rede, destacando-se por sua proximidade com o usuário e a origem dos dados, e é mais adequado para armazenamento e processamento de dados localizados e em pequena escala [11].

Há várias razões principais que estão levando diferentes profissionais a fazer a transição de modelos baseados em nuvem tradicionais para plataformas de computação em borda — dois fatores principais são a baixa latência e a alta largura de banda [66]. No entanto, a computação em borda também oferece vantagens significativas em termos de segurança. Por exemplo, ao enviar dados para um dispositivo de borda, os possíveis atacantes têm menos tempo para lançar um ataque em comparação com a nuvem, devido à menor latência [66, 11]. Além disso, ataques como DDoS, que normalmente seriam debilitantes em um ambiente baseado em nuvem, tornam-se quase inofensivos em um ambiente de computação em borda, pois os dispositivos de borda afetados podem ser removidos da rede sem comprometer a funcionalidade geral da rede [66, 11]. Isso também implica que as redes de borda são muito mais confiáveis, pois não possuem um único ponto de falha. Além disso, as redes de borda são mais facilmente escaláveis devido ao menor porte dos dispositivos [66, 11]. De fato, uma estratégia de escalonamento hori-

zontal oferece às empresas uma maneira atraente de obter bom desempenho com baixo custo. Adicionalmente, alguns desses dispositivos ou centros de dados de borda podem nem precisar ser construídos do zero por uma única empresa. Diferentes partes interessadas podem colaborar para compartilhar os recursos dos dispositivos IoT já existentes na rede de borda [66, 11].

A computação em borda fornece serviços de inteligência artificial para dispositivos terminais em rápido crescimento e dados, tornando os serviços mais estáveis [11]. Por estar próxima à fonte dos dados, como terminais inteligentes, esta tecnologia armazena e processa dados na borda da rede, oferecendo proximidade e conscientização sobre a localização e proporcionando serviços próximos ao usuário [11]. Em termos de processamento de dados, é mais rápida, em tempo real e segura. Além disso, pode resolver o problema do consumo excessivo de energia na computação em nuvem, reduzir custos e aliviar a pressão sobre a largura de banda da rede. A computação em borda é aplicada em diversas áreas, como produção, energia, casas inteligentes e transporte [11].

6.2.3.4. Banco de Dados Federado

Os principais requisitos dos sistemas modernos de bancos de dados federados envolvem uma série de características interligadas [3]. Primeiramente, é essencial que o sistema ofereça encapsulamento de localização e fontes de dados, fornecendo uma interface intuitiva que libere os programadores da necessidade de aprender várias linguagens de consulta e mecanismos de armazenamento. A implantação deve ser adequada às práticas de nuvem, garantindo que a complexidade e os custos associados à instalação, administração e manutenção não sejam excessivos, e que haja mecanismos eficazes para prever e depurar o desempenho, bem como controlar custos [3].

Em termos de linguagem de consulta, o sistema deve ser capaz de operar em armazenamentos de dados heterogêneos, suportar cadeias arbitrárias de consultas — onde os resultados de uma consulta em um banco servem como entrada para consultas em outros — ser independente de esquemas — permitindo a integração de bancos de dados com ou sem esquema — e permitir a transformação de metadados [3]. Além disso, as ferramentas de consulta devem proporcionar interfaces fáceis de usar, linguagens e APIs que se ajustem desde consultas simples *Select-Project-Join* até consultas avançadas específicas de aplicativos [3]. O processamento deve ser eficiente, suportando consultas escaláveis e otimização, combinando técnicas modernas e inovadoras como *joins* vinculados e *semi-joins*, além de conceitos de processamento paralelo de consultas [3].

Em relação ao suporte à decisão em tempo real, o sistema deve permitir o processamento de fluxos de dados, conectando dados históricos e em tempo real. As ferramentas de visualização devem apoiar modelos de dados diversos e novos mecanismos de interação com o usuário, facilitando a exploração de dados com perguntas como “mostre-me algo interessante” [3]. Finalmente, a capacidade de distribuir dados entre *backends* é crucial, permitindo a movimentação de dados e resultados intermediários entre diferentes armazenamentos para otimizar a resposta a consultas e garantir alta performance, utilizando sistemas de monitoramento que ajustam a alocação de dados conforme necessário para melhorar a eficiência das consultas [3].

Em termos de estrutura e armazenamento de dados, o Aprendizado Federado compartilha muitas semelhanças com os bancos de dados federados [41]. No entanto, ao interagir entre si, o sistema de bancos de dados federados não exige proteção de privacidade, e sua equipe de gerenciamento tem acesso total a todas as unidades de banco de dados [41]. Assim, o sistema de banco de dados federado se concentra em operações simples de dados, como adição, busca, exclusão e fusão, enquanto o aprendizado federado visa construir um modelo utilizado normalmente e representar melhor os princípios e legislações obtidos a partir dos dados [41].

6.3. Estratégias para o Aprendizado Federado

O conceito de Aprendizado Federado foi introduzido em [8], devido a preocupações de privacidade com dados de usuários. O Aprendizado Federado permite que os usuários treinem colaborativamente um modelo compartilhado, mantendo os dados pessoais em seus dispositivos. Em geral, existem duas entidades principais no sistema de Aprendizado Federado, que são os proprietários dos dados e o proprietário do modelo. Normalmente, o proprietário do modelo do Aprendizado Federado, chamado de servidor, não tem nenhuma permissão de acesso aos dados dos diferentes proprietários de dados, chamados de usuários ou clientes na arquitetura do Aprendizado Federado. Podemos observar três componentes principais da arquitetura de Aprendizado Federado:

- Dispositivos locais (nós): são os clientes ou dispositivos finais que detêm os dados e realizam o treinamento local. Esses dispositivos podem ser celulares, sensores, ou qualquer outra máquina conectada com capacidade computacional.
- Servidor central: responsável por coordenar o processo de Aprendizado Federado. Ele inicializa os parâmetros do modelo, seleciona os clientes participantes de cada rodada, coleta as atualizações dos modelos locais, realiza a agregação e distribui o modelo global atualizado aos dispositivos locais.
- Canal de Comunicação: é o meio pelo qual os dispositivos locais se comunicam com o servidor. Pode ser uma rede sem fio, como Wi-Fi ou 5G, que deve ser segura e eficiente para minimizar atrasos e garantir a privacidade dos dados.

Além desses componentes, a arquitetura do Aprendizado Federado deve considerar questões de segurança e privacidade, como criptografia dos parâmetros trocados e técnicas de preservação de privacidade, como a adição de ruído.

Durante o treinamento no Aprendizado Federado, cada cliente participante recebe um modelo de Aprendizado de Máquina do servidor com os parâmetros do modelo iniciados com os mesmos valores, geralmente aleatórios. Esses modelos são treinados individualmente no dispositivo local de cada participante e apenas os parâmetros desses modelos são enviados ao servidor, que realiza uma agregação global com base nesses parâmetros. Em termos matemáticos, seja $D_k = \{(x_i, y_i)\}_{i=1}^{n_k}$ o conjunto de dados local do cliente k, onde x_i representa as características de entrada e y_i, a saída correspondente. O objetivo do Aprendizado Federado é minimizar a função de perda global $F(w)$ sobre todos os dados distribuídos, como mostra a Equação 1, onde $F_k(w)$ é a função de perda local no cliente k, w são os parâmetros do modelo a serem aprendidos, K é o número de clientes.

$$F(w) = \frac{1}{K} \sum_{k=1}^{K} F_k(w) \tag{1}$$

A Equação 2 descreve a função de perda local $F_k(w)$, onde w são os parâmetros do modelo e $l(w; x_i, y_i)$ é a perda em um único ponto de dados.

$$F_k(w) = \frac{1}{n_k} \sum_{i=1}^{n_k} l(w; x_i, y_i) \tag{2}$$

Este modelo de aprendizado foi descrito usando o *GBoard*, o teclado do Google, como caso de uso[1], sendo os clientes milhares de celulares Android. Como os celulares tem conexão instável e podem descarregar a qualquer momento, o algoritmo geral de Aprendizado Federado criado em 2017 [8] é baseado em rodadas de aprendizagem e escolhe somente uma porcentagem dos clientes conectados para participarem a cada rodada. Uma rodada é representada por 4 etapas, descritas a seguir:

1. O servidor escolhe uma fração dos clientes conectados e manda os pesos atuais do modelo para estes clientes;

2. Cada cliente treina o modelo por um número pré-determinado de épocas no seu conjunto de dados local e devolve para o servidor os pesos atualizados do modelo;

3. O servidor recebe as atualizações dos clientes participantes, as agrega usando algum algoritmo de agregação do Aprendizado Federado e envia os pesos finais de volta para os clientes participantes;

4. Por fim, cada cliente que recebe estes pesos agregados atualiza o modelo local, testa com seu conjunto de dados de teste e envia as métricas de avaliação (perda, acurácia, escore F1, etc) para o servidor agregá-las e começar a próxima rodada de comunicação.

A agregação dos pesos feita pelo servidor a cada rodada pode ser feita de diversas maneiras. O primeiro algoritmo descrito para tal finalidade é o *Federated Averaging* (FedAVG) [8]. É fundamental o seu entendimento, pois este algoritmo é a base das novas propostas. O FedAVG usa o gradiente descendente estocástico (SGD) nos clientes para treinar o modelo e utiliza uma média ponderada dos parâmetros locais atualizados w_k para calcular os parâmetros globais agregados w_{global}, conforme Equação 3, onde n é o número total de amostras em todos os nós.

$$w_{global} = \frac{1}{K} \sum_{k=1}^{K} \frac{n_k}{n} w_k \tag{3}$$

[1] https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

6.3.1. Tipos de Aprendizado Federado

Falando um pouco dos tipos de Aprendizado Federado, podemos classificar o Aprendizado Federado baseado no tipo dos clientes ou baseado na distribuição dos dados pelos clientes. Quando falamos da classificação baseada no tipo de clientes, podemos ter o Aprendizado Federado entre dispositivos (*Cross-Device*) ou entre silos de dados (*Cross-Silo*). Se os clientes são dispositivos de baixa potência, como telefones móveis [8] ou dispositivos de borda [55], chamamos isso de Aprendizado Federado entre dispositivos (*Cross-Device Federated Learning*). Existem alguns desafios relacionados ao consumo de energia e à conexão nesse tipo de Aprendizado Federado. Por outro lado, se os clientes são empresas (por exemplo, hospitais [53]) com conjuntos de dados semelhantes que desejam criar um modelo central, trata-se de um Aprendizado Federado entre silos (*Cross-Silo Federated Learning*).

Independente do tipo de cliente envolvido no Aprendizado Federado, podemos classificá-lo considerando os atributos dos conjuntos de dados distribuídos e as amostras. Se todos os clientes têm o mesmo conjunto de atributos, mas amostras diferentes, trata-se de um Aprendizado Federado Horizontal [68]. Nesse cenário, todos os clientes utilizam os mesmos atributos como entrada para o modelo. A colaboração entre os clientes é direta, com um servidor centralizado para agregar os pesos do treinamento. Por exemplo, dois bancos podem colaborar para criar um modelo centralizado de detecção de fraudes em transações com cartão de crédito. Ambos possuem o mesmo conjunto de atributos, como por exemplo a renda atual, poupança, ocupação, limite do cartão de crédito e o preço médio de compra.

Por outro lado, se os clientes não têm o mesmo conjunto de atributos, mas possuem as mesmas amostras, eles podem colaborar para criar um modelo mais sofisticado. Por exemplo, um aplicativo de rastreamento de atividade física pode colaborar com um hospital para criar um modelo que entenda a relação entre exercícios e saúde. Esse tipo de Aprendizado Federado é chamado de Aprendizado Federado Vertical [68]. O treinamento é mais desafiador, pois as partes precisam enviar e receber resultados intermediários de treinamento para inseri-los no seu treinamento local.

Existe também o Aprendizado Federado de transferência [40], quando os clientes compartilham parte do conjunto de atributos e/ou parte das amostras. Esse tipo de Aprendizado Federado é comum em casos onde diferentes organizações ou dispositivos possuem dados que podem ser úteis uns para os outros, mas não são completamente compatíveis. Um exemplo deste tipo de Aprendizado Federado é o treinamento de modelo para detectar doença de Parkinson através dos dados obtidos por *smartwatches* [18].

6.4. Aprendizado Federado, LLMs e IA Generativa

O Aprendizado Federado, quando combinado com Modelos de Linguagem de Grande Escala (LLMs) e IA Generativa, oferece uma abordagem inovadora para o desenvolvimento de inteligência artificial avançada. Esta seção explora os componentes principais dessa integração e suas implicações para a privacidade, eficiência e personalização em dispositivos de borda. Primeiro será discutido sobre LLMs

6.4.1. Modelos de Linguagem de Grande Escala

Os LLMs (Large Language Models) são redes neurais profundas treinadas em vastos conjuntos de dados textuais, utilizando arquiteturas como os *Transformers*. Essas arquiteturas permitem que os modelos aprendam padrões linguísticos complexos, possibilitando a realização de tarefas como tradução, sumarização de texto e resposta a perguntas com alto desempenho [71].

6.4.1.1. Arquitetura *Transformers*

Os *Transformers* revolucionaram o campo de NLP. Introduzidos por Vaswani *et al.* [61], os *Transformers* utilizam mecanismos de atenção, permitindo que o modelo foque em diferentes partes de uma sentença ao processar informações. Esse mecanismo foi fundamental para superar as limitações de arquiteturas anteriores, como RNNs e LSTMs, que dependem de processamento sequencial, o que torna o treinamento mais lento e menos eficiente[51].

Cada camada de um *Transformer* é composta por um mecanismo de atenção e redes neurais totalmente conectadas. Isso permite uma representação mais rica dos dados textuais, já que o modelo pode capturar relações entre palavras, independentemente de sua distância no texto. A Figura 6.1 ilustra uma camada individual do Transformer, composta por dois principais componentes: o mecanismo de atenção e a rede totalmente conectada. O fluxo de informação começa com vetores de entrada, que é processado através do mecanismo de atenção própria (*self-attention*). Esse mecanismo avalia a relevância de cada palavra em relação às outras dentro de uma frase, gerando uma representação contextualizada. A saída do mecanismo de atenção é então somado à entrada original através de uma conexão residual, que ajuda a estabilizar o processo de aprendizado em redes profundas. Após essa soma, o resultado é passado para a rede totalmente conectada. Novamente, o resultado final da rede totalmente conectada é somado à entrada original dessa etapa por meio de outra conexão residual. Cada uma dessas partes é precedida por uma normalização de camada (*layer normalization*), que ajusta os valores para manter o equilíbrio do processo de aprendizado.

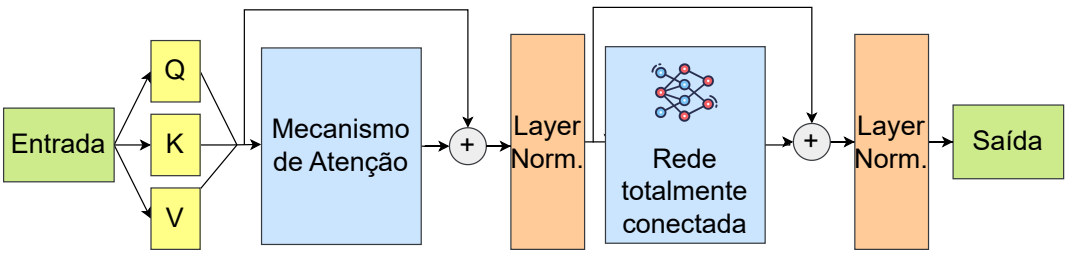

Figure 6.1. Estrutura de uma camada do *Transformer*, destacando o fluxo de informações entre o mecanismo de atenção e a rede totalmente conectada. As conexões residuais ao redor de cada componente somam a entrada original à saída de cada bloco, contribuindo para a estabilidade do aprendizado em redes profundas.

6.4.1.2. Mecanismo de Atenção e *Self-Attention*

O principal avanço dos *Transformers* está no mecanismo de atenção, especificamente a atenção própria (*self-attention*) [61]. Esse mecanismo avalia a relevância de cada palavra em uma sentença em relação a todas as outras palavras da mesma sentença. Isso permite ao modelo capturar dependências de longo alcance e compreender contextos complexos com mais eficiência.

Essa capacidade é crucial para a geração de texto altamente coerente e contextualizado, pois o modelo entende como as palavras se relacionam, mesmo quando estão distantes umas das outras no texto. Esse é um dos motivos pelos quais os *Transformers* tiveram tanto sucesso em várias aplicações de NLP, como na tradução automática, resposta a perguntas e sumarização de textos.

A principal ideia do s*elf-attention* é gerar três vetores para cada palavra da sequência de entrada: *Query* (Q), *Key* (K) e *Value* (V). Esses vetores são utilizados para calcular uma pontuação de atenção que determina o quanto uma palavra deve prestar atenção nas outras palavras da sentença. Dada uma sequência de palavras de entrada, o modelo primeiro transforma essas palavras em vetores de *embedding*. A partir dos vetores de *embedding*, os vetores Q, K e V são obtidos por multiplicação com matrizes de pesos aprendíveis:

$$Q = XW_Q, \qquad K = XW_K, \qquad V = XW_V \tag{4}$$

Onde, X é o vetor de *embeddings* das palavras de entrada, W_Q, W_K e W_V são as matrizes de pesos que serão aprendidas pelo modelo para gerar os vetores de Q, K e V.

Com os vetores de *Query* (Q) e *Key* (K), calcula-se uma pontuação de atenção para cada par de palavras na sequência. Isso é feito com o produto escalar entre o vetor Q de uma palavra e o vetor K das outras palavras. Quanto maior o valor do produto escalar, mais "atento" o modelo estará àquela palavra. A equação para calcular essa pontuação é:

$$\text{Attention}(Q,K) = \frac{QK^T}{\sqrt{d_k}} \tag{5}$$

Onde K^T é o vetor K transposto e d_k é a dimensionalidade dos vetores de Query e Key. O termo $\sqrt{d_k}$ é usado para normalizar os valores e evitar que os produtos escalares cresçam excessivamente com vetores de alta dimensionalidade.

Depois de calcular as pontuações de atenção, aplica-se a função Softmax para normalizar os valores, de modo que todas as pontuações somem 1. Isso permite interpretar as pontuações como "peso" que indicam o quanto uma palavra deve focar nas outras:

$$\alpha_{ij} = \text{Softmax}\left(\frac{Q_i K_j^T}{\sqrt{d_k}}\right) \tag{6}$$

Onde α_{ij} é o peso da palavra j em relação a palavra i.

Agora, com os pesos de atenção α_{ij}, utiliza-se esses valores para ponderar os vetores V correspondentes a cada palavra. O vetor de saída para cada palavra é então a soma ponderada dos valores:

$$\text{Output} = \sum_j \alpha_{ij} V_j \tag{7}$$

Essa operação permite que cada palavra seja representada por uma combinação linear de todas as outras palavras da sequência, levando em consideração sua importância relativa. Essa saída contém informações sobre o contexto global da sentença.

Para capturar diferentes tipos de relações entre as palavras, o mecanismo de atenção geralmente é implementado como *multi-head attention*, ou seja, o processo descrito acima é executado várias vezes em paralelo, com diferentes conjuntos de pesos W_Q, W_K e W_V. Cada "cabeça" de atenção processa a informação de maneira ligeiramente diferente, e os resultados são concatenados no final:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1, \text{head}_2, ..., \text{head}_h) W_O \tag{8}$$

Onde, $\text{head}_i = \text{Attention}(Q,K,V)$ para cada cabeça i e W_O é uma matriz de pesos para combinar as saídas de todas as cabeças de atenção.

O mecanismo de *self-attention* permite que o modelo entenda não só as dependências locais entre palavras vizinhas, mas também relações distantes em uma frase. O *self-attention* facilita o processamento paralelo, já que não depende da ordem sequencial dos dados, tornando os modelos como os *Transformers* muito mais eficientes do que arquiteturas baseadas em RNNs.

Conforme os LLMs aumentaram de tamanho e passaram a ser treinados em conjuntos de dados massivos, surgiram habilidades emergentes que não eram previstas pelas escalas tradicionais de modelos menores. Essas habilidades emergem de forma súbita quando o modelo ultrapassa um determinado limiar de tamanho e complexidade. Um exemplo marcante é a aprendizagem contextual (*In-Context Learning* - ICL), onde os LLMs conseguem resolver tarefas complexas sem treinamento explícito para essas funções. Em vez de exigir ajustes no modelo, eles aprendem com exemplos fornecidos no próprio momento da interação, aplicando padrões já adquiridos durante o treinamento [71]. Isso representa uma capacidade inédita de adaptação a novos contextos com rapidez e eficiência.

Outro avanço fundamental foi o desenvolvimento de técnicas que permitem o alinhamento dos LLMs com as preferências e valores humanos, como o aprendizado por reforço com feedback humano (*Reinforcement Learning from Human Feedback* - RLHF). Por meio dessa abordagem, o modelo é treinado para ajustar suas respostas com base em avaliações humanas, garantindo que suas interações sejam mais coerentes, seguras e alinhadas com as expectativas dos usuários [65]. O RLHF aprimora a capacidade dos LLMs de gerar respostas úteis e eticamente adequadas, mitigando riscos como vieses indesejados ou a geração de informações enganosas, tornando-os mais eficazes e confiáveis em diversas aplicações práticas.

6.4.1.3. Desafios dos LLMs

Apesar de seu impacto transformador, os LLMs enfrentam desafios significativos, especialmente em relação à escalabilidade e ao uso ético. Modelos como o GPT-4, que contam com bilhões de parâmetros, demandam recursos computacionais massivos tanto para treinamento quanto para inferência [71]. Isso resulta em custos elevados de processamento e energia, tornando seu uso inviável para muitos desenvolvedores e organizações com recursos limitados. Além disso, a arquitetura complexa e o grande volume de dados utilizados no treinamento dos LLMs podem amplificar vieses presentes nesses dados, levando a respostas tendenciosas ou inapropriadas em determinados contextos [65]. Esses vieses podem surgir de representações desbalanceadas de grupos sociais, culturas ou gêneros, criando riscos éticos e práticos.

Outro aspecto crítico é a limitação dos LLMs no que diz respeito ao raciocínio lógico e à resolução de problemas complexos. Embora os modelos sejam altamente eficazes em gerar texto fluido e coerente, sua capacidade de raciocínio lógico é inferior à dos seres humanos, especialmente em cenários que envolvem várias etapas de raciocínio ou conclusões abstratas [65]. Isso ocorre porque os LLMs não possuem compreensão semântica profunda nem intencionalidade, o que pode resultar em respostas enganosamente convincentes, mas sem fundamento real.

Além das questões de desempenho, há preocupações éticas cruciais associadas aos LLMs. O risco de disseminação de desinformação é significativo, pois os modelos podem gerar conteúdo plausível, mas incorreto, exacerbando o problema da confiabilidade das informações online. Além disso, a privacidade dos dados utilizados para treinar esses modelos levanta questionamentos importantes, uma vez que o uso de grandes quantidades de dados pessoais pode implicar na violação de normas de privacidade e proteção de dados [9]. Esses desafios éticos destacam a necessidade de regulamentação e supervisão no desenvolvimento e na aplicação dos LLMs para garantir sua utilização de forma responsável e segura.

6.4.2. Integração de LLMs com Aprendizado Federado

Com o crescimento exponencial dos LLMs e suas aplicações em diversas tarefas, surgem desafios relacionados ao uso de dados, como a privacidade e a segurança. Nesse contexto, o Aprendizado Federado oferece uma solução promissora ao permitir o treinamento de LLMs de maneira descentralizada, preservando a confidencialidade dos dados dos usuários e reduzindo a dependência de grandes repositórios centralizados de informações.

O Aprendizado Federado com LLMs combina a capacidade de processamento local com o poder de generalização dos modelos de larga escala. Em vez de transferir grandes volumes de dados para um servidor central, o Aprendizado Federado permite que os modelos sejam treinados diretamente nos dispositivos dos usuários, onde os dados são gerados [69]. Em seguida, apenas as atualizações de parâmetros são enviadas para um servidor central, onde essas atualizações são agregadas para melhorar o modelo global. Isso oferece uma camada de privacidade, já que os dados brutos permanecem localmente nos dispositivos dos usuários [8].

Os benefícios do Aprendizado Federado com LLMs são notáveis, especialmente

em relação à privacidade e eficiência de comunicação [45]. Ao manter os dados localmente nos dispositivos dos usuários, evita-se a transferência de informações pessoais para um servidor central, o que aprimora a segurança dos dados. Essa abordagem é particularmente importante para LLMs, já que o treinamento desses modelos de larga escala, como o GPT-4, exige grandes volumes de dados, muitas vezes sensíveis. Ao utilizar o Aprendizado Federado, os dados não precisam ser movidos, reduzindo o risco de vazamentos ou uso indevido.

Além disso, o Aprendizado Federado oferece uma grande vantagem ao reduzir significativamente a latência e o custo de comunicação. Em vez de transferir grandes conjuntos de dados para servidores centrais, apenas as atualizações mais relevantes dos modelos, como gradientes ou pesos selecionados, são compartilhadas. Isso é especialmente benéfico para LLMs, que possuem bilhões de parâmetros e requerem recursos computacionais massivos para treinamento completo.

Embora o treinamento integral de LLMs não seja viável em dispositivos remotos com baixa capacidade, como smartphones, existem estratégias que permitem o treinamento parcial desses modelos. Existem algumas abordagens de treinamento parcial do modelo como Adaptação de classificação baixa (*Low-Rank Adaptation* — LoRA) e ajuste fino parcial. O LoRA introduz matrizes de baixa classificação que são treinadas durante o ajuste fino, em vez de atualizar todos os parâmetros do modelo. O restante dos parâmetros do LLM fica congelado. Essa técnica foi desenvolvida para ser eficiente em termos de memória e computação [36].

Outra estratégia eficaz no contexto federado é o uso de técnicas de compressão e seleção de parâmetros. Em vez de transmitir todos os pesos ou gradientes após cada rodada de treinamento, apenas os pesos mais importantes – aqueles que sofreram mudanças significativas – são enviados para o servidor central. Isso reduz drasticamente a quantidade de dados transmitidos e minimiza o custo de comunicação [60]. Ao mesmo tempo, mantém-se a integridade do treinamento global do modelo, pois as atualizações críticas são incorporadas ao LLM central durante o processo de agregação.

Essas abordagens permitem que dispositivos com capacidades limitadas contribuam efetivamente para o treinamento do LLM, oferecendo personalizações locais com base em dados específicos dos usuários, sem comprometer a eficiência global do modelo. A combinação de refinamento local e agregação centralizada resulta em um modelo que é ao mesmo tempo personalizado e escalável, mantendo a robustez e a capacidade de generalização que caracterizam os LLMs.

Outro aspecto importante é que o Aprendizado Federado possibilita a diversidade de dados, aproveitando a variação nos dados distribuídos em diferentes dispositivos. Isso permite que os LLMs treinem com informações mais variadas, refletindo diferentes contextos e preferências dos usuários. Mesmo que o treinamento principal ocorra em servidores mais potentes, os ajustes locais enriquecem o modelo com dados únicos, melhorando a generalização e robustez em uma ampla gama de tarefas e cenários de uso.

6.4.2.1. Desafios do Aprendizado Federado com LLMs

Apesar dos benefícios, o Aprendizado Federado com LLMs também apresenta desafios consideráveis, especialmente em função do grande número de parâmetros e da complexidade dos modelos.

No ambiente federado, um dos principais desafios enfrentados é a distribuição desigual dos dados entre os dispositivos dos usuários, criando uma significativa heterogeneidade [45]. Como os dados em dispositivos individuais podem variar drasticamente, o treinamento de LLMs nessas condições pode ser difícil. A variabilidade nas distribuições de dados pode comprometer a convergência do modelo global, uma vez que as atualizações de parâmetros locais são baseadas em diferentes contextos [69]. Isso aumenta a complexidade do treinamento, pois um modelo global precisa lidar com padrões de dados não homogêneos, o que pode gerar instabilidades e lentidão no processo de Aprendizado Federado.

Outro desafio é a limitação dos recursos computacionais disponíveis nos dispositivos dos usuários. Muitos desses dispositivos, como *smartphones* e *tablets*, não possuem o poder computacional necessário para lidar com a enorme complexidade dos LLMs. Isso pode resultar em uma execução mais lenta do treinamento, além de limitar o processamento local que pode ser realizado. A necessidade de aliviar o fardo computacional nesses dispositivos é fundamental para garantir que eles ainda possam participar de forma eficiente no processo federado, muitas vezes utilizando técnicas como o treinamento parcial ou a atualização de parâmetros seletivos. Nesse contexto, as camadas menos pesadas ou mais personalizáveis dos LLMs podem ser ajustadas localmente, enquanto as partes mais complexas são processadas em servidores centrais com maior capacidade computacional.

Além disso, a comunicação e sincronização de parâmetros entre os dispositivos e o servidor central continua a ser um desafio crucial. Mesmo que o aprendizado federado reduza a necessidade de transferir grandes conjuntos de dados, a sincronização dos parâmetros dos modelos exige uma comunicação frequente. No caso dos LLMs, que possuem bilhões de parâmetros, o custo de comunicação pode ser significativo, mesmo quando apenas os gradientes ou atualizações mais relevantes são compartilhados. Para mitigar esse problema, são empregadas técnicas como a compressão de gradientes e a seleção de atualizações importantes, onde apenas os pesos ou gradientes que sofreram mudanças significativas são enviados para agregação, reduzindo a sobrecarga de comunicação sem comprometer a qualidade do modelo global.

Embora esses desafios sejam significativos, a literatura propõe soluções para mitigar os problemas associados ao Aprendizado Federado com LLMs [45, 69, 19, 73]. Técnicas como agregação ponderada, que ajusta as contribuições dos dispositivos com base na qualidade e quantidade dos dados locais, e métodos de compressão de gradientes, que reduzem o volume de dados transmitidos durante as atualizações, estão entre as estratégias exploradas [19]. Além disso, a utilização de treinamento parcial e atualizações seletivas tem mostrado potencial para reduzir o custo computacional em dispositivos com capacidades limitadas, permitindo que esses dispositivos contribuam para o modelo global sem comprometer a eficiência.

No entanto, apesar dessas abordagens promissoras, esses desafios ainda são temas de pesquisa ativa. Questões como a heterogeneidade dos dados, a eficiência da comunicação e a limitação de recursos computacionais continuam a exigir novas soluções que equilibrem a privacidade dos dados, a robustez dos modelos e a eficiência do sistema como um todo. Assim, o Aprendizado Federado com LLMs permanece uma área de investigação em constante evolução, com grandes oportunidades para avanços teóricos e práticos [34, 15, 14].

6.5. Principais Aplicações em Multimídia e Web

O Aprendizado Federado tem mostrado um grande potencial para transformar diversas áreas de multimídia e web, onde a privacidade dos dados e a eficiência no processamento distribuído são cruciais. Serão descritas, nessa seção, as principais aplicações de Aprendizado Federado em multimídia e web, e algumas das consequências éticas que seu uso — ou não — podem trazer.

6.5.1. Sistemas de Recomendação

O primeiro grupo de aplicações que abordaremos são os sistemas de recomendação, que são fundamentais em plataformas de redes sociais, *e-commerce*, e serviços de *streaming*. Tradicionalmente, esses sistemas exigem a centralização de grandes volumes de dados do usuário para treinar modelos que personalizam o conteúdo sugerido. Contudo, essa abordagem centralizada levanta preocupações sobre privacidade e segurança dos dados, além de exigir uma infraestrutura robusta para o armazenamento e processamento dos dados. Com o advento do Aprendizado Federado, é possível treinar modelos personalizados diretamente nos dispositivos dos usuários, sem que os dados precisem sair dos dispositivos locais. Essa abordagem não só melhora a privacidade do usuário, mas também reduz a latência associada à transferência de grandes volumes de dados para servidores centrais.

Para os propósitos desta seção, iremos focar na recomendação de itens com base em feedback implícito ou por meio de filtragem colaborativa de uma classe (OCCF , do inglês). Nesse tipo de abordagem, o foco está em prever as preferências dos usuários analisando seu histórico de interações, como cliques ou visualizações [37]. A suposição subjacente é que os usuários tendem a se interessar por itens semelhantes àqueles que já visualizaram e que usuários com padrões de comportamento semelhantes provavelmente compartilharão interesses parecidos. No cenário da recomendação federada de itens (FIR, do inglês), os dados dos usuários são mantidos localmente em seus dispositivos, garantindo uma maior proteção à privacidade pois não são levados aos servidores centrais. Contudo, essa estratégia descentralizada também introduz desafios significativos para a modelagem dos comportamentos dos usuários e a inferência precisa de suas preferências.

Para isso, utilizamos o Aprendizado Federado em sistemas de recomendação (FedRS), que oferecem uma abordagem promissora para preservar a privacidade dos dados dos usuários no contexto de FIR. No modelo FedRS, os dados dos usuários são armazenados localmente nos dispositivos de borda, enquanto apenas os parâmetros intermediários são enviados ao servidor central [59]. No entanto, um invasor no servidor central ainda pode inferir informações sensíveis com base nos parâmetros intermediários, como identificar itens com os quais o usuário interagiu ou as avaliações feitas pelo usuário. Isso

ocorre porque o servidor pode usar informações sobre gradientes não-nulos e comparações entre gradientes em diferentes iterações para reconstruir dados sobre as interações do usuário.

Para mitigar os problemas de privacidade associados ao Aprendizado Federado em sistemas de recomendação (FedRS), várias técnicas de proteção têm sido desenvolvidas e incorporadas. Entre essas técnicas, destacam-se o uso de itens fictícios e a criptografia homomórfica [59]. No contexto dos sistemas FedRS, os "itens" referem-se aos produtos, serviços ou conteúdos com os quais os usuários podem interagir e fornecer avaliação. Esses itens podem variar amplamente dependendo da aplicação do sistema de recomendação, abrangendo desde produtos em um *e-commerce*, como livros e eletrônicos, até conteúdos em plataformas de *streaming*, como filmes e músicas. Cada interação do usuário com um item, como avaliações, cliques, visualizações ou compras, gera dados que são usados para personalizar as recomendações que o sistema oferece. No FedRS, a interação com itens é crucial, pois as informações sobre essas interações são usadas para treinar modelos de recomendação que ajudam a prever e sugerir itens relevantes para cada usuário. A proteção da privacidade desses dados de interação é um desafio central, uma vez que informações sensíveis sobre as preferências e comportamentos dos usuários podem ser inferidas a partir das interações com os itens. Por isso, técnicas como itens fictícios são usadas para ocultar as interações reais e proteger a privacidade dos usuários.

O uso de itens fictícios é uma abordagem utilizada para proteger as interações reais dos usuários. Nessa técnica, os clientes não enviam apenas gradientes dos itens com os quais interagiram, mas também gradientes de uma amostra de itens que não foram interagidos — ditos itens fictícios [59]. A ideia é criar um "ruído" que dificulte a inferência precisa das interações reais dos usuários. Embora essa abordagem ajude a obscurecer os dados reais, ela pode introduzir ruído no modelo de recomendação, o que pode impactar a precisão das recomendações. O desafio é encontrar um equilíbrio entre a proteção da privacidade e a manutenção da eficácia do sistema de recomendação.

Além dos itens fictícios, a criptografia homomórfica também é uma técnica importante. Essa abordagem permite que operações matemáticas sejam realizadas em dados criptografados sem a necessidade de descriptografá-los [59]. Isso significa que os dados dos usuários permanecem protegidos durante o processamento e análise, oferecendo um nível adicional de segurança. A criptografia homomórfica pode ser aplicada para garantir que os dados sensíveis não sejam expostos, mesmo enquanto são utilizados para treinar o modelo de recomendação. Por exemplo, um servidor pode receber dados criptografados de diferentes dispositivos de usuários, realizar operações de agregação ou Aprendizado de Máquina nesses dados, e devolver os resultados também criptografados, sem nunca ter acesso ao conteúdo real dos dados.

Do ponto de vista prático, pesquisas anteriores apontam o uso de uma metodologia de fatoração de matriz federada segura, onde os gradientes da matriz de vetorização de itens são criptografados antes de ir ao servidor [13, 59]. No entanto, a aplicação dessa técnica pode ser complexa e pode introduzir uma sobrecarga computacional significativa, o que pode afetar o desempenho e a escalabilidade do sistema, além de não haver garantias de que os clientes vão manter a chave secreta em segurança.

Outras técnicas complementares incluem o compartilhamento de pedaços dos pa-

râmetros intermediários. Estes são quebrados em múltiplos pedaços e compartilhados entre os clientes de forma que ninguém terá acesso aos dados de maneira completa [59]. Essa técnica protege o padrão de uso dos usuários de serem inferidos por meio de uma invasão no servidor, mas outros clientes que receberam parte do gradiente podem acessar os itens classificados por terceiros. Além disso, embora o custo computacional seja reduzido por não precisar utilizar criptografia homomórfica, a latência de rede será um problema. Isso ocorre pois cada pedaço dos parâmetros intermediários deverá ser transferido para se reunir em um local. Outras abordagens sugeriram combinar essa estratégia com os itens fictícios [59].

Além dessas técnicas, há uma inovação significativa chamada personalização dual de *embeddings* de itens [72], que se destaca pela capacidade de criar representações diferentes para itens de cada usuário sem comprometer a privacidade. Diferentemente dos métodos anteriores que compartilhavam exatamente os mesmos *embeddings* de itens entre todos os usuários em um sistema federado, a personalização dual permite um ajuste fino desses *embeddings* para cada usuário, gerando visões específicas e personalizadas das representações dos itens. A ideia proposta se baseia na concepção de que a escolha por itens num conjunto de itens é diferente para cada pessoa. Essa abordagem pode ser integrada diretamente em métodos de recomendação federada existentes, que segundo os autores oferece melhorias imediatas em termos de precisão e relevância das recomendações [72].

Existe, ainda, estratégias que abordam a relação estrutural de usuários e itens por meio de *graph neural networks* (GNNs). As GNNs são particularmente eficazes na modelagem das interações estruturais entre usuários e itens para gerar recomendações mais precisas [37]. As GNNs são capazes de aprender a modelar mudanças tanto estruturais quanto mudanças temporais no grafo. Essa característica torna as GNNs naturalmente úteis em contextos de (a) mudança de comportamento ou preferência ao longo do tempo e (b) associação com diferentes grupos que possuem interesses por itens similares. Além disso, a conexão direta entre usuários e itens dentro do grafo permite que o aprendizado estrutural e de *embeddings* ocorra de maneira direta. No entanto, num cenário federado, onde cada cliente possui um subgrafo local, existem desafios para preservar a privacidade dos dados enquanto se mantém a eficácia das recomendações [37].

Para abordar essas dificuldades, algumas estudos propõem expandir o subgrafo local do usuário para obter informações de vizinhança, transmitindo as identidades dos itens de forma criptografada ao servidor. Embora isso ajude a modelar melhor as interações, ainda existem riscos de privacidade, uma vez que certas informações sensíveis, como as semelhanças entre usuários, podem ser reveladas. Além disso, muitas dessas abordagens só conseguem explorar conectividades de ordem inferior, o que implica em olhar para conexões dos subgrafos apenas, isto é, entre usuários e itens nos quais tais usuários interagiram. Isso limita o potencial completo das GNNs em comparação com cenários centralizados.

Com o intuito de superar essas limitações, novas arquiteturas de recomendação federada baseadas em GNNs foram propostas, como o *Privacy-Preserving Graph Convolution Network* (P-GCN) [37]. Essas arquiteturas buscam modelar a conectividade de ordem superior no grafo descentralizado sem comprometer a privacidade dos usuários. A conectividade de ordem superior indica a capacidade de conectar usuários locais com out-

ros usuários, ou ainda conectar usuários com itens de outros usuários. Através de técnicas como agregação segura e estratégias de ocultação em grupo, essas soluções tentam equilibrar a proteção de dados e a qualidade das recomendações, oferecendo uma abordagem mais robusta e segura para sistemas de recomendação federada.

Em resumo, embora essas técnicas representem avanços significativos na proteção da privacidade em FedRS, elas vêm com desafios próprios relacionados ao equilíbrio entre segurança e desempenho. A pesquisa continua a evoluir para encontrar soluções que garantam uma proteção eficaz sem comprometer a qualidade das recomendações.

6.5.2. Personalização de Experiência do Usuário

A personalização da experiência do usuário tornou-se um dos pilares centrais no desenvolvimento de produtos e serviços digitais, especialmente em uma era onde a competição por atenção e engajamento é acirrada. No entanto, esse objetivo levanta um desafio fundamental: o equilíbrio entre generalização e personalização. De um lado, a generalização, onde os dados agregados de uma ampla base de usuários são utilizados para treinar modelos centralizados, permite que se criem sistemas robustos capazes de atender a uma grande diversidade de perfis de usuários. Esses modelos beneficiam-se da riqueza de dados coletados em grande escala, mas podem falhar em capturar as nuances das preferências individuais, resultando em uma experiência menos personalizada. Por outro lado, a personalização busca incorporar dados individuais para moldar a experiência do usuário de forma única, ajustando os serviços para que se alinhem mais precisamente aos desejos e necessidades pessoais. No entanto, isso levanta questões sobre até que ponto os dados individuais devem ser utilizados e como garantir que essa personalização não comprometa a privacidade do usuário. Assim, o desafio reside em desenvolver abordagens que permitam a combinação eficiente desses dois extremos, utilizando a generalização para criar uma base sólida de funcionalidade, ao mesmo tempo que integra a personalização de maneira segura e eficaz para fornecer uma experiência que realmente agrade quem a utiliza.

A disputa entre generalização e especificidade na personalização da experiência do usuário não se limita apenas a uma questão técnica, mas também envolve implicações sociais e éticas profundas. Quando sistemas amplamente utilizados pela população, como o *Google Search*, são encarados como recursos públicos, a maneira como esses sistemas modelam e representam a realidade pode influenciar a percepção dos usuários e reforçar representações hegemônicas de indivíduos ou grupos sociais. Essa dinâmica pode levar a uma forma de totalitarismo algorítmico [30], onde os modelos de Aprendizado de Máquina, funcionando como caixas-pretas, impõem novas "verdades" que podem sufocar a diversidade de opiniões e reforçar a visão dominante. Em particular, as opiniões da cultura dominante, como as hierarquias raciais e relações de poder, frequentemente prevalecem sobre as opiniões dos grupos marginalizados. Por exemplo, ao se realizar uma busca no Google no ano de 2011 com a frase "*black girls*", conteúdo pornográfico foi retornado [49]. Isso demonstra que os algoritmos de personalização do usuário podem perpetuar estereótipos prejudiciais, sobretudo se considerarmos que, atualmente, cada busca que fazemos leva em consideração uma série de características pessoais dos usuários.

Esse cenário levanta uma série de questões cruciais sobre como a sociedade deve abordar a personalização da experiência do usuário. Primeiramente, é essencial educar

o público sobre como essas ferramentas funcionam e sobre o que ocorre com os rastros digitais que deixamos. Além disso, a responsabilidade é um ponto central, especialmente quando os danos causados a grupos marginalizados são justificados como "falhas" no sistema que podem ser "corrigidas" sem haver uma devida responsabilização[2] [49]. A ausência de políticas públicas que regulem essas práticas e a percepção equivocada de neutralidade dos sistemas digitais exacerbam o problema, deixando a sociedade à mercê de softwares manipulativos que reforçam visões de mundo hegemônicas. Portanto, é imperativo que as empresas e desenvolvedores sejam responsabilizados pelos impactos de seus sistemas, e que se promovam modelos de desenvolvimento que priorizem o retorno dos benefícios aos próprios usuários que contribuem com seus dados, evitando práticas predatórias e garantindo uma personalização mais justa e equilibrada.

A personalização da experiência do usuário no contexto do *Google Search* exemplifica claramente os desafios e as controvérsias envolvidas no uso de dados para moldar os resultados de pesquisa de maneira individualizada. O Google utiliza uma vasta quantidade de informações sobre o comportamento do usuário, como histórico de buscas, localização, dispositivos utilizados, e até mesmo as interações em outras plataformas, para refinar os resultados de busca e apresentar conteúdos que supostamente correspondem aos interesses específicos de cada usuário. Essa abordagem permite que o Google entregue resultados mais relevantes e potencialmente mais úteis, otimizando a experiência de busca de acordo com as necessidades percebidas de cada indivíduo. No entanto, essa personalização levanta questões críticas sobre privacidade e a potencial criação de bolhas de filtro [50], onde os usuários são expostos apenas a conteúdos que reforçam suas crenças e interesses atuais, limitando o acesso a uma visão mais ampla e diversificada da informação disponível.

Além disso, a falta de transparência nos critérios utilizados pelos algoritmos de personalização agrava esse problema, pois os usuários não têm como saber quais informações (nem a quantidade) estão sendo usadas para moldar seus resultados de busca, nem como esses algoritmos tomam decisões que afetam diretamente a forma como eles percebem o mundo. Frequentemente, o usuário não possui mecanismos para alterar a maneira como sua identidade é representada, nem tem controle sobre a comercialização dessas informações: eles carecem de justificativa e capacidades de controle para a autodeterminação [24]. Isso perpetua uma visão distorcida da realidade, amplificando erros e reforçando vieses, sem que os usuários tenham o poder de corrigir ou questionar esses processos.

Como exemplo, em uma pesquisa anterior, os autores identificaram que a ordem de classificação dos resultados de uma busca no Google pode alterar as preferências de votos de pessoas em eleições democráticas [25]. Isso se une ao fato de que o *Google Search* permite que páginas ocupem as primeiras posições ao pagar pela exposição. Combinando todos esses fatos com a incapacidade de indivíduos saberem distinguir uma exposição paga de uma comum, além da maioria das pessoas acreditarem que os resultados da busca são de fato verídicos [49], temos a receita de um desastre. Isso demonstra que não apenas existem problemas de privacidade, mas também um problema para o Estado democrático de direito, que sofre em função de políticas de empresas privadas estrangeiras que atuam

[2] A Google nunca foi responsabilizada pelos resultados da busca "*black girls*" em 2011 [49].

sem controle e regulação.

Embora nem todos estes problemas sejam resolvidos, algumas empresas estão adotando estratégias para mitigá-los. A Apple tem explorado o Aprendizado Federado como uma abordagem, permitindo que modelos de Aprendizado de Máquina sejam treinados diretamente nos dispositivos dos usuários, como iPhones e iPads. Essa técnica garante que os dados pessoais permaneçam no dispositivo, enquanto apenas as atualizações de modelos são compartilhadas de volta para os servidores da Apple de forma agregada e anônima. Mais especificamente, a Apple tem uma preocupação com a personalização do *automatic speech recognition* (ASR) [52], onde dados de todos os usuários de dispositivos móveis da Apple ajudam a treinar um modelo central, mas ao mesmo tempo dados individuais devem ser incorporados para haver suficiente personalização. O sistema desenvolvido pela Apple utiliza uma combinação de um sistema ASR do servidor com o resultado do sistema ASR do dispositivo móvel [52]. Isso também entra na disputa entre generalização e personalização: até que ponto todos os usuários podem ajudar a melhorar o desempenho de modelos centralizados, e até que ponto, e como, os dados individuais devem ser incorporados para que o serviço oferecido seja personalizado e agrade quem o usa. Além disso, a Apple integra o Aprendizado Federado com outras técnicas avançadas de preservação de privacidade. Ao aplicar ruído estatístico às atualizações dos modelos antes de enviá-las aos servidores, a Apple consegue impedir que informações específicas de um usuário possam ser inferidas a partir dos dados agregados. Essa combinação de Aprendizado Federado e técnicas de anonimização de dados reforça o compromisso da Apple em oferecer produtos que respeitam a privacidade dos usuários, enquanto continuam a melhorar as funcionalidades e a personalização dos serviços.

A Google também tem explorado o uso de Aprendizado Federado para melhorar a sugestão de texto e de *emojis* no teclado Gboard [32, 54]. Em vez de centralizar os dados de digitação, o modelo é treinado localmente nos dispositivos e apenas atualizações de modelo, não os dados reais, são compartilhadas com o servidor central. Isso preserva a privacidade do usuário enquanto permite a personalização do sistema de sugestão para cada indivíduo.

Por exemplo, assistentes virtuais como a Siri e o *Google Assistant* podem utilizar Aprendizado Federado para melhorar a compreensão do contexto e das preferências linguísticas dos usuários [68]. Assim, esses sistemas podem aprender e se adaptar continuamente às interações dos usuários, proporcionando respostas mais relevantes e personalizadas. Isso é particularmente útil em aplicações de atendimento ao cliente e suporte técnico, onde respostas rápidas e precisas são essenciais.

A personalização da experiência do usuário é um dos aspectos mais valorizados em plataformas de mídia social e websites interativos. Na literatura da filosofia, Hannah Arendt, em suas reflexões sobre a realidade, argumenta que o sentido de realidade é mediado pelo conhecimento compartilhado dentro de uma comunidade [5]. Contudo, se levarmos o mecanismo de personalização ao extremo, de forma que diferentes *timelines*, ou páginas, ofereçam conteúdos completamente diferentes entre usuários, temos um colapso da realidade — os usuários perdem o senso de conhecimento compartilhado que Hannah Arendt argumentou em suas reflexões.

Nesse contexto, o Aprendizado Federado surge como uma solução tecnológica

promissora, sobretudo para a privacidade, mas que ainda permite um alto grau de personalização. Embora esse efeito em plataformas de mídia social possa ser negativo ou catastrófico, é importante ressaltar que algumas plataformas estão oferecendo a possibilidade de trocar entre diferentes exposições: ver conteúdo apenas de sua rede de amigos, ou ver conteúdo personalizado e recomendado de acordo com seus interesses. Portanto, a evolução do Aprendizado Federado reflete uma tentativa de balancear a personalização da experiência do usuário com a manutenção de um senso de conhecimento compartilhado e privacidade. Essa abordagem oferece um meio-termo entre a personalização extrema e a necessidade de uma realidade consensual, conforme discutido por Arendt.

Ainda assim, no entanto, a implementação prática do Aprendizado Federado em plataformas de mídia social apresenta desafios significativos. Um deles é a dificuldade de manter a qualidade do modelo global treinado a partir de dados locais e heterogêneos, uma vez que a diversidade dos dados pode levar a um desempenho desigual entre diferentes segmentos de usuários. Além disso, há questões de segurança e robustez, pois os dispositivos dos usuários podem ser alvo de ataques, comprometendo a integridade dos modelos treinados localmente e, consequentemente, afetando a confiabilidade do sistema como um todo.

A integração do Aprendizado Federado em plataformas de mídia social também requer uma reflexão ética sobre o grau de personalização desejado e os potenciais efeitos sociais adversos. Embora o Aprendizado Federado possa mitigar problemas de privacidade, ele pode, paradoxalmente, exacerbar a fragmentação do conhecimento compartilhado ao fornecer uma experiência de usuário ainda mais personalizada e isolada sem diversidade de informação [23]. No entanto, de maneira generalizada, os algoritmos de Aprendizado Federado para personalização de experiência em mídias sociais possuem um objeto em comum: filtrar aquilo que mais interessa aos usuários [23]. Mas, nem sempre os usuários percebem que tais algoritmos existem ou como eles operam [23], o que dificulta a criação de mecanismos de defesa por parte desses usuários. Assim, é crucial que o desenvolvimento e a implementação dessas tecnologias sejam acompanhados de um debate contínuo sobre suas implicações para a sociedade e o papel das plataformas na mediação da realidade dos usuários.

Algumas metodologias de Aprendizado Federado são capazes de criar uma "clusterização de usuários" [47]. Nesse caso, os usuários são agrupados por interesses em comum, e em vez de treinar um único modelo global, treina-se um modelo para cada cluster de usuários. Essa estratégia seria um meio-termo entre um modelo puramente global e um puramente local com um melhor balanceamento entre generalização e personalização. Nesse cenário, usuários de plataformas de mídias sociais seriam tratados como grupos que possuem interesses em comum. Embora interessante num primeiro momento, o usuário continua sem a capacidade de escolha, de autodeterminação, onde não necessariamente desejaria ser incluído num grupo específico, e para o qual carece da capacidade de controle da sua própria situação.

A Mozilla, por outro lado, tem se destacado ao explorar o uso do Aprendizado Federado como uma estratégia inovadora para treinar modelos de predição diretamente no navegador Firefox, colocando a privacidade dos usuários em primeiro plano [33]. Nesse contexto, o Aprendizado Federado permite que o treinamento dos modelos ocorra

localmente, nos dispositivos dos próprios usuários, em vez de centralizar os dados em servidores remotos. Essa abordagem é particularmente relevante quando se trata de informações sensíveis, como dados de navegação, que incluem padrões de comportamento online, histórico de sites visitados e preferências de uso. Ao manter os dados localmente em cada navegador, os riscos associados a vazamentos de dados e ataques cibernéticos são significativamente reduzidos.

No Firefox, o Aprendizado Federado é empregado para aprimorar funcionalidades como o preenchimento automático de formulários e a sugestão de URLs [33]. Cada instância do navegador pode gerar modelos adaptados ao comportamento específico do usuário, resultando em uma experiência mais fluida e intuitiva. Embora essas tarefas podem parecer simples, elas envolvem a análise de grandes volumes de dados de interação dos usuários com o navegador. Tradicionalmente, esse tipo de análise seria feito em servidores centrais, levantando preocupações significativas sobre privacidade e segurança.

Com efeito, o Aprendizado Federado permite a personalização da experiência do usuário ao treinar modelos de inteligência artificial diretamente nos dispositivos dos usuários, sem necessidade de centralizar os dados em servidores remotos. Isso preserva a privacidade, garantindo que informações sensíveis, como padrões de uso e preferências, permaneçam locais. Aplicações como navegadores podem utilizar essa técnica para otimizar funcionalidades, como o preenchimento automático e a sugestão de URLs, oferecendo uma experiência mais adaptada ao comportamento individual de cada usuário, enquanto reduzem os riscos associados a vazamentos de dados.

6.5.3. Assistentes Virtuais e Chatbots

O avanço dos Large Language Models (LLMs), como o GPT-3 e seus sucessores, tem revolucionado o campo dos Assistentes Virtuais e Chatbots, oferecendo interações cada vez mais naturais e eficientes. No entanto, a aplicação dessas tecnologias em um contexto que respeite a privacidade dos usuários é um desafio que precisa ser abordado com atenção. Nesse cenário, o Aprendizado Federado se apresenta como uma solução inovadora, possibilitando o treinamento de modelos de linguagem de forma distribuída, sem comprometer a confidencialidade dos dados dos usuários. Essa abordagem é especialmente relevante quando se trata de preservar a polidez e a qualidade das interações, um aspecto crucial na aceitação e no sucesso dos assistentes virtuais. Neste cenário em particular, a capacidade de especialização de uma LLM pode ser a diferença entre a adoção ou não dos usuários, uma vez que diálogos entre uma LLM e o usuário possuem um caráter altamente intimista.

A polidez nas interações com assistentes virtuais é mais do que uma simples questão de boas maneiras; ela desempenha um papel fundamental na construção de uma experiência de usuário positiva e na consolidação da confiança nos sistemas de IA [21]. A polidez não apenas suaviza as interações, mas também é vital para assegurar que as respostas geradas sejam culturalmente sensíveis e respeitosas [20]. No entanto, o desafio reside em treinar LLMs para compreender e reproduzir essas nuances de polidez sem comprometer a privacidade dos dados dos usuários, que é frequentemente coletada e analisada para melhorar a qualidade das respostas.

O Aprendizado Federado oferece uma resposta a esse dilema ao permitir que os modelos de linguagem sejam treinados diretamente nos dispositivos dos usuários. Nesse

sistema, os dados pessoais nunca deixam o dispositivo, e apenas as atualizações dos parâmetros do modelo são enviadas para o servidor central. Essa abordagem permite que os LLMs aprendam com um grande volume de interações reais, ajustando suas respostas para serem mais polidas e adequadas a diferentes contextos culturais, sem a necessidade de acessar dados sensíveis. Dessa forma, assistentes virtuais e chatbots podem melhorar continuamente suas capacidades de comunicação, mantendo ao mesmo tempo altos padrões de privacidade e segurança.

Apesar das vantagens evidentes, a implementação do Aprendizado Federado em LLMs não é isenta de desafios. Um dos principais obstáculos é a necessidade de recursos computacionais robustos e distribuídos para suportar o treinamento local em grande escala. Além disso, a harmonização das atualizações dos modelos provenientes de diversos dispositivos pode ser complexa, exigindo técnicas avançadas de agregação e gerenciamento de modelos. Contudo, as oportunidades que surgem dessa integração são significativas. Assistentes virtuais capazes de entender e respeitar nuances de polidez em diferentes regiões e contextos podem proporcionar uma experiência de usuário significativamente melhorada, estabelecendo um novo padrão na interação entre humanos e máquinas.

Alguns estudos anteriores apontam uma resposta diferente aos problemas de privacidade e segurança. Por exemplo, utilizando o Aprendizado Federado Descentralizado (DFL, do inglês) que se apoia no uso de contratos inteligentes da tecnologia de *blockchain* [58]. A tecnologia *blockchain* é um sistema de registro distribuído e imutável que permite a criação de uma cadeia de blocos, onde cada bloco contém um conjunto de transações ou dados. Essa cadeia é mantida por uma rede de nós (computadores) que trabalham em conjunto para validar e adicionar novos blocos, garantindo que todos os registros sejam consistentes e protegidos contra adulterações. O aspecto descentralizado do *blockchain* elimina a necessidade de uma autoridade central que é frequentemente um problema das metodologias tradicionais de Aprendizado Federado, aumentando a segurança e a transparência do sistema.

No contexto do Aprendizado Federado Descentralizado, o *blockchain* pode ser utilizado para registrar as atualizações dos modelos de aprendizado de forma segura e auditável, servindo como orquestrador para o processo de treinamento [58]. Além disso, permite a seleção de participantes (clientes) fazendo com que seus dados sejam mantidos localmente e garantindo um armazenamento público e visível dos parâmetros do modelo central. Desta forma, cada atualização feita por um cliente é registrada em um bloco, criando um histórico imutável das contribuições de cada participante. Isso não apenas aumenta a segurança, mas também promove a confiança entre os participantes, pois todas as atualizações podem ser verificadas por qualquer membro da rede. Note, por meio da Figura 6.2, como o modelo é treinado localmente e enviado para a *blockchain*. Note que o modelo de Aprendizado de Máquina também é obtido pelos clientes locais por meio da *blockchain*.

Os contratos inteligentes, por sua vez, são programas que residem na *blockchain* e executam automaticamente ações predefinidas quando determinadas condições são atendidas. Os contratos são códigos no qual sua funcionalidade é transparente e todos os membros da rede podem acessá-lo publicamente. No Aprendizado Federado descentralizado, os contratos inteligentes podem ser utilizados para automatizar processos como a

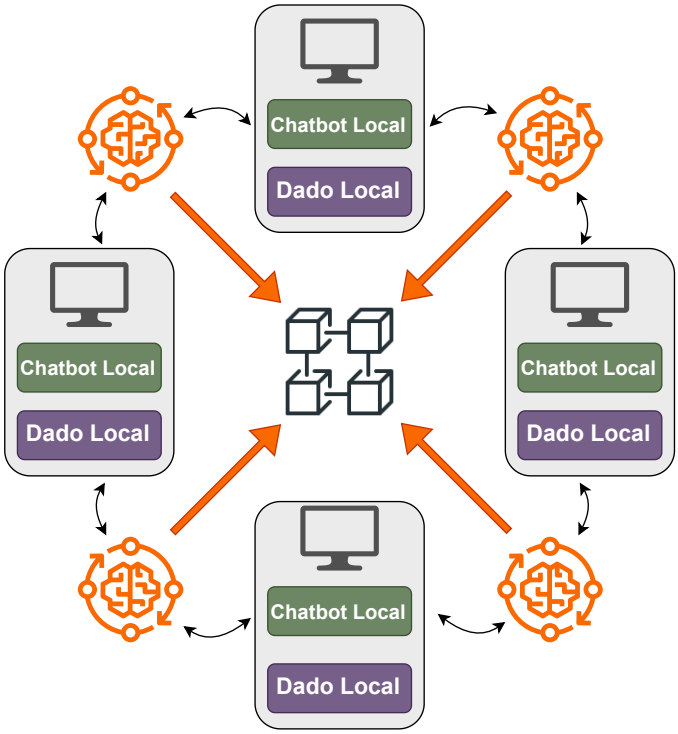

Figure 6.2. Arquitetura geral de um Aprendizado Federado Descentralizado (DFL). Figura adaptada de Su et al. (2024) [58].

agregação de modelos, a distribuição de recompensas financeiras para os participantes e a aplicação de regras de consenso. Por exemplo, um contrato inteligente pode ser programado para distribuir recompensas apenas quando uma determinada quantidade de contribuições válidas for alcançada ou quando o modelo atingir um nível específico de precisão. Além disso, contratos inteligentes podem garantir que as atualizações sejam realizadas de acordo com as regras acordadas, sem a necessidade de intermediários.

No entanto, o uso de *blockchains* apresenta desafios de escalabilidade, especialmente devido à alta latência associada aos mecanismos de consenso utilizados pelas *blockchains* [27]. Esse problema é exacerbado em aplicações que exigem respostas rápidas, como assistentes virtuais e chatbots, onde a latência no processamento pode comprometer significativamente a experiência do usuário. Em um cenário onde o tempo de resposta é crucial para manter o fluxo do diálogo e engajar o usuário, qualquer atraso pode resultar em uma experiência frustrante e menos eficiente. Portanto, encontrar soluções para reduzir essa latência sem comprometer a segurança e a integridade dos dados se torna uma prioridade.

Uma das soluções propostas para mitigar os problemas de escalabilidade e latência no uso de blockchains para Aprendizado Federado Descentralizado é o uso do *InterPlanetary File System* (IPFS). O IPFS é um protocolo de armazenamento distribuído que permite o compartilhamento e a distribuição de dados de maneira descentralizada e eficiente [58]. Ele funciona por meio de um sistema de endereçamento baseado no conteúdo, onde os arquivos são fragmentados em pequenas partes e distribuídos entre os nós da rede. Cada fragmento é identificado por um hash único, e esses hashes são usados para

recuperar os dados de forma rápida e confiável, independentemente de onde eles estão armazenados na rede.

Integrar o IPFS com *blockchains* e contratos inteligentes pode ajudar a aliviar a carga sobre a rede, reduzindo a quantidade de dados que precisa ser processada diretamente na *blockchain* e, consequentemente, diminuindo a latência. Em vez de armazenar grandes volumes de dados na *blockchain*, que podem aumentar o tempo de confirmação das transações, o IPFS permite que os dados sejam armazenados e recuperados de maneira mais ágil, mantendo a eficiência necessária para aplicações de tempo real, como chatbots. Além disso, o IPFS pode garantir a disponibilidade contínua dos dados, mesmo que alguns nós da rede estejam inativos, aumentando a robustez e a resiliência do sistema [58].

Do ponto de vista da aplicação prática, a educação a distância ampliou de maneira notável, sobretudo nos anos da pandemia de COVID-19. Com a transição repentina para o ambiente virtual, tanto instituições de ensino quanto estudantes precisaram se adaptar rapidamente às novas ferramentas de aprendizado online. Esse cenário criou uma demanda crescente por soluções tecnológicas que pudessem proporcionar suporte eficaz e personalizado aos alunos. Dentro desse contexto, os chatbots inteligentes ganharam destaque como ferramentas valiosas para auxiliar na comunicação, no acesso a informações e no acompanhamento do progresso dos estudantes.

Esses chatbots, alimentados por algoritmos de Aprendizado de Máquina e Aprendizado Profundo, têm o potencial de melhorar significativamente a experiência de aprendizado ao oferecer respostas rápidas e precisas às dúvidas dos estudantes, além de fornecer orientação personalizada com base em dados individuais de desempenho. Sistemas como o Boulez exemplificam essa aplicação [22], ao integrar uma rede de chatbots que colaboram entre si utilizando técnicas de Aprendizado Federado. Essa abordagem permite que os chatbots compartilhem conhecimentos e melhorem continuamente sua capacidade de resposta, sem comprometer a privacidade dos dados dos usuários. Como resultado, a interação entre os alunos e os sistemas de aprendizado torna-se mais fluida e envolvente, contribuindo para um ambiente educacional online mais dinâmico e adaptado às necessidades individuais dos estudantes. Neste cenário, as particularidades de interação de cada chatbot, com cursos ou alunos individuais, podem ser aproveitadas para melhorar a interação com todos os chatbots de modo geral.

Portanto, a combinação de LLMs com Aprendizado Federado no desenvolvimento de assistentes virtuais e chatbots representa um passo importante para a criação de sistemas de IA mais seguros, privados e culturalmente sensíveis. Ao preservar a privacidade dos usuários e ao mesmo tempo melhorar continuamente a polidez e a adequação das respostas, essas tecnologias têm o potencial de transformar a maneira como interagimos com máquinas, tornando essas interações mais humanas e eficazes. Assim, o futuro dos assistentes virtuais e chatbots parece promissor, com a promessa de um equilíbrio entre inovação tecnológica e respeito aos valores humanos fundamentais.

6.6. Principais Desafios de Pesquisa do Aprendizado Federado com IA Generativa e LLMs em Multimídia e Web

Nesta seção, exploram-se os principais desafios de pesquisa que surgem da combinação do FL com IA Generativa e LLMs em cenários de Multimídia e Web. O foco desta seção

é abordar desafios críticos como a complexidade da comunicação e a latência inerente ao FL, as exigências de privacidade e segurança dos dados cada vez mais rigorosas e a necessidade de equilibrar a personalização dos modelos com a generalização para diferentes contextos e usuários. Esses desafios são amplificados pelo grande volume e pela diversidade dos dados multimídia, que exigem técnicas avançadas para lidar com a complexidade do processamento. Além disso, as demandas por respostas em tempo real em ambientes de Web tornam a pesquisa nessa área não apenas complexa, mas também essencial para o futuro dessas aplicações baseadas em IA.

6.6.1. Custo de Comunicação e Latência

Um dos principais desafios enfrentados no Aprendizado Federado é o alto custo de comunicação e a latência. A troca de atualizações de modelos entre os dispositivos dos participantes e o servidor de agregação aumenta a demanda por comunicação, especialmente em redes de baixa largura de banda ou com conectividade instável. Embora no Aprendizado Federado seja compartilhado apenas os modelo locais dos participantes, a transferência de LLMs para o servidor de agregação consome muita banda, elevando o custos de comunicação.

Diversos trabalhos na literatura [46, 70, 64, 42, 10, 60] buscam aprimorar a eficiência da comunicação no Aprendizado Federado por meio de três abordagens principais: i) aumentar a computação local, reduzindo, assim, a frequência das rodadas de comunicação; ii) aplicar técnicas de compressão ao modelo local, diminuindo o volume de dados transmitidos ao servidor; e iii) adotar atualizações baseadas em importância, em que apenas os parâmetros com alterações significativas durante a atualização local são enviadas para agregação.

Entre os trabalhos que visam aumentar a computação local, para reduzir o custo de comunicação, Liu *et al.* propuseram um algoritmo que permite aumentar a quantidade de atualizações locais antes da agregação global [46]. O algoritmo possui mecanismos para garantir a convergência através da calibração do número ideal de atualizações locais. De forma semelhante, Yao *et al.* introduziram um modelo de dois fluxos para computação aumentada, utilizando a máxima discrepância média (*Maximum Mean Discrepancy* – MMD) para alinhar os modelos locais com o modelo global durante a atualização local [70]. O objetivo é convergir em menos rodadas de agregação. Wang *et al.*, por sua vez, propuseram um algoritmo para ajustar dinamicamente a frequência de agregação global com base na distribuição de dados e nas características do sistema, otimizando o uso dos recursos disponíveis dos participantes e minimizando a função de perda [64].

Embora os métodos de aumentar a computação local possam reduzir o número total de rodadas de comunicação, esquemas de compressão de modelos também podem ser utilizados para diminuir o volume de dados transmitidos no Aprendizado Federado. Exemplos desses esquemas incluem esparsificação, amostragem parcial e quantização, que reduzem significativamente o tamanho das mensagens enviadas em cada rodada de agregação. Konevcny *et al.* introduziram técnicas como a atualização estruturada e a atualização esboçada, que impõem estruturas predefinidas ou utilizam codificação compactada para minimizar as informações enviadas durante cada rodada de agregação [42]. O método de atualização estruturada utiliza matrizes de baixa classificação e máscaras aleatórias para compactar as atualizações, enquanto a atualização esboçada envolve codi-

ficação e subamostragem antes da transmissão. Caldas *et al.* estenderam esses conceitos ao propor uma compressão com perdas para reduzir ainda mais os custos de comunicação [10]. Compressão com perdas refere-se a técnicas de compressão em que parte dos dados originais é perdida durante o processo, resultando em uma versão compactada que não é idêntica ao original, mas que ainda mantém uma qualidade aceitável para a aplicação específica. Esses métodos têm mostrado eficácia na diminuição do volume de dados transmitidos sem comprometer significativamente o desempenho do modelo [42, 10].

A técnica de atualização baseada em importância é outra estratégia utilizada para reduzir a quantidade de dados transmitidos em Aprendizado Federado. A ideia principal é enviar apenas os gradientes e parâmetros mais relevantes para o servidor de agregação. Tao *et al.* propuseram o algoritmo *edge Stochastic Gradient Descent* (eSGD), que identifica e transmite apenas a fração do gradiente que tem maior impacto na minimização da função perda na atualização local [60]. Da mesma forma, Wang *et al.* desenvolveram um algoritmo que seleciona atualizações relevantes do modelo local comparando-as com o modelo global antes da transmissão [63]. Esse algoritmo visa reduzir os custos de comunicação e garantir a convergência.

6.6.2. Segurança e Privacidade dos Dados

O Aprendizado Federado é vulnerável a ataques que podem comprometer o desempenho do treinamento colaborativo. Esta subseção aborda os principais tipos de ataques direcionados ao Aprendizado Federado, incluindo ataques que afetam o desempenho do modelo e ataques que comprometem a privacidade dos dados.

6.6.2.1. Ataque ao Desempenho do Modelo

Ataques que comprometem o desempenho do modelo em Aprendizado Federado podem impactar negativamente tanto a eficácia do treinamento colaborativo quanto a qualidade das aplicações multimídia e web. Participantes maliciosos podem enviar parâmetros incorretos ou corrompidos, alterando o modelo global durante a agregação e prejudicando seu desempenho. Os ataques ao desempenho do modelo podem ser direcionados, como ataques de *backdoor*, ou não direcionados, como ataques bizantinos [19]. Em ataques de *backdoor*, um participante malicioso pode introduzir modelos envenenados que causam falhas em previsões específicas, como erros na classificação de imagens ou na recomendação de conteúdo [62]. Por outro lado, ataques bizantinos visam desestabilizar o modelo global de forma mais ampla, sem focar em tarefas específicas, resultando em uma atualização incorreta do modelo global e comprometendo todo o treinamento colaborativo [39]. Além disso, participantes podem tentar obter benefícios do modelo global sem contribuir adequadamente, enviando parâmetros aleatórios para a agregação (*free-riding*). Mesmo sem intenção maliciosa, o *free-riding* pode prejudicar o treinamento colaborativo, afetando negativamente a eficácia de sistemas de multimídia e web baseados em Aprendizado Federado.

O servidor de agregação não pode garantir que os participantes tenham usado dados reais durante o treinamento local, o que expõe o sistema a ataques de participantes mal-intencionados [29]. Em contextos de multimídia e web, tais ataques podem compro-

meter significativamente a qualidade das aplicações. Um método comum é o envenenamento do conjunto de dados, onde participantes maliciosos introduzem dados rotulados incorretamente para distorcer o treinamento do modelo global e gerar parâmetros falsificados [28]. Dados rotulados incorretamente referem-se a instâncias onde os rótulos atribuídos não correspondem à verdadeira identidade ou características dos dados, como quando imagens são rotuladas de forma errada ou recomendações são distorcidas. Uma abordagem para realizar esse ataque envolve a geração de amostras falsificadas e a incorporação delas nas atualizações do modelo local, o que pode impedir ou sabotar a convergência adequada do modelo global, afetando negativamente a eficácia das aplicações multimídia e web baseadas em Aprendizado Federado.

Um dos ataques mais eficazes ao desempenho do modelo global, superando até mesmo o envenenamento de dados, é o envenenamento do modelo. Nesse tipo de ataque, um participante mal-intencionado manipula as atualizações do modelo local antes de enviá-las ao servidor de agregação, com o objetivo de comprometer diretamente o modelo global [29]. O adversário busca fazer com que o modelo global classifique incorretamente entradas específicas com elevada confiança, o que é alcançado por meio da alteração deliberada do processo de treinamento. Além disso, o invasor pode amplificar seus próprios parâmetros para dominar a média das atualizações, aumentando sua influência sobre o modelo global. Estudos anteriores [4, 74, 62] têm demonstrado que os ataques de envenenamento de modelo são consideravelmente mais eficazes do que os ataques de envenenamento de dados.

Embora os ataques de envenenamento de dados e de modelo sejam intencionalmente projetados para comprometer o desempenho global, o *free-riding* também pode prejudicar o treinamento colaborativo, mesmo sem ter como objetivo explícito a degradação do modelo. Em Aprendizado Federado, *free-riding* ocorre quando um participante se beneficia dos avanços do modelo global sem contribuir de maneira adequada para o seu treinamento [19]. Um *free-rider* pode, por exemplo, utilizar apenas uma pequena fração de seus dados reais ou até mesmo ruído aleatório, visando economizar recursos computacionais. Isso sobrecarrega os participantes honestos, que precisam compensar essa falta de contribuição, resultando em um modelo global de qualidade inferior e potencialmente menos robusto.

6.6.2.2. Ataque à Privacidade dos Dados

Uma das principais motivações em utilizar o Aprendizado Federado em contextos de multimídia e web é proteger a privacidade dos usuários durante o treinamento colaborativo. No entanto, ataques de privacidade podem comprometer essa proteção ao permitir a inferência dos dados armazenados nos dispositivos dos participantes. Em particular, qualquer entidade com acesso aos modelos locais pode potencialmente descobrir informações sensíveis sobre os dados dos usuários [26]. O servidor de agregação, que consolida as atualizações dos modelos locais, é especialmente vulnerável a esses ataques, representando uma ameaça significativa à suposição de privacidade do Aprendizado Federado, pois os dados dos participantes podem acabar sendo expostos.

A inversão de modelo é um ataque que visa comprometer a privacidade dos da-

dos dos participantes em um ambiente de Aprendizado Federado. Nesse tipo de ataque, um adversário que tem acesso ao modelo treinado, como, por exemplo, o servidor de agregação, utiliza os parâmetros do modelo para tentar reconstruir o conjunto de dados original usado para treinar o modelo. A técnica explora a correlação existente entre as características dos dados de entrada e as previsões geradas pelo modelo, permitindo ao invasor inferir informações sensíveis sobre os dados utilizados no treinamento.

Ao manipular o modelo, o atacante pode gradualmente revelar detalhes específicos, como características demográficas ou outras informações pessoais que foram usadas para treinar o modelo. Mesmo que o modelo global não tenha acesso direto aos dados individuais dos participantes, o conhecimento dos parâmetros atualizados do modelo local permite que o invasor, de maneira iterativa, reconstrua os dados originais ou algo muito próximo deles. Esse tipo de ataque, portanto, representa uma séria ameaça à privacidade, pois viola o princípio de que os dados dos participantes devem permanecer seguros e inacessíveis ao longo do processo de treinamento federado.

O ataque de reconstrução com Rede Adversária Generativa (*Generative Adversarial Network* — GAN) representa uma categoria de ataques de privacidade em Aprendizado Federado que supera a eficácia dos ataques de inversão de modelo [35]. Enquanto ataques de inversão de modelo enfrentam dificuldades para inferir dados em cenários com estruturas de Aprendizado Profundo mais complexas, o ataque de reconstrução GAN, introduzido por Hitaj *et al.* [35], demonstra que um participante mal-intencionado pode efetivamente reconstruir os dados dos outros participantes. Nesse ataque, o adversário cria uma réplica do modelo global para atuar como discriminador e treina um gerador para replicar os dados dos participantes. O processo envolve a inserção de dados gerados no discriminador, a medição da perda nas saídas do discriminador, e subsequente ajuste do gerador. Este método permite ao atacante inferir dados dos participantes, mesmo quando técnicas de privacidade diferencial são aplicadas, embora seja importante notar que um aumento na privacidade diferencial pode resultar em uma diminuição do desempenho do modelo global. Privacidade diferencial é um técnica onde é introduzida uma quantidade controlada de ruído aleatório nas respostas ou nas operações sobre os dados, de modo que seja difícil identificar ou inferir informações sobre qualquer pessoa específica, mesmo com acesso ao conjunto de dados processado.

Em aplicações de multimídia e web, um ataque à privacidade dos dados pode ter consequências particularmente graves, dada a natureza sensível dos dados envolvidos. Por exemplo, em uma aplicação de Aprendizado Federado usada para personalização de conteúdo em plataformas de *streaming* de vídeo ou música, os dados de entrada podem incluir preferências pessoais, histórico de visualizações ou até mesmo informações biométricas capturadas por dispositivos inteligentes. Se um atacante conseguir reconstruir os dados dos usuários, será possível acessar informações íntimas e comportamentais desses usuários.

Em aplicações web, como em redes sociais ou *e-commerce*, os dados utilizados para treinar os modelos podem conter informações pessoais, como hábitos de navegação, histórico de compras, interações sociais e até comunicações privadas. A capacidade de um invasor de reconstruir esses dados a partir de um modelo treinado compromete diretamente a privacidade do usuário, podendo levar a situações como roubo de identidade,

exposição de preferências e comportamentos sensíveis, ou mesmo chantagem.

6.6.3. Heterogeneidade Estatística dos Dados

O Aprendizado de Máquina Distribuído e o Aprendizado Federado são abordagens distintas para a execução de algoritmos de Aprendizado de Máquina em ambientes com dados dispersos por múltiplos dispositivos. A principal diferença entre essas abordagens reside na forma como os dados são manipulados e acessados. No Aprendizado de Máquina Distribuído, um servidor central tem acesso total ao conjunto de dados de treinamento, podendo subamostrar e distribuir esses dados em subconjuntos menores com distribuições semelhantes para os nós participantes, utilizando estruturas como *Apache Spark* [3] e *Apache Hadoop* [4] para esse fim. Em contraste, o Aprendizado Federado opera sob a premissa de que o servidor central não tem acesso direto aos dados. Neste modelo, os dados permanecem nos dispositivos que os geraram, e apenas os parâmetros do modelo treinado são trocados entre os dispositivos.

Portanto, enquanto o Aprendizado de Máquina Distribuído permite o acesso centralizado aos dados e facilita a manipulação e a subamostragem para treinamento, o Aprendizado Federado enfrenta desafios significativos relacionados à heterogeneidade dos dados. A falta de acesso centralizado aos dados e a necessidade de manter os dados localizados nos dispositivos podem resultar em uma variabilidade substancial nas distribuições de dados entre os participantes. Isso pode impactar negativamente o treinamento do modelo, levando a uma redução na precisão e na eficiência do modelo global, além de dificultar a convergência e a generalização em contextos de dados não independentes e identicamente distribuídos (não-IID). Assim, é crucial desenvolver estratégias para mitigar esses efeitos e garantir que o Aprendizado Federado possa alcançar um desempenho robusto e eficaz.

Dados não-IID refere-se a uma situação em que os dados distribuídos entre diferentes participantes ou dispositivos não seguem a mesma distribuição estatística, nem são independentes entre si. Em contextos de Aprendizado Federado, isso significa que os dados em cada dispositivo podem ter diferentes distribuições, características e padrões, o que pode resultar em dados heterogêneos e potencialmente enviesados. Com base em pesquisas anteriores sobre o desafio da heterogeneidade estatística dos dados [45, 69], pode-se observar os seguintes impactos em aplicações multimídia e web:

1. **Desempenho do Modelo**: Em aplicações de multimídia e web, como reconhecimento de imagem, recomendação de conteúdo e personalização de interfaces, dados não-IID podem levar a modelos que não generalizam bem para novas amostras ou usuários. Por exemplo, se um modelo de recomendação for treinado com dados que têm diferentes padrões de comportamento entre usuários, pode haver uma degradação na precisão das recomendações para novos usuários ou conteúdos.

2. **Convergência do Modelo**: A heterogeneidade dos dados pode dificultar a convergência do modelo global, resultando em um treinamento mais lento e menos estável. Em cenários como a análise de vídeos ou a classificação de imagens, isso

[3] Disponível em `https://spark.apache.org/`. Acessado em 26/08/2024

[4] Disponível em `https://hadoop.apache.org/`. Acessado em 26/08/2024

pode comprometer a capacidade do modelo de aprender representações úteis e precisas, afetando a qualidade dos serviços oferecidos.

3. **Personalização e Qualidade do Serviço**: Para aplicações web que dependem de personalização, como motores de busca e assistentes virtuais, a presença de dados não-IID pode levar a uma experiência menos personalizada e relevante para os usuários. A diferença nas preferências e comportamentos entre os dados de diferentes dispositivos pode resultar em um modelo que não reflete adequadamente as necessidades e preferências individuais dos usuários.

Portanto, lidar com a não-IID é crucial para garantir que as aplicações de multimídia e web baseadas em Aprendizado Federado possam oferecer desempenho robusto e uma experiência de usuário consistente e de alta qualidade.

Um tópico emergente sobre heterogeneidade estatística dos dados é a personalização vs. generalização [31]. Para lidar com essa heterogeneidade, o Aprendizado Federado deve equilibrar dois objetivos principais: generalização e personalização. A generalização refere-se à capacidade do modelo global de fazer previsões confiáveis para classes de dados que foram observadas em vários clientes. Isso é particularmente importante quando os dados de entrada durante a inferência se assemelham à distribuição global de treinamento. Por outro lado, a personalização envolve adaptar o modelo para melhor refletir as características específicas dos dados locais de cada cliente. Essa abordagem é crucial quando os dados de um cliente são significativamente diferentes da distribuição global, como no caso de sensores de atividades físicas para diferentes esportes.

No entanto, essas duas metas muitas vezes entram em conflito [67]. Modelos globalmente generalizados podem não desempenhar bem em dados altamente específicos ou não vistos localmente, enquanto modelos personalizados podem falhar ao enfrentar dados que não estão representados no conjunto local de treinamento [75]. Abordagens recentes em Aprendizado Federado personalizado tentam mitigar esses desafios ao permitir que o modelo global se adapte às características locais de cada cliente, ao mesmo tempo em que mantém uma certa capacidade de generalização [31, 67, 17]. Isso pode ser alcançado através da personalização de parâmetros específicos dentro de um modelo global ou utilizando classificadores globais e personalizados em conjunto.

Portanto, a questão central no Aprendizado Federado onde os dados são não-IID é como criar um modelo que consiga equilibrar de forma eficaz a generalização e a personalização [17]. Abordagens como o compartilhamento de partes do modelo ou o ajuste de parâmetros específicos visam harmonizar esses objetivos, mas ainda há uma dificuldade em manter o desempenho geral do modelo enquanto se adapta às particularidades locais [75]. Esse equilíbrio é fundamental para melhorar o desempenho do Aprendizado Federado em cenários de dados diversos e desiguais.

6.7. Considerações Finais e Perspectivas Futuras

O Aprendizado Federado combinado com IA Generativa e Modelos de Linguagem de Grande Escala (LLMs) representa uma evolução significativa no desenvolvimento de tecnologias de inteligência artificial aplicadas a multimídia e web. Esta abordagem não só preserva a privacidade dos dados dos usuários, mas também democratiza o acesso ao

treinamento de modelos complexos, permitindo que um maior número de instituições e dispositivos participem do processo de aprimoramento dos modelos. No entanto, desafios como o custo de comunicação, a heterogeneidade dos dados e a segurança das atualizações do modelo permanecem e exigem soluções inovadoras. Futuras pesquisas devem focar em otimizar a eficiência computacional, desenvolver protocolos de segurança mais robustos e criar algoritmos de agregação que possam lidar com a variabilidade dos dados não-IID. A integração bem-sucedida dessas tecnologias pode transformar significativamente a personalização e a interação dos usuários com plataformas de multimídia e web, promovendo uma experiência mais segura, eficiente e adaptativa. Com a contínua evolução e refinamento dessas abordagens, espera-se que o Aprendizado Federado com IA Generativa e LLMs se torne um pilar central na criação de sistemas inteligentes e colaborativos.

Referências

[1] S. AbdulRahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet of Things Journal*, 8(7):5476–5497, 2020.

[2] S. Angra and S. Ahuja. Machine learning and its applications: A review. In *2017 international conference on big data analytics and computational intelligence (ICBDAC)*, pages 57–60. IEEE, 2017.

[3] L. G. Azevedo, E. F. de Souza Soares, R. Souza, and M. F. Moreno. Modern federated database systems: An overview. *ICEIS (1)*, pages 276–283, 2020.

[4] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. How to backdoor federated learning. In *Proceedings of Machine Learning Research*, volume 108, pages 2938–2948, Online, 26–28 Aug 2020. PMLR.

[5] A. Ballesteros. Digitocracy: Ruling and being ruled. *Philosophies*, 5(2):9, 2020.

[6] Y. Bengio, Y. Lecun, and G. Hinton. Deep learning for ai. *Communications of the ACM*, 64(7):58–65, 2021.

[7] F. Bourse, M. Minelli, M. Minihold, and P. Paillier. Fast homomorphic evaluation of deep discretized neural networks. In H. Shacham and A. Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 483–512, Cham, 2018. Springer International Publishing.

[8] H. Brendan McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*, volume 54, 2017.

[9] J. Cabrera, M. S. Loyola, I. Magaña, and R. Rojas. Ethical dilemmas, mental health, artificial intelligence, and llm-based chatbots. In *International Work-Conference on Bioinformatics and Biomedical Engineering*, pages 313–326. Springer, 2023.

[10] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*, 2018.

[11] K. Cao, Y. Liu, G. Meng, and Q. Sun. An overview on edge computing research. *IEEE access*, 8:85714–85728, 2020.

[12] X. Cao, T. Başar, S. Diggavi, Y. C. Eldar, K. B. Letaief, H. V. Poor, and J. Zhang. Communication-efficient distributed learning: An overview. *IEEE journal on selected areas in communications*, 41(4):851–873, 2023.

[13] D. Chai, L. Wang, K. Chen, and Q. Yang. Secure federated matrix factorization. *IEEE Intelligent Systems*, 36(5):11–20, 2020.

[14] T. Che, J. Liu, Y. Zhou, J. Ren, J. Zhou, V. S. Sheng, H. Dai, and D. Dou. Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization. *arXiv preprint arXiv:2310.15080*, 2023.

[15] C. Chen, X. Feng, J. Zhou, J. Yin, and X. Zheng. Federated large language model: A position paper. *arXiv preprint arXiv:2307.08925*, 2023.

[16] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor. Distributed learning in wireless networks: Recent progress and future challenges. *IEEE Journal on Selected Areas in Communications*, 39(12):3579–3605, 2021.

[17] M. Chen, M. Jiang, Q. Dou, Z. Wang, and X. Li. Fedsoup: Improving generalization and personalization in federated learning via selective model interpolation. In H. Greenspan, A. Madabhushi, P. Mousavi, S. Salcudean, J. Duncan, T. Syeda-Mahmood, and R. Taylor, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, pages 318–328, Cham, 2023. Springer Nature Switzerland.

[18] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems*, 35(4):83–93, 2020.

[19] H. N. Cunha Neto, J. Hribar, I. Dusparic, D. M. F. Mattos, and N. C. Fernandes. A survey on securing federated learning: Analysis of applications, attacks, challenges, and trends. *IEEE Access*, 11:41928–41953, 2023.

[20] M. de Souza Monteiro and L. C. de Castro Salgado. Conversational agents: a survey on culturally informed design practices. *Journal on Interactive Systems*, 14(1):33–46, 2023.

[21] M. de Souza Monteiro, V. C. Pereira, and L. C. de Castro Salgado. Investigating politeness strategies in chatbots through the lens of conversation analysis. In *Anais do XXII Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais*. SBC, 2023.

[22] S. D'Urso, F. Sciarrone, and M. Temperini. Boulez: A chatbot-based federated learning system for distance learning. In *2023 27th International Conference Information Visualisation (IV)*, pages 210–215, 2023.

[23] R. Eg, Ö. D. Tønnesen, and M. K. Tennfjord. A scoping review of personalized user experiences on social media: The interplay between algorithms and human factors. *Computers in Human Behavior Reports*, 9:100253, 2023.

[24] S. Engelmann, V. Scheibe, F. Battaglia, and J. Grossklags. Social media profiling continues to partake in the development of formalistic self-concepts. social media users think so, too. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 238–252, 2022.

[25] R. Epstein and R. E. Robertson. The search engine manipulation effect (seme) and its possible impact on the outcomes of elections. *Proceedings of the National Academy of Sciences*, 112(33):E4512–E4521, 2015.

[26] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 17–32, 2014.

[27] X. Fu, H. Wang, and P. Shi. A survey of blockchain consensus algorithms: mechanism, design and applications. *Science China Information Sciences*, 64:1–15, 2021.

[28] C. Fung, C. J. Yoon, and I. Beschastnikh. Mitigating sybils in federated learning poisoning. *arXiv preprint arXiv:1808.04866*, 2018.

[29] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Madry, B. Li, and T. Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2022.

[30] B.-C. Han. *Infocracy: Digitization and the crisis of democracy*. John Wiley & Sons, 2022.

[31] D.-J. Han, D.-Y. Kim, M. Choi, C. G. Brinton, and J. Moon. Splitgp: Achieving both generalization and personalization in federated learning. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, pages 1–10, 2023.

[32] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage. Federated learning for mobile keyboard prediction, 2019.

[33] F. Hartmann, S. Suh, A. Komarzewski, T. D. Smith, and I. Segall. Federated learning for ranking browser history suggestions, 2019.

[34] A. Hilmkil, S. Callh, M. Barbieri, L. R. Sütfeld, E. L. Zec, and O. Mogren. Scaling federated learning for fine-tuning of large language models. In E. Métais, F. Meziane, H. Horacek, and E. Kapetanios, editors, *Natural Language Processing and Information Systems*, pages 15–23, Cham, 2021. Springer International Publishing.

[35] B. Hitaj, G. Ateniese, and F. Perez-Cruz. Deep models under the gan: Information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 603–618, New York, NY, USA, 2017. Association for Computing Machinery.

[36] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[37] P. Hu, Z. Lin, W. Pan, Q. Yang, X. Peng, and Z. Ming. Privacy-preserving graph convolution network for federated item recommendation. *Artificial Intelligence*, 324:103996, 2023.

[38] C. Janiesch, P. Zschech, and K. Heinrich. Machine learning and deep learning. *Electronic Markets*, 31(3):685–695, 2021.

[39] M. S. Jere, T. Farnan, and F. Koushanfar. A taxonomy of attacks on federated learning. *IEEE Security & Privacy*, 19(2):20–28, 2021.

[40] Y. Jin, Y. Liu, K. Chen, and Q. Yang. Federated learning without full labels: A survey, 2023.

[41] F. A. KhoKhar, J. H. Shah, M. A. Khan, M. Sharif, U. Tariq, and S. Kadry. A review on federated learning towards image processing. *Computers and Electrical Engineering*, 99:107818, 2022.

[42] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[43] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[44] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen. Multi-key privacy-preserving deep learning in cloud computing. *Future Generation Computer Systems*, 74:76 – 85, 2017.

[45] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):2031–2063, 2020.

[46] Y. Liu, Y. Kang, X. Zhang, L. Li, Y. Cheng, T. Chen, M. Hong, and Q. Yang. A communication efficient vertical federated learning framework. *arXiv preprint arXiv:1912.11187*, 2019.

[47] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

[48] C. Mawela. A web-based solution for federated learning with llm based automation. Master's thesis, C. Mudiyanselage, 2024.

[49] S. U. Noble. Algorithms of oppression. In *Algorithms of Oppression*. New York University Press, 2018.

[50] E. Pariser. *The filter bubble: What the Internet is hiding from you*. penguin UK, 2011.

[51] N. Patwardhan, S. Marrone, and C. Sansone. Transformers in the real world: A survey on nlp applications. *Information*, 14(4):242, 2023.

[52] M. Paulik, M. Seigel, H. Mason, D. Telaar, J. Kluivers, R. van Dalen, C. W. Lau, L. Carlson, F. Granqvist, C. Vandevelde, et al. Federated evaluation and tuning for on-device personalization: System design & applications. *arXiv preprint arXiv:2102.08503*, 2021.

[53] S. Rajendran, J. S. Obeid, H. Binol, R. D'Agostino, K. Foley, W. Zhang, P. Austin, J. Brakefield, M. N. Gurcan, and U. Topaloglu. Cloud-Based Federated Learning Implementation Across Medical Centers. *JCO Clinical Cancer Informatics*, 5:1–11, 2021. PMID: 33411624.

[54] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays. Federated learning for emoji prediction in a mobile keyboard, 2019.

[55] J. Ren, W. Ni, G. Nie, and H. Tian. Research on resource allocation for efficient federated learning, 2021.

[56] L. Sani, A. Iacob, Z. Cao, B. Marino, Y. Gao, T. Paulik, W. Zhao, W. F. Shen, P. Aleksandrov, X. Qiu, et al. The future of large language model pre-training is federated. *arXiv preprint arXiv:2405.10853*, 2024.

[57] P. P. Shinde and S. Shah. A review of machine learning and deep learning applications. In *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*, pages 1–6. IEEE, 2018.

[58] H. Su, C. Xiang, and B. Ramesh. Towards confidential chatbot conversations: A decentralised federated learning framework. *The Journal of The British Blockchain Association*, 2024.

[59] Z. Sun, Y. Xu, Y. Liu, W. He, L. Kong, F. Wu, Y. Jiang, and L. Cui. A survey on federated recommendation systems. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[60] Z. Tao and Q. Li. esgd: Communication efficient distributed deep learning on the edge. In {*USENIX*} *Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[62] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16070–16084. Curran Associates, Inc., 2020.

[63] L. Wang, W. Wang, and B. Li. Cmfl: Mitigating communication overhead for federated learning. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 954–964, 2019.

[64] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019.

[65] J. Wu, S. Yang, R. Zhan, Y. Yuan, D. F. Wong, and L. S. Chao. A survey on llm-gernerated text detection: Necessity, methods, and future directions. *arXiv preprint arXiv:2310.14724*, 2023.

[66] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li. A survey of federated learning for edge computing: Research problems and solutions. *High-Confidence Computing*, 1(1):100008, 2021.

[67] C. Xie, D.-A. Huang, W. Chu, D. Xu, C. Xiao, B. Li, and A. Anandkumar. Perada: Parameter-efficient federated learning personalization with generalization guarantees. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23838–23848, June 2024.

[68] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–19, 2019.

[69] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[70] X. Yao, C. Huang, and L. Sun. Two-stream federated learning: Reduce the communication costs. In *2018 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, 2018.

[71] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 4(2):100211, 2024.

[72] C. Zhang, G. Long, T. Zhou, P. Yan, Z. Zhang, C. Zhang, and B. Yang. Dual personalization on federated recommendation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 4558–4566. ijcai.org, 2023.

[73] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

[74] X. Zhou, M. Xu, Y. Wu, and N. Zheng. Deep model poisoning attack on federated learning. *Future Internet*, 13(3), 2021.

[75] G. Zhu, X. Liu, J. Niu, S. Tang, X. Wu, and J. Zhang. Dualfed: enjoying both generalization and personalization in federated learning via hierachical representations. *arXiv preprint arXiv:2407.17754*, 2024.