Aprendizado Federado Incremental e Sensível ao Risco para Modelos de Ranqueamento em Cenários com Distribuições Heterogêneas de Dados

Gestefane Rabbi[1], Celso França[1], Daniel Xavier de Sousa[2], Thierson Couto Rosa[3], Jussara M. Almeida[1], Marcos André Gonçalves[1]

[1] Dep. de Ciência da Computação - Universidade Federal de Minas Gerais (UFMG)
[2] Instituto Federal de Goiás (IFG)
[3] Instituto de Informática (UFG)

{gestefane,celsofranca,jussara,mgoncalv}@dcc.ufmg.br
daniel.sousa@ifg.edu.br, thierson@ufg.br

Resumo. *Este trabalho propõe uma nova estratégia de Aprendizado Federado para Ranqueamento (FL2R) em cenários com dados não independentes e não identicamente distribuídos (não-IID) entre clientes. Apresentamos o* ***FedRisk****, um método de agregação sensível ao risco que pondera as contribuições dos clientes conforme sua confiabilidade, aliado a um mecanismo de reutilização de parâmetros do modelo global anterior, para mitigar os efeitos da heterogeneidade dos dados. Experimentos com o conjunto MSLR-WEB10K mostram que o* ***FedRisk*** *supera o FedProx — baseline mais robusto — ao reduzir a diferença de desempenho entre modelos federados e centralizados. O FedRisk alcançou uma melhoria de* ***15.6%*** *no nDCG@5 em relação ao FedProx e reduziu substancialmente a variância, aumentando a estabilidade entre rodadas. Além disso, para métricas como nDCG@10, o* ***FedRisk*** *igualou o desempenho do modelo centralizado — feito inédito entre os métodos comparados, sobretudo em um cenário federado não-IID.*

Abstract. *This work proposes a new Federated Learning for Ranking (FL2R) strategy in scenarios with non-independent and non-identically distributed (non-IID) data across clients. We present* ***FedRisk****, a risk-aware aggregation method that weights client contributions according to their reliability, combined with a mechanism for reusing parameters from the previous global model, to mitigate the effects of data heterogeneity. Experiments with the MSLR-WEB10K ensemble show that* ***FedRisk*** *outperforms FedProx, the most robust baseline, by reducing the performance gap between federated and centralized models. FedRisk achieved a* ***15.6%*** *improvement on nDCG@5 over FedProx and substantially reduced variance, increasing inter-round stability. Furthermore, for metrics such as nDCG@10,* ***FedRisk*** *matched the performance of the centralized model — a first among the compared methods, especially in a non-IID federated setting.*

1. Introdução

Tradicionalmente, o treinamento de modelos de aprendizado de máquina é realizado de forma centralizada, com todos os dados concentrados em um único computador. Contudo, em muitos cenários, os dados estão distribuídos em diferentes dispositivos, como, por exemplo, em smartphones que armazenam localmente os padrões de escrita

de texto ou padrões de busca para cada usuário. Nesses casos, enviar todos os dados para um servidor central com o objetivo de executar o treinamento dos modelos torna-se indesejável ou até proibitivo, seja por questões de privacidade ou pelo grande volume de informações [Neto et al. 2020, Hejazinia et al. 2022]. Outro problema dessa abordagem centralizada é o risco de falha do servidor, deixando os dispositivos sem respostas [Jeong et al. 2022]. Além disso, os dados distribuídos geralmente não seguem um padrão único de representação, implicando em alto custo computacional de padronização para um aprendizado centralizado sobre todos os dados [Tong et al. 2021].

Neste contexto de paradigma de aprendizado distribuído, com dados geograficamente distribuídos, o aprendizado federado (*federated learning* (FL)) visa lidar com a impossibilidade ou a inviabilidade de reunir dados em um único local para o treinamento de modelos. No FL, múltiplos clientes — como dispositivos móveis, servidores de instituições ou organizações descentralizadas — colaboram para treinar um modelo global de forma coletiva, sem compartilhar seus dados locais.

No contexto do *aprendizado federado*, os dados permanecem descentralizados e armazenados localmente por cada cliente, refletindo padrões específicos de comportamento, preferências e contextos de uso. Idealmente, o modelo federado deveria ser tão efetivo quanto um modelo treinado com todos os dados centralizados [Wang and Zuccon 2022]. Contudo, atingir exatamente o mesmo nível de desempenho de um modelo centralizado é extremamente desafiador, devido à natureza descentralizada e heterogênea dos dados entre os clientes. Essa natureza descentralizada frequentemente resulta em dados, que quando considerados coletivamente, são *não independentes e não identicamente distribuídos* (não-IID). Logo, aproximar o desempenho do modelo federado ao do modelo centralizado é considerado um resultado altamente positivo, pois demonstra que mesmo sem acesso completo aos dados, é possível construir modelos globais eficazes preservando a privacidade dos dados locais.

Federated Learning to Rank (FL2R) aplica o aprendizado federado à tarefa de ranqueamento de documentos. Nesse contexto, a heterogeneidade se expressa principalmente pela distribuição desigual dos rótulos de relevância e pela variação no número de amostras por cliente. Por exemplo, de um conjunto com cinco níveis de relevância (*irrelevante* (0) a *relevante* (4)), um cliente pode conter apenas exemplos com rótulo *parcialmente irrelevante (1)*, enquanto outro possui apenas rótulos 3 e 4, resultando em um aprendizado local enviesado a partir de subconjuntos limitados do espaço de rótulos. Além disso, a quantidade de exemplos disponíveis por cliente pode variar substancialmente, acentuando o desequilíbrio — caracterizando a parte *não identicamente distribuída* do cenário. Essa heterogeneidade impacta diretamente os valores de parâmetros aprendidos localmente e dificulta sua agregação em um modelo global que represente adequadamente o conjunto de clientes. [Wang and Zuccon 2022] mostram que distribuições não-IID podem resultar em modelos globais altamente enviesados, favorecendo desproporcionalmente os dados de determinados clientes.

A fase de agregação dos modelos locais para compor o modelo global representa um dos principais desafios de FL2R e de aprendizado federado em geral. A combinação eficiente de modelos treinados sobre dados heterogêneos é complexa e impacta diretamente a estabilidade e o desempenho do modelo central. Diversas estratégias têm sido propostas para mitigar esse problema, destacando-se o FedAvg [Liu et al. 2021], abordagem amplamente adotada que realiza a média ponderada dos parâmetros dos

modelos locais, com base no número de amostras de treinamento de cada cliente.

Apresentamos neste artigo uma nova proposta de agregação dos parâmetros em modelos federados, com aplicação para FL2R. Nossa proposta baseia-se em duas principais hipóteses: (i) *os parâmetros dos clientes podem ser ponderados em função da Sensibilidade ao Risco* [Rodrigues et al. 2025], uma métrica que captura o grau de confiabilidade (menores taxas de erro) nas predições do modelo de cada cliente – *parâmetros dos modelos de clientes com menor risco terão maior peso em relação aos modelos mais suscetíveis a erro*; e (ii) *os modelos federados podem aproveitar o conhecimento adquirido a cada rodada de interação, fazendo uso do histórico dos parâmetros durante o treinamento para reduzir a variabilidade e a aumentar previsibilidade.*

A primeira hipótese mitiga a ausência de avaliação quando há falta de confiabilidade no desempenho dos clientes [Divi et al. 2021]. A segunda hipótese trata de um cenário muito frequente na literatura, que é ignorar o aprendizado dos parâmetros do modelo global no processo de otimização, descartando informações úteis como o aprendizado histórico, especialmente em cenários com dados não-IID.

Para prover evidências para as nossas hipóteses, definimos duas principais questões de pesquisa, as quais procuramos responder empiricamente:

QP1: *O uso da Sensibilidade ao Risco (como descrito em [Rodrigues et al. 2025]), computado a partir dos erros de predição gerados pelos clientes como um fator de ponderação na agregação dos parâmetros, melhora a eficácia do modelo global?*

Com base em trabalhos que exploram o conceito de sensibilidade ao risco [Rodrigues et al. 2022, Rodrigues et al. 2025], propomos um mecanismo que avalia a variabilidade do desempenho dos modelos locais. Especificamente, cada cliente envia ao servidor seus erros de predição, calculados por meio do erro quadrático médio (MSE – *Mean Squared Error* [Hastie et al. 2009]), a partir dos quais o servidor estima um coeficiente de risco individual. Esse coeficiente é então utilizado na ponderação dos modelos durante a agregação: clientes mais estáveis (com menor risco) têm maior influência na composição do modelo global, enquanto clientes mais instáveis têm peso reduzido. Assim, a agregação passa a considerar não apenas os parâmetros locais, mas também a qualidade com que foram aprendidos, potencialmente promovendo maior estabilidade e consistência nas métricas de desempenho ao longo das rodadas.

QP2 - *A inclusão de parâmetros históricos no processo de agregação pode contribuir para mitigar os efeitos negativos de cenários onde os dados são não-IID?*

Avaliamos a melhoria potencial na agregação ao adicionar conhecimento histórico (parâmetros do modelo global da rodada anterior) aos parâmetros dos clientes na rodada atual, atuando de forma complementar ao fator de Sensibilidade ao Risco para mitigar ainda mais a redução de desempenho do modelo global.

Conseguimos mostrar empiricamente que a ponderação por sensibilidade ao risco reduziu significativamente a variabilidade no desempenho do modelo global e promoveu maior estabilidade nas métricas ao longo das rodadas. Além disso, a adição de memória histórica ao processo de agregação contribuiu para estabilizar a convergência, reduzindo os intervalos de confiança. O **FedRisk** melhorou o nDCG@5 em **15,6%** em relação ao FedProx, o *baseline* mais efetivo e estável, e superou outros *baselines* com margens ainda maiores, como 18,0% sobre o FedAvg em nDCG@10 e 39,7% sobre o FedAdagrad em nDCG@10. Mais ainda, para determinadas métricas como o nDCG@10, o **FedRisk** con-

Algorithm 1: FedAvg - Federated Averaging

Servidor executa:

Input: Número total de clientes K, rodadas T, quantidade C de clientes por rodada
Output: Modelo global θ_G

Inicializar θ_G^0 // **no servidor**
for *cada rodada* $t = 1, ..., T$ **do**
  Selecionar subconjunto S_t com C clientes aleatórios
  for *cada cliente* $k \in S_t$ ***em paralelo*** **do**
    $\theta_k^t \leftarrow$ **ClientTrain**(k, θ_G^{t-1})
  $n \leftarrow \sum_{k \in S_t} n_k$
  $\theta_G^t \leftarrow \sum_{k \in S_t} \frac{n_k}{n} \hat{\theta}_k^t$
return θ_G^t

Algorithm 2: ClientTrain(k, θ) – Treinamento local no cliente k

Cliente executa:

Input: Cliente k, parâmetros globais θ, épocas E, batch size B, taxa de aprendizado η
Output: Parâmetros locais θ_k

Dividir os dados locais P_k em batches de tamanho B
Inicializar modelo local: $\theta_k \leftarrow \theta$
for *épocas* $e = 1, ..., E$ **do**
  for *cada batch* $b \in P_k$ **do**
    Atualizar θ_k com SGD nos dados do batch b
return θ_k

Figura 1. Algoritmo FedAvg (esquerda): executado pelo servidor. Algoritmo ClientTrain (direita): execução local em cada cliente durante o treinamento federado.

seguiu igualar a performance do modelo centralizado, um feito que nenhum outro *baseline* foi capaz, principalmente devido à natureza não-IID de dados em cenários federados.

2. Trabalhos Relacionados

Esta seção discute trabalhos relevantes em FL e os desafios associados, como a agregação de modelos, o impacto da natureza não-IID dos dados e o uso de métricas para avaliação da qualidade do aprendizado.

Estado da Arte em Aprendizado Federado: A maioria dos métodos de agregação em FL tem como base o Federated Averaging (FedAvg) [Jiang et al. 2020], amplamente reconhecido como o estado da arte. O processo começa com a inicialização dos parâmetros do modelo global θ_G^0 no servidor central, onde θ representa o conjunto de valores internos (como pesos e vieses) que definem o comportamento do modelo de aprendizado de máquina. A cada rodada t, o servidor seleciona aleatoriamente uma fração C dos K clientes, envia a eles os parâmetros θ_G^{t-1}, e cada cliente realiza uma atualização local utilizando *Stochastic Gradient Descent* (SGD) [Bottou 2010], um método de otimização que ajusta gradualmente esses parâmetros com base em pequenas amostras dos dados. Essa atualização é realizada por E épocas, sendo que cada época corresponde a uma passagem completa pelos dados do cliente k, permitindo que o modelo refine suas previsões antes de retornar seus parâmetros atualizados θ_k^t ao servidor. Após o treinamento de todas as épocas, os clientes retornam os parâmetros ao servidor, que calcula a média ponderada pelas quantidades de dados locais n_k, produzindo o novo modelo global $\theta_G^t = \sum_{k \in S_t} \frac{n_k}{n} \theta_k^t$, onde S_t é o subconjunto de clientes selecionados para treino na rodada t e n é o total de exemplos da rodada. Esse modelo é então usado na próxima iteração. Contudo, a média ponderada pode sofrer com dados não-IID dos clientes, causando baixa efetividade do modelo global.

A Figura 1 apresenta o algoritmo clássico de agregação em aprendizado federado, conhecido como *FedAvg*, juntamente com o procedimento de treinamento local realizado em cada cliente. Utilizamos a seguinte notação ao longo dos algoritmos: θ_G^t representa os parâmetros do modelo global na rodada t; θ_k^t representa os parâmetros do modelo local do cliente k na rodada t; S_t é o subconjunto de clientes selecionados para a rodada t; n_k é o número de exemplos de treino do cliente k; $n = \sum_{k \in S_t} n_k$ é o número total de exemplos utilizados na rodada; P_k indica o conjunto de dados locais do cliente k.

Melhoria na Efetividade do Aprendizado Federado: Diversas estratégias fo-

ram propostas para aprimorar modelos em FL, especialmente no contexto de dados não-IID. Entre as principais estratégias desenvolvidas, destaca-se o FedProx [Li et al. 2020], que introduz uma penalização proximal nos modelos locais para reduzir a divergência entre os parâmetros locais e globais durante o treinamento. [Karimireddy et al. 2020] propõem a utilização de variáveis de controle para corrigir o desvio introduzido pela heterogeneidade dos dados. As variáveis são atualizadas e compartilhadas entre servidor e clientes, influenciando tanto a atualização local quanto a agregação. Ainda, FedNova [Wang and Liu 2020] ajusta as contribuições dos clientes proporcionalmente ao número de atualizações realizadas localmente. A normalização introduzida por esse método reduz o impacto da heterogeneidade, mas não elimina as diferenças de desempenho entre clientes com volumes e qualidades de dados distintos.

Sensibilidade ao Risco: Em RI[1], Sensibilidade ao Risco é definida como técnicas que buscam reduzir a probabilidade de resultados ruins em consultas específicas, enquanto maximizam a qualidade geral dos resultados [Wang et al. 2012].

Diferente desse conceito, estudos que exploram fatores de qualidade do desempenho dos clientes envolvendo análise de risco (*risk-aware*) têm sido desenvolvidos, mas nenhum aborda especificamente o conceito de Sensibilidade ao Risco [Wang et al. 2012]. Por exemplo, [Zhao et al. 2024] propõem o FRAL-CSE, que estima centralmente a sensibilidade dos clientes para orientar a agregação global, mas sem o uso de métricas explícitas de sensibilidade ao risco. [Ads et al. 2024] desenvolvem uma abordagem de FL acelerado que leva em conta fatores de risco de transmissão e confiabilidade na seleção e ponderação de clientes. Já [Chen et al. 2021] introduzem um modelo para minimizar riscos associados a decisões incorretas em ambientes de *crowdsensing*. Esses trabalhos representam uma tendência emergente de incorporação de noções de risco no contexto federado, mas ainda não consideram estratégias baseadas na sensibilidade ao risco nem a ponderação dinâmica de clientes com base na qualidade preditiva.

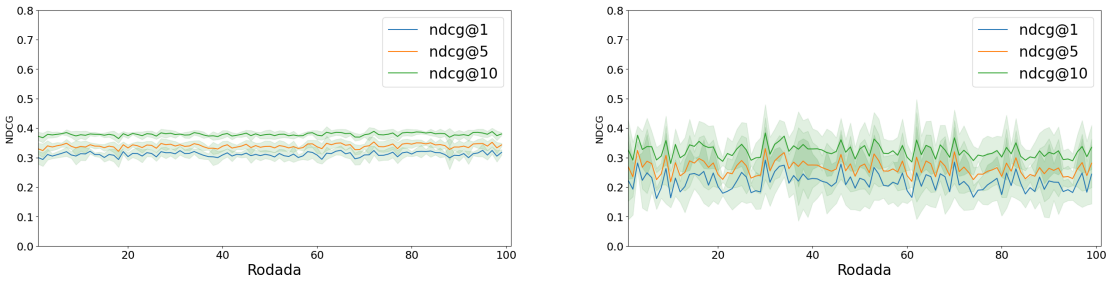
Em nossa abordagem, exploramos os trabalhos recentes de [Rodrigues et al. 2022, Rodrigues et al. 2025] que propõem a *RiskLoss*, uma função derivável para otimizar a sensibilidade ao risco em Redes Neurais Profundas, no contexto de RI. Na prática, a função *RiskLoss* utiliza a variação entre vários modelos de *ranking* e diversas consultas durante o treinamento, reduzindo a possibilidade de resultados muito ruins e, em alguns casos, melhorando os resultados gerais. Nesse contexto, utilizamos a função *RiskLoss* em conjunto com o conhecimento histórico dos parâmetros em aprendizado federado, melhorando o fator de ponderação dos parâmetros dos clientes na formação do modelo global.

3. Abordagens Propostas

A Figura 2 ilustra o impacto da distribuição dos dados no aprendizado federado utilizando FedAvg. O gráfico à esquerda apresenta o cenário com dados IID, em que as métricas de desempenho (nDCG@1, nDCG@5 e nDCG@10) exibem maior estabilidade ao longo das rodadas, com intervalos de confiança (IC) mais estreitos. Em contraste, o gráfico à direita mostra o caso não-IID, evidenciando alta variabilidade nas métricas e ausência de convergência, com ICs amplos e flutuações significativas.

A Figura 2 destaca o problema central que buscamos resolver neste trabalho: *reduzir a variabilidade e melhorar a estabilidade no desempenho do modelo global*

[1]Recuperação de Informação.



(a) - FedAvg com dados IID **(b) - FedAvg com dados não-IID**

Figura 2. Desempenhos médios em 5 *folds* do FedAvg em cenários com dados IID e não-IID.

em cenários com dados não-IID. Para isso, propomos investigar duas estratégias complementares: (i) a introdução de um fator de Sensibilidade ao Risco para ajustar o peso de cada cliente na agregação, considerando a confiabilidade de seu modelo; (ii) a incorporação do conhecimento histórico ao processo de agregação, aproveitando parâmetros do modelo global de rodadas anteriores. A seguir, apresentamos em detalhes cada uma das abordagens propostas.

3.1. Uso de Sensibilidade ao Risco na Agregação

Na primeira estratégia, propomos substituir a técnica tradicional de agregação do FL — baseada na proporção da quantidade de amostras de cada cliente em relação ao total de amostras dos clientes da rodada ($\frac{n_k}{n}$, no Algoritmo 1) — por uma estratégia que utiliza o fator de Sensibilidade ao Risco [Rodrigues et al. 2022] como ponderador na combinação dos modelos locais. Esse fator é calculado pelo servidor com base nos erros de predição enviados por cada cliente e reflete a confiabilidade do seu modelo em uma rodada. O objetivo dessa abordagem é reduzir a variabilidade no desempenho do modelo global e aumentar a estabilidade da convergência, mesmo sem necessariamente melhorar a efetividade. Essa abordagem é ilustrada nos algoritmos 3 e 4 que apresentam o fluxo do treinamento federado com agregação sensível ao risco. O Algoritmo 3 descreve o procedimento executado no servidor, incluindo a seleção dos clientes, o ajuste ponderado pelos riscos individuais e a combinação do modelo agregado com o modelo global anterior.

A estratégia tradicional de ponderar pela quantidade de amostras assume que clientes com maior volume de dados devem ter maior influência na formação do modelo global. No entanto, essa premissa pode ser falha: um cliente com muitos dados, mas concentrados em poucas classes ou com baixa diversidade, pode não aprender padrões relevantes ou generalizáveis. Dessa forma, o maior número de amostras não necessariamente se traduz em melhor desempenho local, o que compromete a efetividade da agregação baseada apenas no volume de dados. Com nossa estratégia, a agregação passa a privilegiar clientes que apresentam menor risco de erro nas predições, atribuindo a eles maior peso na formação do modelo global. Já contribuições associadas a maiores riscos de erro têm sua influência reduzida, tornando o processo de agregação mais alinhado à qualidade preditiva dos modelos locais.

$$\tilde{\theta}^t = \frac{1}{|S_t|} \sum_{k \in S_t} \left((1 - risk_k) \cdot \theta_k^t \right) \tag{1}$$

Na Equação (1), que expressa formalmente a estratégia de ponderação proposta, cada modelo local θ_k^t é ponderado por um fator inversamente proporcional ao risco estimado do cliente k. A média resultante $\tilde{\theta}^t$ representa a versão ajustada do modelo global, que incorpora não apenas os parâmetros locais, mas também a avaliação relativa

Algorithm 3: Treinamento Federado com Agregação Sensível ao Risco

Servidor executa

Input: Número total de clientes K, rodadas T, quantidade C de clientes por rodada, α proporção do modelo agregado, β proporção do modelo global anterior
Output: Modelo global θ_G

1 Inicializar θ_G^0 // Modelo global inicial
2 **for** *rodada* $t = 1$ **to** T **do**
3 Selecionar subconjunto $S_t \subseteq \{1, ..., K\}$ de C clientes aleatórios
4 **for** *cada cliente* $k \in S_t$ **do in parallel**
5 $\theta_k^t, risk_k \leftarrow$ **ClientTrain**(k, θ_G^{t-1}) // Chamadas assíncronas
6 **Agregação Global:**
7 $\tilde{\theta}^t \leftarrow \frac{1}{|S_t|} \sum_{k \in S_t} (1 - risk_k) \cdot \theta_k^t$ // Ponderação pelo risco
8 $\theta_G^t \leftarrow \alpha \cdot \tilde{\theta}^t + \beta \cdot \theta_G^{t-1}$ // Combinação do modelo agregado com o anterior
9 **return** θ_G^T
10 **servidor_calcula_riscos**(mse_vector)
11 // calcula riscos dos clientes conforme **Seção 3.1.1**
12 **return** $risks$

Em vermelho destacamos as alterações propostas nos algoritmos baseados em FedAvg, para produzir os resultados alcançados nesse estudo.

Algorithm 4: ClientTrain(k, θ)

Cliente executa

Input: Cliente k, parâmetros globais θ, épocas E, batch size B, taxa de aprendizado η
Output: Parâmetros locais θ_k, risco $risk_k$

1 $\theta_k \leftarrow \theta$ // copia o modelo global para o cliente
2 $\mathcal{R}_k \leftarrow \emptyset$ // riscos por batch
3 **for** *época* $e = 1$ **to** E **do**
4 **for** *batch* $b = 1$ **to** B **do**
5 Obter Y_k^b
6 Calcular saídas: $outputs \leftarrow net(X_k^b)$
7 $P_k^b \leftarrow \arg\max(outputs)$
8 $mse\_vector_k^b = [(p_i - y_i)^2 \mid \forall i \in b]$
9 $risks \leftarrow$ servidor_calcula_risks$(mse_vector_k^b)$
10 $risk_k^b \leftarrow risks[k]$
11 $\mathcal{R}_k \leftarrow \mathcal{R}_k \cup \{risk_k^b\}$
12 Otimizar modelo local θ_k (backpropagation)
13 **if** $|\mathcal{R}_k| > 0$ **then**
14 $risk_k \leftarrow$ mediana$(\mathcal{R}_k)$
15 **else**
16 $risk_k \leftarrow 0$
17 **return** $\theta_k, risk_k$

Notação: X_k^b: features do cliente k no batch b, P_k^b: predições, Y_k^b: rótulos verdadeiros, $mse\_vector_k^b$: erros quadráticos, $risk_k^b$: risco por batch.

de desempenho dos clientes. Essa equação substitui diretamente o cálculo da média ponderada tradicional do *FedAvg* e foi implementada na linha 7 do Algoritmo 3.

Para que o fator de risco seja calculado de forma personalizada para cada cliente k, propomos uma modificação no processo de treinamento local, apresentada no Algoritmo 4. Como cada cliente já possui os rótulos verdadeiros dos pares `<consulta, documento>`, ele pode calcular localmente os erros de predição — utilizando, por exemplo, o erro quadrático médio (MSE)[2] — linha 8 do Algoritmo 4 — e enviá-los ao servidor após cada *batch*. Com os vetores de erros de todos os clientes, o servidor executa o cálculo do fator de Sensibilidade ao Risco, que passa então a compor a ponderação na etapa de agregação global. Essa alteração no fluxo tradicional do FL é uma das principais contribuições deste trabalho, pois permite ao servidor ajustar o peso das contribuições de cada cliente de forma proporcional ao seu fator de risco de erro nas predições locais.

3.1.1. Cálculo do Fator de Risco

Inspirados na estratégia de otimização sensível ao risco de [Rodrigues et al. 2022, Rodrigues et al. 2025], propomos uma adaptação para regular a agregação de parâmetros em nossa abordagem, modelando o cálculo do fator de Sensibilidade ao Risco assim: dado um conjunto $S_t \subseteq \{1, 2, ..., K\}$, clientes aleatoriamente selecionados em uma rodada t, com $|S_t| < K$. Para cada cliente $k \in S_t$, consideramos um lote (*batch*) local de b instâncias (pares consulta-documento). Cada cliente k calcula localmente o vetor de erros quadráticos médios sobre o lote, gerando uma entrada da matriz $M \in \mathbb{R}^{|S_t| \times b}$, $M_{kj} = (p_{kj} - y_{kj})^2$, onde p_{kj} é a predição do cliente k para a instância j; e y_{kj} é o rótulo verdadeiro de j.

A matriz M contém os erros observados de cada cliente sobre suas instâncias

[2]Nesse contexto, o erro quadrático médio é definido como $MSE(P_i^b, Y_i^b) = \left[(p_i - y_i)^2 \mid i \in \text{batch } b\right]$, onde p_i representa a predição e y_i o rótulo verdadeiro.

locais. Seguindo [Rodrigues et al. 2025], definimos o erro esperado e_{kj} para cada célula da matriz como $e_{kj} = \frac{S_k \cdot T_j}{N}$, onde $S_k = \sum_{j=1}^{b} M_{kj}$ é o somatório dos erros do cliente k para todas as instâncias; $T_j = \sum_{k=1}^{C} M_{kj}$ é o somatório dos erros da instância j entre todos os clientes; e $N = \sum_{k=1}^{C} \sum_{j=1}^{b} M_{kj}$ é o somatório total da matriz.

O desvio padrão do erro observado em relação ao esperado é dado por $z_{kj} = \frac{M_{kj} - e_{kj}}{\sqrt{e_{kj}}}$. Com esses desvios, define-se a métrica a ser usada para calcular o risco geométrico do cliente k (ou seja, o grau com que os valores z_{kj} daquele cliente se desviam negativamente da média global), chamada $ZRisk(k)$, como:

$$ZRisk(k) = \sum_{j \in J^-} z_{kj} + (1 + \alpha) \sum_{j \in J^+} z_{kj} \tag{2}$$

onde: $J^+ = \{j \mid z_{kj} \geq 0\}$ são os desvios positivos (erros maiores do que o esperado) e $J^- = \{j \mid z_{kj} < 0\}$ são os desvios negativos. A partir de $ZRisk(k)$, é calculado o risco final do cliente usando a função GeoRisk:

$$GeoRisk(k) = \sqrt{\left(\frac{1}{b} \sum_{j=1}^{b} M_{kj}\right) \cdot \Phi\left(\frac{ZRisk(k)}{b}\right)}, \tag{3}$$

onde $\Phi(\cdot)$ representa a função de distribuição acumulada da normal padrão.

A métrica ZRisk captura o risco estatístico de um sistema ao avaliar, com base em múltiplos sistemas de referência, a frequência e a gravidade dos desvios negativos de desempenho por consulta, sendo sensível à variância e à forma da distribuição dos resultados. No entanto, por ser independente da média de desempenho global, ZRisk não permite uma comparação direta entre sistemas com níveis médios de efetividade distintos. Para resolver essa limitação, foi proposta a métrica GeoRisk [Dincer et al. 2016], que combina o ZRisk com a média de desempenho por meio de uma média geométrica, permitindo uma avaliação comparativa mais completa entre sistemas em termos de efetividade e propensão ao risco.

Por fim, o fator de sensibilidade ao risco do cliente k é obtido comparando o seu risco com o risco de um sistema ideal Z, definido como a média dos erros por instância:

$$Risk(Z, P_k) = GeoRisk(Z) - GeoRisk(k) \tag{4}$$

onde: $Z = \left(\frac{1}{C} \sum_{k=1}^{C} M_{kj}\right)_{j=1}^{b}$ é o sistema ideal (média por coluna) e P_k é a linha k da matriz M, com os erros individuais do cliente k. Este valor final $Risk(Z, P_k)$ é o fator de Sensibilidade ao Risco utilizado para ponderar a contribuição do cliente na agregação global, que em nosso trabalho aplicamos para a tarefa de FL2R.

3.2. Adição Incremental do Modelo Global na Agregação:

A abordagem proposta busca aproveitar os valores dos parâmetros do modelo global da rodada anterior θ_G^t, somando-os aos parâmetros do modelo resultante da agregação por média, conforme expressa a Equação 5.

$$\theta_G^t = \alpha \cdot \tilde{\theta}^t + \beta \cdot \theta_G^{t-1} \tag{5}$$

Aqui, o modelo global da rodada t, denotado por θ_G^t, é obtido por meio de uma combinação linear entre o modelo agregado atual com risco $\tilde{\theta}^t$ (baseado na média dos parâmetros e no fator risco dos clientes) e o modelo global da rodada anterior θ_G^{t-1}. Os coeficientes α e β controlam, respectivamente, a influência do modelo atual $\tilde{\theta}^t$ e do modelo his-

tórico θ_G^{t-1} no processo de agregação. Esta estratégia está implementada na linha 8 do Algoritmo 3. Com essa proposta, espera-se que a reutilização do modelo anterior atue como uma memória estável ajudando acelerar a convergência do modelo ao longo das rodadas.

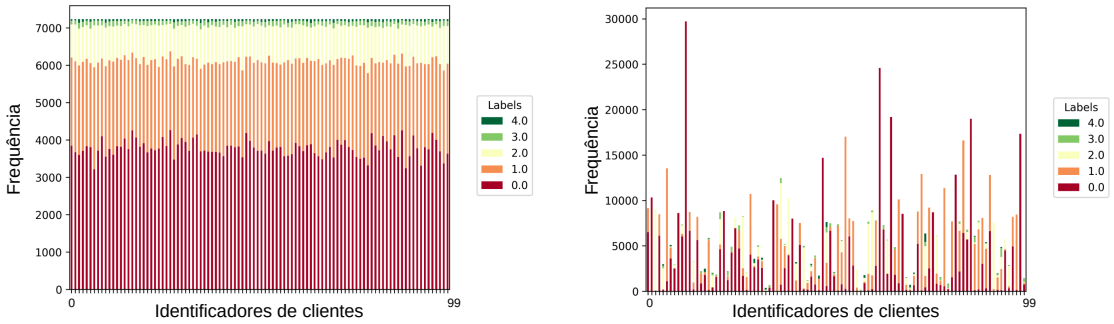
4. Resultados

Nesta seção, apresentamos a configuração experimental adotada e os resultados combinando sensibilidade ao risco e reaproveitamento de parâmetros históricos – respondendo às perguntas de pesquisa propostas. A análise está organizada em quatro partes: descrição da configuração experimental, comparação com *baselines*, avaliação da redução da variabilidade entre os clientes e a avaliação do impacto individual de cada componente da estratégia **FedRisk** no resultado final.

4.1. Configuração Experimental

Dados e Biblioteca: Os experimentos utilizaram o dataset MSLR-WEB10K [Qin and Liu 2013], um benchmark amplamente adotado em aprendizado para ranqueamento [Köppel et al. 2019], com cerca de 10.000 consultas associadas a múltiplos documentos representados por vetores de 135 atributos numéricos extraídos de páginas da web. Para simular o ambiente federado, foi usada a biblioteca Flower [Beutel et al. 2020], que permite experimentos escaláveis de aprendizado federado.

Distribuição dos Dados IID e não-IID: As distribuições IID e não-IID utilizadas nos experimentos foram geradas com base na amostragem Dirichlet, conforme implementado na biblioteca Flower. A Figura 3 ilustra essas distribuições dos dados entre 100 clientes ao todo, e dentre estes, são selecionados aleatoriamente 10, a cada rodada de treinamento. No cenário IID (Figura 3-a), a distribuição das classes de rótulos (0 a 4) é aproximadamente igual entre todos os clientes, tanto em quantidade total de exemplos quanto na proporção de cada classe. Já no cenário não-IID (Figura 3-b), observa-se uma forte variação: alguns clientes concentram exemplos de apenas uma ou poucas classes, enquanto outros possuem amostras de classes distintas em proporções muito desbalanceadas.



(a) - Dados IID (b) não-IID

Figura 3. Distribuições IID e não-IID de dados entre todos os clientes (100 ao todo).

4.2. Comparação com *Baselines*

Para avaliar o desempenho dos modelos nas tarefas de ranqueamento, foram utilizadas as métricas nDCG@k (*Normalized Discounted Cumulative Gain*) [Järvelin and Kekäläinen 2002, Wang et al. 2013] e MRR (*Mean Reciprocal Rank*) [Voorhees et al. 1999]. A métrica nDCG@k avalia a qualidade do ranqueamento considerando a relevância e a posição dos itens, dando maior peso aos relevantes nas primeiras posições, sendo útil quando há diferentes níveis de relevância. Já a MRR

#	Estratégia de Agregação	Clientes por rodada	nDCG (x100)			MRR (x100)		
			@1	@5	@10	@1	@5	@10
1	**FedRisk**	10	27.0 (0.8)	**31.8 (1.0)**	**37.3 (1.2)**	**48.0 (1.5)**	**63.9 (1.3)**	**64.9 (1.3)**
2	FedAvgM	10	**27.4 (11.3)**	29.8 (10.0)	34.2 (8.5)	46.4 (16.2)	61.6 (14.0)	62.8 (13.2)
3	FedProx (μ=0.9)	10	24.1 (6.0)	27.5 (4.1)	32.8 (3.6)	41.8 (8.2)	58.2 (7.5)	59.5 (7.1)
4	FedOpt	10	24.1 (9.5)	27.5 (7.4)	32.7 (7.0)	42.6 (14.3)	58.5 (11.6)	59.9 (11.0)
5	FedTrimmedAvg	10	20.8 (5.3)	26.2 (5.1)	32.5 (4.6)	37.3 (9.4)	54.9 (8.4)	56.4 (7.9)
6	FedAvg	10	23.0 (8.1)	26.2 (7.0)	31.6 (6.7)	39.8 (11.7)	56.0 (11.2)	57.5 (10.5)
7	FedAdam	10	17.8 (3.3)	22.8 (3.5)	28.8 (3.5)	32.9 (5.5)	50.5 (5.9)	52.3 (5.4)
8	FedMedian	10	17.9 (3.0)	22.6 (3.4)	28.3 (3.8)	33.1 (4.9)	50.3 (5.6)	52.0 (5.2)
9	FedYogi	10	17.0 (1.1)	21.3 (1.2)	27.1 (1.2)	31.0 (1.7)	47.8 (2.0)	49.8 (1.9)
10	FedAdagrad	10	16.7 (2.0)	20.9 (1.4)	26.7 (1.6)	30.8 (2.7)	47.3 (2.6)	49.3 (2.5)
11	Centralizado	–	32.4 (3.9)	35.0 (3.2)	35.9 (2.2)	58.5 (5.7)	73.2 (4.1)	73.7 (4.0)

Tabela 1. Comparação do desempenho entre modelos em cenários com dados não-IID (linhas 1-10) na MSLR-WEB10K; 5-folds com intervalo de confiança de 95%; 100 rodadas de treinamento.

mede a posição do primeiro item relevante, indicando a capacidade do sistema em trazer boas recomendações no topo da lista. Em ambas, valores mais altos indicam melhor desempenho no ranqueamento.

Comparamos o desempenho do **FedRisk** com os de métodos de agregação *baselines* da literatura. Seguindo o protocolo experimental baseado na técnica de validação cruzada, dividimos os dados em 5 *folds* e avaliamos o desempenho médio do modelo global, nos *folds* de teste, para cada uma das estratégias de agregação. A Tabela 1 apresenta uma coluna com o número de clientes por rodada de treinamento (10 selecionados aleatoriamente de 100 possibilidades) e, nas colunas seguintes, os valores de desempenhos **médios**, ao final das 100 rodadas, nas métricas nDCG(@1,@5,@10) e MRR(@1,@5,@10), juntamente com seus respectivos intervalos de confiança (valores entre parênteses)[3].

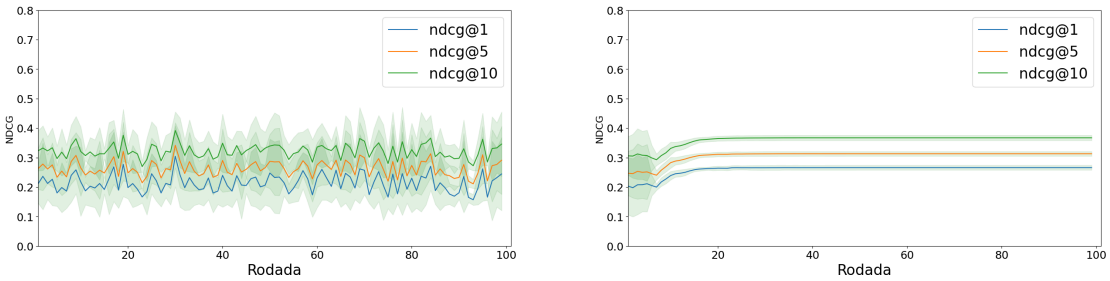
Como mostrado na Tabela 1, em 5 das 6 métricas avaliadas, o nosso método, **FedRisk**, superou todas as estratégias de agregação consideradas nos experimentos nos cenários com dados não-IID. Em nDCG@5, por exemplo, o **FedRisk** alcançou **31.8**, enquanto o FedProx, que foi o melhor *baseline* entre os comparados (considerando efetividade e estabilidade), atingiu 27.5, um ganho de **15.6%**. Em relação ao FedAdagrad, na mesma métrica, o ganho foi ainda maior, **52.1%**.

Por fim, vale ressaltar também que o **FedRisk** foi o modelo que mais se aproximou da performance do modelo centralizado (última linha da Tabela 1), principalmente em relação ao nDCG@5 e nDCG@10, inclusive com um valor absoluto superior ao centralizado nesta última métrica, se tornando um resultado altamente relevante, ainda mais considerando o cenário não-IID tratado pelo **FedRisk**. Esse comportamento pode ser atribuído à regularização mais ponderada e suave nos parâmetros, evitando alterações bruscas no modelo global. Comportamento obtido com a agregação por sensibilidade ao risco e ao uso do histórico de modelos.

4.3. Análise da Variabilidade e Convergência

No aprendizado federado, a natureza não-IID dos dados acentua a importância de analisar a variabilidade dos resultados, devido às flutuações entre rodadas e *folds*. Como observa [Spiegelhalter 2024], métricas com boas médias podem mascarar alta incerteza e baixa consistência quando há grande variabilidade, comprometendo a confiabilidade. Um exemplo é o FedAvgM na Tabela 1: apesar dos bons valores médios em NDCG@1, o intervalo de confiança (IC) elevado revela instabilidade entre particionamentos e baixa

[3]Utilizamos as métricas nDCG e MRR nas posições 1, 5 e 10, que são amplamente reconhecidas na literatura para medir a qualidade de sistemas de ranqueamento [Järvelin and Kekäläinen 2002, Voorhees 1999].



(a) - Agregação com FedProx. **(b) - Agregação com FedRisk**

Figura 4. **Aplicação de FedRisk em proporções máximas ($\alpha = 1.0, \beta = 1.0$) do modelo com sensibilidade ao risco e do modelo anterior. | 10 clientes por rodada.**

consistência.

Utilizamos os IC sobre os resultados entre os *folds* para cada rodada, para mostrar a variabilidade dos modelos FedProx e **FedRisk** [Brownlee 2018]. As sombras das curvas nas Figuras 4-a e 4-b correspondem às amplitudes dos intervalos de confiança. Como podemos observar na Figura 4-b, o **FedRisk** promove uma redução significativa na variabilidade dos resultados dos cinco *folds*, tornando os intervalos de confiança visivelmente mais estreitos em todas as curvas. A Figura 4-a, por sua vez, mostra que o FedProx apresenta alta variabilidade dos resultados. Vale ressaltar também a rápida convergência do **FedRisk** que com poucas rodadas (< 20) atinge o máximo de performance e estabilidade, enquanto o FedProx não parece convergir.

Esses resultados reforçam que a combinação das estratégias promoveu ganhos consistentes tanto em desempenho quanto em estabilidade, posicionando o **FedRisk** como a abordagem mais robusta entre as avaliadas, respondendo positivamente às QP1 e QP2.

4.4. Análise da Contribuição dos Componentes da Solução

Investigamos como o desempenho do **FedRisk** é influenciado pelos pesos de seus dois componentes: o modelo médio dos clientes na rodada atual ($\tilde{\theta}_t$) e o modelo global da rodada anterior (θ_G^{t-1}), conforme a Equação 5. Para isso, variamos os pesos α e β em diferentes proporções, explorando diferentes cenários de influência dos componentes.

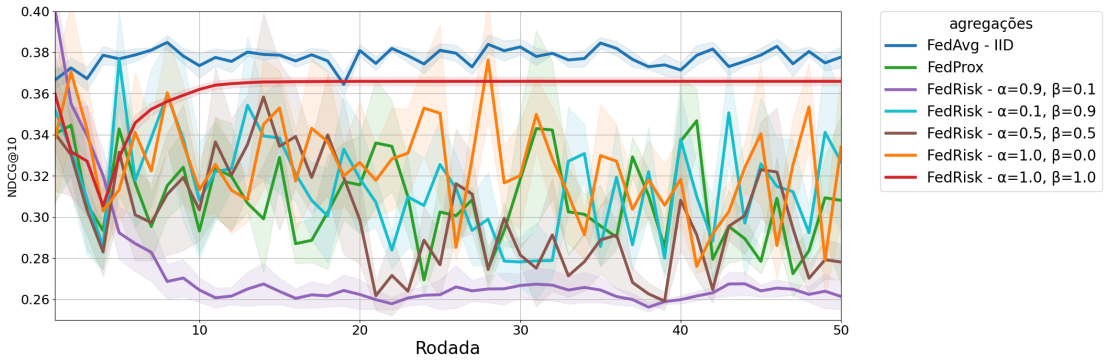


Figura 5. **Comparação entre diferentes proporções de cada componente na composição do modelo global, usando 5 clientes por rodada.**

A Figura 5 apresenta os resultados da análise, comparando as configurações em termos da métrica nDCG@10 ao longo das rodadas, incluindo dois modelos de referência: o FedAvg com dados IID e o *baseline* FedProx. Note que o cenário de simplesmente "desligar" o componente $\tilde{\theta}^t$, fazendo $\alpha = 0$, não foi considerado pois é equivalente a desconsiderar o aprendizado e manter o mesmo modelo global por todas as rodadas.

Nossa análise de componentes considerou as seguintes configurações:

- ($\alpha = 0.9$, $\beta = 0.1$): Maior peso do modelo agregado e menor peso do modelo anterior privilegia o aprendizado dos clientes. A curva indica que só a agregação por risco não estabiliza o desempenho do modelo global nem reduz a variabilidade.
- ($\alpha = 0.1$, $\beta = 0.9$): Maior contribuição do modelo anterior e menor do agregado enfraquece o aprendizado dos clientes. A curva indica que a memória histórica somente não melhora a efetividade do modelo global nem reduz a variabilidade.
- ($\alpha = 0.5$, $\beta = 0.5$): Mesmo peso para as contribuições do modelo agregado e do modelo anterior. A curva sugere que o equilíbrio entre memória e aprendizado atual sem considerar o máximo de importância para os dois fatores ainda não é suficiente para alcançar os melhores resultados.
- ($\alpha = 1.0$, $\beta = 0.0$): Peso total do modelo baseado no risco e nenhum da memória prioriza apenas o aprendizado dos clientes, mas a curva mostra que isso não melhora o desempenho nem reduz a variabilidade.
- ($\alpha = 1.0$, $\beta = 1.0$): Peso total aos dois componentes, produzindo uma curva altamente estável, *com valores próximos ao do FedAvg com dados IID e muito próximo ao ideal*. **Esta configuração apresenta o melhor compromisso entre eficácia e consistência.**

Os resultados indicam que técnicas isoladas, como a ponderação baseada em risco ou a incorporação do modelo histórico, contribuem de forma limitada para a estabilidade global. No entanto, a combinação de ambas mostra-se significativamente mais eficaz, reforçando a hipótese de que o efeito desejado decorre da sinergia entre essas estratégias. O melhor cenário é obtido quando ambas têm peso igual a um (valor máximo).

Cada experimento, com 5 *folds*, 100 clientes, 100 rodadas e 5 épocas locais, levou cerca de 12 horas em uma estação com processador AMD Ryzen Threadripper PRO 5955WX (16 núcleos, 32 threads), 512 GB de RAM e GPU NVIDIA RTX A6000 (48 GB). Apesar das extensões do FedRisk, a complexidade global segue linear no número de clientes, ou seja $\mathcal{O}(n)$.

5. Conclusão e Trabalhos Futuros

Este trabalho propôs a ponderação por sensibilidade ao risco e a reutilização incremental da memória do modelo global como estratégias para mitigar os efeitos de dados não-IID em FL2R. Para a QP1, introduzimos um diagnóstico local de erro, no qual clientes calculam vetores de MSE que permitem ao servidor estimar um fator de sensibilidade ao risco. A QP2 examinou a reutilização da memória anterior do modelo, aliada à sensibilidade ao risco, com resultados expressivos. Como resultado, o método proposto alcança um ganho relativo de **15.6%** em NDCG, e iguala ao desempenho do modelo centralizado. Uma limitação do FedRisk refere-se ao *overhead* computacional introduzido pelo cálculo das métricas de sensibilidade ao risco. Esse processamento adicional aumenta o custo das iterações locais e pode impactar a eficiência do algoritmo, especialmente em cenários com grande número de clientes ou rounds. Trabalhos futuros podem explorar otimizações nesse cálculo, visando reduzir o *overhead* sem comprometer a robustez do método.

Agradecimentos

Agradecemos ao apoio do CNPq(443011/2023-0 e 03184/2021-5), Capes, Fapemig, Fapesp, AWS e Instituto Nacional de Ciência e Tecnologia em Inteligência Artificial Responsável para Linguística Computacional, Tratamento e Disseminação de Informação (INCT-TILD-IAR; 408490/2024-1).

Referências

Ads, Z. et al. (2024). Risk-aware accelerated federated learning over heterogeneous wireless networks. *arXiv preprint arXiv:2401.09267*.

Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., Sani, L., Kwing, H. L., Parcollet, T., Gusmão, P. P. d., and Lane, N. D. (2020). Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer.

Brownlee, J. (2018). *Statistical Methods for Machine Learning*. Machine Learning Mastery.

Chen, S. et al. (2021). Risk-aware federated learning in crowdsensing systems. *arXiv preprint arXiv:2101.01266*.

Dincer, B., Zhu, Y., Craswell, N., and Zhang, M. (2016). Risk-sensitive evaluation and learning to rank using multiple baselines. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 483–492.

Divi, S., Lin, Y.-S., Farrukh, H., and Celik, Z. B. (2021). New metrics to evaluate the performance and fairness of personalized federated learning.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Hejazinia, M. et al. (2022). Fel: High capacity learning for recommendation and ranking via federated ensemble learning. *arXiv preprint arXiv:2206.03852*.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Jeong, J., Kim, H., Park, J., Lee, S., and Yoon, D. N. (2022). Fedcc: Boosting robustness of federated learning against model poisoning attacks. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 861–875. ACM.

Jiang, J. C., Kantarci, B., Oktug, S., and Soyata, T. (2020). Federated learning in smart city sensing: Challenges and opportunities. *Sensors*, 20(21):6230.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S. U., and Suresh, A. T. (2020). Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning (ICML)*.

Köppel, M., Segner, A., Wagener, M., Pensel, L., Karwath, A., and Kramer, S. (2019). Pairwise learning to rank by neural networks revisited: Reconstruction, theoretical analysis and practical performance. *arXiv preprint arXiv:1909.02768*.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2020). Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, pages 429–450.

Liu, S., Celik, E., and Widmer, J. (2021). Label-aware aggregation for improved federated learning. In *Proceedings of the 2021 20th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–13. IEEE.

Neto, H. N. C., Mattos, D. M. F., and Fernandes, N. C. (2020). Privacidade do usuário em aprendizado colaborativo: Federated learning, da teoria à prática. In *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSEG)*.

Qin, T. and Liu, T. (2013). Introducing LETOR 4.0 datasets. *CoRR*, abs/1306.2597.

Rodrigues, P. H. S., de Sousa, D. X., França, C., Rabbi, G., Couto Rosa, T., and Gonçalves, M. A. (2025). Risk-sensitive optimization of neural deep learning ranking models with applications in ad-hoc retrieval and recommender systems. *Information Processing & Management*, 62(4):104126.

Rodrigues, P. H. S., Xavier Sousa, D., Couto Rosa, T., and Gonçalves, M. A. (2022). Risk-sensitive deep neural learning to rank. In *ACM SIGIR Conference*, SIGIR '22, page 803–813.

Spiegelhalter, D. (2024). *The Art of Uncertainty: How to Navigate Chance, Ignorance, Risk and Luck*. Pelican Books.

Tong, Y. et al. (2021). An efficient approach for cross-silo federated learning to rank. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*.

Voorhees, E. M. (1999). The trec-8 question answering track report. In *Proceedings of the Eighth Text Retrieval Conference (TREC-8)*. National Institute of Standards and Technology (NIST).

Voorhees, E. M. et al. (1999). The trec-8 question answering track report. In *TREC*, volume 8.

Wang, J. and Liu, M. (2020). Tackling the objective inconsistency problem in heterogeneous federated optimization. In *NeurIPS*.

Wang, L., Bennett, P. N., and Collins-Thompson, K. (2012). Robust ranking models via risk-sensitive optimization. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, page 761–770, New York, NY, USA. Association for Computing Machinery.

Wang, S. and Zuccon, G. (2022). Is non-iid data a threat in federated online learning to rank? In *ACM SIGIR Conference*, SIGIR '22, page 2801–2813.

Wang, Y., Li, T.-Y., Wang, D., and Zhu, M. (2013). A theoretical analysis of ndcg type ranking measures. *Journal of Machine Learning Research*, 14:25–54.

Zhao, S. et al. (2024). Federated risk-aware learning with central sensitivity estimation. *arXiv preprint arXiv:2502.17694*.