UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA TELEINFORMÁTICA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMATICA

MESTRADO ACADÊMICO EM ENGENHARIA DE TELEINFORMÁTICA

ERNESTO GURGEL VALENTE NETO

A NEW METHODOLOGY FOR EDGE INTELLIGENCE DATA QUALITY EVALUATION IN IID AND NON-IID DATASETS IN FEDERATED LEARNING

FORTALEZA

2024

ERNESTO GURGEL VALENTE NETO

A NEW METHODOLOGY FOR EDGE INTELLIGENCE DATA QUALITY EVALUATION IN IID AND NON-IID DATASETS IN FEDERATED LEARNING

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia de TELEINFORMÁTICA do Programa de Pós-Graduação em ENGENHARIA de Teleinformatica do Centro de TECNOLOGIA da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia de Teleinformatica. Área de Concentração: Sinais e Sistemas - Reconhecimento de Padrões e Sistemas Dinâmicos

Orientador: Prof. Dr. Julio César Santos dos Anjos

Coorientador: Dr. Solon Alves Peixoto

FORTALEZA

2024

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

V249n Valente Neto, Ernesto Gurgel..
A new methodology for edge intelligence data quality evaluation in iid and non-iid datasets in federated learning / Ernesto Gurgel. Valente Neto. – 2025.
117 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2025.
Orientação: Prof. Dr. Julio César Santos dos Anjos.
Coorientação: Prof. Dr. Solon Alves Peixoto.

1. Data Quality. 2. Deep Learning. 3. Federated Learning. 4. Internet of Things. I. Título.
CDD 621.38

ERNESTO GURGEL VALENTE NETO

A NEW METHODOLOGY FOR EDGE INTELLIGENCE DATA QUALITY EVALUATION IN IID AND NON-IID DATASETS IN FEDERATED LEARNING

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia de TELEINFORMÁTICA do Programa de Pós-Graduação em ENGENHARIA de Teleinformatica do Centro de TECNOLOGIA da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia de Teleinformatica. Área de Concentração: Sinais e Sistemas - Reconhecimento de Padrões e Sistemas Dinâmicos

Aprovada em: 21 de Fevereiro de 2025

BANCA EXAMINADORA

Prof. Dr. Julio César Santos dos Anjos (Orientador)
Universidade Federal do Ceará (UFC)

Dr. Solon Alves Peixoto (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Guilherme Barreto
Universidade Federal do Ceará (UFC)

Prof. Dr. Victor Albuquerque
Universidade Federal do Ceará (UFC)

Prof. Dr. Valderi Leithardt
Iscte - Instituto Universitário de Lisboa, Portugal

AGRADECIMENTOS

À Instituição CAPES, pelo apoio financeiro com a manutenção da bolsa de auxílio.

Minha mãe e minha irmã me apoiaram e ficaram ao meu lado durante minha dedicação aos estudos e sempre acreditaram em mim. Ao meu amigo e professor de inglês, Felipe Costa.

Aos meus antigos colegas de trabalho, Aloísio Queiroz, Edson Martins e Isaac Barbosa, os quais me incentivaram a fazer o mestrado.

Quero expressar minha profunda gratidão ao Prof. Dr. Julio César Santos dos Anjos, que me transformou, ensinando e incentivando com profundo zelo e paciência, focando no meu desenvolvimento e crescimento.

Dr. Solon Alves Peixoto, que foi um conselheiro, colega e amigo, apontando muitas soluções e novas ideias. Ao Prof. Dr. Guilherme Barreto, que construiu minha base e fundamentação teórica e científica como um acadêmico e questionador. Ao Prof. Dr. Victor Albuquerque, que me auxiliou na dissertação. Aos demais professores, por todo o conhecimento e sabedoria que compartilharam comigo e também pelo papel significativo no meu desenvolvimento.

Ao Prof. Dr. Tobias Rafael Fernandes Neto, coordenador do Laboratório de Sistemas Motrizes (LAMOTRIZ) onde este *template* foi desenvolvido. Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC). À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo financiamento da pesquisa de mestrado via bolsa de estudos.

“Deus não joga dados com o Universo.”

(Albert Einstein)

RESUMO

A geração maciça de dados a partir de dispositivos da Internet das Coisas (IoT) aumenta a procura de uma análise de dados eficiente para extrair informações significativas. O Aprendizado Federado (FL) permite que os dispositivos IoT colaborem em modelos de Inteligência Artificial (IA), preservando a privacidade dos dados. No entanto, a seleção de dados de alta qualidade para formação continua a ser um desafio crítico em ambientes de FL com dados não independentes e identicamente distribuídos (non-iid). Dados de baixa qualidade introduzem erros, atrasam a convergência e aumentam os custos computacionais. Para enfrentar esses desafios, este estudo propõe um algoritmo para análise da qualidade dos dados em ambientes centralizados e de FL. O algoritmo proposto reduz os custos computacionais, elimina o processamento desnecessário de dados e acelera a convergência do modelo de IA. As experiências utilizaram os conjuntos de dados MNIST, Fashion-MNIST, CIFAR-10 e CIFAR-100, e a avaliação do desempenho baseou-se nas principais métricas da literatura, como a *Accuracy*, *Recall*, *F1-score* e *Precision*. Os resultados mostram, no melhor dos casos, reduções no tempo de execução de até 56,49%, com uma perda de acurácia de cerca de 0,50%.

Palavras-chave: qualidade de dados; aprendizado profundo; aprendizado federado; internet of things (iot).

ABSTRACT

Massive data generation from Internet of Things (IoT) devices increases the demand for efficient data analysis to extract meaningful insights. Federated Learning (FL) allows IoT devices to collaborate in Artificial Intelligence (AI) training models while preserving data privacy. However, selecting high-quality data for training remains a critical challenge in FL environments with non-independent and identified distributed (non-iid) data. Poor-quality data introduce errors, delay convergence, and increase computational costs. This study develops a data quality analysis algorithm for FL and centralized environments to address these challenges. The proposed algorithm reduces computational costs, eliminates unnecessary data processing, and accelerates AI model convergence. The experiments used the MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100 datasets, and performance evaluation was based on main literature metrics like accuracy, recall, F1 score, and precision. Results show the best case execution time reductions of up to 56.49%, with an accuracy loss of around 0,50%.

Keywords: qualidade de dados; aprendizado profundo; aprendizado federado; internet of things (iot).

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

SUMÁRIO

1 INTRODUÇÃO

No campo de *Machine Learning* (ML), a qualidade, quantidade e relevância das informações estão intimamente ligadas ao uso de recursos computacionais. Essa relação é evidente em dois cenários principais: centralizado e descentralizado. Por exemplo, em um cenário centralizado, os dados são agregados em um único local, o que envolve custos de transferência de dados, computação e energia. Em contraste, em um cenário descentralizado, os dados são processados em dispositivos com recursos limitados, menor capacidade de armazenamento e largura de banda de comunicação restrita.

Como resultado, o grande número de dispositivos e o alto volume de dados exigem processamento e análise em alta velocidade para gerar *insights* valiosos, ao mesmo tempo em que cumprem os requisitos legais de proteção de dados privados e confidenciais (Vailshery, 2023). Esses aspectos envolvem desafios técnicos e de design de sistemas (Huynh; Nippa; Aichner, 2023). Consequentemente, a qualidade dos dados é crucial em aprendizado de máquina (Munappy et al., 2022), especialmente em áreas com limitações de dados, como a descoberta de medicamentos, que frequentemente opera com conjuntos de dados pequenos e restritos (Tilborg et al., 2024).

Contudo, neste cenário, a privacidade e a segurança dos dados tornam-se questões centrais. Assim, estudos focados em segurança da informação e sistemas distribuídos sugerem uma estratégia para superar esses desafios, adotando um novo conceito conhecido como FL (Khan et al., 2921). Nesse processo, os dados permanecem na borda; os dispositivos locais recebem redes neurais para treinamento e, após esse processo, compartilham os pesos dos neurônios com o servidor, onde ocorre a agregação sem a necessidade de compartilhar dados sensíveis.

No entanto, essa abordagem introduz novos desafios, incluindo o manuseio de dados Independent and Identically Distributed (iid) iid e Non-Independent and Identically Distributed (non-iid) (Sun et al., 2020). Os desafios mais proeminentes incluem o alto custo de comunicação causado pela dispersão dos dados e a complexidade de selecionar clientes para treinamento, o que pode afetar negativamente a eficiência do aprendizado. Além disso, estabelecer incentivos para colaboração e gerenciar a heterogeneidade dos dispositivos representa obstáculos significativos (Abdulrahman et al., 2020; Ma et al., 2022).

Além dos desafios relacionados à qualidade dos dados, a heterogeneidade dos dados em FL apresenta obstáculos significativos, particularmente em relação às distribuições non-iid,

que afetam diretamente a precisão e a convergência dos modelos de FL (Anjos, et al., 2023). A presença de dados non-iid, frequentemente resultante de dependências temporais, também apresenta riscos de viés e desempenho inconsistente nos modelos treinados (Bassiouni; Chakraborty; Sallam; Hussain, 2024) . Esses desafios impactam a seleção de clientes e o desenvolvimento de métodos eficazes de fusão de modelos (Wen et al., 2023).

Dessa forma, estudos têm investigado a influência da entropia na qualidade dos dados (Li; Chao; Ercisli, 2022), abordando as ambiguidades (Bustincio; Souza; Costa; Bittencourt, 2023) e minimizando os custos computacionais (Orlandi et al., 2023). Assim, esses trabalhos buscam reduzir os atrasos de comunicação e diminuir o tempo de execução dos algoritmos.

Motivados por esses estudos sobre propriedades de entropia, analisamos as distribuições de imagens. Observamos que as amostras apresentaram um comportamento de cauda longa com valores extremos, o que contribuiu para a dispersão em torno da média. O algoritmo proposto, EnBaSe, lida com a dispersão de valores anômalos e a distorção das caudas em dados non-iid. Além disso, a abordagem de avaliação da qualidade dos dados dentro da classe não altera a dispersão da distribuição e mantém a representatividade do domínio da amostra. O impacto dessa abordagem é a manutenção das propriedades da distribuição estocástica e a redução de cálculos não significativos nos nós, o que, como resultado, possibilita o aumento do desempenho do FL em computação de borda e economiza custos de energia.

Embora a proposta seja voltada diretamente para a borda do sistema de FL, ela contribui para mitigar desafios típicos da camada superior, como a presença de variabilidade nos dados, a eficiência do processo de agregação global e o custo computacional na borda.

Em geral, na camada superior, o **Modelo de FL e o Servidor de Agregação** orquestra os clientes (nós), detecta eventos raros e garante resistência a ataques de envenenamento ou falhas, como o custo de comunicação (Itahara et al., 2021). As responsabilidades podem incluir técnicas como extração de características (Chen; Vikalo et al., 2023), regularização dinâmica (Yu et al., 2022), seleção de nós (Tu; Zhao; Deng, 2023), clusterização de clientes (Li; Lin; Shang; Wu, 2023), amostragem de clientes (Zheng; Ye; Li; Gao, 2023), contribuições dos clientes (Sun et al., 2023) e seleção adaptativa (Bustincio; Souza; Costa; Bittencourt, 2023, além de maior justiça na colaboração entre os clientes e mecanismos de defesa contra ataques de envenenamento (Zhang et al., 2022; Rodríguez-Barroso; Martínez-Cámara; Luzón. Herrera, 2022).

Por exemplo, aspectos da **camada superior**, como a orquestração de clientes e o gerenciamento de problemas de conexão, são comumente abordados em estudos sobre algoritmos

de agregação global e no servidor de agregação. Alternativamente, as soluções podem integrar criptografia, blockchain e gerenciamento de conexões.

Além disso, a taxonomia no Continuum da Internet das Coisas (Al-Dulaimy et al., 2024; Ullah et al., 2023) atribui essas responsabilidades ao orquestrador, incluindo como gerenciar conectividade, recursos de rede, gerenciamento de recursos, gerenciamento da rede e segurança através das camadas distribuídas de borda, névoa e nuvem. Algumas responsabilidades são compartilhadas entre os algoritmos de FL e o orquestrador, especialmente em cenários nos quais a coordenação, a disponibilidade de dados e a resiliência do sistema são cruciais para os processos distribuídos de treinamento e agregação.

Este trabalho propõe um **algoritmo agnóstico** (isto é, uma nova camada de Inteligência Artificial (IA) como etapa de preparação de dados na borda) **sem interferir** na execução do algoritmo de **FL nem** no **servidor** onde ocorre a agregação. O algoritmo **baseia-se na avaliação da qualidade dos dados** na borda, removendo dados sem informações relevantes para o treinamento e melhorando a convergência dos algoritmos de treinamento por meio da seleção das informações mais significativas dos dados de entrada com base na métrica de entropia.

Portanto, o **algoritmo proposto** pode ser integrado a diferentes pipelines de FL, independentemente das estratégias ou do gerenciamento de agregação adotados pelo servidor. Por essa razão, a detecção de eventos raros, a segurança em FL, os ataques que comprometem a integridade ou a confiabilidade do modelo neural hospedado no servidor de agregação, e os problemas de comunicação **estão fora do escopo deste trabalho.**

Conforme ilustrado na Infografia 1, apresenta-se a metodologia adotada neste trabalho, a qual se estrutura nas seguintes etapas: 1) Estudo do modelo teórico, com ênfase no conceito de entropia; 2) Definição dos conjuntos de métodos empregados na pesquisa; 3) Avaliação e discussão dos resultados obtidos.

1.1 Definição do Problema

A crescente quantidade de dados, impulsionada pelo número crescente de usuários da internet e dispositivos IoT, aumenta a necessidade de um gerenciamento, análise e processamento de dados. Os avanços na área de gestão de volume de dados, redução de tráfego de dados e proteção da privacidade são contribuições para este cenário (Anjos et al., 2020; Rosendo; Costan; Valduriez; Antoniu, 2022; Fernandes et al., 2020; Muralidharan et al., 2022).

Além disso, a quantidade de dados criados, capturados, copiados e consumidos em

Figura 1 – A infografia deste trabalho.

Fonte: Elaboração Própria.

todo o mundo está prevista para aumentar rapidamente. Em 2020, alcançou 64,2 zettabytes, e até 2025, deve crescer para mais de 180 zettabytes. Desta forma, o número global de dispositivos IoT deve dobrar até 2030 (Taylor, 2023; Taylor, 2022; Vailshery, 2023).

Além disso, a implementação da *General Data Protection Regulation* (GDPR) na Europa em 2018, e a subsequente criação da Lei Geral de Proteção de Dados Pessoais (LGPD) no Brasil, reflete preocupações com privacidade e segurança de dados. O modelo de FL, embora não criado especificamente em resposta a essas leis, alinha-se aos seus princípios, focando em privacidade, eficiência e descentralização de dados. Ele mantém os dados nas fontes originais, promove a descentralização do processamento e reduz o uso de recursos de rede, transmitindo apenas os gradientes da rede neural, proporcionando uma solução (Zhang et al., 2021; Du et al., 2020).

Contudo, o modelo FL apresenta inerentemente uma latência elevada, causando atrasos na convergência, redução da frequência de atualização, problemas de sincronização, impacto na precisão e aumento do consumo de energia. Além disso, uma das limitações atuais do ambiente federado é a seleção de nós que frequentemente abandonam o sistema e são substituídos por novos nós que introduzem informação de forma incremental, sem nenhuma validação.

Neste contexto, a entrada ignora a qualidade dos dados selecionados, concentrando-se antes na capacidade de processamento e na disponibilidade dos nós (clientes). Os dados que contribuem com um valor incremental mínimo para o modelo resultam em consequências ao nível do processamento de borda. Como resultado, esses dados são subutilizados e não produzem uma atualização significativa nos pesos do modelo local, levando à computação desnecessária na borda, a requisitos de comunicação excessivos e aumento do consumo de energia.

Como nenhum desses novos conjuntos de nós é especificamente validado, eles

permitem dados de entrada de qualidade variável, o que pode potencialmente aumentar a latência e dificultar a obtenção de uma elevada precisão e baixas perdas.

Portanto, a motivação deste trabalho consiste em validar a entrada de dados e reduzir o elevado custo de convergência das redes neurais em dispositivos IoT, com foco na minimização dos custos computacionais e energéticos indiretamente associados.

Assim, o modelo proposto EnBaSe dá prioridade tanto à qualidade dos nós como à capacidade de processamento. Assim, o algoritmo de entropia EnBaSe exclui informações que não contribuem significativamente para o modelo ou que possam ter uma contribuição limitada, selecionando os dados com as informações mais relevantes.

Por fim, na Tabela 1, são detalhadamente apresentados os principais desafios enfrentados no contexto debatido até agora. Esta tabela sintetiza as questões-chave, fornecendo uma visão estruturada dos obstáculos.

Tabela 1 – Desafios do aprendizado federado em dispositivos IoT.

Desafio	**Descrição**
Diversidade do Sistema	Desafios em lidar com dados heterogêneos non-iid.
Limitações dos Dispositivos	Restrições de processamento, memória e energia em dispositivos IoT.
Custo de Convergência	Tempo e recursos adicionais necessários para a convergência de dados distribuídos.
Segurança Contra Ataques	Necessidade de proteção contra manipulações maliciosas.
Eficiência na Comunicação	Desafios relacionados ao tráfego de dados em redes limitadas.

Fonte: Elaboração Própria.

1.2 Objetivos Gerais e Especificos

Esta pesquisa concentra-se no uso da entropia, um conceito originado na teoria da informação, como critério para a seleção de dados em sistemas distribuídos, tais como o FL, aplicado a dispositivos de IoT. O objetivo é otimizar o processo de treinamento de redes neurais para classificação de imagens, ao mesmo tempo em que se busca reduzir o consumo de recursos computacionais, promovendo economia de energia e tempo de processamento — por meio de comparações com abordagens tradicionais, como amostragem aleatória estratificada, e com o uso de métricas como acurácia, F1-Score, recall, sensibilidade e resultados encontrados na literatura do mesmo domínio.

E objetivos específicos:

- Desenvolver um algoritmo capaz de validar a qualidade dos dados e reduzir o custo

computacional de processamento com o mínimo de perda de acurácia possível.

- Comparar o desempenho do modelo proposto em conjunto com outras abordagens existentes na literatura; e
- Avaliar o modelo em termos de precisão e eficiência computacional.

1.2.1 Principais Contribuições

Esta subseção apresenta as principais contribuições deste estudo, destacando os avanços alcançados por meio desta pesquisa:

- Uma nova abordagem da seleção de dados;
- Uma comparação detalhada com as abordagens existentes, destacando as vantagens do modelo proposto; e
- Uma avaliação abrangente da eficácia do algoritmo proposto, considerando tanto a precisão quanto a eficiência no uso de recursos.

1.2.2 Desafios e Limitações

Nesta seção, destacamos os principais desafios e limitações enfrentados neste estudo. Eles definem o escopo e os limites deste estudo, proporcionando uma compreensão clara das áreas que requerem atenção especial e das restrições sob as quais a pesquisa foi conduzida.

O objetivo é desenvolver um algoritmo capaz de reduzir o tempo de treinamento de uma rede neural, assim como os custos computacionais e energéticos, minimizando ao máximo a perda de precisão. Este algoritmo empregará o conceito de entropia para quantificar a previsibilidade da informação, servindo como um mecanismo de seleção de dados de alta qualidade em ambientes iid e non-iid. Os desafios incluem:

- **Complexidade da Entropia em Sistemas Distribuídos:** A aplicação da entropia como uma ferramenta para selecionar dados e a complexidades da natureza distribuída e heterogênea desses sistemas;
- **Desenvolvimento do Algoritmo:** Desenvolver um algoritmo baseado na entropia que seja eficaz em minimizar o uso de recursos ao mesmo tempo em que mantém uma alta precisão; e
- **Equilíbrio entre Precisão e Eficiência de Recursos:** Encontrar um equilíbrio entre a eficiência no uso de recursos e a precisão da rede neural.

A investigação é limitada aos tipos de dados e modelos específicos de ML examina-

dos. Além disso, foi conduzida sob a premissa de recursos computacionais limitados disponíveis. Assim, as limitações do estudo são:

- **Escopo da Análise de Dados:** A análise está limitada aos tipos de dados e aos modelos de aprendizado de máquina testados (e.g., MNIST, Fashion-MNIST, CIFAR-10, CIFAR-100 e as *Convolutional Neural Network* (CNN)); e
- **Redes Neurais:** A análise não tem como objetivo criar uma rede neural ótima para solucionar desafios relacionados ao estado da arte, mas sim desenvolver uma rede com baixo viés e precisão aceitável para a comparação de experimentos.

Por fim, concluímos que os **problemas mitigados** por este estudo incluem a redução do custo computacional na borda, a aceleração da convergência da rede neural na borda e a diminuição da heterogeneidade dos dados, melhorando a convergência das redes neurais.

1.3 Metodologia de busca de artigos

Nesta subseção, será apresentado os artigos avaliados no decorrer da Revisão Sistemática da Literatura (RSL), organizados com base nos critérios, métricas e metodologias discutidas nas seções a seguir. A seleção dos artigos teve como base o tema em estudo, bem como fatores de impacto, qualidade e relevância já discutidos. A subseção 1.3.1 detalha os objetivos da pesquisa; a Seção 1.3.2 aborda as questões de pesquisa, e, na mesma seção, a Tabela 1 ilustra a relação entre as questões de pesquisa e os tópicos abordados.

A estratégia adotada nesta pesquisa é descrita na seção seguinte 1.4, que inclui a Tabela 3, apresentando os veículos de pesquisa e seus respectivos tópicos. O método de pesquisa e avaliação da pesquisa são expostos na seção 1.4.1.

1.3.1 Revisão Sistemática da Literatura

O objetivo principal é explorar o uso da entropia como uma ferramenta para selecionar dados. Este processo tem o potencial de economizar recursos, minimizando os custos associados à computação, ao consumo de energia e ao tempo. O estudo está situado no contexto de sistemas distribuídos, abrangendo áreas como IoT e ML. Os objetivos específicos desta pesquisa são:

- Investigar como diferentes padrões de distribuição de dados podem influenciar a eficácia das redes neurais;

- Investigar como diversas estratégias de gerenciamento de dados impactam a eficiência energética e o desempenho computacional em ambientes de IoT;
- Realizar uma investigação abrangente para comparar métodos existentes sob diferentes cenários e métricas de desempenho;
- Expandir o escopo da pesquisa para incluir o FL e o processamento de imagens; e
- Investigar os desafios encontrados nas abordagens atuais de FL, identificando áreas para futuras melhorias.

1.3.2 Questões de Pesquisa

As questões abordam a influência dos padrões de distribuição de dados, o impacto das estratégias de gerenciamento de dados na eficiência energética e no desempenho computacional, a organização de conjuntos de dados (especialmente de imagens) e a integração do FL.

1. **Aprendizado Federado e Processamento de Imagens**
    a. Quais são os principais desafios enfrentados no uso do aprendizado federado?
    b. Quais são os principais desafios e limitações encontrados nessas abordagens?
    c. Como essas estratégias impactam a eficiência energética e o desempenho computacional?
    d. De que maneira os diferentes padrões de distribuição de dados afetam a eficácia do FL?

Este questionário busca explorar aspectos relacionados aos dados em FL, especialmente no contexto de IoT e processamento de imagens. A Tabela 2 apresenta a relação entre as ideias abordadas nos questionários e seus respectivos tópicos, destacando a correlação entre as questões estabelecidas e os tópicos que elas buscam explorar, fornecendo, assim, uma visão geral do estudo.

1.4 Estratégias da RSL

A consulta realizada na plataforma Acesso disponível em *QUALIS CAPES* Sucupira possibilitou a identificação das revistas científicas e editoras classificadas na categoria Qualis A1. Essa classificação contribui para garantir a qualidade e a precisão acadêmica nas pesquisas. Através dessa plataforma, obteve-se acesso a 668 páginas de revistas científicas com classificação Qualis A1.

Tabela 2 – Correlação entre os questionários e os temas.

Questionário de Pesquisa	**Temas**
Aprendizado Federado e Processamento de Imagens	Investigação dos desafios do aprendizado federado
Desafios no uso do aprendizado federado	Exploração das limitações das abordagens atuais
Desafios e limitações nas abordagens	Avaliação do impacto dessas estratégias na eficiência energética e no desempenho computacional
Impacto na eficiência energética e no desempenho computacional	Análise da variação na distribuição de dados e seu efeito
Influência dos padrões de distribuição de dados na eficácia da rede neural	Estudo da influência dos padrões de distribuição de dados

Fonte: Elaboração Própria.

A busca por artigos relevantes envolveu o uso de diferentes fontes de informação. A Tabela 3 apresenta a estrutura de pesquisa selecionada, destacando os veículos utilizados e os tópicos pesquisados em cada um deles. A pesquisa foi conduzida em duas fontes principais: Google Scholar e as bases de dados do IEEE Xplore, Springer, Science Direct, ACM, ACM Digital Library, Nature e Wiley.

Tabela 3 – Veículos e Tópicos Pesquisados.

Veículo	**Tópicos Pesquisados**
Google Scholar	• Technical and Privacy Challenges in Federated Learning • Data Privacy and Security in Federated Systems • Energy Efficiency and Computational Performance in Federated Models • Signal Processing and Denoising in Federated Learning • Transfer Learning and Image Data Augmentation for Deep Learning • Surveys on Deep Learning, IoT, Distributed Machine Learning, and Signal Processing
IEEE Xplore, Springer, Science Direct, ACM, ACM Digital Library, Nature, Wiley	• Federated Learning and its Challenges (Privacy, Efficiency, Energy Optimization) • Federated Learning for Image Processing • Convolutional Neural Networks in Federated Models • Data Distribution and Heterogeneous Data in Federated Systems • Data Filtering in Federated Learning • Surveys on Deep Learning, IoT, Distributed Machine Learning, and Signal Processing

Fonte: Elaboração Própria.

1.4.1 Método de Pesquisa e Documentação da RSL

O *template* ilustrado na Figura 2 representa a estrutura que fundamenta a análise de qualidade dos artigos científicos examinados neste trabalho. Composto por diversas seções, ele detalha os aspectos essenciais a serem considerados em um artigo, facilitando uma análise objetiva e direta. As seções são descritas a seguir:

Figura 2 – Layout da Documentação de Artigos - Template.

Nome do Artigo

Síntese — Resumo direto baseado no abstract.

I. Resumo do Artigo

Resumo do artigo, baseado em tópicos gerais e na introdução.

II. Descrição do Problema

Principais problemas que o artigo busca solucionar.

III. Descrição do Modelo/Solução

Descrição da solução abordada pelo autor.

A Tabela I, a seguir apresenta os equipamentos, algoritmos, técnicas e arquiteturas empregadas na solução.

Tabela I: Tecnologias Relacionadas

Equipamento Relacionados	
Sensores	**Descrição**
Conjunto de Técnicas de Avaliação	
Técnica de Análise	**Descrição**
Arquiteturas Empregadas	
Arquitetura de Rede Neural	**Descrição**

IV. Descrição dos Pontos Fortes

Os pontos fortes descritos a seguir:

- Descrição dos pontos fortes na pesquisa.

V. Descrição dos Pontos Fracos

Os pontos fracos descritos a seguir:

- Descrição dos pontos fracos na pesquisa.

VI. Em Aberto Na Literatura

Documentação do que está em aberto na literatura.

VII. Tabela de Avaliação

A seguir, apresentamos a Tabela II com a resolução, avaliação e pontuação, contendo notas com intervalo de 1 a 5.

Tabela II: Tabela de Avaliação

TABELA DE AVALIAÇÃO	NOTA
1. O tema é importante para a comunidade científica?	
2. O paper é tecnicamente sólido?	
3. A cobertura do tema é suficientemente abrangente e equilibrada?	
4. Avalie a profundidade técnica do artigo?	
5. Como você avaliaria a novidade técnica do paper?	
6. Organização geral do paper?	
7. Adequação do título	
8. Adequação do Abstract	
9. Motivação	
10. Solução/Modelo	
11. Resultados	
12. Bibliografia	
DADOS DO ARTIGO	**Quantidade**
Qualis	--
Citações	--
Visualizações	--
Link	--

Fonte: Elaboração Própria.

1. Síntese: equivale a um *abstract* do artigo, apresentando de forma resumida os principais pontos abordados para revisão literaria e consultas.
2. Resumo do Artigo: essa seção resume de forma concisa a introdução do artigo, destacando os principais pontos abordados.
3. Descrição do Problema: nessa seção, é feita uma descrição clara e objetiva do problema abordado no artigo, fornecendo contexto e relevância para o estudo realizado.
4. Descrição do Modelo/Solução: nesta seção, é apresentada a solução proposta pelo artigo, incluindo a abordagem adotada e os métodos utilizados para resolver o desafio proposto.
5. Tecnologias Relacionadas: esse tópico contem uma tabela que é adaptada individualmente para cada artigo, registrando os equipamentos, algoritmos, métricas, arquiteturas e técnicas utilizados na solução proposta. Ela fornece uma visão geral das tecnologias empregadas, auxiliando na identificação das abordagens relevantes.
6. Descrição dos Pontos Fortes: nessa seção, são destacados os pontos fortes do artigo, ressaltando as contribuições significativas e os aspectos positivos do trabalho.
7. Descrição dos Pontos Fracos: nessa seção, são mencionados os pontos fracos do artigo, apontando possíveis limitações ou aspectos que poderiam ser melhorados.
8. Em Aberto Na Literatura: essa seção aborda questões ou lacunas que ainda não foram totalmente exploradas ou solucionadas. Essa forma de documentação auxilia a monitorar a relevância do tema e pode fornecer informações para pesquisas futuras.
9. Tabela de Avaliação: essa tabela apresenta uma avaliação geral do artigo, incluindo critérios como a importância do tema, solidez técnica, cobertura do assunto, entre outros. Além disso, o artigo é avaliado quanto ao seu *Qualis*, quantidade de citações e visualizações por parte dos pesquisadores. Essa forma oferece uma visão geral da qualidade e relevância do artigo analisado.

1.5 Organização deste Trabalho

Este trabalho está organizado da seguinte forma: o background, juntamente com os trabalhos relacionados, é apresentado no Capítulo 2. O algoritmo proposto é detalhado no Capítulo 3. Os materiais e métodos são apresentados no Capítulo 4. Os resultados são discutidos no Capítulo 5. Por fim, o Capítulo 6 aborda as conclusões e as perspectivas futuras.

1.6 Considerações Finais

Esse capítulo apresentou uma breve introdução ao tema, assim como ideias gerais sobre como este estudo foi construído, seus objetivos, suas contribuições e limitações.

2 BACKGROUND E TRABALHOS RELACIONADOS

Neste capítulo, são explorados o *background* e os trabalhos relacionados. Dessa forma, cada um desses conceitos contribui para o desenvolvimento deste estudo. Os conceitos serão apresentados a seguir, com a exemplificação de algumas de suas características e detalhados nas seções subsequentes.

2.1 Internet of Things (IoT)

IoT combina dispositivos equipados com sensores, capacidade de processamento e software, facilitando a troca de dados entre dispositivos e sistemas através da Internet ou outras redes. Esta tecnologia integra áreas como eletrônica, comunicação e engenharia da computação, representando um avanço significativo na interação com objetos cotidianos.

O desenvolvimento da IoT é impulsionado por tecnologias emergentes como computação ubíqua, sensores acessíveis, sistemas embarcados avançados e aprendizagem de máquina. Campos como sistemas embarcados, redes de sensores sem fio e automação (residencial e predial) também contribuem significativamente. No âmbito doméstico, a IoT se manifesta em produtos de casa inteligente (iluminação, termostatos, segurança, câmeras), controláveis via smartphones e alto-falantes inteligentes. Além disso, sua aplicação também se estende à saúde, proporcionando inovações significativas.

A Internet, essencial em setores como educação, pesquisa, negócios e uso pessoal, tem sua funcionalidade ampliada com o crescimento da IoT. Esta evolução permite a integração de objetos do dia a dia (carteiras, relógios, geladeiras, carros) em redes para monitoramento e controle remotos, aprimorando a segurança do usuário. O impacto da IoT no cotidiano das pessoas é profundo, afetando o pensamento, a vida e o trabalho (Singh, 2023).

Os componentes fundamentais da IoT incluem:

- **Sensores e Atuadores:** Dispositivos que coletam dados do ambiente;
- **Conectividade:** Wi-Fi, Bluetooth e redes celulares, essenciais para conectar dispositivos IoT à internet;
- **Processamento de Dados:** Capacidade de analisar e interpretar os dados coletados, frequentemente utilizando plataformas em nuvem; e
- **Interface de Usuário:** Aplicações e sistemas que facilitam a interação dos usuários com dispositivos IoT.

Setores que utilizam aplicações da IoT:

- **Doméstico:** Automação residencial, com sistemas de segurança inteligentes e termostatos conectados;
- **Industrial:** Monitoramento de cadeias de suprimentos e gestão de inventário;
- **Saúde:** Dispositivos médicos conectados a sistemas de alerta; e
- **Cidades Inteligentes:** Gerenciamento de tráfego, sistemas de iluminação pública eficientes e monitoramento ambiental.

2.2 A natureza dos dados e suas propriedades estatísticas

Nas subseções a seguir, serão discutidas e apresentadas as definições de homogeneidade e heterogeneidade dos dados, bem como suas relações e definições em relação aos dados iid, non-iid, e ao Teorema do Limite Central (TLC).

2.2.1 Dados homogêneos e dados heterogêneos

Os dados homogêneos são caracterizados por compartilhar propriedades quantitativas ou qualitativas, o que implica uma variação aceitável ou previsível, refletindo uma tendência semelhante. Assim, quando os dados têm características estatisticamente homogêneas, as redes neurais tendem a convergir para valores ótimos do *Stochastic Gradient Descent* (SGD) (Ju; Zhang; Toor; Hellander, 2024).

Ao contrário dos dados homogéneos, que apresentam uniformidade, os dados heterogéneos apresentam uma grande variabilidade de tipos e formatos (Gao; Yao. Yang, 2022). Estes dados incluem frequentemente variabilidade, valores atípicos e inconsistências no formato e no significado (Kamm et al., 2023). Por exemplo, a diversidade de sensores IoT e dispositivos de coleta de dados resulta numa variabilidade e desequilíbrio significativos nos dados gerados, criando dados heterogêneos (Xu; Qu; Xiang; Gao, 2023).

2.2.2 Dados iid e dados non-iid

Os dados iid referem-se a observações independentes que seguem a mesma distribuição probabilística, o que melhora a convergência e o desempenho do modelo ao manter a uniformidade nas amostras de treino (Zhu; Xu; Liu; Jin, 2021). Em ML, esses dados são frequentemente processados em servidores centralizados, assumindo homogeneidade. Neste contexto,

os modelos tendem a convergir para um ótimo global devido à semelhança nas distribuições de amostras (Li; Diao; Chen; He, 2022).

Ao contrário, os dados non-iid introduzem heterogeneidade estatística, com variações na quantidade e distribuição de amostras (Ma et al., 2022). Neste cenário, cada nó ou dispositivo tem um conjunto de dados único. A variabilidade na quantidade de dados e nas classes entre dispositivos introduz enviesamento, dificultando o treino da rede neural (Jamali-Rad; Abdizadeh; Singh, 2022; Cao, 2022).

2.2.3 Amostragem Aleatória

No início do século XX, o campo da *Artificial Intelligence* (AI) começou a se beneficiar significativamente de modelos matemáticos, como, por exemplo, o *backpropagation*. Dentre esses modelos, um modelo matemático comumente aplicado em redes neurais é uma técnica estatística e probabilística utilizada em ML, a amostragem aleatória (Rajput; Wang; Chen, 2023), que contribui para promover a diversidade na seleção da amostra e permite que a rede neural aprenda diferentes aspectos dos dados, o que leva a previsões mais precisas (Macnell et al., 2023).

Esta técnica consiste em selecionar aleatoriamente um conjunto de dados, de modo que cada elemento tenha uma probabilidade igual ou conhecida de ser selecionado. Dessa forma, o objetivo é criar um conjunto representativo de dados, evitando o enviesamento de escolhas não aleatórias. Esta abordagem aumenta a robustez do modelo ao diminuir a dependência de configurações específicas da amostra (Chen et al., 2022).

2.2.4 Teorema do Limite Central

O Teorema do Limite Central (TLC), que afirma, a distribuição de variáveis da soma (ou média) tende a se aproximar de uma distribuição normal, ou seja, uma gaussiana à medida que o tamanho de uma amostra de uma população aumenta. Desta forma, o TLC é expresso matematicamente, seja $X_1, X_2, \ldots, X_n$ variáveis aleatórias iid com média μ e variância σ^2. Então, a média amostral $\bar{X}_n$ é dada por:

$$\text{TCL de Lindeberg:} \quad \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$$

Conforme n se torna grande, a distribuição de $\bar{X}_n$ se aproxima de uma distribuição

normal com média μ e variância $\frac{\sigma^2}{n}$. Em outras palavras, a padronização da média amostral se aproxima de uma distribuição normal padrão $N(0,1)$ conforme o tamanho da amostra n aumenta.

Desta forma, para dados homogêneos iid, ou seja, os dados independentes e identicamente distribuídos. A independência garante que os valores das variáveis X não estejam associados aos valores de outras variáveis Y, as observações de uma amostra não influenciam as outras.

A identidade na distribuição significa que todos os dados seguem a mesma distribuição com os mesmos parâmetros, como média, variância, etc. Assim, a antítese dessa ideia são dados non-iid (por exemplo, heterogêneos), ou seja, não são independentes e/ou não são identicamente distribuídos. A falta de independência significa que os valores das variáveis podem estar associados a outras variáveis e que as observações de uma amostra podem influenciar outras. Desta forma, a heterogeneidade da distribuição pode fazer com que as distribuições dos parâmetros, média, variância e outras medidas divirjam.

Por fim, em relação ao TLC, depende das suposições de independência e identidade na distribuição das variáveis serem aceitas. Assim como no contexto do TLC, a homogeneidade está relacionada às variáveis aleatórias que seguem a mesma distribuição de parâmetros, como média e variância. Porém, em cenários non-iid, as suposições do TLC são violadas, onde os dados não seguem uma distribuição normal e a presença da heterogeneidade, junto com a violação da pressuposição de normalidade, criam um cenário desafiador.

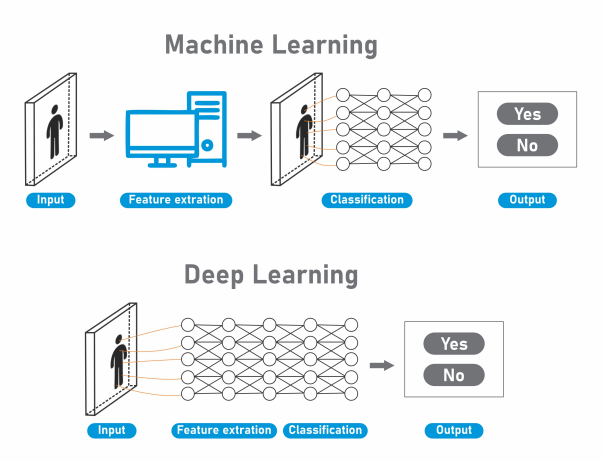
2.3 Breve Contextualização do Aprendizado Profundo

O *Deep Learning* (DL), um subcampo ML, caracteriza-se pelo uso de redes neurais artificiais com várias camadas, possibilitando o processamento de grandes volumes de dados e a extração de padrões complexos. Diferentemente do aprendizado de máquina tradicional, que muitas vezes requer entrada manual de características e é limitado na capacidade de processar dados não estruturados, o DL automatiza a extração de características e é eficaz em trabalhar com dados não estruturados, como imagens e texto.

Essas redes neurais profundas utilizam múltiplas camadas, com unidades neurais organizadas em camadas. Estas unidades aplicam funções de ativação para processar não-linearidades e aprender relações complexas entre dados. A otimização dos parâmetros da rede é feita por algoritmos como o SGD, visando minimizar erros em tarefas de reconhecimento de padrões e classificação de dados (Li; Li; Chen; Sun, 2023). A Figura 3 ilustra a estrutura e o

funcionamento do DL.

Figura 3 – Exemplo de extração de características de uma rede neural profunda.



Fonte: Adaptado de (Mittal; Srivastava; Jayanth, 2023).

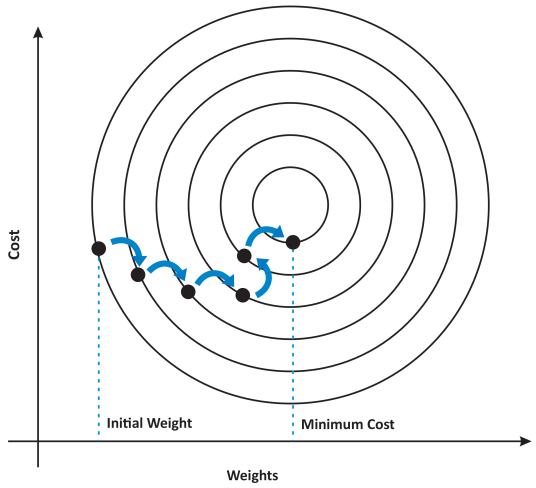
2.3.1 Técnicas para a Otimização de Treinamentos

Em relação à otimização, normalmente é utilizada para buscar soluções para problemas complexos que exigem soluções eficientes e eficazes em termos de computação, por exemplo. Problemas como estes normalmente são classificados como NP-difíceis, necessitando de métodos computacionais e técnicas para minimizar os custos e maximizar a eficiência.

O campo da otimização abrange uma variedade de metodologias para lidar com problemas complexos e NP-difíceis em diversos setores, incluindo transporte, logística, fabricação, cidades inteligentes, entre outros. Este espectro inclui metaheurísticas, matheurísticas, simheurísticas, heurísticas biased-randomised (BR) e *learn heuristics*. Essas abordagens são essenciais para encontrar soluções eficientes em cenários dinâmicos e incertos, especialmente em áreas como aprendizado de máquina e DL (Juan et al., 2023).

Para AI, o SGD torna-se um método otimizador no treinamento de modelos de aprendizado de máquina. Em redes neurais DL, o SGD adota uma abordagem adaptativa. Ele atualiza os parâmetros de forma iterativa, usando apenas um subconjunto ou mesmo um único exemplo do conjunto de dados, tornando-o mais eficiente e menos propenso a ficar preso em mínimos locais. Na Figura 4, demonstramos o SGD, onde a cada operação atualiza iterativamente os parâmetros do modelo, de modo a reduzir o erro nas previsões.

Figura 4 – Modelo de Gradiente Descendente Estocástico.



Fonte: Adaptado de (Shalev-Shwarrtz; Ben-David, 2014)

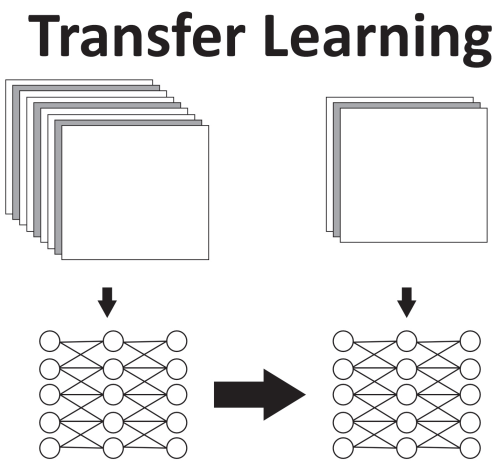
Além do SGD, outros métodos amplamente utilizados incluem Adam, Adagrad, RMSProp e *Nesterov Accelerated Gradient* (NAG). Cada um possui características específicas e é aplicado conforme as particularidades de cada problema, otimizando o desempenho de maneira direcionada.

2.3.2 Transfer Learning

TL, é uma técnica poderosa no campo de ML, onde modelos desenvolvidos para uma tarefa são reutilizados como ponto de partida para modelos em outras tarefas. Desta forma, as redes neurais treinadas em grandes conjuntos de dados de imagens são ajustadas para tarefas específicas, como o reconhecimento de objetos em contextos específicos.

Essa aplicação diversificada do TL é possível devido à sua capacidade de adaptar modelos a partir de um domínio-fonte rico em dados para um domínio-alvo com dados mais escassos (Niu; Liu; Wang; Song, 2020). Na Figura 5, é apresentado o modelo de TL. Neste modelo, a partir de um domínio com ampla disponibilidade de dados, os pesos da rede neural são empregados como base para um novo domínio, caracterizado por ter uma menor quantidade de dados disponíveis.

Figura 5 – Exemplo de aplicação de TL.



Fonte: Adaptado de (Zhu; Lin; Jain; Zhou, 2023).

2.3.3 Data Augmentation

A técnica de DA é fundamental em DL, visando aumentar a quantidade e diversidade dos dados de treinamento. Esta técnica inclui transformações como rotação, escalonamento e inversão de imagens. Além disso, essa técnica contribui para prevenir o *overfitting*, que ocorre quando um modelo se ajusta excessivamente aos dados de treinamento.

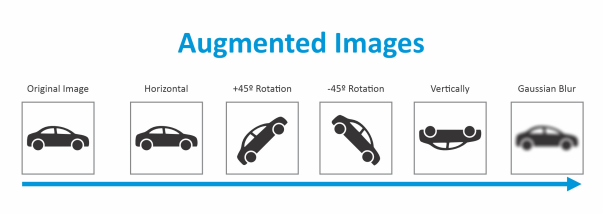
Por exemplo, DA é aplicado na análise de sentimentos e em linguagem, onde sua eficácia melhora a precisão dos modelos de análise de sentimentos devido à diversidade e riqueza dos dados (Shorten; Khoshgoftaar; Furht, 2021). Outra aplicação da DA é na radiologia e radioterapia, onde expande conjuntos de dados limitados, contribuindo para uma melhor generalização dos modelos (Chlap et al., 2021).

Na Figura 6, ilustra-se o conceito de DA aplicado a uma imagem. Esta figura demonstra como o DA pode manipular a imagem original para gerar um conjunto ampliado de dados. O processo de DA envolve a criação de novas instâncias de dados a partir dos dados existentes.

2.3.4 Aprendizado Centralizado e FL

No ML centralizado, os dados são originados e armazenados, analisados e processados num servidor dedicado ou numa localização centralizada. Esta arquitetura promove a eficiência na modelação estatística e na detecção de padrões. Este modelo facilita a aplicação de algoritmos ML que requerem grandes quantidades de dados para generalizar bem e obter

Figura 6 – Exemplo de aplicação de DA.



Fonte: Adaptado de (Ottoni; Amorim; Novo; Costa, 2023).

resultados preditivos precisos.

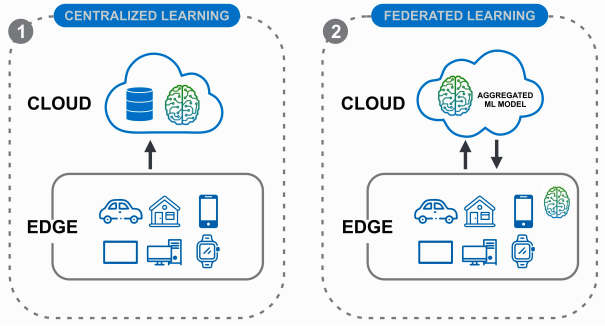
No entanto, os modelos centralizados apresentam desafios, principalmente relacionados com a segurança dos dados e a coleta e centralização de grandes volumes de dados. Como resultado, isto expõe informações sensíveis a riscos como a fuga de dados ou ciberataques (Coutinho-Almeida; Cruz-Correia; Rodrigues, 2024). A centralização também leva a problemas relacionados com a latência, onde os dados de diferentes fontes são centralizados num servidor, consumindo uma quantidade significativa de comunicação, especialmente com dados geograficamente distribuídos (Rao et al., 2024).

Ao contrário da abordagem centralizada, FL segue uma estratégia descentralizada para treinar modelos ML. Dados de diferentes fontes contribuem para o treinamento de vários dispositivos ou nós em uma rede (clientes), como smartphones, tablets, sensores IoT e outros dispositivos de computação de borda. Cada dispositivo utiliza os seus dados para treinar um modelo AI e, em seguida, envia atualizações dos parâmetros do modelo para um servidor central que agrega as atualizações dos parâmetros da rede neural. Nesta abordagem, os dados privados permanecem nos dispositivos da borda, nunca sendo compartilhados diretamente, respeitando as perspectivas éticas e legais em contextos em que os dados são sensíveis. Além disso, reduz as transferências massivas de dados e o risco de fugas de dados em grande escala (Yuan et al., 2024).

A Figura 7 mostra o desenho comparativo das arquiteturas de ML para ambos os contextos. A Figura 7.(1) mostra uma abordagem de aprendizado centralizado, onde dados de diferentes dispositivos são transferidos, centralizados e armazenados para treinar o modelo de ML. Em contraste, na Figura 7.(2), o aprendizado ocorre de forma descentralizada: os dados permanecem nos dispositivos locais, enquanto o modelo de ML é processado na borda. O modelo é compartilhado e atualizado com valores de parâmetros enviados por cada dispositivo (Anjos et al., 2023).

Assim, as **contribuições desta área**, neste trabalho, incluem um algoritmo agnóstico,

Figura 7 – Cenário de dados centralizados e descentralizados.



Fonte: Elaborado pelo autor.

na etapa de pré-processamento, que pode ser integrado a um servidor centralizado para reduzir o custo computacional e acelerar a convergência do algoritmo no domínio de imagens. Além disso, no cenário de borda, o algoritmo pode ser embarcado localmente, reduzindo os custos computacionais de dispositivos com baixo poder de processamento e limitações energéticas.

2.4 Teoria da Informação

A entropia é um conceito fundamental na teoria da informação para entender a transmissão e comunicação eficiente de informações; portanto, fornece uma maneira quantitativa de avaliar a incerteza. Considere um sistema simples que usa lançamentos de moeda. A probabilidade de ocorrência para cada lado deste sistema é a mesma. Onde $H(X)$ representa as faces e a entropia da moeda, e X é a variável aleatória que representa o resultado de cada lançamento de moeda, onde podemos quantificar a incerteza dos dados gerados pelos lançamentos pela seguinte fórmula:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{2.1}$$

Onde:

- X representa o conjunto de todos os valores de símbolos possíveis.
- $p(x_i)$ é a probabilidade de ocorrência do i-ésimo valor do símbolo.
- O $\sum_{i=1}^{n}$ abrange todos os valores possíveis.
- $\log_2 p(x_i)$ é o logaritmo com base b dois da probabilidade $p(x_i)$, tornando a unidade de entropia em bits.

Neste caso, em cada lançamento da moeda, onde há duas possibilidades (cara ou coroa), a entropia $H(X)$ fornece 1 bit de informação. Quanto maior o valor da entropia, maior a

imprevisibilidade e a incerteza; quanto menor o valor, menor a imprevisibilidade e a incerteza.

No entanto, neste sistema de lançamentos de moedas, é impossível determinar o resultado precisamente, já que cada lançamento possui o mesmo grau de incerteza. Porém, se a entropia for reduzida, o resultado indicará menos surpresa na informação a cada novo lançamento devido à maior previsibilidade dos resultados predeterminados.

Portanto, quando aplicamos a entropia no campo da *Computer Vision* (CV), cada X passa a representar um pixel e os valores de cor, intensidade de uma cor ou intensidade específica. Assim, a probabilidade de ocorrência de cada valor de cor ou intensidade, isto é, cada pixel único, pode ser calculada com base no número de vezes que cada cor ou intensidade específica aparece na imagem. Assim, uma alta entropia representa uma grande diversidade de pixels, indicando alta complexidade na textura, variação significativa e pouca previsibilidade da informação. Por outro lado, uma baixa entropia aponta para maior homogeneidade da imagem, isto é, maior uniformidade para identificar regiões, facilitando a segmentação de elementos em uma cena.

Finalmente, a entropia pode ser ilustrada através das seguintes analogias: inicialmente, há um conjunto desorganizado de imagens caracterizado por alta entropia e grande incerteza. Ao organizar e separar esse conjunto de imagens, é possível dividi-lo em segmentos de baixa entropia, tornando-o altamente previsível. Em contraste, outra parte é caracterizada por sua alta entropia e imprevisibilidade contínua.

2.5 Modelos de filtros

A filtragem de ruído em dados é uma etapa crucial em várias aplicações de processamento de sinais e análise de dados. Ela visa remover ou atenuar componentes indesejáveis ou irrelevantes (ruídos) dos dados, melhorando assim a qualidade da informação para análise subsequente. A eficácia da filtragem de ruído é determinante para o desempenho de algoritmos de aprendizado de máquina e sistemas de AI, especialmente em ambientes desafiadores como IoT. Nesta seção, abordaremos diferentes tipos de filtragem de ruído, cada um adequado a situações específicas e tipos de dados.

- **Filtros de Passa-Baixa:** São usados para remover altas frequências do sinal, que geralmente incluem ruído. Estes filtros são eficazes em suavizar dados, como em imagens e sinais de áudio, mantendo as informações importantes que geralmente residem em frequências mais baixas.
- **Filtros de Passa-Alta:** Em contraste com os filtros de passa-baixa, estes filtros são

utilizados para realçar características de alta frequência, como bordas em imagens, e são menos eficientes na remoção de ruídos de alta frequência.

- **Filtros Adaptativos:** Adaptam-se dinamicamente às características do sinal, sendo ideais para ambientes onde o ruído pode variar significativamente. Eles são comumente utilizados em sistemas de comunicação e processamento de sinais em tempo real.
- **Filtragem Baseada em Wavelet:** Utiliza transformadas wavelet para decompor o sinal em diferentes níveis de frequência e tempo, permitindo uma filtragem seletiva e mais eficiente de ruído, especialmente em sinais não estacionários.
- **Filtragem Baseada em Aprendizado de Máquina:** Para filtrar ruídos, especialmente em imagens e vídeos, redes neurais convolucionais aprendem a remover ruídos usando exemplos treinados.

A filtragem e o processamento de sinais são utilizados no *Edge Computing* (EC) com IoT para diagnóstico de falhas em máquinas, melhorando a eficiência computacional e reduzindo a sobrecarga em servidores na nuvem (Lu et al., 2023). Em outro contexto, (Abel; Dhanalakshmi; Kumar, 2023) destacam a importância das técnicas de filtragem adaptativa, incluindo métodos lineares e não lineares, para a obtenção eficiente do Eletrocardiograma Fetal (FECG) a partir de gravações abdominais.

Finalmente, (Chaudhary; Gupta; Pachori, 2023) destacam a importância da representação de Fourier-Bessel no processamento de sinais. Esta técnica permite a visualização de sinais em diversos domínios, especialmente no domínio da frequência, aplicada no processamento de sinais de fala, incluindo aprimoramento, identificação de gênero e compressão de fala.

2.6 Trabalhos Relacionados

Nesta Seção serão apresentados os trabalhos relacionados com este estudo e discutidas as oportunidades e desafios de pesquisa encontrados.

2.6.1 Pubicações Selecionadas

A Tabela 4 apresenta as publicações selecionadas. Respectivamente, as colunas da tabela listam os temas abordados nas pesquisas científicas publicadas. Esses temas foram utilizados pelos autores como fundamentos para as problemáticas exploradas e discutidas em suas respectivas pesquisas e no estado da arte.

Tabela 4 – Visão geral dos estudos de aprendizagem federada por características do sistema e técnicas de otimização.

		Properties				Strategies							
Author	Date	Deep Learning	Machine Learning	Edge Computing	IoT Devices	Algorithm Optimization	Automatic Adjustment	Data/Client Selection	Data Distribution	Image Data Analysis	Data Quality Analysis	Aggregation Method Compatible	Embedded Compatible
Li, Beibei et al.(**??**)	2020	x				x				x			
Kang, Jiawen et al.(KANG et al., 2020)	2020	x											
Du, Zhaoyang et al.(DU et al., 2020)	2020	x		x	x				x				
Itahara, Sohei, et al. (ITAHARA et al., 2021)	2021	x	x		x	x		x	x				
Criado, Marcos F. et al.(CRIADO et al., 2022)	2022	x	x						x				
Gafni, Tomer et al.(GAFNI et al., 2022)	2022	x		x	x	x			x				
Al-Saedi, Ahmed A et al.(AL-SAEDI; BOEVA; CASALICCHIO, 2022)	2022	x		x	x	x			x	x			
Yu, Xi et al.(**??**)	2022					x	x		x	x			
Ullah, Shan et al.(**??**)	2022	x				x			x	x			
Xu, Jian et al.(**??**)	2022	x				x			x	x			
Wolfrath, Joel et al.(**??**)	2022	x		x	x	x		x	x	x			
Li, Yang, et al. (LI; CHAO; ERCISLI, 2022)	2022	x								x	x		
Zhang, Yu, et al. (**??**)	2023	x	x			x				x			
Condori Bustincio, et al. (BUSTINCIO; SOUZA; COSTA; BITTENCOURT, 2023)	2023	x	x		x			x	x				
Orlandi, Fernanda C. et al.(ORLANDI et al., 2023)	2023	x		x	x	x	x	x	x	x			
Lo, Sin Kit et al.(LO et al., 2023)	2023	x		x	x	x		x	x	x			
Hossain, Md Zarif et al.(HOSSAIN; IMTEAJ, 2023)	2023			x	x	x			x	x			
Tao, Zeyi et al.(TAO; WU; LI, 2023)	2023	x		x	x	x			x	x			
Lee, Hyeongok et al.(LEE, 2023)	2023					x			x	x			
Tu, Chengwu et al.(**??**)	2023	x		x	x	x		x	x	x			
Li, Boyuan et al.(LI; CHEN; YU, 2023)	2023	x		x	x	x			x	x			
Yang, Wei-Jong et al.(**??**)	2023								x	x			
Chen, Huancheng et al.(CHEN; VIKALO et al., 2023)	2023	x		x	x	x			x				
Zheng, Shu et al.(**??**)	2023					x			x	x			
Huang, Chenxi et al.(HUANG et al., 2023)	2023	x				x			x	x			
Dolaat, Khalid Mahmoud Mohammad et al.(DOLAAT; ERBAD; IBRAR, 2023)	2023		x	x	x	x			x	x			
Qiao, Yu et al.(QIAO; LE; HONG, 2023)	2023			x	x	x			x	x			
Sabah, Fahad et al.(SABAH et al., 2023)	2023	x	x	x				x	x				
Li, Zexi et al.(LI; LIN; SHANG; WU, 2023)	2023	x	x			x		x	x	x			
Sun, Qiheng et al.(SUN et al., 2023)	2023					x		x	x	x			
Iyer, Venkataraman Natarajan et al.(IYER, 2024)	2024		x						x	x			
Milan Ilić et al.(ILIĆ; IVANOVIĆ; KURBALIJA; VALACHIS, 2024)	2024	x	x	x		x							
Yan, Litao, et al. (**??**)	2024		x	x	x	x					x		
Hamidi, Shayan Mohajer, et al. (HAMIDI; TAN; YE; YANG, 2024)	2024	x	x		x	x	x	x	x				
Nosso Modelo	2024	x	x	x	x		x	x	x	x	x	x	x

Fonte: Elaboração Própria.

O FedAVO (Hossain; Imteaj, 2023), inspirado em estratégias de otimização natural para melhorar a eficiência da comunicação em FL, visa reduzir a sobrecarga de comunicação através de métodos inspirados na natureza. O método deste trabalho EnBaSe se concentra na eficiência da seleção e qualidade; essa metodologia utiliza a entropia para selecionar dados de alta qualidade, reduzindo não apenas a comunicação, mas também os custos computacionais e

energéticos. Em relação ao Fedco (Al-Saedi; Boeva; Casalicchio, 2022), que utiliza a otimização de agrupamento para aumentar a eficiência da comunicação em FL, gerenciando e reduzindo a sobrecarga de comunicação, o algoritmo EnBaSe usa a entropia para seleção de dados, otimizando a informação transferida no canal de informação e assegurando que somente dados de alta qualidade sejam processados.

Quanto ao FedWNS (Tu; Zhao; Deng, 2023) utiliza a seleção de nós baseada na distribuição de dados através do aprendizado por reforço, ao utilizar uma estratégia de seleção de nós para obter melhores resultados. O método EnBaSe realiza uma seleção de dados mais granular, focando em qualidade e balanceamento, resultando em uma otimização do uso de recursos.

O algoritmo proposto por (Yu et al., 2022) para ajustar automaticamente os pesos para desempenho, busca eficiência e precisão do aprendizado. Esta técnica é relevante para ambientes FL, pois pode aumentar a eficácia do treinamento dos modelos sem a necessidade de intervenção manual frequente. Em paralelo, o algoritmo EnBaSe proposto se diferencia ao selecionar dados de alta qualidade antes do treinamento, otimizando o processo de aprendizado desde o início e reduzindo significativamente os custos computacionais e energéticos.

O FL precondicionado introduziu um método para precondicionar o aprendizado para impulsionar o desempenho (Tao; Wu; Li, 2023). Precondicionado propõe precondicionar ambientes de aprendizado ou dados, visando melhorar o desempenho do FL. Essa abordagem prepara previamente os dados ou o ambiente, facilitando o processo de treinamento. A entropia, como critério de seleção, assegura que o conjunto de dados utilizado seja o mais informativo possível com qualidade, otimizando tanto o tempo quanto a eficiência do treinamento.

Outra abordagem considera a heterogeneidade e foca na seleção de clientes (Wolfrath et al., 2022). Esse método considera a heterogeneidade e foca na seleção de clientes agrupados para acelerar o processo de FL, abordando os desafios da heterogeneidade dos clientes durante o aprendizado. Ao utilizar a entropia (EnBaSe) para a seleção de dados, garante-se que os dados mais informativos sejam priorizados. Essa estratégia difere da seleção de clientes agrupados, pois foca na qualidade dos dados, e não apenas na organização dos clientes. Com essa abordagem, a precisão do modelo é aprimorada, uma vez que se assegura que os dados utilizados no treinamento são de alta qualidade, independentemente da heterogeneidade dos clientes.

Por fim, o último método pesquisado compartilha a distribuição dos dados de treinamento para enriquecer o processo de otimização global (Li; Chen; Yu, 2023). Desta forma, os

clientes não apenas enviam seus modelos para o servidor, mas também compartilham a distribuição de seus dados de treinamento. O EnBaSe, comparado ao trabalho proposto, seleciona os dados antes do treinamento, reduzindo a necessidade de compartilhamento extensivo de dados. Isso otimiza o processo de treinamento ao garantir que apenas dados relevantes e de alta qualidade sejam utilizados. Além disso, ao reduzir a quantidade de dados transferidos, há uma diminuição significativa nos custos de comunicação e processamento.

Estes métodos procuram principalmente alcançar eficiência na comunicação, otimização na seleção de clientes, robustez e resiliência em cenários em que a segurança dos dados é crítica. Servem assim de motivação e base para investigação futura, apontando os principais desafios discutidos na literatura atual sobre IoT e FL. Desta forma, apresentam uma visão geral do estado da arte. Em paralelo, outras abordagens têm sido baseadas na análise da homogeneidade do sistema e da entropia dos dados da rede neural, como mostra a Tabela 5.

Tabela 5 – Resumo dos Trabalhos Relacionados - Abordagens Similares.

Autor	**Vantagens**	**Desvantagens**
(ITAHARA et al., 2021)	Robustez contra ataques e ruídos	Perda de precisão
(LI; CHAO; ERCISLI, 2022), 2022.	Reduz redundância em conjuntos de dados	Restrito a multiclasse
(BUSTINCIO; SOUZA; COSTA; BITTENCOURT, 2023)	Reduz a sobrecarga de comunicação	Generalização limitada
(**??**)	Robustez na seleção de características	Alta complexidade
(ORLANDI et al., 2023)	Mitiga de dados non-iid, reduz consumo de energia	Leve redução na precisão
(HAMIDI; TAN; YE; YANG, 2024)	Alta precisão em dados desbalanceados	Aumento da complexidade

Fonte: Elaboração Própria.

Nestes trabalhos um autor aplica o conceito de entropia em sua arquitetura para diferenciar entre dados relevantes e irrelevantes. O método gera uma imagem perturbada a partir de um protótipo estatístico, onde os valores de entropia são usados como indicadores de qualidade (Li; Chao; Ercisli, 2022). Outro estudo utiliza a redução de entropia para mitigar ambiguidades e melhorar a precisão das saídas do modelo, usando um método chamado Entropy Reduction Aggregation (ERA) (Itahara et al., 2021).

Além disso, um autor explora estratégias para minimizar a sobrecarga de comunicação e a heterogeneidade dos dados, utilizando entropia na seleção de clientes em dispositivos IoT com dados non-iid (Bustincio; Souza; Costa; Bittencourt, 2023). Em outra abordagem, a teoria da informação é integrada ao algoritmo, desenvolvendo um classificador de alta precisão baseado na combinação ótima de características (Zhang et al., 2023).

Ao avaliar a entropia nas extremidades, o FedAvg-BE busca reduzir o tempo de execução em ambientes de FL com dados non-iid, alcançando uma redução de 26% para o CIFAR-10 (Orlandi et al., 2023). Em contraste com o FedAvg-BE, que foca em minimizar

o tempo de execução, o algoritmo proposto neste estudo, EnBaSe, emprega a entropia para selecionar dados de alta qualidade antes do processamento. Por fim, a teoria da informação é aplicada ao contexto de FL para minimizar a dispersão de classes minoritárias e monitorar a concentração das classes utilizando métricas específicas (Hamidi; Tan; Ye; Yang, 2024).

Ao se comparar estes **trabalhos relacionados** ao algoritmo proposto, observa-se que a solução apresentada é um **algoritmo agnóstico** na etapa de **pré-processamento**, aplicável a cenários de **aprendizado centralizado e descentralizado (i.e., FL)**. Em um servidor centralizado, ele desempenha o papel de pré-processamento e, ao ser embarcado na borda, pode servir como pipeline para algoritmos de FL.

2.6.2 Visão Geral do Problema

O modelo FL emprega ML descentralizado, abordando desafios como gargalos de conexão, atualizações infrequentes, latência de rede e atrasos de convergência. Estes fatores têm um impacto significativo no consumo de energia, particularmente em dispositivos de baixo desempenho, como smartphones, tablets, etc. A falta de validação de novos conjuntos de dados aumenta o tempo de convergência dos modelos de redes neurais treinados e não garante que os novos dados melhorem a precisão. Algoritmos concebidos para validar a qualidade dos dados poderiam, portanto, acelerar a convergência de modelos treinados e reduzir os custos de energia para dispositivos que operam sob tais restrições.

O modelo proposto neste estudo concentra-se na validação da informação antes de iniciar o processo de treinamento da rede neural. Dessa forma, apenas os dados mais relevantes são utilizados, reduzindo o volume de informações necessárias para o treinamento e impactando diretamente nos custos associados a esse processo. Por fim, a principal diferença entre o EnBaSe e os outros modelos no estado da arte é que o EnBaSe funciona como uma camada integrada na arquitetura da rede neural na borda. Além disso, motivados por estes estudos sobre as propriedades da entropia, analisámos as distribuições das imagens.

Observámos que as amostras apresentavam um comportamento de cauda longa com valores extremos, o que contribuía para a dispersão em torno da média. O algoritmo EnBaSe proposto lida com a dispersão de valores anómalos e a distorção das caudas em dados non-iid. Além disso, a abordagem em torno da avaliação da qualidade dos dados no interior da classe não altera a dispersão da distribuição e mantém a representatividade do domínio da amostra. O impacto da abordagem é a manutenção das propriedades da distribuição estocástica e a

diminuição da computação não significativa dos nós, o que, como resultado, permite um maior desempenho do FL na computação periférica e poupa custos de energia.

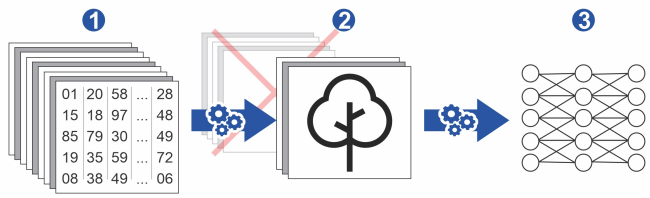
2.7 Considerações Finais

Este capítulo apresentou a fundamentação teórica, background, abordando os assuntos relacionados a este trabalho, assim como onde foram também abordados temas do estado da arte relacionados à pesquisa e trabalhos relacionados e seu impacto que foram utilizados no desenvolvimento desta pesquisa, conduzindo também uma discussão sobre o assunto.

3 MODELO PROPOSTO

Esta seção introduz um algoritmo para processamento e seleção de imagens em DL, baseado na teoria da informação. O modelo centraliza-se na entropia, uma medida chave para quantificar a informação e avaliar a incerteza em conjuntos de dados. O modelo é estruturado em três etapas principais: 1) cálculo da entropia de cada imagem em um conjunto; 2) seleção de imagens baseada na distribuição da classe, utilizando a mediana como critério; e 3) aplicação da seleção e envio dos dados a uma rede neural artificial. Conforme ilustrado na Figura 8, estas etapas formam a base do processo de seleção.

Figura 8 – Proposta de modelo conceitual.



Fonte: Elaboração Própria.

3.0.1 Hipótese

A hipótese para a elaboração do algoritmo proposto neste trabalho envolve a aplicação da entropia no campo da CV, onde cada conjunto de pixels passa a representar valores de cor ou intensidade específicos. Dessa forma, a probabilidade de cada valor de cor ou intensidade — bem como de cada ocorrência única de pixel — pode ser calculada com base no número de vezes que cada cor ou intensidade específica aparece na imagem.

Desta forma, uma alta Entropia representa uma grande diversidade de pixels, indicando alta complexidade de textura, variação significativa e pouca previsibilidade da informação. Por outro lado, uma baixa entropia indica maior homogeneidade na imagem, ou seja, melhor uniformidade na identificação de regiões com pouca ou nenhuma informação relevante, facilitando a segmentação de elementos em uma cena.

Por fim, assume-se que uma alta entropia adiciona pouco valor à rede neural e introduz ruído no processo de treinamento. A seguinte analogia ilustra essa ideia: inicialmente,

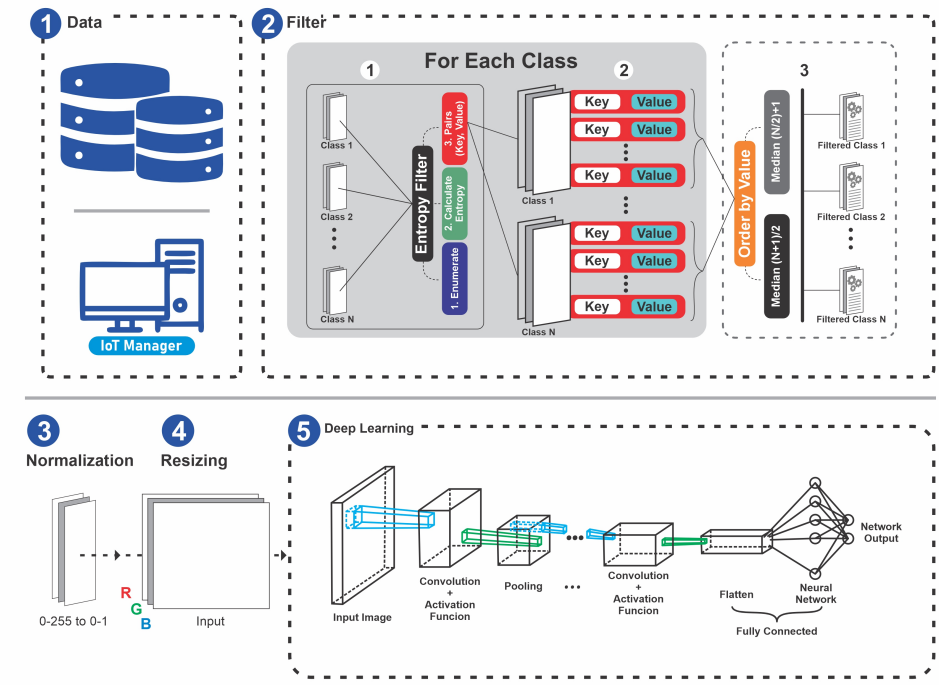
um conjunto desorganizado de imagens apresenta alta entropia e grande incerteza. Ao organizar e separar esses conjuntos, é possível dividi-los em segmentos com baixa entropia, tornando-os altamente previsíveis.

Em contraste, a outra parte exibe alta Entropia e imprevisibilidade contínua. Consequentemente, dados com alta Entropia tendem a ser vistos como de baixa qualidade ou ruidosos e, portanto, são excluídos do processo de treinamento para melhorar o desempenho do modelo.

3.1 Representação do Fluxo de Seleção a Treinamento

Detalha-se a seguir o modelo que representa o fluxo do algoritmo proposto, estruturado em cinco etapas distintas: 1) Representação de todos os conjuntos de dados de entrada, nos quais a entropia será calculada; 2) Desenvolvimento do processo de computação e seleção; 3) Implementação de processos de pré-processamento para a normalização dos dados; 4) Alteração das dimensões dos dados; 5) Treinamento da CNN, empregada no aprendizado de máquina.

Figura 9 – Representação gráfica do fluxo do algoritmo proposto em embarcado.



Fonte: Elaborado pelo autor.

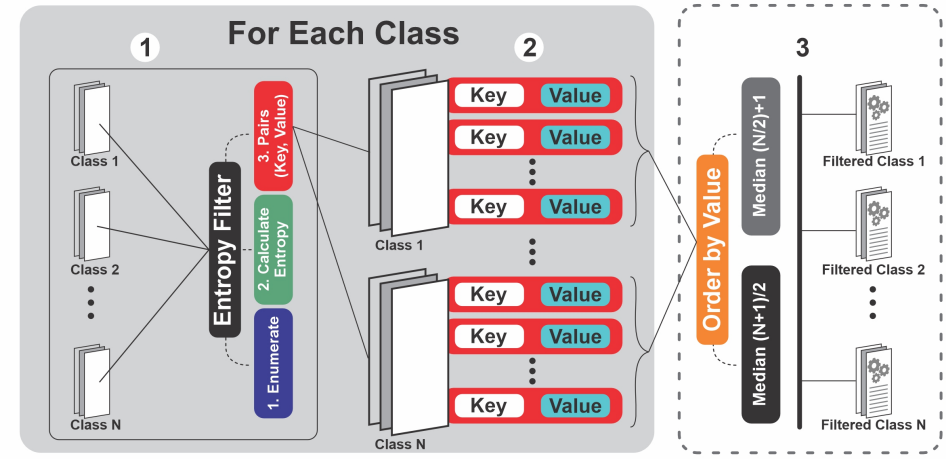
Conforme ilustrado na Figura 9, o algoritmo inicia-se com o recebimento de imagens de uma fonte de dados. Essas imagens, representadas por matrizes 2D, possuem múltiplos canais de cores *Red, Green and Blue* (RGB) ou estão em escalas de cinza. O algoritmo processa estas

imagens selecionando-as com base em um modelo construído segundo a teoria da informação. Posteriormente, são executadas operações de normalização e alteração de dimensão. Adicionalmente, técnicas como DA e TL podem ser empregadas antes da etapa de treinamento do modelo de aprendizado de máquina.

Como ilustrado na Figura 9, o principal objetivo do algoritmo é selecionar um conjunto de dados, segmentando-o em uma amostra com previsibilidade e menor grau de surpresa. Esta seleção facilita o envio dos dados para as etapas subsequentes do processo de aprendizado de máquina. A entropia desempenha um papel crucial neste contexto: uma entropia mais elevada indica um maior grau de surpresa e imprevisibilidade, implicando na necessidade de mais informações para processamento. Inversamente, uma entropia mais baixa indica um menor grau de surpresa e maior previsibilidade, resultando na necessidade de menos dados.

Conforme ilustrado na Figura 10, o algoritmo consiste nas seguintes etapas: 1) Cálculo da entropia para cada classe, resultando no número da matriz da imagem 2D, o valor de entropia correspondente e o resultado da combinação, que forma um grupo de chave e valor; 2) Neste grupo, para cada classe, cada imagem é associada a uma chave, que indica a imagem correspondente, e um valor de entropia; 3) Conforme a etapa anterior, os valores são ordenados dentro de cada classe e, posteriormente, divididos conforme sua distribuição probabilística, ou seja, a mediana, resultando nas classes selecionadas.

Figura 10 – Algoritmo proposto - Entropy-Based Selection (EnBaSe).



Fonte: Elaboração Própria.

Assim, os valores inferiores à mediana de cada classe são selecionados e armazenados como representantes dessa classe. Este procedimento é aplicado a todas as classes do conjunto de dados. Em resumo, o algoritmo organiza os dados por chave de imagem, calcula e ordena

os valores da entropia para cada matriz 2D e seleciona aqueles que indicam menor incerteza. Consequentemente, o resultado para cada classe consiste em uma amostra com menor incerteza e surpresa, indicando a necessidade de menos informação para a previsão da saída.

3.1.1 Algoritmo Proposto para Seleção dos Dados

O Algoritmo 1, conforme ilustrado abaixo, explicita o processo utilizado na metodologia proposta. Além disso, o método pode ser aplicado previamente às etapas de normalização, redimensionamento e à técnica de DA, conforme será discutido na Seção 5.

Algoritmo 1: EnBaSe. Where K denotes the total number of classes.

Require : $\mathscr{X}_{\text{train}}$, $\mathscr{Y}_{\text{train}}$, K
Ensure : Selected classes based on entropy
$\mathscr{X}_{\text{selected}} \leftarrow \emptyset$;
$\mathscr{Y}_{\text{selected}} \leftarrow \emptyset$;
for *label* $\leftarrow 0$ **to** $K-1$ **do**
  $\mathscr{C} \leftarrow$ Retrieve indices belonging to class label;
  $\mathscr{M}_{\text{Entropy}} \leftarrow \emptyset$;
  for *each sample* $\in \mathscr{C}$ **do**
    $\mathscr{M}_{\text{Entropy}} \leftarrow (\text{key}, \text{ComputeEntropy}(\text{image}))$;
  end
  Sort $\mathscr{M}_{\text{Entropy}}$ **by** ComputeEntropy(image);
  Calculate the median of $\mathscr{M}_{\text{Entropy}}$;
  $\mathscr{I}_{\text{Qualified}} \leftarrow \emptyset$;
  for *each key* $\in \mathscr{M}_{Entropy}$ **do**
    if *key.entropy* $\leq$ *median* **then**
      Append key.index to $\mathscr{I}_{\text{Qualified}}$;
    end
  end
  for $i \in \mathscr{I}_{Qualified}$ **do**
    Append $\mathscr{X}_{\text{train}}[i]$ to $\mathscr{X}_{\text{selected}}$;
    Append $\mathscr{Y}_{\text{train}}[i]$ to $\mathscr{Y}_{\text{selected}}$;
  end
end
return $\mathscr{X}_{selected}$, $\mathscr{Y}_{selected}$;
Function `ComputeEntropy(`*image*`):`
  $H \leftarrow -\sum_d p(\text{image}) \log_2(p(\text{image}))$;
  return H;

Entrada:

- *xTrain*: representa as imagens do conjunto de dados de treinamento.
- *yTrain*: são os rótulos correspondentes para essas imagens.
- K: é o número total de classes.

A seguir, exploraremos em detalhes o funcionamento do algoritmo EnBaSe, destacando suas principais características:

i. **Inicialização**: São criados dois conjuntos vazios ($\mathscr{X}_{\text{selected}}$ e $\mathscr{Y}_{\text{selected}}$) para armazenar os dados selecionados e suas respectivas classes.

ii. **Iteração sobre as classes**: O algoritmo itera por cada classe presente no conjunto de treinamento ($\mathscr{X}_{\text{train}}$ e $\mathscr{Y}_{\text{train}}$), identificando os índices associados a cada classe.

iii. **Cálculo da entropia**: Para cada elemento na classe atual, a função COMPUTEENTROPY calcula a entropia da amostra com base na distribuição de probabilidade de seus atributos. Os resultados aparecem como pares $(\text{índice}, \text{ComputeEntropy})$ em $\mathscr{M}_{\text{Entropy}}$, que também é referida como o mapa de entropia.

iv. **Ordenação e seleção baseada na mediana**: Os pares em $\mathscr{M}_{\text{Entropy}}$ são ordenados pela entropia e o valor mediano da entropia é calculado. Somente os elementos com valores de entropia menores ou iguais à mediana são selecionados, garantindo a inclusão dos dados mais representativos e menos redundantes.

v. **Atualização dos subconjuntos selecionados**: Os índices selecionados são usados para copiar os dados correspondentes de $\mathscr{X}_{\text{train}}$ e $\mathscr{Y}_{\text{train}}$ para os subconjuntos $\mathscr{X}_{\text{selected}}$ e $\mathscr{Y}_{\text{selected}}$.

vi. **Saída final**: Ao final da iteração sobre todas as classes, os subconjuntos $\mathscr{X}_{\text{selected}}$ e $\mathscr{Y}_{\text{selected}}$ contêm os dados mais informativos e homogêneos, otimizados para o treinamento do modelo.

Dessa forma, o algoritmo organiza os dados por chave de imagem, calcula e ordena os valores de entropia para cada matriz 2D, e seleciona aqueles que indicam menor incerteza. Consequentemente, o resultado para cada classe consiste em uma amostra com menor incerteza e surpresa, indicando a necessidade de menos informações para a provisão de saída.

A seguir, apresentamos uma **breve discussão** na análise dos experimentos e resultados documentados na **Seção de Resultados (Cap. 5).**

A imagem é fornecida à função como uma matriz contendo valores brutos variando de 0 a 255 ou valores normalizados entre 0 e 1. Imagens em tons de cinza e canais RGB são tratadas como uma distribuição de probabilidade unificada, em vez de distribuições separadas para diferentes canais de cor. Consequentemente, generalizamos as probabilidades associadas aos canais de cor — intensidade para o azul e contraste para o verde e o vermelho — permitindo uma análise da complexidade dos pixels sem distinções específicas de canal. Essa abstração possibilita a medição da variação geral dos pixels por meio de uma única matriz numérica.

Os valores obtidos dessa análise fornecem a complexidade visual das imagens. De acordo com a teoria da informação, quando a entropia apresenta uma variação baixa, há pouca

incerteza ou os padrões são previsíveis. Por outro lado, quando a variação é alta, isso pode indicar um excesso de detalhes ou uma previsibilidade difícil.

Observamos em experimentos que algumas classes de diferentes conjuntos de dados apresentam valores extremos de entropia, distorções. Portanto, para evitar que a média seja influenciada por esses valores extremos ao separar baixa entropia de alta entropia, escolhemos a mediana como critério de separação, pois ela minimiza o impacto desses valores extremos, evitando distorções na distribuição.

3.1.2 Formulação Matemática do EnBaSe

Nesta subseção, apresenta-se a formulação matemática baseada na hipótese descrita na subseção 3.0.1 e nos princípios da Teoria da Informação, com o objetivo de extrair dados com maior qualidade informativa (ver 2.4). Em particular, aplica-se a entropia de Shannon para quantificar o grau de desordem em cada imagem, permitindo a identificação de amostras com baixa entropia que melhor representam o domínio da rede neural. As formulações a seguir definem a base matemática do pseudocódigo apresentado na Subsecção 3.1.1.

Seja o espaço amostral I representado por uma matriz de valores de pixels de uma imagem, e $p(x_i)$ a distribuição de probabilidade dos valores de pixel x_i na imagem. Então, a entropia de Shannon $H(I)$ é definida como:

$$H(I) = -\sum_{i=1}^{d} p(x_i) \log_2 p(x_i) \tag{3.1}$$

A entropia $H(I)$ quantifica o grau de incerteza nos novos dados: quanto maior a incerteza, maior a quantidade de informação associada. Ela é calculada utilizando o logaritmo $\log_2$ e é medida em bits. Assim, um valor baixo de $H(I)$ indica um baixo grau de incerteza na imagem. Portanto, um valor baixo de entropia implica alta previsibilidade, o que beneficia o treinamento de redes neurais quando estas se especializam em um subconjunto específico de dados. Essa abordagem permite que a rede neural aprenda de forma mais eficiente com menos dados de entrada.

Dada uma classe $c \in \{1,\ldots,K\}$ com um conjunto de amostras $\mathscr{I}_c = \{I_{c_1}, I_{c_2}, \ldots, I_{c_n}\}$, o conjunto ordenado das entropias das imagens é definido como:

$$\mathscr{H}_c = \left\{ H(I_{c_j}) \;\middle|\; j = 1,\ldots,n \right\} \tag{3.2}$$

O subconjunto selecionado $\mathscr{S}_c$ da classe c é definido com base na mediana da entropia do conjunto $H(I_{c_j})$, e consiste na coleção de imagens cujas entropias são menores ou iguais a essa mediana:

$$\mathscr{S}_c = \{I_{c_j} \in \mathscr{I}_c \mid H(I_{c_j}) \leq \mathscr{H}_c\} \tag{3.3}$$

Os dados selecionados para o treinamento da rede neural são representados pela união dos subconjuntos selecionados de todas as classes $\mathscr{S}_c$:

$$\mathscr{S} = \bigcup_{c=1}^{K} \mathscr{S}_c \tag{3.4}$$

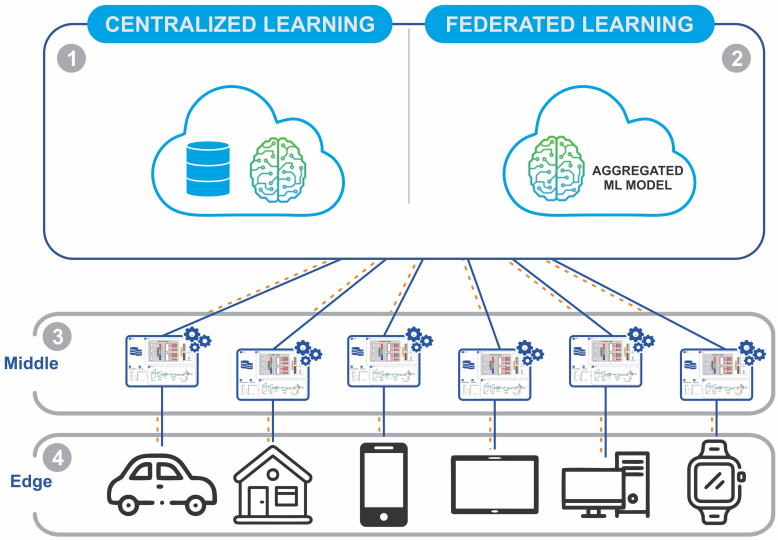
O EnBaSe seleciona amostras com base na entropia que é menor ou igual ao valor mediano de $\mathscr{H}_c$ para cada classe c, construindo assim um subconjunto mais informativo $\mathscr{S}_c$. O conjunto final de treinamento $\mathscr{S}$ é obtido como $\mathscr{S} = \bigcup_{c=1}^{K} \mathscr{S}_c$, combinando todos os subconjuntos ao longo das K classes. A maior previsibilidade e o menor ruído dos dados reduzem o problema do gradiente que desaparece e permitem que o modelo aprenda de forma mais eficaz com os dados. Como resultado, o subconjunto selecionado ($\mathscr{S}$) reduz indiretamente os custos computacionais e acelera a convergência dos modelos de rede neural.

3.2 Embedding em FL

A Figura 11 ilustra o modelo proposto, sendo embarcado no *middleware* de um dispositivo IoT, onde:

1. Figura 11 (4) O algoritmo embarcado seleciona o conjunto de dados apropriado dentro do dispositivo. Após o treinamento, o modelo envia os pesos para o servidor de agregação; e
2. Figura 11 (3) A seleção dos dados de treinamento ocorre localmente em cada dispositivo, que é acionado para iniciar o treinamento;
3. Figura 11 (1-2) O servidor, por sua vez, realiza a agregação dos modelos recebidos. Finalmente, o modelo global atualizado é distribuído a todos os dispositivos participantes. Alternativamente, os dados podem ser transferidos para um servidor centralizado, onde a rede neural é treinada.

Figura 11 – Aplicação do EnBaSe no aprendizado federado.



Fonte: Elaboração Própria.

3.2.1 Camada de Middleware e Modelo Proposto

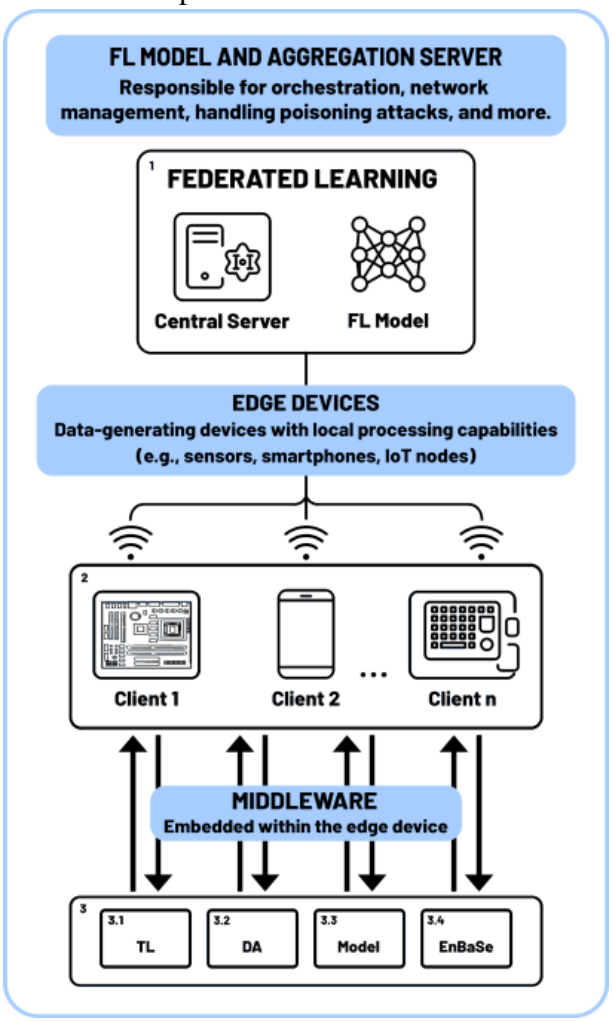
Nesta subseção, detalhamos a conclusão do algoritmo proposto, a fim de descrever os principais componentes do **protótipo de middleware** utilizado no experimento — com ênfase no **EnBaSe** — e sua **relação com a arquitetura de FL** e os testes realizados. Além disso, são especificadas as funcionalidades implementadas no middleware, destacando sua interação com o algoritmo de FL.

No **Modelo de FL e Servidor de Agregação**, representado na Figura 12 (1), são executadas funções que, embora **tipicamente atribuídas ao orquestrador do *continuum*, são frequentemente implementadas por algoritmos de FL ou por ambos**, conforme descrito a seguir:

- Orquestrar os nós clientes com base em políticas de conexão;
- Gerenciar a seleção de clientes para participação nas rodadas de treinamento;
- Manter resiliência contra falhas, como conectividade intermitente ou abandono dos clientes;
- Garantir comunicação de rede robusta entre os clientes e o servidor;
- Mitigar ataques de envenenamento e garantir a integridade do modelo;
- Detectar eventos raros e anomalias;
- Monitorar o comportamento dos clientes, respeitando princípios de privacidade;
- Aplicar protocolos de segurança, como criptografia e autenticação;
- Aplicar integração com blockchain (quando aplicável); e

- Definir a estratégia de agregação dos pesos dos modelos dos clientes.

Figura 12 – Arquitetura do algoritmo proposto com middleware embarcado para aprendizado federado.



Fonte: Elaboração Própria.

Na Figura 12 (2), a **camada de borda** é definida como a região do *continuum* onde os dados podem ser processados localmente, **atendendo a demandas que não podem ser satisfeitas por nuvens centralizadas**. Essa camada representa a **capacidade de decidir onde processar os dados, gerenciar recursos físicos e virtuais** e executa aplicações especializadas que dão suporte a domínios específicos (Al-Dulaimy et al., 2024).

Complementarmente, essa camada executa suas funções onde a computação ocorre, frequentemente, a poucos saltos de distância dos dispositivos IoT, ou até mesmo *embutida no próprio dispositivo conectado* (Ullah et al., 2024).

Na Figura 12 (3), conforme definido por (Ullah et al., 2023), a Camada 3 representa as *Soluções de Nível Inferior – Middleware*. Em outras palavras, **são soluções que atuam**

sobre uma camada de abstração subjacente e de baixo nível, relacionada à **configuração de recursos** e à infraestrutura a ser utilizada. **Essas soluções não fornecem funções essenciais de orquestração, como implantação distribuída e reconfiguração dinâmica.**

Assim, essa camada é responsável por:

- Gerenciar os dados localmente;
- Realizar inferências locais com base nos modelos recebidos;
- Executar pré-processamento inteligente conforme especificações da camada superior;
- Integrar-se a soluções de orquestração superiores, como a FL;
- Executar algoritmos de *Transfer Learning* sob demanda;
- Executar *Data Augmentation* com parâmetros definidos estaticamente ou dinamicamente;
- Utilizar o algoritmo *EnBaSe*, quando aplicável;
- Receber e executar modelos enviados pela camada superior; e
- Coletar os resultados dos modelos treinados.

Assim, o **protótipo de middleware** implementado no experimento tem a função de receber diferentes algoritmos de FL, conforme demonstrado na Figura 12 e descrito a seguir:

- Figura 12 (3-1): Recebimento da arquitetura da rede neural e desbloqueio da arquitetura de TL, conforme instruções enviadas ao dispositivo responsável pelo treinamento.
- Figura 12 (3-2): Execução de técnicas de DA localmente, conforme a demanda e requisição recebidas, seja com base em modelos adaptativos ou em conjuntos específicos de instruções, quando solicitado.
- Figura 12 (3-3): Recebimento de uma rede neural previamente especificada para os dispositivos, permitindo o treinamento com arquiteturas distintas de redes neurais; e
- Figura 12 (3-1): Executar o algoritmo proposto, EnBaSe, **a partir de duas perspectivas de execução**: a **Execução Normal** (ou seja, do algoritmo de FL sem o EnBaSe) e a execução **com EnBaSe**.

Dessa forma, concluímos a apresentação das **funcionalidades da arquitetura gerenciada na qual o EnBaSe atua**, por meio do desenvolvimento de um **protótipo de middleware** voltado à **camada inferior da arquitetura**.

3.3 Conclusão deste Capítulo

Este capítulo descreveu o modelo proposto para ambientes FL, modelo conceitual e fluxo. Inclui também detalhes do algoritmo desenvolvido, seu pseudocódigo e relação com

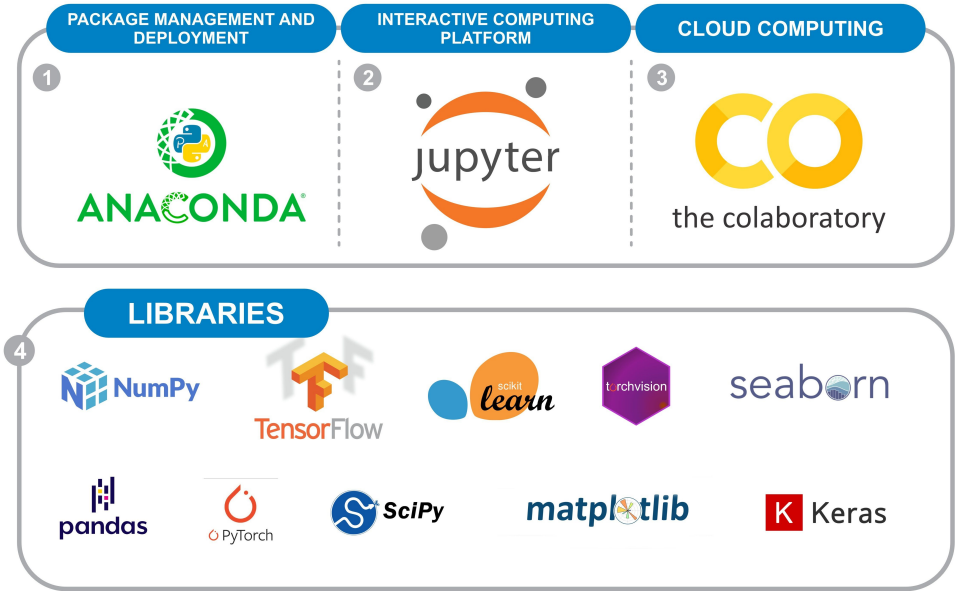
dados não identicamente distribuídos non-iid.

4 MATERIAIS E MÉTODOS

4.1 Configuração do Experimento

O objetivo desta subseção é fornecer uma visão geral das abordagens metodológicas usadas para realizar este experimento. Na Figura 13, é apresentada uma visão geral das ferramentas e bibliotecas utilizadas para compor e gerenciar o ambiente de programação para treinamento de modelos, tanto em ambientes centralizados quanto em **Computação em Nuvem (CN)** (**1, 2, 3, 4**). Os códigos-fonte estão disponíveis no Github [1] para fins de reprodução.

Figura 13 – Configurações específicas.



Fonte: Elaboração Própria.

Para a Figura 13 (1), inicialmente no escopo desta pesquisa, utilizamos o Anaconda, que é uma distribuição que engloba as linguagens Python e R, usadas em ciência de dados e computação científica. Para a Figura 13 (2), utilizamos o Jupyter Notebook para criação de códigos, equações e visualizações. Este ambiente também permite análises científicas em ciência de dados, modelagem estatística e aprendizado de máquina. Para a Figura 13 (3), é utilizado o Google Colab, uma plataforma na nuvem para execução de Jupyter Notebooks com acesso a recursos de hardware.

[1] <https://github.com/ernesto-arq/Experiments-with-entropy-and-Artificial-Intelligence>

4.1.1 Setting Details

Para determinar a arquitetura mais influente em termos de custo-benefício, realizamos uma análise comparativa usando o conjunto de dados MNIST, cujos resultados são apresentados na Tabela 6. Essa comparação nos permite identificar qual configuração oferece o melhor equilíbrio entre desempenho e investimento em nosso experimento. Dessa forma, escolhemos a arquitetura T4 devido ao desempenho e custo computacional.

Tabela 6 – Comparação de Arquiteturas de GPU e Ambiente.

Ambiente	**RAM (GB)**	**GPU (GB)**	**Disco (GB)**	**Tempo Total (s)**	**Tempo Médio (s)**	**Dif. (CPU) (%)**
CPU	51	-	225.8	573.65	57.37	-
TPU	35.2	-	225.8	389.55	38.96	32.0
T4	12.7	15.0	201.2	367.78	36.78	36.0
V100	51	16.0	201.2	269.60	26.96	53.0
A100	83.5	40.0	201.2	261.14	26.11	54.0

Fonte: Elaboração Própria.

4.1.2 Datasets

Durante a revisão bibliográfica, encontramos, no período de 2021 a 2023, 23 artigos com *datasets* usados em experimentos de IoT com FL, envolvendo MNIST, Fashion-MNIST, IMDb, Reuters, SVHN, USPS, Office-31, Bing-Caltech256, COREL5000, CIFAR-10, CIFAR-100, *Agricultural pest images*, Clothing1M, Fed-ISIC2019, LEAF, *Adult Income*, *Body Signal of Smoking*, 25PDB, FC699, D1189, D640, TinyImageNet, simulações físicas e matemáticas. Assim, escolhemos os **mais comumente utilizados (MNIST, Fashion-MNIST, CIFAR-10 e CIFAR-100)**. A Tabela 7 representa os conjuntos de dados escolhidos:

Tabela 7 – *Datasets* Selecionados.

Conjunto de Dados	**Descrição**	**Formato**	**Treinamento/Teste**
MNIST	Dígitos manuscritos	28x28	50.000/10.000
Fashion-MNIST	Vestuário	28x28	60.000/10.000
CIFAR-10	Diversos	32x32	50.000/10.000
CIFAR-100	20 Superclasses	32x32	50.000/10.000

Fonte: Elaboração Própria.

Onde:

- **MNIST:** Compreende imagens de dígitos escritos à mão (0-9), em escala de cinza, divididos em conjuntos de treinamento e de teste;

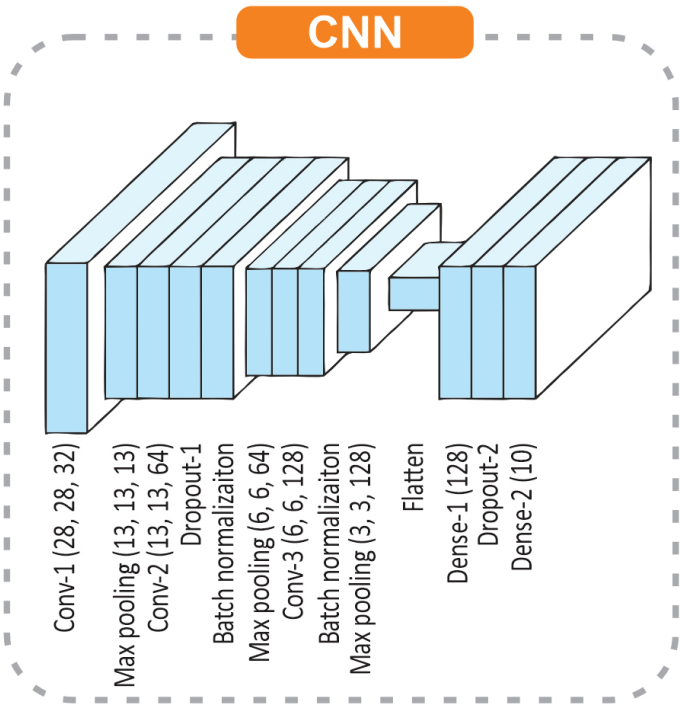
- **Fashion-MNIST:** Este conjunto inclui imagens de itens de vestuário em dez categorias distintas, tais como camisas e calças, todas em escala de cinza;
- **CIFAR-10:** Contém imagens coloridas distribuídas em dez categorias variadas, que incluem automóveis e animais; e
- **CIFAR-100:** Este conjunto contém 100 classes que abrangem desde pessoas até elementos naturais.

4.2 Arquitetura das Redes Neurais para o cenário IID

A seguir, apresentamos a arquitetura específica das redes neurais profundas criadas para cada conjunto de dados escolhido. As descrições servem como modelo para a replicabilidade deste trabalho e dos experimentos. As configurações das redes neurais, assim como os hiperparâmetros, foram desenvolvidas por meio de experimentação empírica, até que a rede neural demonstrasse equilíbrio entre a acurácia e a perda, evidenciando capacidade de generalização.

A arquitetura CNN é apresentada na Figura 14 para MNIST sem aplicação de TL ou DA. O otimizador escolhido foi o SGD, com taxa de aprendizado de 0,01 e momento de 0,9. A função de perda aplicada é a entropia cruzada categórica, com *batch* de 32 e 10 *epochs*.

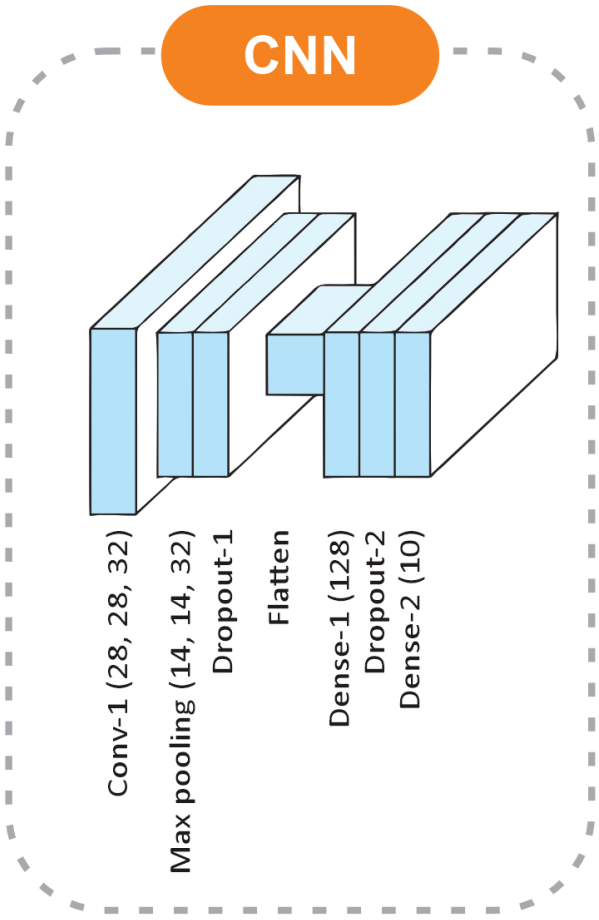
Figura 14 – Arquitetura CNN aplicada ao MNIST.



Fonte: Elaboração Própria.

Para o Fashion-MNIST foi criada uma CNN apresentada na Figura 15 sem aplicação de TL ou DA. O otimizador escolhido foi o SGD, com taxa de aprendizado de 0,01 e momento de 0,9. A função de perda aplicada é a entropia cruzada categórica, com *batch* de 128 e 10 *epochs*.

Figura 15 – Arquitetura CNN aplicada ao Fashion MNIST.

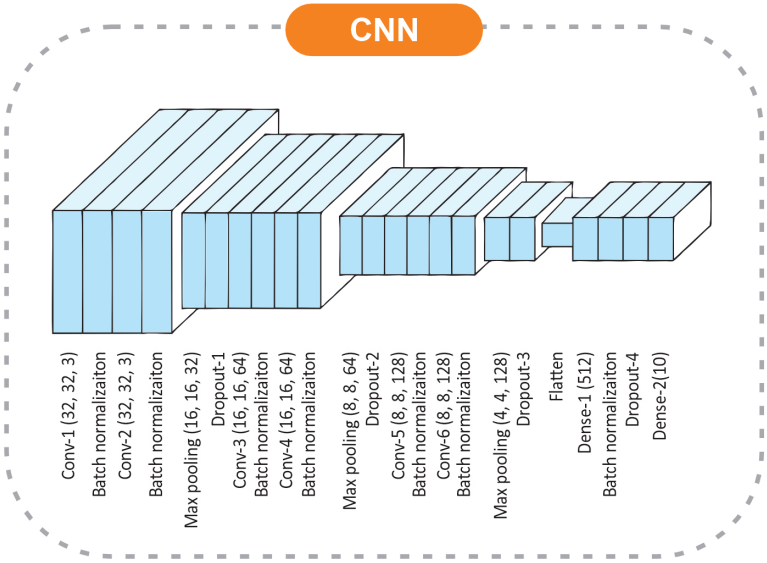


Fonte: Elaboração Própria.

Para o CIFAR-10, a técnica de DA inclui operações de rotação de até 15 graus, inversão horizontal e ajustes de até 10% na largura e altura das imagens. Além disso, foi criada uma CNN apresentada na Figura 20. O otimizador escolhido foi o *Adam*, com uma taxa de aprendizado de $0,001$, β_1 de $0,9$, β_2 de $0,999$ e ε de 1×10^{-8}. A função de perda utilizada foi a entropia cruzada categórica, com *batch* de 128 e 50 *epochs*.

Por fim, para o CIFAR-100 com arquitetura representada na Figura 20, TL é aplicada usando a ResNet50. Na configuração, a camada superior é removida, os pesos são inicializados com *ImageNet* e o formato de entrada definido é (224, 224) com 3 canais RGB. A técnica de DA

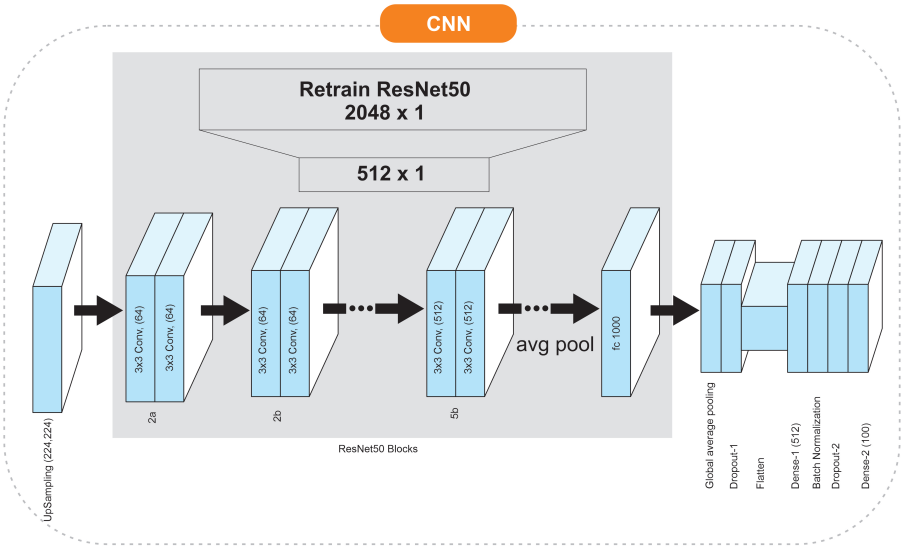
Figura 16 – Arquitetura CNN aplicada ao CIFAR-10.



Fonte: Elaboração Própria.

é aplicada com transformações nas imagens, como rotação de até 15 graus, inversão horizontal e ajustes de tamanho. O otimizador escolhido foi o SGD, com taxa de aprendizado de 1×10^{-3} e momento de 0,9. A função de perda aplicada é a entropia cruzada categórica. O treinamento foi realizado em *batches* de 128, durante 50 *epochs*. Um *callback* foi aplicado durante o processo de treinamento para monitoramento e ajustes automáticos dos hiperparâmetros.

Figura 17 – Arquitetura CNN aplicada ao CIFAR-100.



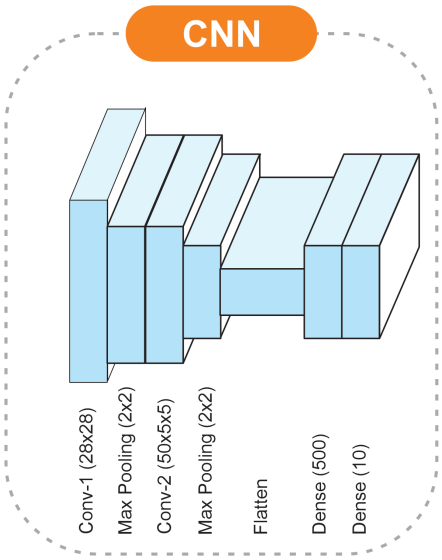
Fonte: Elaboração Própria.

4.3 Arquitetura das Redes Neurais para o cenário non-iid

Para os conjuntos de dados MNIST e Fashion-MNIST, foram utilizadas CNNs sem aplicação de TL ou DA, otimizadas por SGD. Na normalização do MNIST, são utilizados uma média de 0,1307 e um desvio padrão de 0,3081. No Fashion-MNIST, os parâmetros são uma média de 0,2860 e um desvio padrão de 0,3530, ambos com *batch* de 64.

A arquitetura CNN é apresentada na Figura 18 para MNIST e Fashion-MNIST sem aplicação de TL ou DA. O otimizador escolhido foi o SGD, com taxa de aprendizado de 0,01 e momento de 0,5.

Figura 18 – Arquitetura CNN aplicada ao MNIST e Fashion-MNIST.



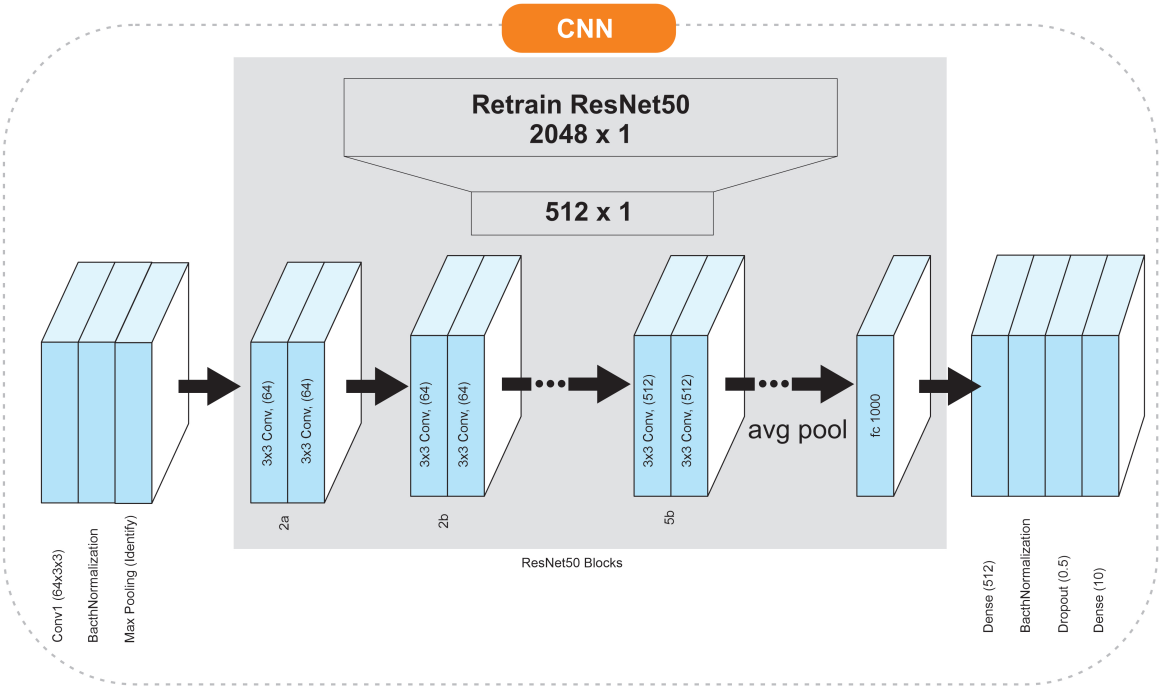
Fonte: Elaboração Própria.

Para os conjuntos de dados CIFAR-10 e CIFAR-100, foi utilizado o modelo ResNet-50 adaptado, incorporando DA. As técnicas de DA utilizadas incluem inversão horizontal, rotação com um limite máximo de 15 graus e uma transformação afim aleatória (0, (0,1; 0,1)), sendo o modelo otimizado por SGD, ambos com *batch* de 64. Na normalização do CIFAR-10, são utilizados valores médios de 0,4914, 0,4822 e 0,4465, e desvios padrão de 0,2023, 0,1994 e 0,2010, para os canais RGB. No CIFAR-100, as médias são 0,5071, 0,4867 e 0,4408, e os desvios padrão são 0,2675, 0,2565 e 0,2761, também para os canais RGB.

A arquitetura CNN foi utilizada para os conjuntos de dados CIFAR-10 e CIFAR-100, conforme apresentadas na Figura 19. O treinamento foi realizado com 50 *epoch*, com uma taxa

de aprendizado de 0,01 e momento de 0,5.

Figura 19 – Arquitetura CNN aplicada ao CIFAR-10. e CIFAR-100.



Fonte: Adaptado de (Wang; Zhu, 2023)

As arquiteturas foram inicialmente testadas no CIFAR-100, onde apresentaram desempenho estável. Em seguida, foram aplicadas ao CIFAR-10, obtendo também resultados satisfatórios. O mesmo procedimento foi repetido para os conjuntos Fashion-MNIST e MNIST, com resultados igualmente aceitáveis.

4.4 Configuração do Experimento IID

Os conjuntos de dados do MNIST, Fashion-MNIST, CIFAR-10 e CIFAR-100 foram divididos da seguinte forma:

- Alocação de dados para validação:
  - **MNIST**: 12.000 dados, correspondendo a 20% do total.
  - **MNIST-Fashion**: 12.000 dados, correspondendo a 20% do total.
  - **CIFAR-10**: 5.000 dados, correspondendo a 10% do total.
  - **CIFAR-100**: 5.000 dados, correspondendo a 10% do total.

A diferença na proporção de dados alocados para validação reflete a menor quantidade disponível e a maior complexidade inerente aos conjuntos CIFAR-10 e CIFAR-100.

4.5 Configuração do Experimento non-iid

Para a criação do cenário non-iid no FL, em que os dados não são independentes e identicamente distribuídos, o método de criação de conjuntos de dados baseia-se na assimetria da distribuição das características, na assimetria da distribuição das etiquetas e na assimetria da quantidade (Li et al., 2021); estes tipos de distribuições enviesadas implicam características de dados heterogêneos (Lee; McLachilan, 2022), conforme apresentadas no GitHub.

Para criar esse tipo de distribuição, foram aplicadas técnicas que consideram cada tipo de assimetria — características, etiquetas e quantidade de dados (Zhang et al., 2022; Sheng et al., 2023). Esses aspectos, como descrito a seguir, contribuem para a criação de um ambiente heterogêneo de FL, permitindo a diversidade de contribuições e características:

i. **Distorção de Características:** A distorção de características refere-se ao desequilíbrio entre diferentes quantidades de etiquetas em vários clientes relativamente a um cliente específico. Por exemplo, o mesmo caráter pode ser escrito em vários estilos, como variações na largura do traço ou na inclinação, o que leva a representações heterogêneas das mesmas etiquetas.
ii. **Distorção de Etiquetas:** A distorção de etiquetas ocorre quando diferentes clientes (nós) em locais distintos apresentam distribuições variadas devido a diferenças demográficas. Estas variações resultam de fatores demográficos e contextuais que afetam a frequência de ocorrência de etiquetas em cada cliente (nó).
iii. **Distorção de Quantidade:** A distorção de quantidade refere-se a um desequilíbrio no número de etiquetas específicas de um cliente, o que afeta a quantidade de dados disponíveis para um único cliente (nó). Este desequilíbrio cria uma sub-representação ou uma sobre-representação de etiquetas específicas, afetando assim o equilíbrio do modelo.

Desta forma, o objetivo é garantir que qualquer variação entre grupos (nós) resulte de um fator aleatório e não de um fator sistemático, refletindo assim a variabilidade inerente aos ambientes IoT ao permitir que os diferentes nós contribuam de forma desigual. Assim, é apresentada uma representação mais realista dos diversos dados encontrados nos dispositivos IoT do mundo real, representando as condições heterogêneas.

4.5.0.1 Algoritmo de Agregação Global

Os algoritmos de agregação global FedAvg (McMahan et al., 2017) e FedProx (Li et al., 2020) foram selecionados como os modelos clássicos que servem de *baseline* no estado da arte, proporcionando uma compreensão mais abrangente dos resultados. A seguir, são descritos os algoritmos de agregação utilizados em FL, que representam parte deste trabalho:

i. **FedAvg**: calcula a média ponderada de cada atualização do cliente. Neste cálculo, os pesos correspondem tipicamente ao volume de dados que cada cliente possui, ajustando assim quaisquer desequilíbrios no conjunto de dados; e

ii. **FedProx**: o FedProx introduz uma nova abordagem para a função de perda local, adicionando um termo de regularização. Esse termo penaliza ajustes nos pesos do modelo local que se desviam significativamente dos pesos do modelo global. Assim, o FedProx procura minimizar o impacto da heterogeneidade dos dados e dos dispositivos, promovendo uma aprendizagem mais harmoniosa

4.6 Métricas e Técnicas de Avaliação

Esta seção avalia os métodos para identificar uma rede neural imparcial e de alta qualidade. O processo de avaliação está organizado da seguinte forma:

- **Precisão:** A precisão mede a proporção de previsões corretas em relação ao total de amostras, calculada dividindo os acertos pelo total de amostras.
- **Recall:** O recall avalia a proporção de verdadeiros positivos corretamente identificados, calculado dividindo os números positivos precisos pelo total de números positivos mais os falsos negativos.
- **F1-Score:** O F1-Score é o valor harmônico entre precisão e recall, indicando um equilíbrio entre eles. Valores mais altos indicam melhor desempenho do modelo. Neste caso específico, será usado como uma métrica adicional no contexto de FL devido à alta heterogeneidade dos dados.
- **Perda:** A função de perda mede o erro entre previsões e resultados típicos, com funções específicas para cada problema (por exemplo, entropia cruzada para classificação). O objetivo é minimizar essa perda para melhorar o modelo.
- **Validação Cruzada:** A validação cruzada é utilizada para avaliar a capacidade de generalização de um modelo. Assim, o conjunto de dados é dividido em *folds*, sendo o modelo

treinado e validado em diferentes combinações. Cada *fold* pode apresentar variações na quantidade, qualidade e distribuição dos dados (até o desenvolvimento da presente pesquisa, não existe modelo capaz de avaliar K-*folds* em ambiente FL, mantendo a privacidade, então o método está restrito apenas ao aprendizado centralizado).

- **Curva de Aprendizado:** Representa o desempenho do modelo ao longo do tempo, comparando treinamento e validação para detectar *overfitting* e *underfitting*, e verificar a convergência do modelo.
- **Matriz de Confusão:** A matriz de confusão resume os valores previstos e os erros de classificação, desta forma, permite que ela indique o equilíbrio e desequilíbrio de classificação do modelo (até o desenvolvimento da presente pesquisa, não existe modelo capaz de criar matriz de confusão de *n* dispositivos IoT em ambiente FL, mantendo a privacidade, então o método está restrito apenas ao aprendizado centralizado).

4.7 Parametros de Comparação de Performance

Nesta subseção, descrevemos a metodologia de comparação dos experimentos realizados neste trabalho, abordando a comparação entre o conjunto de dados completo, o algoritmo proposto e um modelo matemático do estado da arte, utilizado como *baseline* de performance.

Utilização do conjunto de dados: Inicialmente, o conjunto de dados completo é utilizado para analisar o desempenho, fornecendo uma visão geral abrangente das capacidades da rede neural (no experimento, será denominado como Execução Normal).

Método de seleção aleatória: Foi aplicado um método de seleção aleatória na entrada dos dados para garantir a diversidade, a imparcialidade e a robustez da rede neural, conforme discutido no *Backgournd* Subseção 2.2.3. Esta técnica foi aplicada por dois motivos importantes no experimento: primeiro, para comparar o EnBaSe com o modelo matemático que busca garantir a robustez da rede neural; segundo, porque este método também selecionará metade dos conjuntos de dados das classes, permitindo medir o grau de eficácia do algoritmo EnBaSe em termos de custo computacional e performance.

4.8 Considerações Finais

Neste capítulo, foi apresentada a metodologia deste trabalho, juntamente com os conjuntos de tecnologias, os conjuntos de dados, as técnicas envolvidas, assim como as métricas

utilizadas na avaliação.

5 RESULTADOS

Neste capítulo, detalhamos os resultados obtidos com o método proposto na Seção 3, juntamente com a metodologia de métricas e técnicas apresentada na Seção 4. Iniciamos o experimento com o algoritmo proposto neste trabalho, analisando o comportamento da entropia sob a ótica de transformações matemáticas, distribuição, cenário iid, e, em seguida, no ambiente FL com dados non-iid, apresentando alta heterogeneidade.

5.1 Resultados do Experimento com Entropia

5.1.1 Custo Computacional

Foram coletados os tempos médios do total de 10 simulações para cada conjunto de dados, conforme são representados na Tabela 8.

Tabela 8 – Tempos Médios para o Cálculo de Entropia.

Dataset	**Tempo Médio Total (s)**	**Tamanho do Dataset**	**Imagens por (s)**
MNIST	≈ 2.771	60.000	≈ 21652.8
Fashion MNIST	≈ 3.216	60.000	≈ 18656.7
CIFAR-10	≈ 5.578	50.000	≈ 8963.8
CIFAR-100	≈ 5.499	50.000	≈ 9092.56

Fonte: Elaboração Própria.

Assim, a Tabela 8 representa o custo computacional da entropia em segundos, indicando quanto tempo demora um computador, em média, para realizar os cálculos necessários para a separação dos conjuntos de dados em baixa entropia e alta entropia, conforme discutido no algoritmo proposto.

5.1.2 Entropia e Normalização de Dados

A análise comparativa da entropia antes e após a normalização, que consiste na divisão dos valores por 255, equivalente aos tamanhos dos pixels, tem como objetivo avaliar o impacto dessa normalização na seleção de amostras, tanto pré quanto pós-aplicação do cálculo da entropia.

Observou-se que a entropia, sendo a soma das probabilidades, resultou em valores quase idênticos, com média de 14ª casas decimais similares. Isso representa, em média, 75% de equivalência nos valores antes e após a normalização. Nos 25% restantes, observou-se uma

variação mínima, geralmente no 13º ao 15º dígito após a vírgula, conforme demonstrado na Tabela 9.

Tabela 9 – Comparação da Entropia Antes e Depois da Normalização.

Índice da Imagem	Entropia Antes da Normalização	Entropia Depois da Normalização
27582	5,150483033018236	5,150483033018237
5760	5,242222088792437	5,242222088792438
29284	5,247872784912092	5,247872784912093
4484	5,291615062825755	5,2916150628257554
18188	5,3789068676650285	5,3789068676650290
18304	5,3941323322039985	5,3941323322039980
37629	5,402272628167595	5,402272628167594
27772	5,4121732428076275	5,4121732428076270
26492	5,4147344664719230	5,4147344664719235
53310	5,418346152065549	5,418346152065550

Fonte: Elaboração Própria.

Para a verificação da propriedade discutida na tabela 9, avaliamos a entropia de dois conjuntos de dados, em formas normalizada e não normalizada. Foi utilizada uma função para análise com variável, para registrar diferenças entre os conjuntos A e B, e uma variável para contar ocorrências de valores idênticos em cada imagem. Um acumulador soma as diferenças decimais, e outro registra se os valores de cada imagem são iguais; se não, registramos o par e somamos a diferença decimal ao acumulador.

Tabela 10 – Análise dos Dados Normalizados.

Conjunto de Dados	Conjunto de Dados	Idênticos	Diferentes	Porcentagem (%)
MNIST	60.000	45.812	14.188	76,36%
Fashion-MNIST	60.000	43.230	16.770	72,05%
CIFAR-10	50.000	36.377	13.623	72,75%
CIFAR-100	50.000	36.312	13.688	72,62%

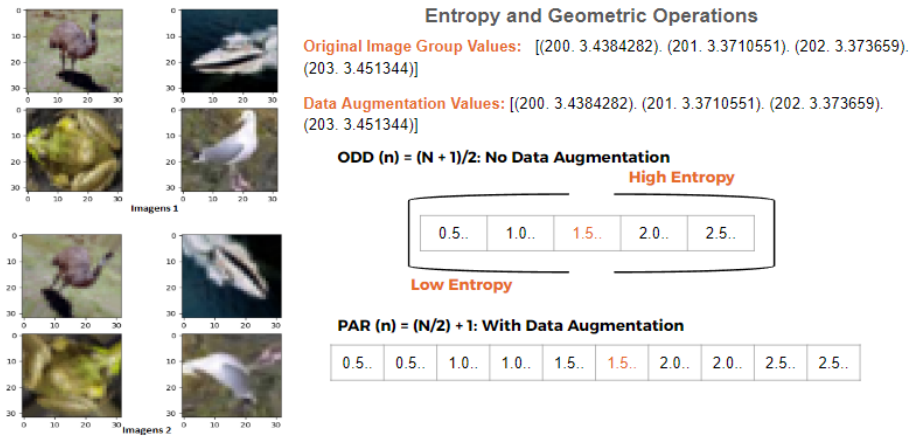
Assim, concluímos que os elementos são, em média, 72% a 76% idênticos quando aplicada a normalização, com variação entre a 14ª e a 15ª casa decimal, atribuindo essa possível variação ao fenômeno conhecido como erro de ponto flutuante ou imprecisão numérica.

5.1.3 Entropia e DA

Na Figura 20, são apresentadas imagens antes e depois da aplicação da técnica de DA, descritas na figura como *Original Image Groupe Values* e *Data Augmentation Values*. Com

base nisso, também é demonstrado que os valores obtidos do DA seriam apenas duplicados quando o algoritmo separa a amostra a partir da mediana, resultando em uma seleção equivalente dos mesmos valores.

Figura 20 – Técnica de aumento de dados com o algoritmo proposto. Imagens extraídas do *dataset* CIFAR-100.



Fonte: Elaborado pelo autor.

Para entender melhor a Figura 20 é preciso observar que os números das imagens (200, 201, 202 e 203) são mostrados na figura, com seus respectivos valores de entropia (por exemplo, valor original e o valor da entropia com DA). A entropia mede o grau de desordem ou incerteza em um conjunto de dados representado pelo valor dos pixels, intensidade e sua distribuição na matriz de entrada. Dessa forma, transformações geométricas como rotação, escala e translação não alteram o conteúdo de sua distribuição, permanecendo com a mesma distribuição estatística.

Isso permite concluir que o algoritmo EnBaSe pode ser aplicado primeiro e, em seguida, aplicado DA, assim economizando processamento de ampliação de dados em subconjuntos que não serão selecionados.

5.2 Comportamento da Entropia em Conjuntos de Dados de Imagem

Utilizando a técnica discutida na seção 4.5 para adicionar viés aos conjuntos de dados e observar a forma como a entropia lidava com a heterogeneidade em um conjunto de dados, utilizou-se a criação de um histograma para observar as propriedades de sua distribuição estatística, utilizando os testes de Shapiro-Wilk, Kolmogorov-Smirnov e D'Agostino and Pearson, apresentados na Tabela 11, onde foram realizadas 5 simulações com 400 nós (clientes).

Tabela 11 – Resultados dos testes estatísticos para diferentes conjuntos de dados.

Dataset	**Shapiro-Wilk**	**Kolmogorov**	**D'Agostino**
MNIST	344	399	342
Fashion	181	391	223
CIFAR-10	37	318	71
CIFAR-100	7	203	17

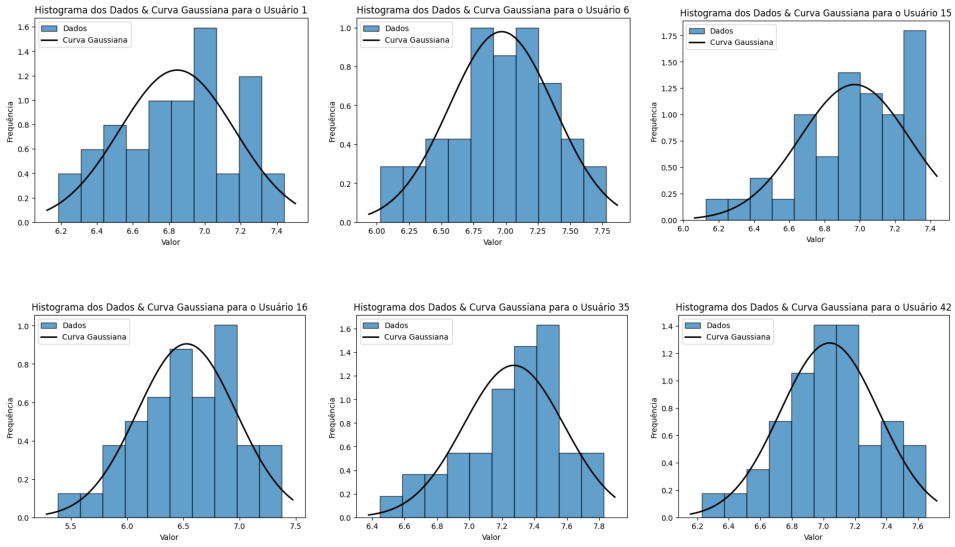
Fonte: Elaboração Própria.

Concluiu-se que:

- **MNIST:** A distribuição de classes no MNIST é homogênea e equilibrada. A entropia tende a apresentar uma distribuição mais próxima de uma curva gaussiana.
- **Fashion-MNIST:** No Fashion-MNIST, a distribuição gaussiana se apresenta de forma uniforme em uma quantidade considerável de clientes.
- **CIFAR-10:** No conjunto CIFAR-10, uma pequena proporção dos clientes exibiu dados normalmente distribuídos.
- **CIFAR-100:** Para o CIFAR-100, em algumas métricas estatísticas, um número mínimo de clientes exibiu dados normalmente distribuídos. Este conjunto se revelou o mais desafiador em termos de aproximação à distribuição gaussiana.

Na Figura 21, são apresentados os histogramas das distribuições do conjunto de dados MNIST e os valores da entropia das imagens após a aplicação do algoritmo EnBaSe. Nesse experimento, busca-se determinar a entropia global de todo o conjunto de dados.

Figura 21 – Aplicação da seleção por entropia para o *dataset* MNIST.

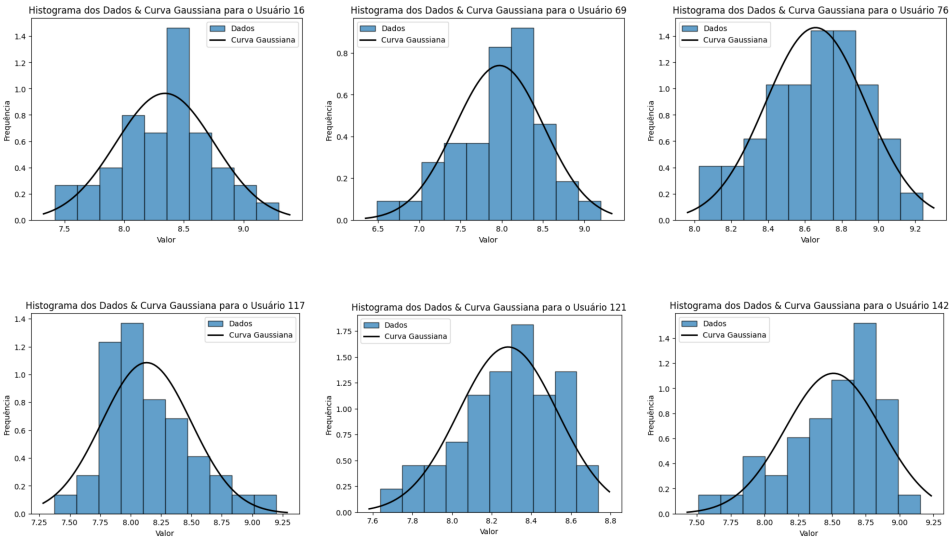


Fonte: Elaboração Própria.

Observou-se que, em um cenário com um conjunto de dados mais simplificado, a entropia exerce uma influência significativa na distribuição dos dados, assemelhando-se à curva

gaussiana. Essa tendência é claramente observada nas imagens, onde a aplicação do EnBaSe em dados desbalanceados revela uma aproximação à distribuição gaussiana no conjunto de dados MNIST. De forma semelhante, na Figura 22, ilustramos a distribuição do conjunto de dados Fashion-MNIST.

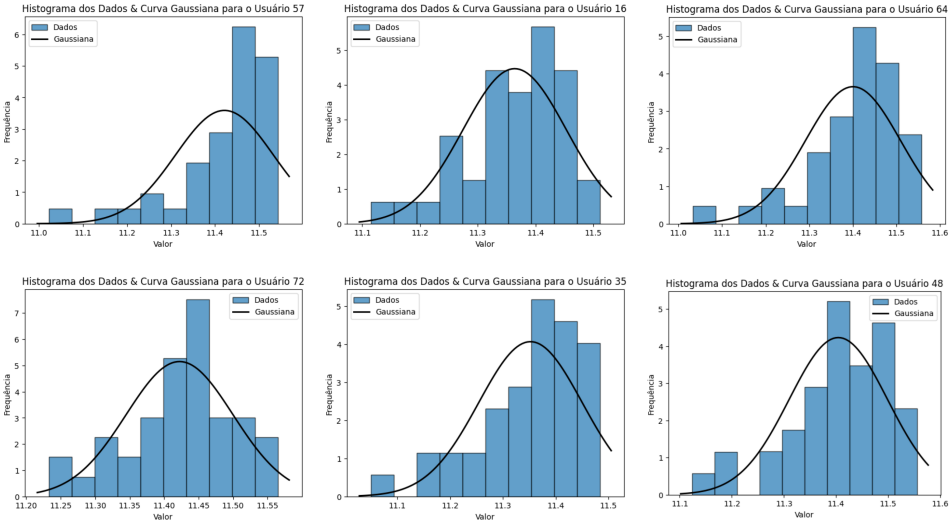
Figura 22 – Aplicação da seleção por entropia para o *dataset* Fashion-MNIST.



Fonte: Elaboração Própria.

No Fashion-MNIST, um conjunto de dados inspirado no MNIST, mas concebido para ser mais desafiador, verifica-se que o método de seleção de imagens adotado resulta em uma distribuição gaussiana menos uniforme nos conjuntos de dados de cada máquina selecionada. De forma similar, o mesmo experimento é ilustrado na Figura 23 para o CIFAR-10.

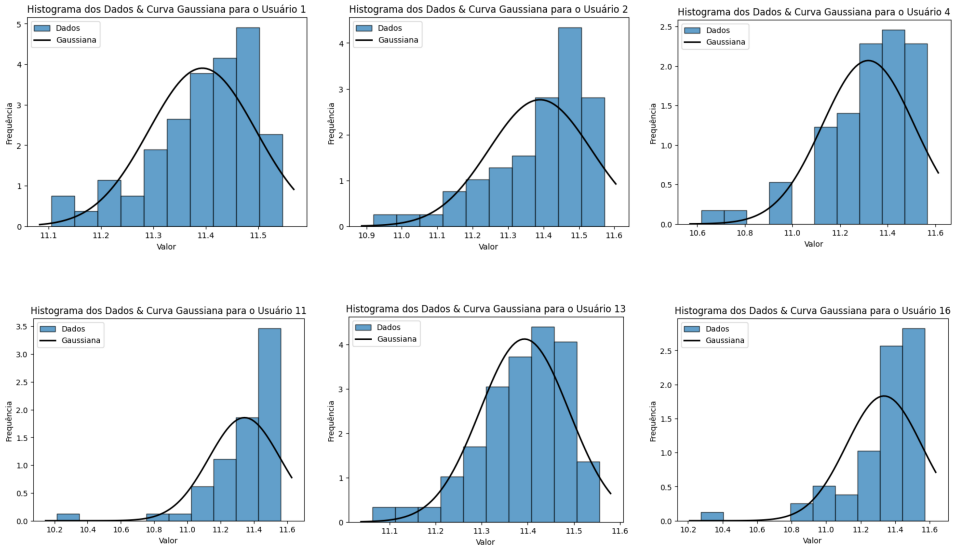
Figura 23 – Aplicação da seleção por entropia para o *dataset* CIFAR-10.



Fonte: Elaboração Própria.

Nos testes estatísticos aplicados ao conjunto de dados CIFAR-10, observou-se uma redução significativa na robustez da distribuição estatística, caracterizando-o como um conjunto mais desafiador. Por fim, o experimento é replicado no CIFAR-100, como representado na Figura 24.

Figura 24 – Aplicação da seleção por entropia para o *dataset* CIFAR-100.



Fonte: Elaboração Própria.

Para o conjunto de dados CIFAR-100, considerado o mais desafiador dentre os analisados, os resultados foram semelhantes aos observados em todos os experimentos anteriores. Estatisticamente, alguns testes indicam uma aproximação a uma distribuição gaussiana, enquanto o gráfico do histograma demonstra algumas dessas características.

5.3 Resultados do experimento com algoritmo EnBaSe

Nesta seção, são apresentados os resultados obtidos pelo algoritmo proposto para os conjuntos de dados MNIST, Fashion-MNIST, CIFAR-10 e CIFAR-100 no cenário iid. A motivação deste experimento baseia-se na hipótese de que, caso o algoritmo não seja capaz de desempenhar com eficiência em um ambiente com dados homogêneos, dificilmente será capaz de lidar com a heterogeneidade dos dados em uma simulação de um ambiente IoT real, com alto grau de heterogeneidade de dados non-iid. Consequentemente, o algoritmo EnBaSe será exposto ao cenário heterogêneo com dados non-iid, conforme apresentado na Subseção 4.5 e 4.5.0.1. Nesta seção, a análise inclui a descrição dos resultados após a aplicação do EnBaSe, utilizando as métricas estabelecidas na Subseção 4.6.

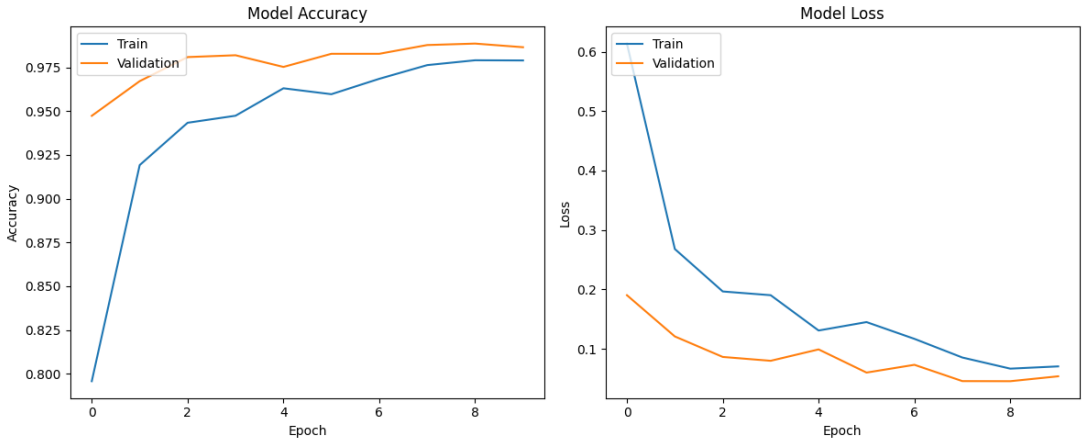
5.3.1 EnBaSe aplicado no cenário iid

Esta seção discutirá os resultados obtidos a partir do experimento conduzido sob o cenário iid. Prosseguiremos com uma apresentação detalhada dos resultados alcançados, conforme especificado na Subseção 4.6. Um total de 120 experimentos foi conduzido, distribuído igualmente entre os conjuntos de dados selecionados, agrupados nas categorias Execução Normal, Aleatório (Amostragem Aleatória) e Entropia (EnBase), com cada categoria compreendendo dez experimentos.

Nos experimentos realizados com o conjunto de dados MNIST, destacamos a aplicação da entropia (EnBaSe) desenvolvida durante esta pesquisa, como evidenciado na Figura 25. Neste contexto, focamos em dados iid associados ao MNIST, onde 30 experimentos foram conduzidos. Dez desses experimentos foram realizados usando Execução Normal, dez com seleção aleatória e os 10 restantes empregaram o modelo do algoritmo EnBaSe proposto.

A Figura 25 apresenta os resultados alcançados pelo algoritmo EnBaSe, e a Tabela 12 compila os resultados médios desses experimentos. Especificamente, as configurações Execução Normal, Aleatória e EnBaSe no MNIST apresentaram tempos médios de execução de 73,5 segundos, 36,2 segundos e 32,0 segundos, respectivamente, incluindo as fases de treinamento, teste e validação. Em termos de acurácia, observou-se que a técnica de Seleção Aleatória resultou em uma redução de 0,1% na acurácia em comparação à abordagem Execução Normal.

Figura 25 – Performance do EnBaSe para o conjunto de dados do MNIST.



Fonte: Elaboração Própria.

Ao mesmo tempo, o método EnBaSe demonstrou uma diminuição de 0,3% na acurácia comparado à configuração Execução Normal. Observando a Figura 25, nota-se que as curvas de treinamento e validação estão próximas, e particularmente quando a curva de

Tabela 12 – Tabela de Resultados Médios para o MNIST.

Tipo	Análise	Tamanho	Accuracy	Recall	Loss	Tempo (s)
Execução Normal	Treinamento	48.000	≈ 99.5	≈ 99.5	≈ 0.038	≈ 73.5
	Validação	12.000	≈ 98.9	≈ 98.9	≈ 0.049	
	Teste	10.000	≈ 99.1	≈ 99.1		
Aleatório	Treinamento	48.000	≈ 99.4	≈ 99.4	≈ 0.054	≈ 36.2
	Validação	12.000	≈ 98.7	≈ 98.7	≈ 0.058	
	Teste	10.000	≈ 98.9	≈ 89.0		
EnBaSe	**Treinamento**	**48.000**	$\approx$ **99.2**	$\approx$ **99.2**	$\approx$ **0.069**	$\approx$ **32.0**
	Validação	**12.000**	$\approx$ **98.6**	$\approx$ **98.6**	$\approx$ **0.050**	
	Teste	**10.000**	$\approx$ **97.6**	$\approx$ **97.6**		

Fonte: Elaboração Própria.

validação ultrapassa a de treinamento, isso pode revelar várias percepções importantes sobre o comportamento do modelo. A proximidade entre essas curvas, acompanhada por um bom desempenho no conjunto de validação, indica que o modelo está efetivamente generalizando para novos dados. Tipicamente, espera-se que o modelo tenha um desempenho ligeiramente melhor no conjunto de treinamento, já que foi especificamente ajustado para esses dados.

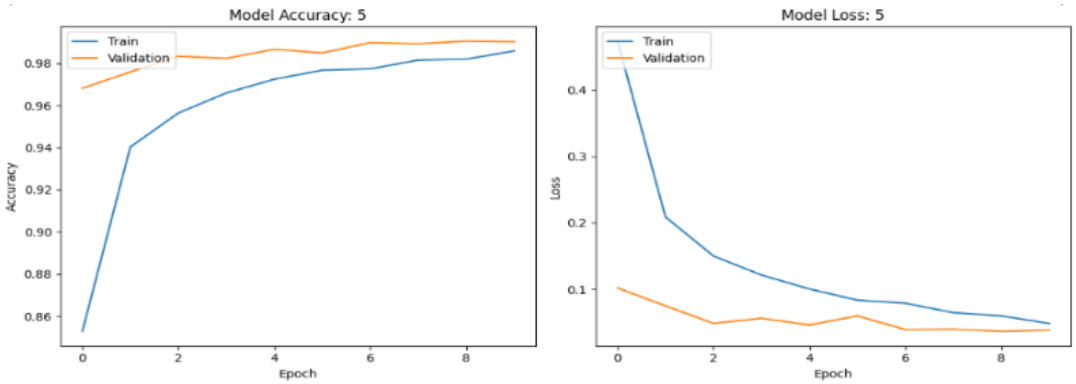
No entanto, uma curva de validação consistentemente mais alta em comparação à de treinamento pode sinalizar duas possíveis condições: a primeira é que o conjunto de validação pode ser inerentemente mais fácil para o modelo prever em relação ao conjunto de treinamento; a segunda é que o modelo pode estar se beneficiando de técnicas de regularização ou outras estratégias que melhoram sua capacidade de generalização, resultando em um desempenho inesperadamente melhor nos dados de validação em determinadas circunstâncias.

Por fim, o leve aumento na curva de perda e a redução na curva de aprendizado para os dados de treinamento e validação podem indicar que o modelo está começando a memorizar os dados de treinamento. Isso também pode sugerir a necessidade de ajustar a taxa de aprendizado ou a possibilidade de uma degradação dessa taxa. Nesse contexto, a degradação no aprendizado implica que as camadas mais profundas do modelo podem começar a aprender padrões menos relevantes ou até prejudiciais à medida que o treinamento avança.

A seguir, é apresentada a validação cruzada para o conjunto de dados MNIST, utilizando o algoritmo EnBaSe, representada nas Figuras 26 e 27. Dessa forma, a validação cruzada emprega diferentes proporções do conjunto de dados para treinamento e teste em várias iterações, estimando o grau de acurácia do modelo. Foi utilizada a validação cruzada com $k = 5$. De modo geral, todos os exemplos da validação cruzada são consistentes e semelhantes aos resultados apresentados a seguir.

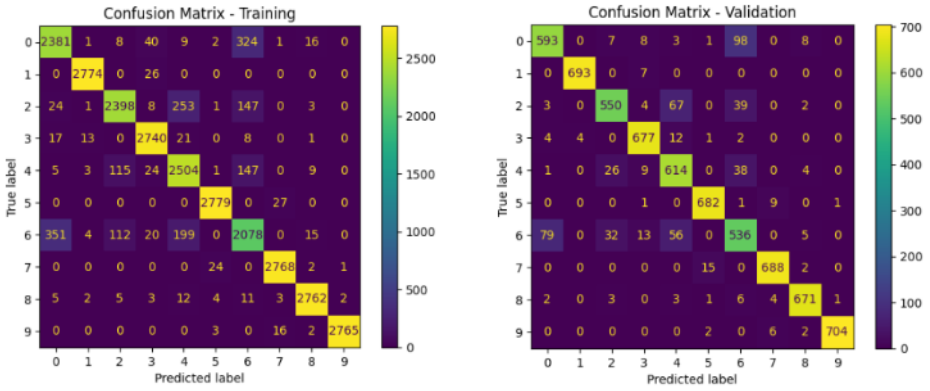
Na Tabela 13, são exibidos os resultados das iterações de $k = 1$ até $k = 5$. Esta tabela

Figura 26 – Convergência do algoritmo EnBaSe em validação cruzada no *dataset* MNIST.



Fonte: Elaboração Própria.

Figura 27 – Matriz de confusão do algoritmo EnBaSe em validação cruzada no *dataset* MNIST.



Fonte: Elaboração Própria.

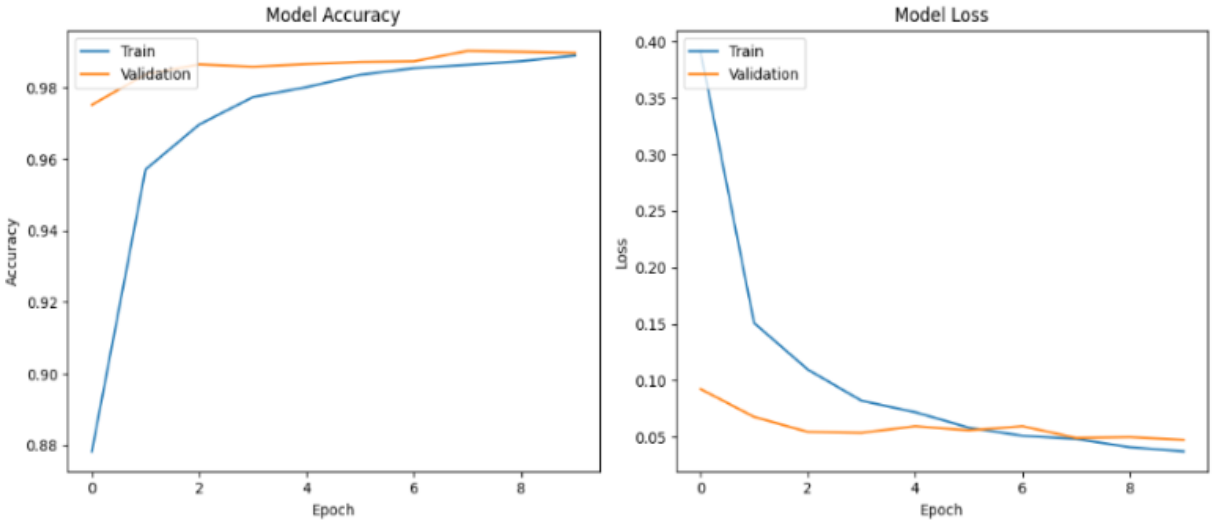
inclui, respectivamente, o valor de k, acurácia de treino, *recall* de treino, *loss* de treino, acurácia de validação, *recall* de validação e *loss* de validação. A consistência registrada que pode ser observada na tabela indica que o modelo é estável. Isso significa que o modelo não está se ajustando excessivamente a nenhum subconjunto específico de dados, mas sim que generaliza bem para qualquer dobra de $k = 1$ até $k = 5$. Os valores de alta precisão para cada k indicam que há um bom ajuste para o problema ao qual foi treinado, com baixa taxa de perda tanto no treinamento quanto na validação, apontando que o modelo conseguiu minimizar bem o erro ao longo do processo de aprendizagem. Por fim, os valores de *recall* constantes indicam que o modelo consegue identificar bem as classes relevantes e classificá-las corretamente.

Tabela 13 – Tabela de resultados de validação cruzada para a entropia no MNIST.

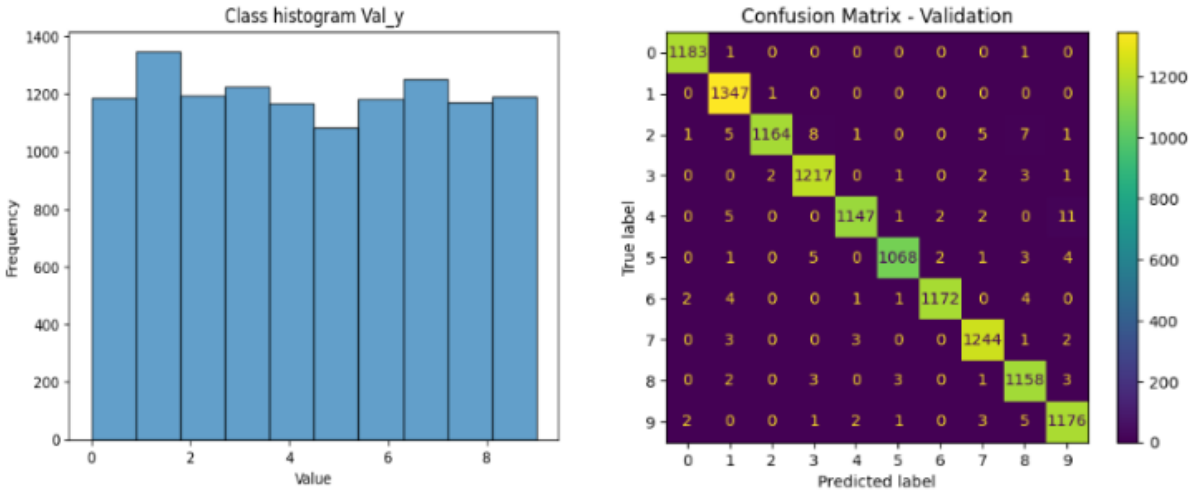
Crossvalidation K	**Train**			**Validation**		
	Accuracy	**Recall**	**Loss**	**Accuracy**	**Recall**	**Loss**
1	92.1	92.1	0.268	91.18	91.2	0.241
2	93.1	93.1	0.248	91.65	91.8	0.237
3	93.2	93.3	0.239	91.38	91.1	0.237
4	93.0	93.3	0.255	91.34	91.3	0.240
5	92.6	92.7	0.253	91.54	91.6	0.236

Fonte: Elaboração Própria.

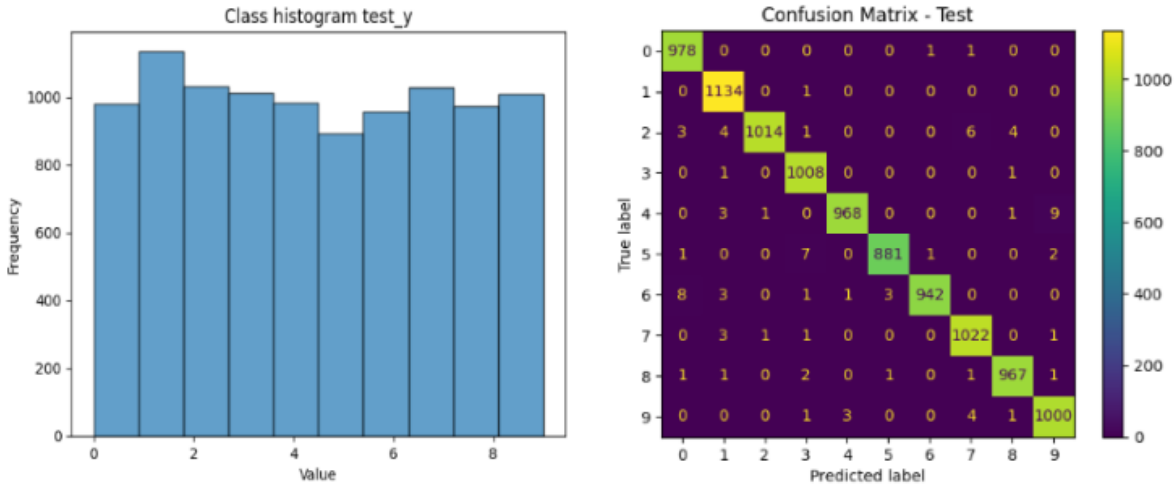
A seguir, são apresentados os resultados das matrizes de confusão para a Execução Normal e para o conjunto de dados em que foi utilizado o algoritmo EnBaSe, respectivamente nas Figuras 28 (a), (b), (c) e 29 (a), (b), (c). Estes resultados são do treinamento realizado com o conjunto de dados inteiro, servem como base de comparação para a análise dos efeitos da entropia.



(a) Convergência para treino com Execução Normal para MNIST.



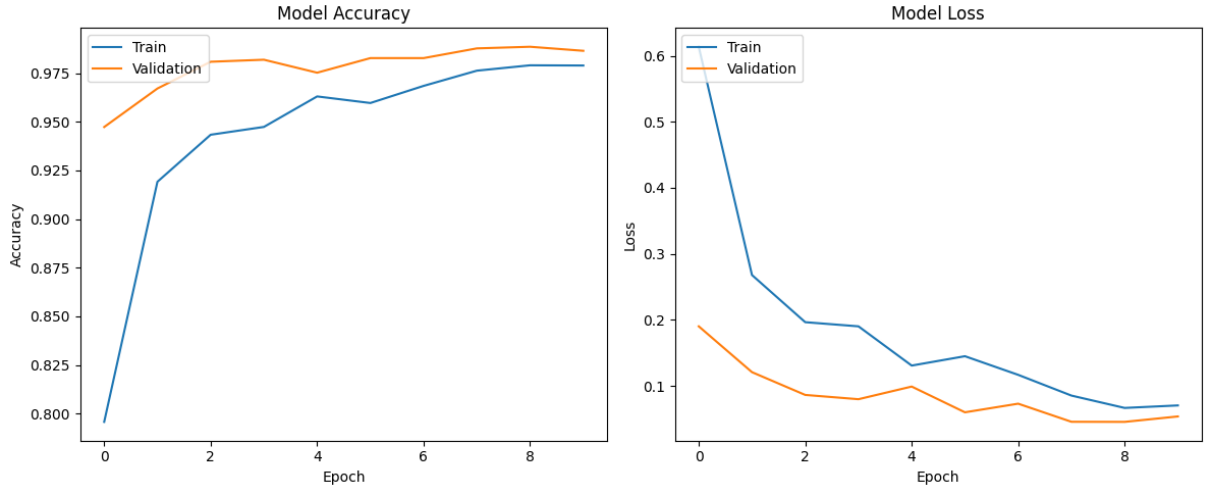
(b) Histograma de classes e matriz de confusão para o conjunto de validação para MNIST.



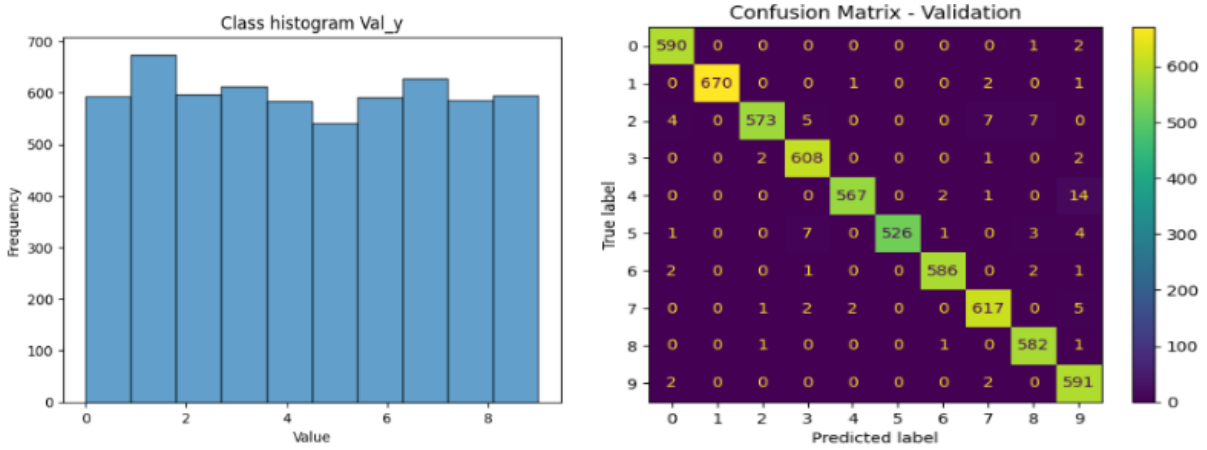
(c) Histograma de classes e matriz de confusão para o conjunto de teste para MNIST.

Figura 28 – Análise da Execução Normal para o MNIST.

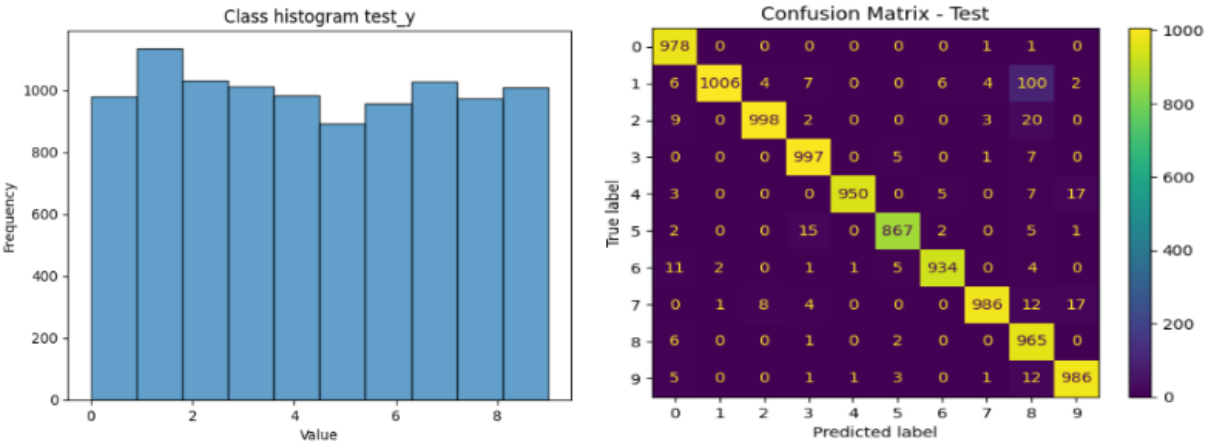
Fonte: Elaboração Própria.



(a) Curva de convergência para o treino do algoritmo EnBaSe no MNIST.



(b) Histograma de classes e matriz de confusão para o conjunto de validação com EnBaSe.



(c) Histograma de classes e matriz de confusão para o conjunto de teste com EnBaSe para MNIST.

Figura 29 – Análise do conjunto de dados MNIST com o algoritmo EnBaSe.
Fonte: Elaboração Própria.

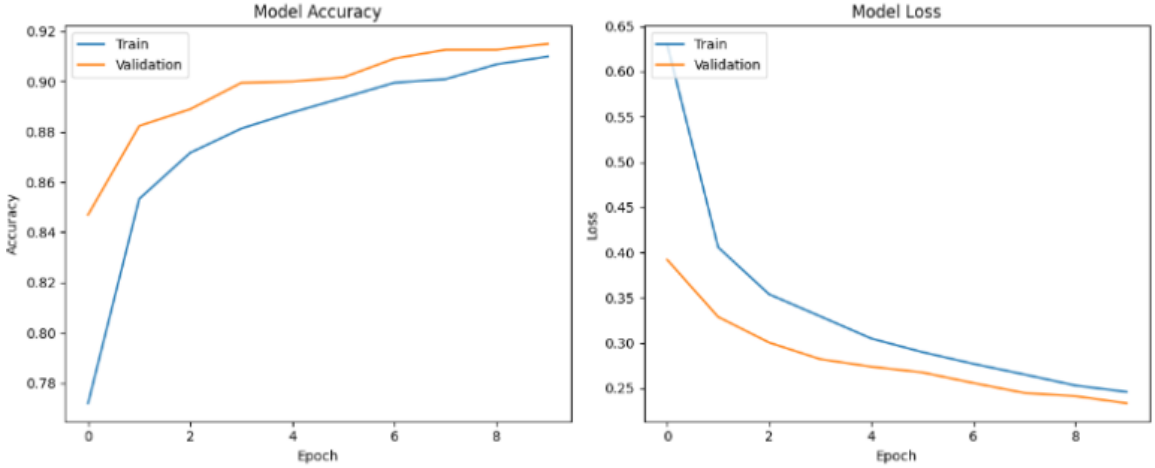
Pode-se observar, nas Figuras 28 e 29, respectivamente, o conjunto de dados de treinamento completo e o conjunto utilizando o algoritmo EnBaSe. Observa-se que o algoritmo

EnBaSe, ao analisar o mapa de calor e os valores da matriz de confusão, apresenta resultados semelhantes aos obtidos em ambos os casos para o conjunto de dados MNIST.

No estudo envolvendo o conjunto de dados Fashion-MNIST, destacado na Figura 34a, aplicamos o EnBaSe e apresentamos uma curva de aprendizado. A Tabela 14 exibe os resultados médios de 30 experimentos, distribuídos igualmente entre Execução Normal, Aleatório e o algoritmo EnBaSe.

No contexto do Fashion-MNIST, observamos que os tempos médios de execução para as configurações Execução Normal, EnBaSe e Aleatória foram de 17,32 segundos, 10,36 segundos e 10,83 segundos, respectivamente. Esses resultados ressaltam a eficácia das técnicas de seleção utilizadas. Curiosamente, a técnica Aleatória registrou um ganho de 0,1% na acurácia, enquanto o EnBaSe apresentou um aumento de 0,3% na acurácia. Além disso, notamos que a curva de validação permaneceu consistentemente mais alta do que a curva de treinamento no Fashion-MNIST.

Figura 30 – Análise dos conjuntos de dados Fashion-MNIST para o algoritmo EnBaSe.



Fonte: Elaboração Própria.

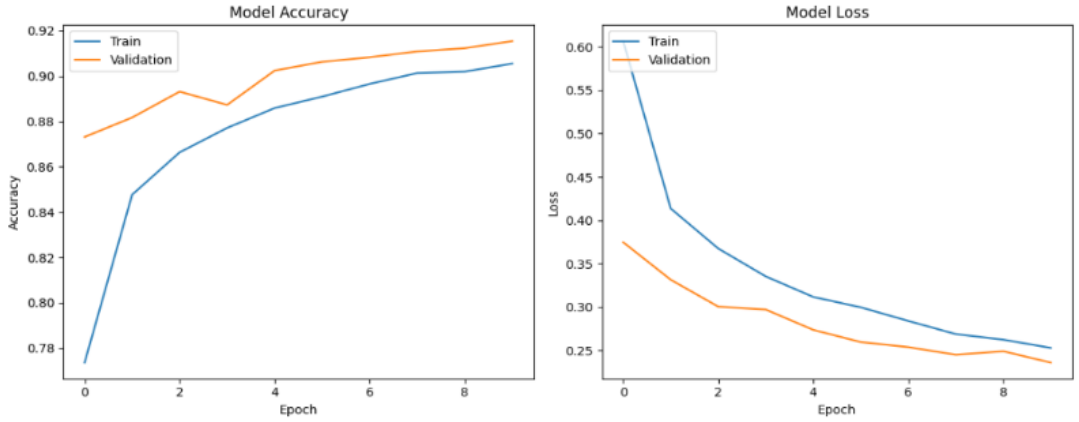
Tabela 14 – Tabela de Resultados médios para o Fashion-MNIST.

Tipo	Análise	Tamanho	Accuracy	Recall	Loss	Tempo (s)
Execução Normal	Treinamento	48.000	≈ 92.1	≈ 92.1	≈ 0.278	≈ 17.3
	Validação	12.000	≈ 90.7	≈ 90.7	≈ 0.253	
	Teste	10.000	≈ 89.7	≈ 89.7		
Aleatório	Treinamento	48.000	≈ 91.2	≈ 91.2	≈ 0.308	≈ 10.3
	Validação	12.000	≈ 88.9	≈ 88.9	≈ 0.303	
	Teste	10.000	≈ 88.4	≈ 88.4		
Entropia	**Treinamento**	**48.000**	**≈ 92.4**	**≈ 92.4**	**≈ 0.262**	**≈ 10.8**
	Validação	**12.000**	**≈ 91.0**	**≈ 91.0**	**≈ 0.246**	
	Teste	**10.000**	**≈ 79.6**	**≈ 79.6**		

Fonte: Elaboração Própria.

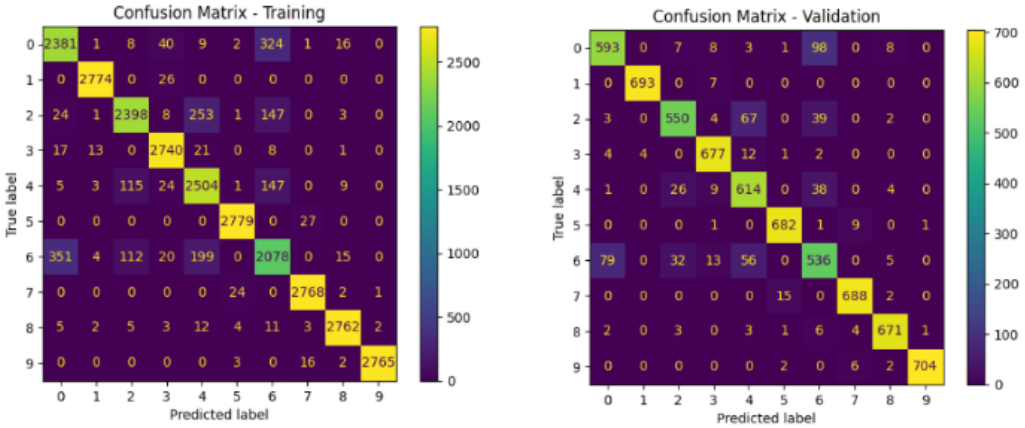
É apresentada, a seguir, a validação cruzada para o conjunto de dados Fashion-MNIST, utilizando o algoritmo deste estudo — EnBaSe. Dessa forma, a validação cruzada utiliza diferentes proporções do conjunto de dados para treinamento e testes em diferentes iterações, estimando o grau de acurácia do modelo. Foi utilizada validação cruzada para $k = 5$. De modo geral, todos os exemplos da validação cruzada são consistentes e semelhantes aos resultados apresentados a seguir.

Figura 31 – Convergência do algoritmo EnBaSe em validação cruzada no *dataset* Fashion-MNIST.



Fonte: Elaboração Própria.

Figura 32 – Matriz de confusão do algoritmo EnBaSe em validação cruzada no *dataset* Fashion-MNIST.



Fonte: Elaboração Própria.

Na Tabela 15, apresentada a seguir, encontram-se os resultados da validação cruzada realizada pelo algoritmo EnBaSe. Esta tabela inclui, respectivamente, o valor de k, acurácia de treino, *recall* de treino, *loss* de treino, acurácia de validação, *recall* de validação e *loss* de validação. Ao observar a tabela, é possível identificar que o classificador mostra consistência,

variando pouco entre os valores de acurácia, o que indica um modelo estável. Além disso, também é possível, para todos os k, identificar uma boa capacidade de generalização, demonstrando valores minimizados do erro com uma consistência constante para a acurácia de validação. Dessa forma, conclui-se que, através da validação cruzada, o modelo é estável e robusto, e as mínimas variações entre os diferentes valores de k indicam que o modelo generaliza bem para novos dados.

Tabela 15 – Resultados da validação cruzada para EnBaSe no Fashion-MNIST.

Crossvalidation K	**Train**			**Validation**		
	Accuracy	**Recall**	**Loss**	**Accuracy**	**Recall**	**Loss**
1	92.1	92.1	0.268	91.1	91.2	0.241
2	93.1	93.1	0.248	91.6	91.8	0.237
3	93.2	93.3	0.239	91.3	91.1	0.237
4	93.0	93.0	0.255	91.3	91.3	0.240
5	92.6	92.7	0.253	91.5	91.6	0.236

Fonte: Elaboração Própria.

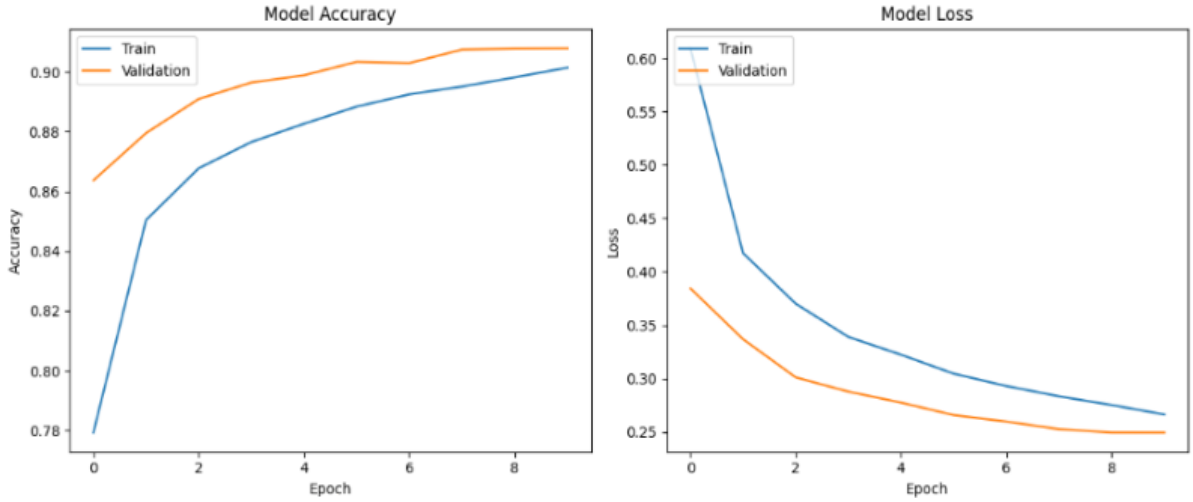
São apresentados, a seguir, os resultados obtidos para a Execução Normal do Fashion-MNIST. Para este conjunto de dados, introduzimos a aplicação da entropia (EnBaSe), proposta neste estudo, assim como, respectivamente, a matriz de confusão para a Execução Normal, apresentada na Figura 33, e para o conjunto ao qual foi aplicado o algoritmo EnBaSe, na Figura 34.

É interessante notar que, em ambos os modelos, algumas classes no mapa de calor da matriz de confusão apresentam tonalidades mais intensas, indicando alta precisão. Por outro lado, outras classes exibem tonalidades mais claras no mapa de calor, sugerindo uma precisão menor. Essas variações podem, inicialmente, ser atribuídas ao desbalanceamento entre as classes. No entanto, o algoritmo EnBaSe garante que a mediana de cada classe seja selecionada, o que nos permite descartar essa hipótese.

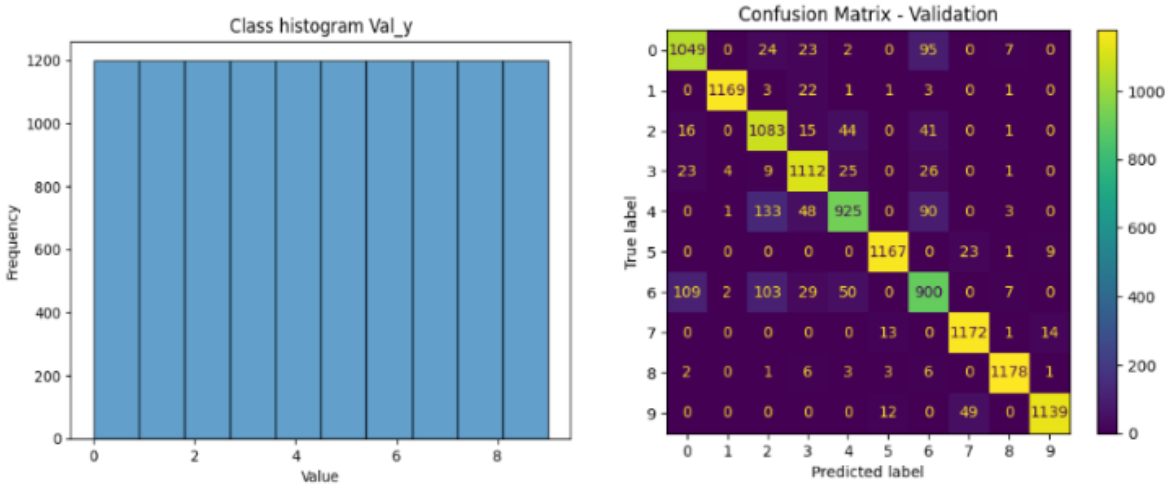
Portanto, podemos indicar que, na matriz de confusão, os erros de classificação entre as classes podem ser devidos à semelhança entre elas, tornando difícil distingui-las. Outras possíveis causas incluem inconsistências ou erros nos dados de treinamento, onde alguns conjuntos podem não representar tão bem o modelo, ou à complexidade do modelo, indicando um subajuste, no qual a rede neural precisa de mais ajustes para ser capaz de solucionar o problema.

A hipótese de subajuste pode ser a mais válida, como observado na Figura 33a, onde

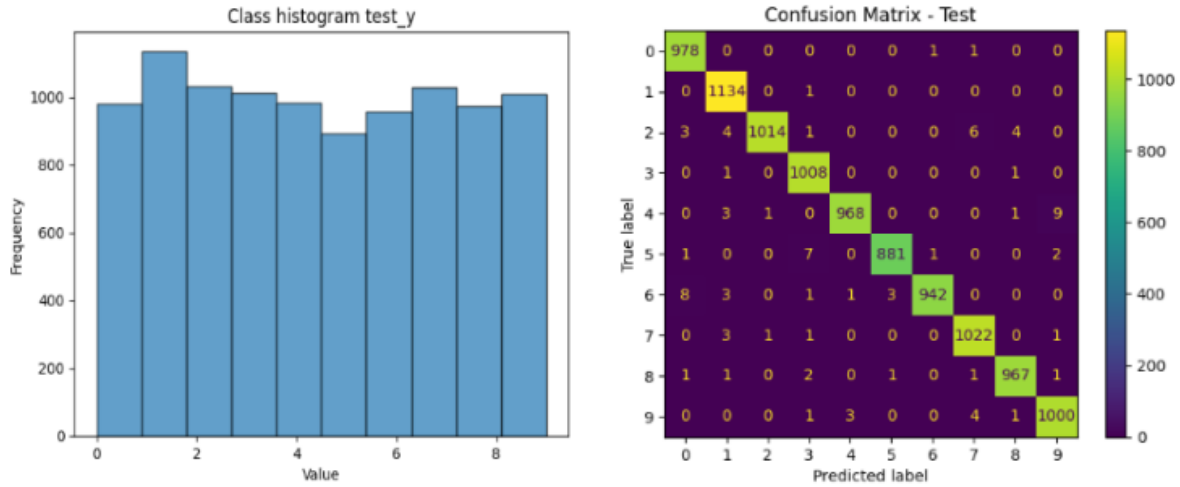
podemos ver que o modelo de treinamento apresenta uma performance inferior à do modelo de validação, apontando que ainda existe espaço para refinamento da rede neural e que esta não atingiu o platô para a tarefa para a qual foi elaborada. Dessa forma, explicam-se melhor os erros de classificação.



(a) Curva de convergência para treino com Execução Normal para Fashion-MNIST.

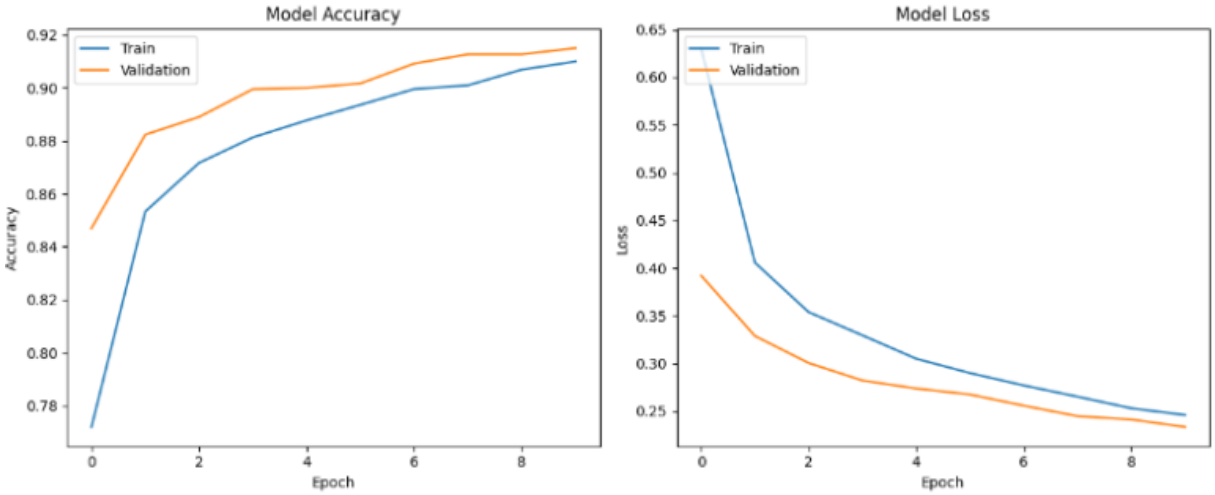


(b) Histograma de classes e matriz de confusão para validação da Execução Normal para Fashion-MNIST.

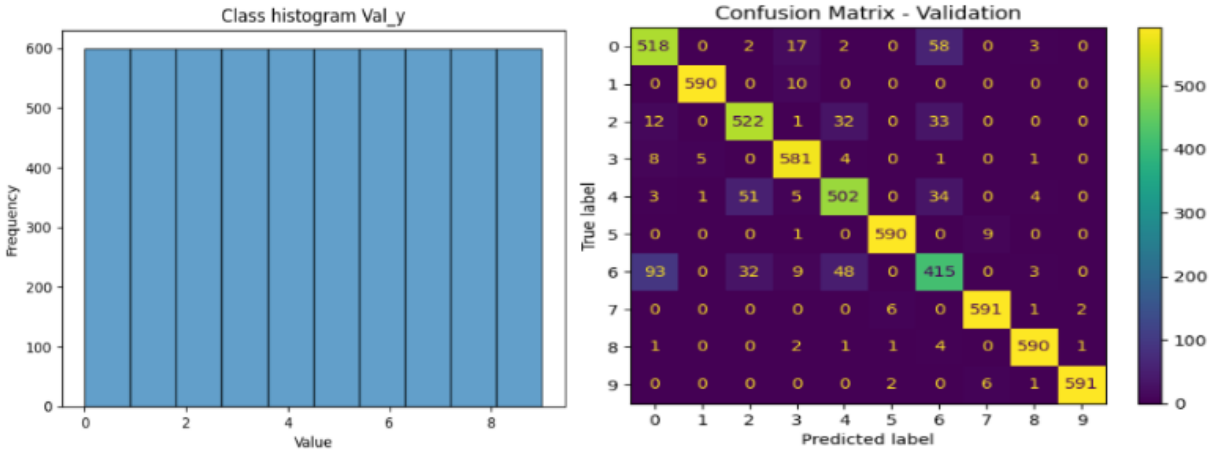


(c) Histograma de classes e matriz de confusão para o teste com Execução Normal para Fashion-MNIST.

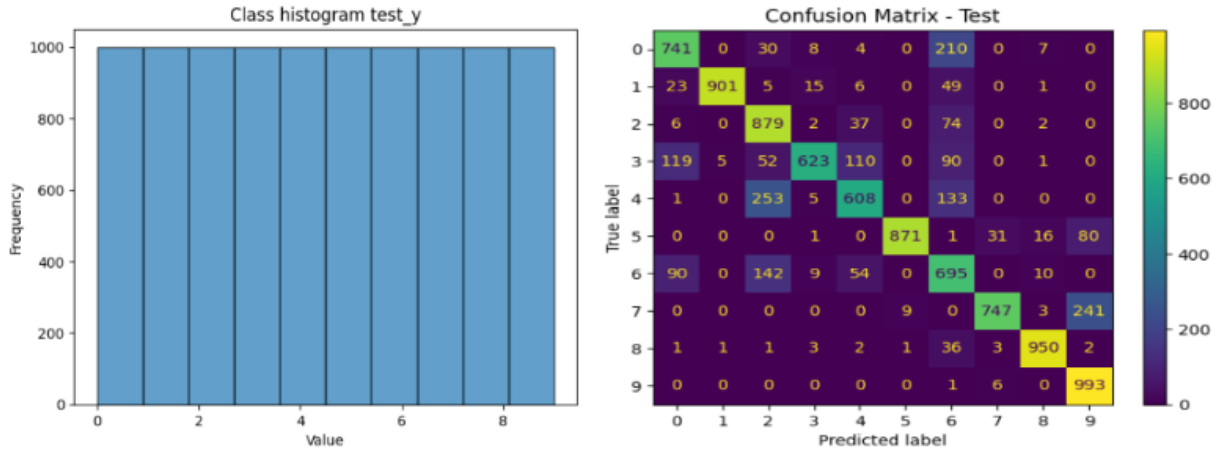
Figura 33 – Análise do conjunto completo de dados Fashion-MNIST.
Fonte: Elaboração Própria.



(a) Curva de convergência para o treino do algoritmo EnBaSe no Fashion-MNIST.



(b) Histograma de classes e matriz de confusão para validação do algoritmo EnBaSe no Fashion-MNIST.



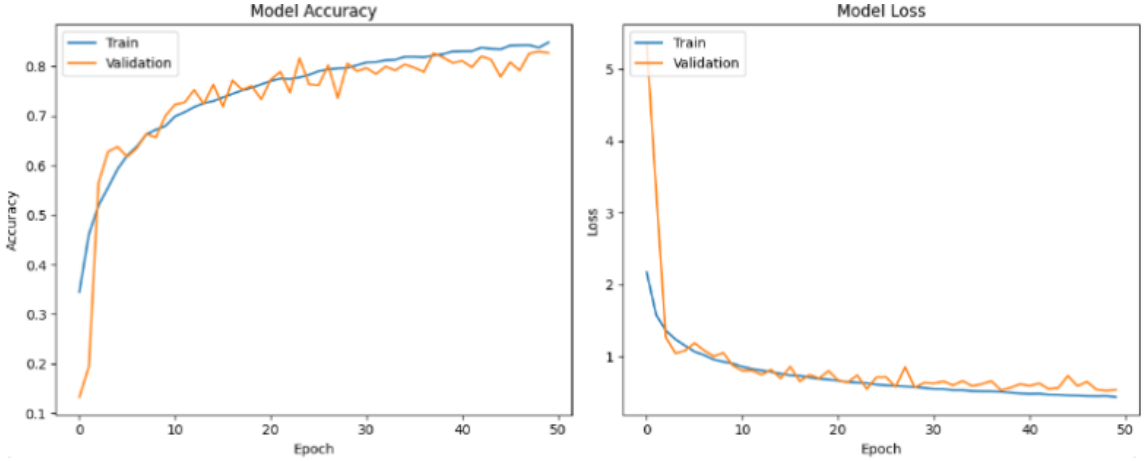
(c) Histograma de classes e matriz de confusão para o conjunto de teste com EnBaSe no Fashion-MNIST.

Figura 34 – Análise do conjunto de dados Fashion-MNIST utilizando o algoritmo EnBaSe.
Fonte: Elaboração Própria.

Para o CIFAR-10, representado pela Figura 35, apresentamos o comportamento de

treinamento do algoritmo EnBaSe, conforme mostrado na Tabela 16, que apresenta a média de 10 experimentos para Execução Normal, EnBaSe e Aleatório. No CIFAR-10, as médias dos tempos de treinamento foram de 1019,2 s para Execução Normal, 505,8 s para Aleatório e 501,1 s para EnBaSe, respectivamente, destacando as diferenças de eficiência. A técnica Aleatória apresentou uma perda de 3,6% na acurácia, enquanto a técnica EnBaSe mostrou uma perda de 1,7%.

Figura 35 – Análise do conjunto de dados CIFAR-10 para o algoritmo EnBaSe.



Fonte: Elaboração Própria.

Tabela 16 – Tabela de Resultados médios para CIFAR-10.

Tipo	Análise	Tamanho	Accuracy	Recall	Loss	Tempo (s)
Execução Normal	Treinamento	40.000	≈ 90.9	≈ 90.9	≈ 0.395	≈ 1019.2
	Validação	5.000	≈ 86.4	≈ 86.4	≈ 0.403	
	Teste	10.000	≈ 85.8	≈ 85.9		
Aleatório	Treinamento	20.000	≈ 89.3	≈ 89.3	≈ 0.448	≈ 505.8
	Validação	5.000	≈ 82.2	≈ 82.2	≈ 0.554	
	Teste	10.000	≈ 81.7	≈ 81.7		
EnBaSe	**Treinamento**	**20.000**	**≈ 89.2**	**≈ 89.2**	**≈ 0.441**	**≈ 501.1**
	Validação	**5.000**	**≈ 82.0**	**≈ 82.0**	**≈ 0.558**	
	Teste	**10.000**	**≈ 78.9**	**≈ 78.9**		

Fonte: Elaboração Própria.

Na Figura 35, a sobreposição das curvas de treinamento e validação, com apenas leves flutuações na interseção, é geralmente um indicador positivo. Isso demonstra que o modelo tem um desempenho consistente nos dados de treinamento e validação. A proximidade entre essas curvas indica que o modelo está generalizando eficientemente, sendo capaz de transferir o conhecimento adquirido durante o treinamento para os dados de validação. Esse cenário sugere um equilíbrio entre viés e variância.

Um baixo viés revela que o modelo possui a capacidade necessária para entender a

complexidade inerente aos dados. Ao mesmo tempo, uma baixa variância indica que o modelo não está superajustado aos dados de treinamento, permitindo um bom desempenho em novos dados. Por fim, a curva de aprendizado, mostrando uma taxa de aprendizado constante e alta, e a curva de perda, apresentando valores reduzidos, indicam que o modelo de treinamento alcançou, ou está próximo de alcançar, seu potencial máximo dentro das restrições impostas por sua arquitetura e configurações de treinamento.

A validação cruzada para o algoritmo EnBaSe foi realizada com $k = 5$. Esse método analisa a estatística e a capacidade de generalização do modelo, testando diferentes proporções do conjunto de dados em várias iterações para estimar a acurácia. Uma amostra dos resultados é apresentada a seguir, na Figura 36 para a curva de convergência e na Figura 37 para a matriz de confusão.

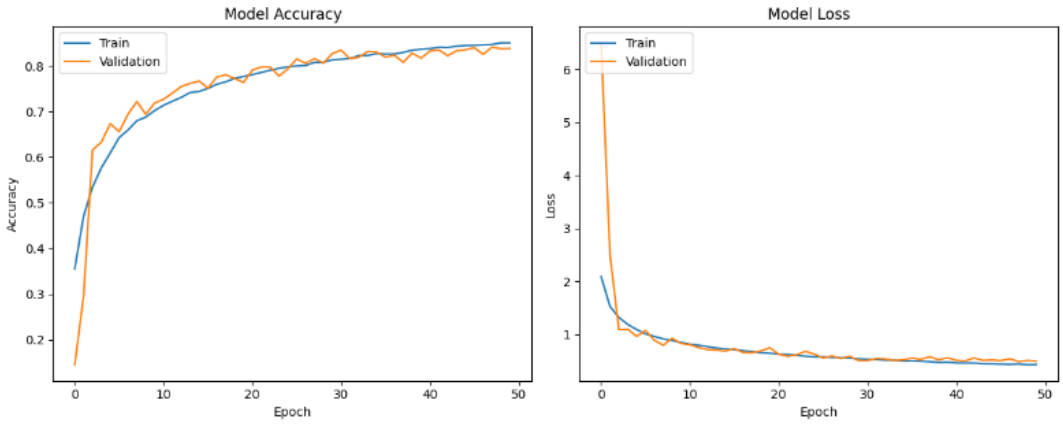


Figura 36 – Convergência do algoritmo EnBaSe na validação cruzada para o CIFAR-10.

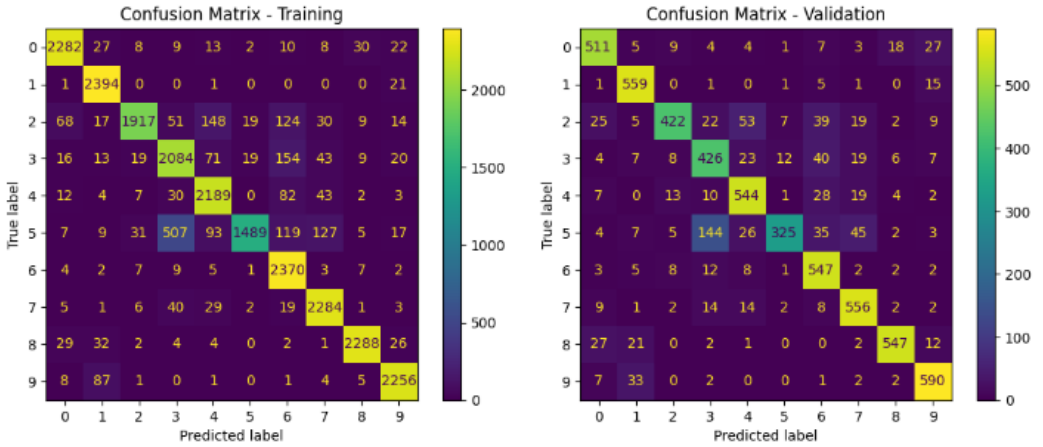


Figura 37 – Matriz de confusão do algoritmo EnBaSe em validação cruzada no CIFAR-10.
Fonte: Elaboração Própria.

A Tabela 17 mostra os resultados da validação cruzada para o algoritmo EnBaSe. É

possível observar, na tabela, que o modelo foi capaz de classificar corretamente o conjunto de dados durante o treinamento, apresentando uma leve oscilação na previsão, o que indica que a arquitetura da rede neural se mostrou razoavelmente estável, com uma flutuação de acurácia ao lidar com novos dados.

Além disso, ao observar a sensibilidade do modelo, os valores mantêm-se constantes, praticamente idênticos aos da acurácia, o que sugere que o modelo é capaz de identificar corretamente as classes positivas tanto no treinamento quanto na validação. Por fim, ao analisar os valores de *loss*, nota-se uma leve degradação ao ser apresentado a novos dados, porém os valores próximos aos originais indicam que o modelo não sofre de *overfitting* significativo.

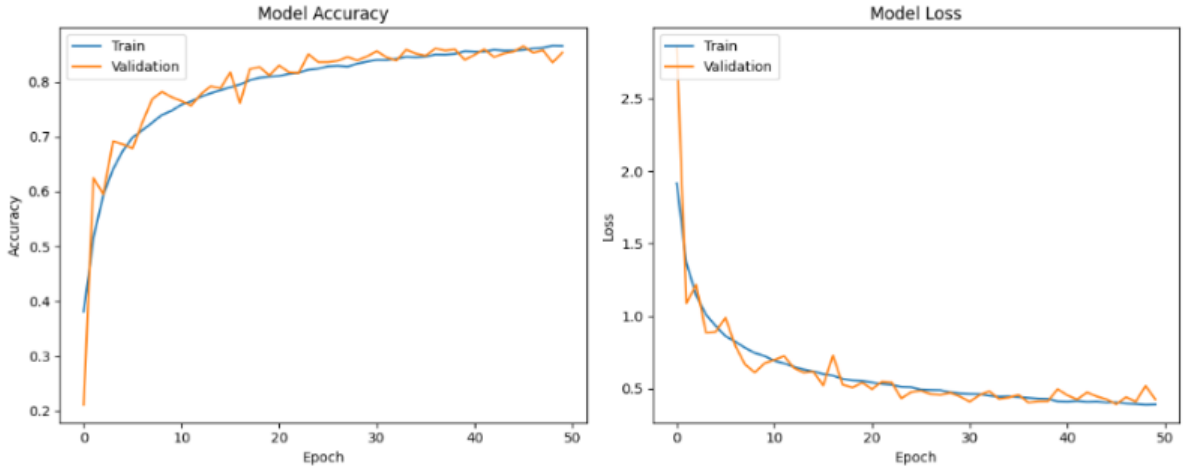
Tabela 17 – Resultados da validação cruzada para o CIFAR-10 do algoritmo EnBaSe.

Validação Cruzada K	**Treinamento**			**Validação**		
	Accuracy	**Recall**	**Loss**	**Accuracy**	**Recall**	**Loss**
1	90.2	90.2	0.429	83.5	83.7	0.511
2	90.2	90.2	0.421	83.9	84.0	0.494
3	91.1	91.1	0.420	84.4	84.5	0.482
4	89.4	89.4	0.417	82.6	82.6	0.538
5	89.8	89.8	0.430	83.7	83.7	0.494

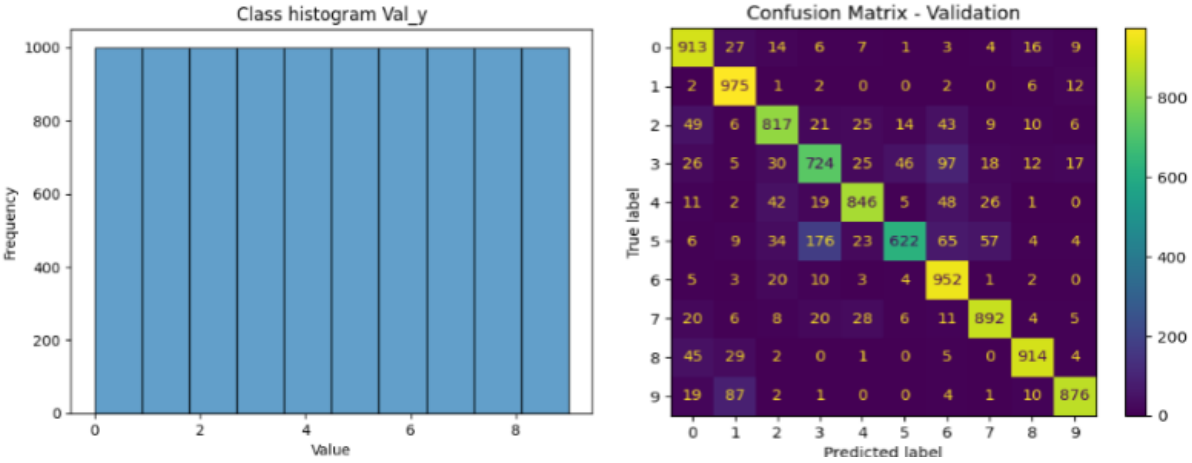
Fonte: Elaboração Própria.

Nas Figuras 38, que apresentam a rede neural treinada com Execução Normal e utilizando o algoritmo EnBaSe, são mostrados os mapas de calor para ambas as arquiteturas. Para esses resultados, é possível observar, nos mapas de calor de ambos os modelos, altas precisões em ambos os cenários, apresentando-se dentro da margem esperada de perda de precisão, conforme proposto pelo algoritmo EnBaSe, mantendo a máxima acurácia possível.

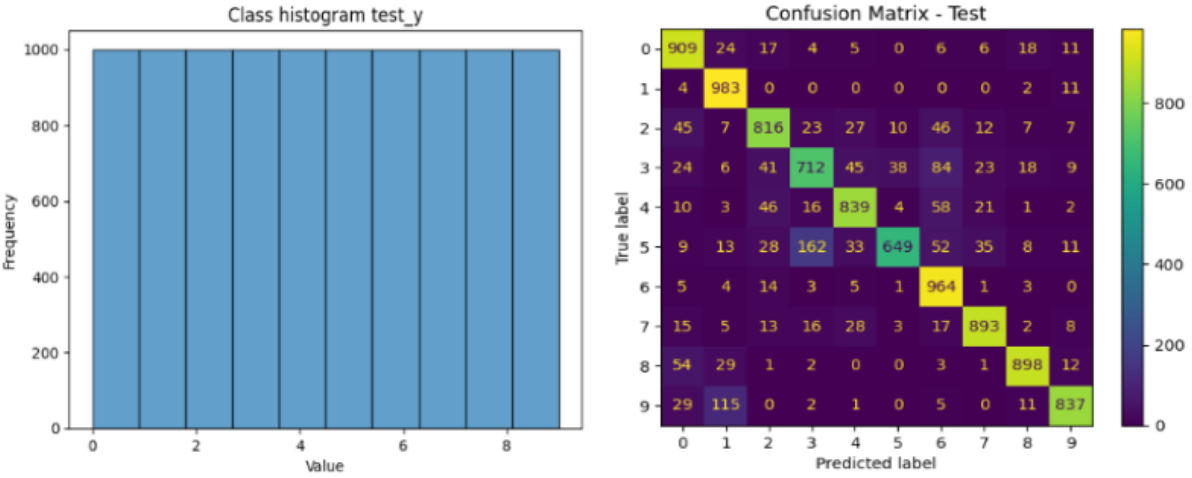
Isso indica que a rede neural está bem estruturada, atingindo seu platô, generalizando bem para novos dados e apresentando alta precisão para os casos verdadeiros em ambos os cenários.



(a) Curva de Convergência para treino com Execução Normal para CIFAR-10.



(b) Histograma de classes e matriz de confusão para validação com Execução Normal para CIFAR-10.



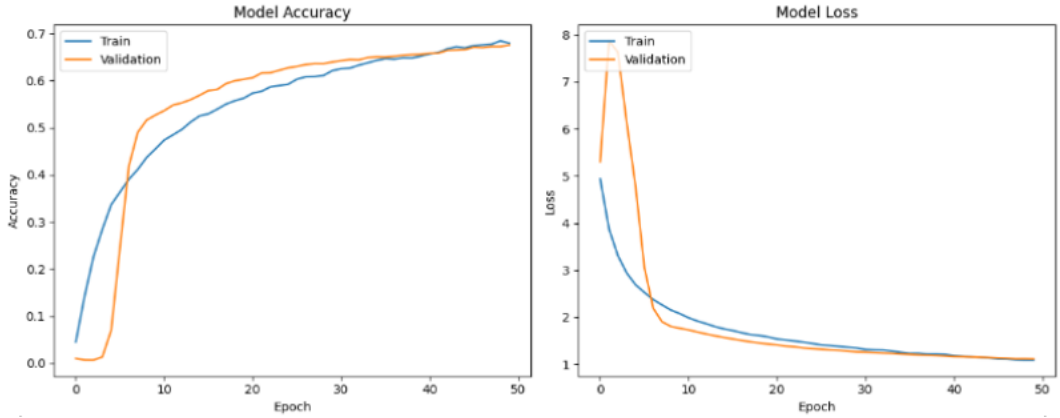
(c) Histograma de classes e matriz de confusão da Execução Normal para CIFAR-10.

Figura 38 – Análise da Execução Normal para CIFAR-10.
Fonte: Elaboração Própria.

Por fim, para o conjunto de dados CIFAR-100, a aplicação do algoritmo EnBaSe está representada na Figura 38. Na Tabela 18, é possível observar a média de 10 experimentos

para o conjunto de dados iid em cada tipo de método, demonstrando os resultados obtidos.

Figura 39 – Análises do conjunto de dados CIFAR-100 com EnBaSe.



Fonte: Elaboração Própria.

Tabela 18 – Tabela de Resultados médios no CIFAR-100.

Tipo	**Análise**	**Tamanho**	**Accuracy**	**Recall**	**Loss**	**Tempo (s)**
Execução Normal	Treinamento	40.000	≈ 79.3	≈ 79.3	≈ 0.952	≈ 16393.2
	Validação	5.000	≈ 72.2	≈ 72.3	≈ 1.151	
	Teste	10.000	≈ 71.6	≈ 71.7		
Aleatório	Treinamento	20.000	≈ 77.0	≈ 77.0	≈ 0.952	≈ 8232.7
	Validação	5.000	≈ 67.5	≈ 67.5	≈ 1.151	
	Teste	10.000	≈ 66.5	≈ 66.6		
EnBaSe	**Treinamento**	**20.000**	**≈ 77.7**	**≈ 77.7**	**≈ 0.952**	**≈ 8778.7**
	Validação	**5.000**	**≈ 68.1**	**≈ 68.1**	**≈ 1.151**	
	Teste	**10.000**	**≈ 64.9**	**≈ 64.9**		

Fonte: Elaboração Própria.

Os tempos médios de treinamento para o CIFAR-100 foram de 16.393,26 s para Execução Normal, 8.778,76 s para EnBaSe e 8.232,70 s para Aleatório. A técnica Aleatória apresentou uma perda de 2,3% em acurácia, enquanto a técnica EnBaSe registrou uma diminuição de 1,6%.

Como ilustrado na Figura 39, observamos um padrão interessante em que a curva de validação começa acima da curva de treinamento. Isso pode ocorrer quando os conjuntos de treinamento e validação são compostos de maneira que facilitam o aprendizado do modelo ou quando refletem as características gerais dos dados. Essa característica inicial pode fazer com que a curva de validação apresente um desempenho inicial superior.

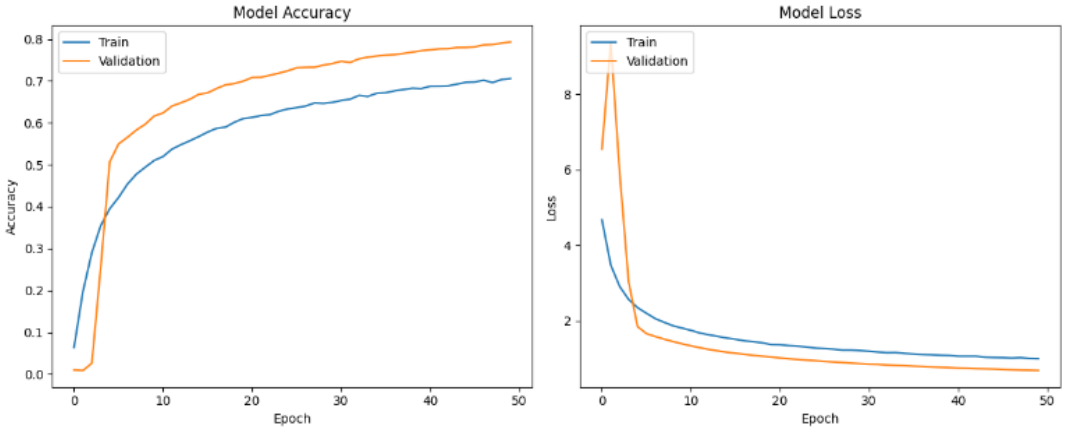
À medida que o treinamento avança, o modelo aprimora sua capacidade de otimizar essas representações genéricas. Dessa forma, o resultado é uma convergência entre as curvas de treinamento e validação. Além disso, a curva de perda para treinamento e validação mostra uma

tendência de declínio ao longo do tempo, reforçando a noção de que o modelo está aprendendo efetivamente a partir dos dados fornecidos.

Portanto, pode-se afirmar que, para o CIFAR-100 — um benchmark reconhecido em testes de reconhecimento de padrões de imagem — o algoritmo EnBaSe exibiu desempenho notável. Ele alcançou esse resultado sem introduzir um viés que comprometesse o desempenho, o aprendizado ou a capacidade de generalização do modelo treinado.

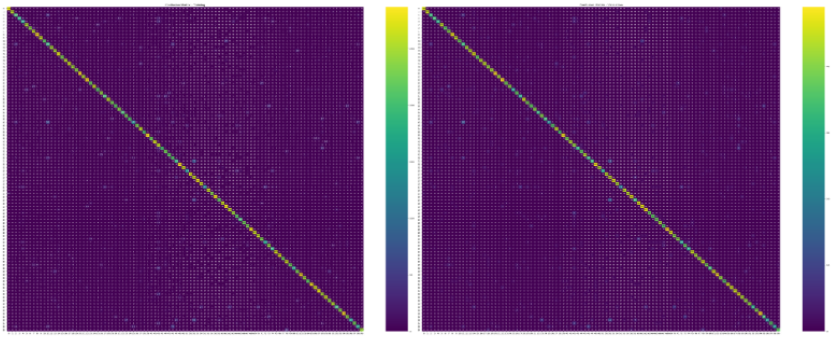
A seguir, apresentamos a validação cruzada para EnBaSe no CIFAR-100, utilizando diferentes proporções de dados para treinamento e teste em várias iterações, com o objetivo de estimar a acurácia do modelo. Utilizamos $k = 5$ na validação cruzada. Os resultados mostram que, de modo geral, os exemplos de validação cruzada são consistentes e similares.

Figura 40 – Convergência do algoritmo EnBaSe em validação cruzada no CIFAR-100.



Fonte: Elaboração Própria.

Figura 41 – Matriz de confusão do algoritmo EnBaSe em validação cruzada no CIFAR-100.



Fonte: Elaborado pelo autor.

A Tabela 19 exibe os resultados da validação cruzada da entropia das classes no CIFAR-100, incluindo o valor de k, acurácia de treino, *recall* de treino, *loss* de treino, acurácia de validação, *recall* de validação, e *loss* de validação.

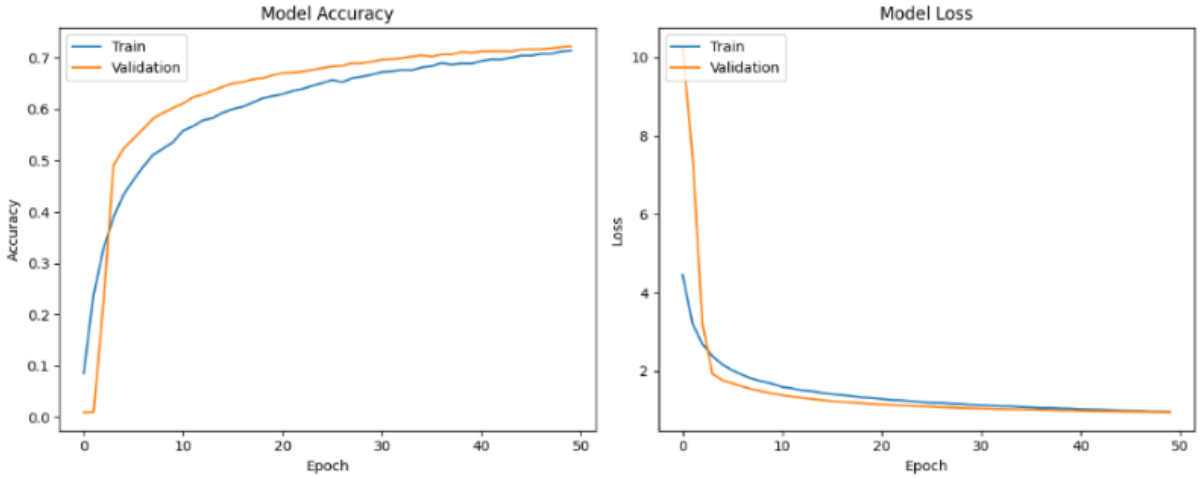
Tabela 19 – Resultados da validação cruzada para o CIFAR-100 com o algoritmo EnBaSe.

Crossvalidation K	**Train**			**Validation**		
	Accuracy	**Recall**	**Loss**	**Accuracy**	**Recall**	**Loss**
1	78.8	78.9	0.689	78.7	78.8	0.689
2	79.0	79.0	0.682	79.6	79.6	0.682
3	79.0	79.1	0.691	79.1	79.1	0.691
4	79.1	79.1	0.681	79.6	79.6	0.681
5	78.9	78.9	0.685	79.3	79.3	0.685

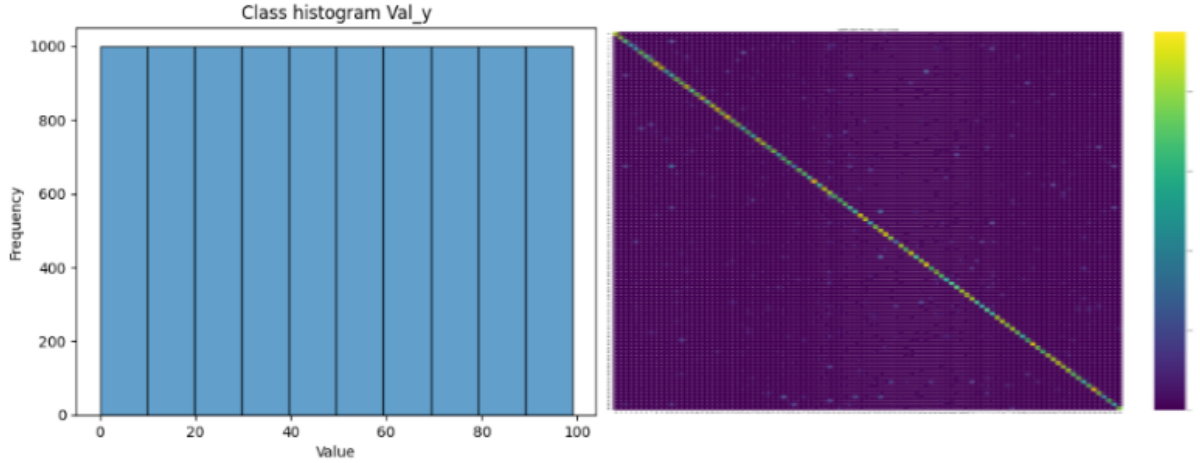
Fonte: Elaboração Própria.

Na tabela, podemos observar que os valores de *recall* e *loss* para todos os *k* encontram-se próximos, indicando que o modelo generaliza bem para novos dados, uma vez que não há grandes discrepâncias entre os resultados. Além disso, todos os valores obtidos na validação cruzada são relativamente altos, indicando uma capacidade constante de previsão, assim como uma precisão correta, evidenciada pelos valores de *recall*. Dessa forma, pode-se concluir que o modelo apresenta boa estabilidade e capacidade de generalização, sem evidências de *overfitting* ou *underfitting*.

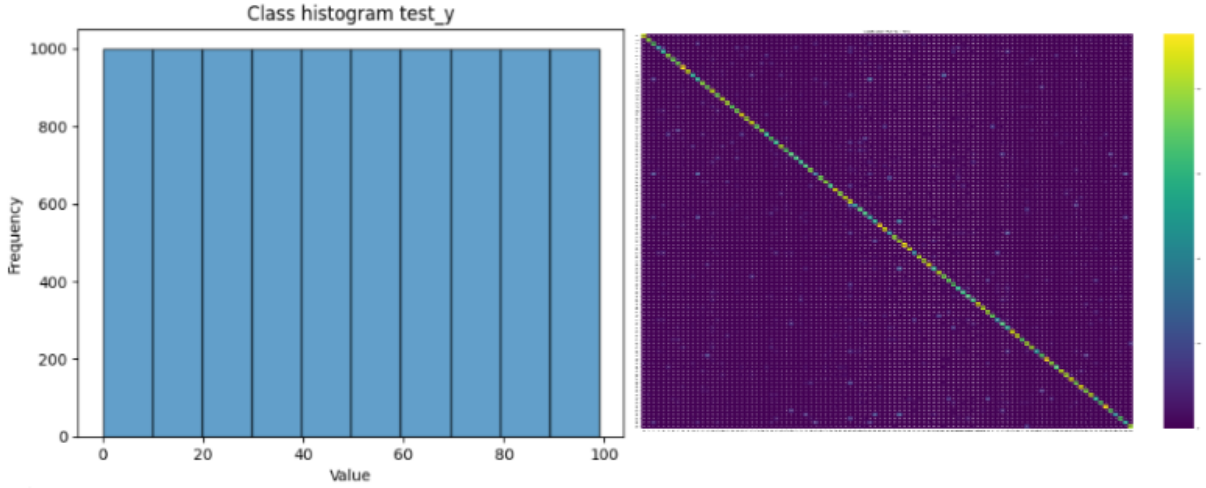
A seguir, na Figura 42 e na Figura 43, são apresentadas, respectivamente, as curvas de convergência para os modelos da Execução Normal e EnBaSe, bem como as matrizes de confusão correspondentes. Devido à complexidade do CIFAR-100, é difícil analisar o mapa de calor da matriz de confusão; no entanto, ao observar os pontos de calor, é possível afirmar, de maneira geral, que ambas as matrizes de confusão, tanto para Execução Normal quanto para o modelo com o algoritmo EnBaSe, apresentam resultados semelhantes. Isso se reflete também em outras análises, como a validação cruzada e as métricas de comparação presentes na tabela.



(a) Curva de Convergência para treino com Execução Normal para CIFAR-100.



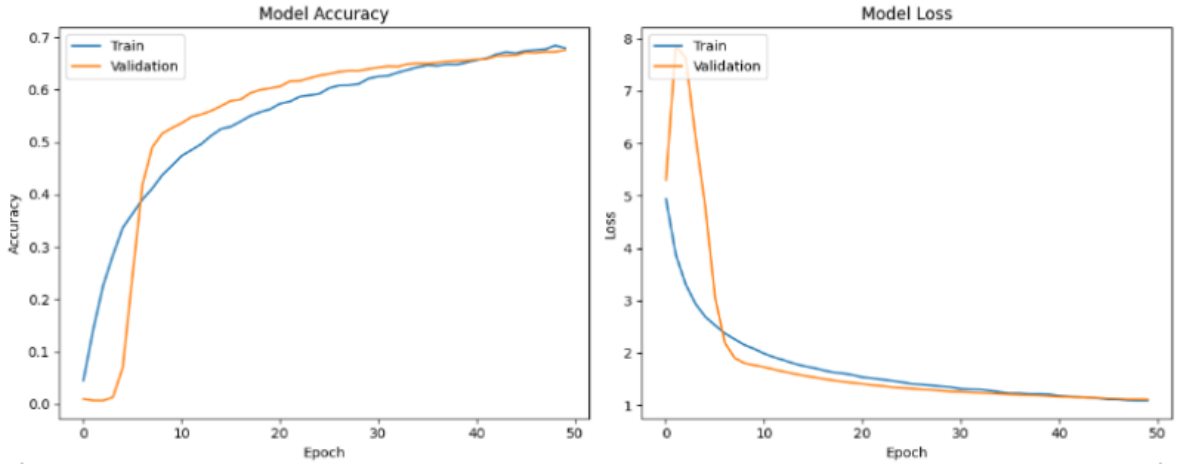
(b) Histograma de classes e matriz de confusão para validação do conjunto completo de dados CIFAR-100.



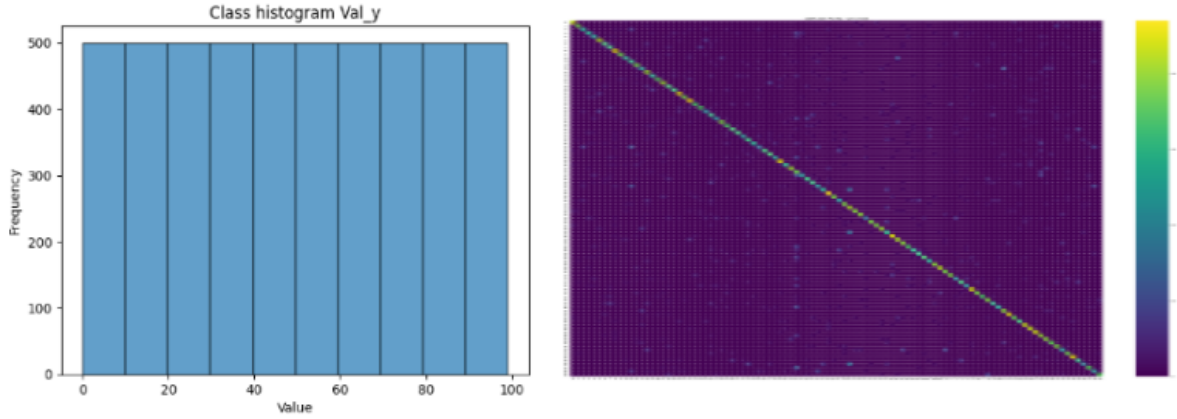
(c) Histograma de classes e matriz de confusão para o teste da Execução Normal para CIFAR-100.

Figura 42 – Análise da Execução Normal do CIFAR-100.

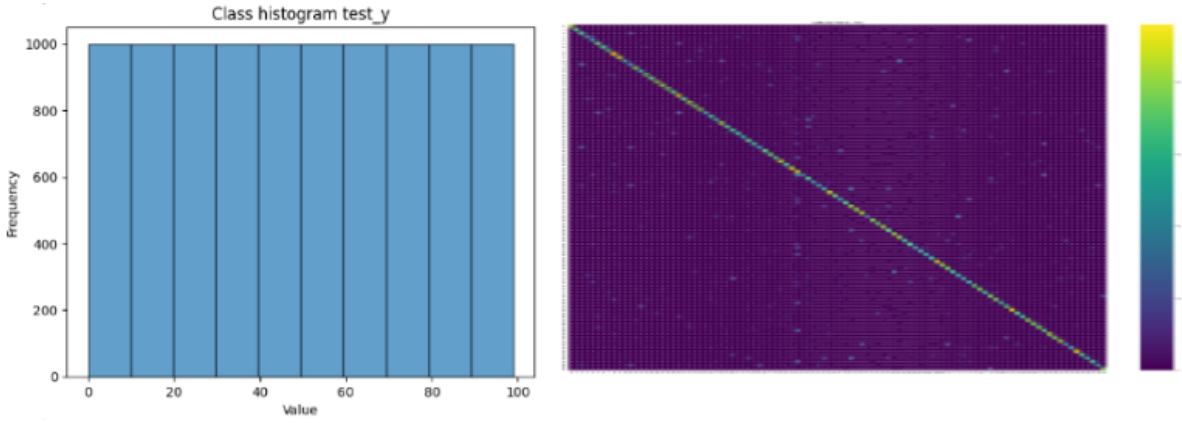
Fonte: Elaboração Própria.



(a) Curva de convergência para o treino do algoritmo EnBaSe no CIFAR-100.



(b) Histograma de Classes e Matriz de Confusão para Validação do algoritmo EnBaSe no CIFAR-100.



(c) Histograma de classes e matriz de confusão para o conjunto de teste com EnBaSe no CIFAR-100.

Figura 43 – Análise do conjunto de dados CIFAR-100 utilizando o algoritmo EnBaSe.
Fonte: Elaboração Própria.

5.3.2 EnBaSe aplicado no cenário non-iid

Esta seção discutirá os resultados dos experimentos conduzidos sob o cenário non-iid. Prosseguiremos com uma apresentação detalhada dos valores alcançados pelas métricas de

avaliação, conforme especificado na Seção 4.6. Adicionalmente, realizaremos uma análise aprofundada das características e da eficácia das redes neurais utilizadas, conforme introduzido na Seção 4.2. Utilizando a distribuição apresentada na subseção 4.5 e os algoritmos de agregação global discutidos na subseção 4.5.0.1.

Nesta seção, avaliamos o desempenho dos algoritmos FedAvg e FedProx nos conjuntos de dados mencionados na Seção 4.1.2, examinando o efeito da complexidade dos dados. Foram utilizados critérios de seleção variados (Execução Normal, Entropia (EnBaSe) e Aleatório (Amostragem Aleatória)) em 240 experimentos. O objetivo é determinar a eficiência dos algoritmos para o treinamento de modelos em um ambiente FL, com foco na otimização dos dados, bem como na redução do custo computacional e energético.

Detalhamos os resultados para o MNIST na Tabela 20, avaliando o impacto dos diferentes métodos de agregação quando aplicado o algoritmo EnBaSe.

Tabela 20 – Resultados médios com FedAvg e FedProx no conjunto de dados MNIST.

Conjunto de Dados	Algoritmo	Seleção	Precisão (%)	Recall (%)	F1-Score (%)	Accuracy (%)	Tempo (s)
MNIST	FedAvg	Normal	≈ 74.4	≈ 74.5	≈ 71.8	≈ 85.7	≈ 1096.4
MNIST	**FedAvg**	**EnBaSe**	**≈ 65.9**	**≈ 66.4**	**≈ 62.7**	**≈ 78.9**	**≈ 586.5**
MNIST	FedAvg	Aleatório	≈ 42.5	≈ 40.3	≈ 64.6	≈ 56.8	≈ 555.9
MNIST	FedProx	Normal	≈ 69.3	≈ 71.4	≈ 67.5	≈ 81.7	≈ 1180.4
MNIST	**FedProx**	**EnBaSe**	**≈ 69.3**	**≈ 68.9**	**≈ 66.0**	**≈ 81.2**	**≈ 629.4**
MNIST	FedProx	Aleatório	≈ 42.5	≈ 43.0	≈ 37.8	≈ 59.1	≈ 606.1

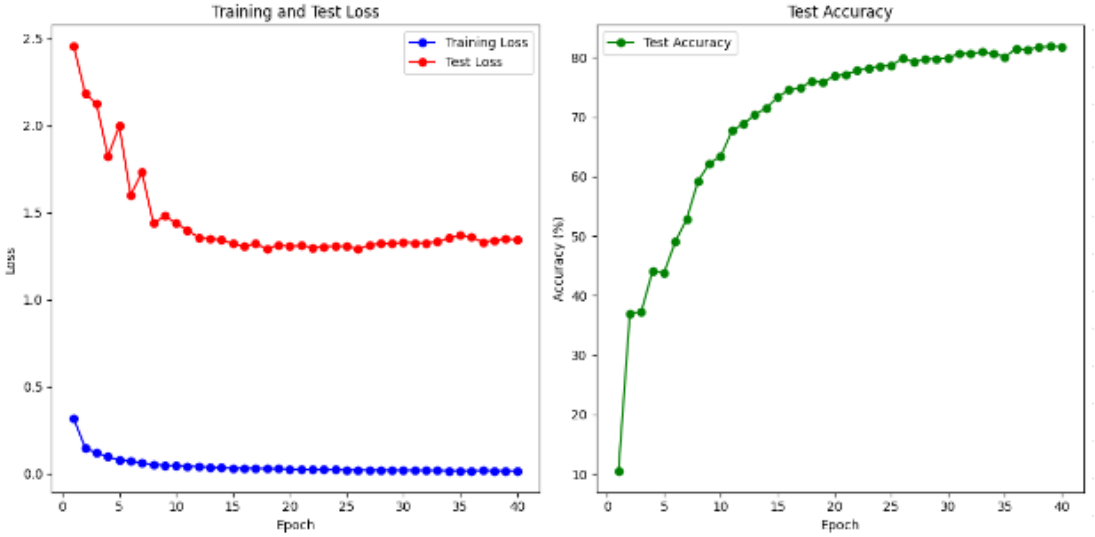
Fonte: Elaboração Própria.

Na Tabela 20, é possível observar que os algoritmos de agregação FedAvg, tanto para o EnBaSe quanto o Aleatório, demonstram notável eficiência temporal, reduzindo o tempo de execução em 46,5% e 49,2% do original, respectivamente. No entanto, enquanto o EnBaSe apresenta uma redução de acurácia de 6,8%, o Aleatório exibe uma perda mais significativa, atingindo 28,9% de acurácia. Essa comparação destaca a eficiência temporal proporcionada por ambos os métodos e revela um compromisso entre economizar tempo e manter a acurácia do modelo.

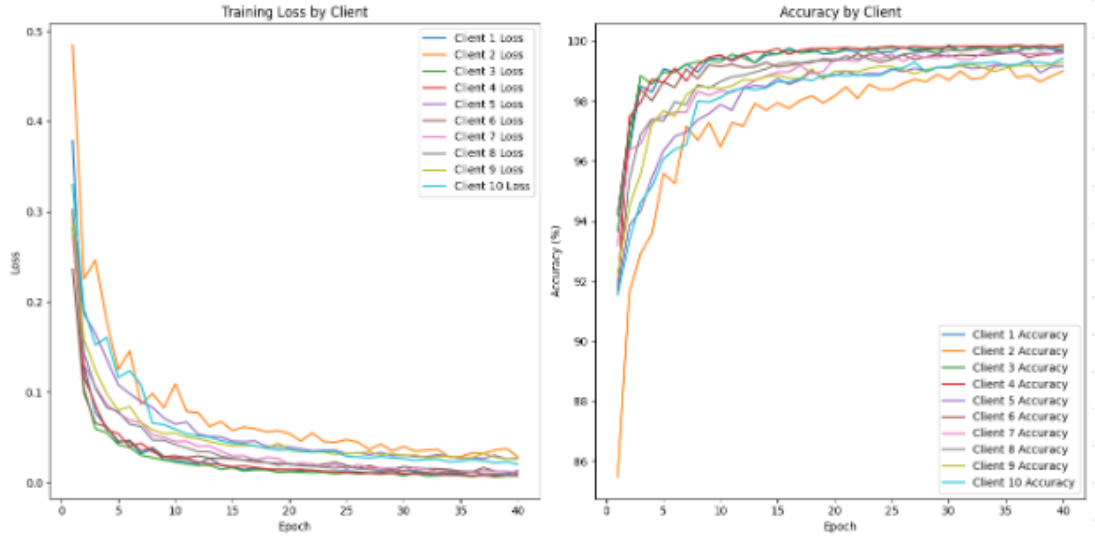
Por outro lado, quando aplicado o algoritmo de agregação FedProx, um padrão semelhante em termos de redução do tempo de execução é observado, com o EnBaSe e Aleatório alcançando 46,6% e 48,6% do tempo original, respectivamente, demonstrando também considerável eficiência temporal. No entanto, em termos de acurácia, o EnBaSe mostra um desempenho significativamente superior, com uma diminuição de apenas 0,5% de acurácia, enquanto o método Aleatório registra uma perda de de 22,6%.

Este resultado sugere que a diversidade aleatória em um cenário heterogêneo pode causar inconsistências nas atualizações do modelo global quando agregadas no servidor central. Assim, por si só, a aleatoriedade das amostras não é capaz de lidar com a heterogeneidade de dados non-iid. Além disso, é possível perceber que o algoritmo EnBaSe está, de fato, lidando com dados non-iid, conseguindo manter o balanceamento necessário para treinar uma rede neural, enquanto reduz o custo computacional.

Na Figura 44, observa-se a dinâmica de convergência do modelo global no contexto de um cenário non-iid. O algoritmo FedProx foi selecionado pela sua eficiência superior em comparação com as variantes do FedAvg testadas neste experimento. Esta figura revela a tendência de aprendizado do modelo global à medida que os clientes avançam em seu treinamento local.



(a) Modelo global utilizando o FedProx no MNIST com EnBaSe.



(b) Convergência dos clientes do algoritmo FedProx com EnBaSe.

Figura 44 – Análise da convergência do modelo global MNIST e dos clientes do algoritmo FedProx com EnBaSe.

Fonte: Elaboração Própria.

Um padrão ruidoso é identificado na atualização e no treinamento do modelo global, resultando em uma curva de convergência igualmente ruidosa. "Ruidoso", neste contexto, indica uma alta oscilação nos níveis. Esse fenômeno ocorre porque, em um contexto de treinamento distribuído, cada cliente (ou nó) treina o modelo com seu conjunto de dados local. Tais conjuntos de dados podem apresentar diferenças significativas em distribuição e volume, afetando suas contribuições ao modelo global de maneiras distintas.

Dessa forma, para cada dispositivo que está sendo treinado individualmente com dados não homogêneos, os pesos das sinapses são ajustados especificamente para ele. Posteriormente, os pesos de diferentes modelos, treinados com tipos distintos de dados, são enviados para

agregação. Isso implica no envio de pesos ajustados para conjuntos de dados diversos, o que dificulta a convergência. As variações exigem que cada nó faça ajustes específicos, considerando as propriedades de seus dados.

Essa diversidade pode causar inconsistências nas atualizações do modelo global quando agregadas no servidor central. Consequentemente, essas disparidades geram uma curva de convergência mais ruidosa para o modelo global, à medida que ele tenta se ajustar simultaneamente às várias características apresentadas por cada conjunto de dados local.

A Tabela 21 compara as estratégias FedAvg e FedProx aplicadas ao conjunto de dados Fashion-MNIST, enfatizando diferentes estratégias de seleção de dados. No contexto do FedAvg, a implementação das seleções EnBaSe e Aleatório reduz o tempo de execução para 48,7% e 47,0% do tempo original, respectivamente.

Tabela 21 – Resultados médios para FedAvg e FedProx no conjunto de dados Fashion-MNIST.

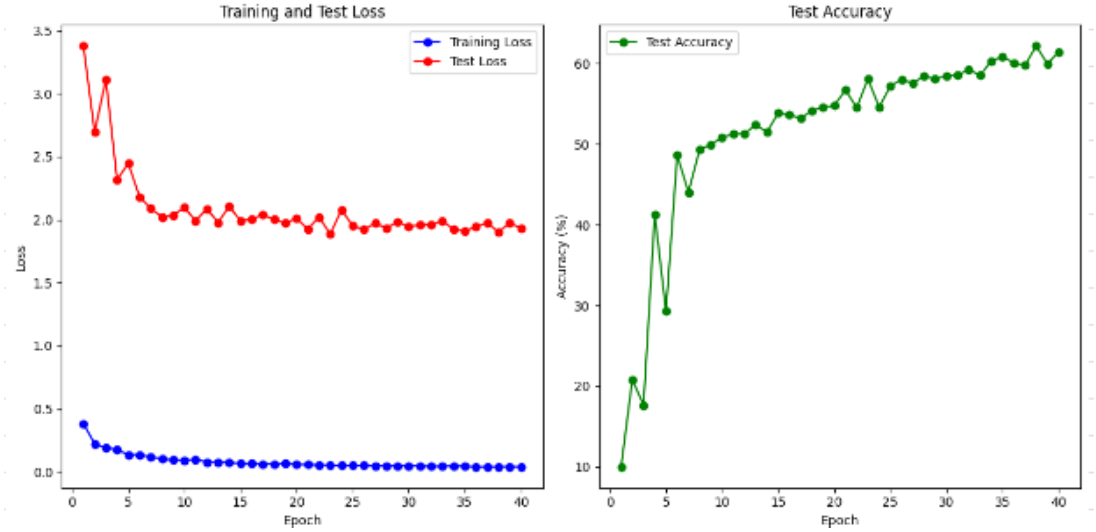
Conjunto de Dados	Algoritmo	Seleção	Precisão (%)	Recall (%)	F1-Score (%)	Accuracy (%)	Tempo (s)
Fashion	FedAvg	Execução Normal	≈ 56.7	≈ 52.8	≈ 47.4	≈ 62.2	≈ 1084.8
Fashion	**FedAvg**	**EnBaSe**	≈ **53.7**	≈ **49.7**	≈ **45.4**	≈ **58.8**	≈ **556.4**
Fashion	FedAvg	Aleatório	≈ 27.8	≈ 25.7	≈ 20.6	≈ 39.8	≈ 574.6
Fashion	FedProx	Execução Normal	≈ 60.3	≈ 57.0	≈ 52.3	≈ 66.4	≈ 1125.5
Fashion	**FedProx**	**EnBaSe**	≈ **55.7**	≈ **52.3**	≈ **47.5**	≈ **60.8**	≈ **601.3**
Fashion	FedProx	Aleatório	≈ 26.4	≈ 24.7	≈ 18.8	≈ 35.9	≈ 632.3

Fonte: Elaboração Própria.

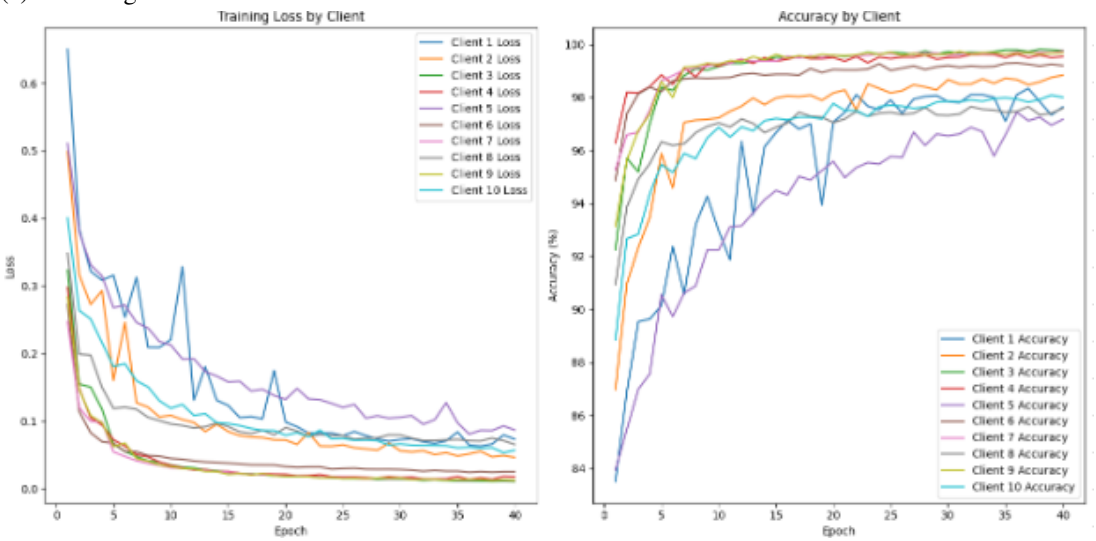
Para o Fedvag, o algoritmo EnBaSe registra uma perda de 3,4%, enquanto a seleção Aleatória apresenta uma queda mais acentuada de 22,3% na acurácia. Esse resultado indica que, embora ambas as estratégias ofereçam vantagens em eficiência temporal similar, porém EnBaSe é capaz de conservar a capacidade de generalização para a rede neural.

Por outro lado, quando aplicado ao FedProx, o EnBaSe e Aleatória reduzem o tempo de execução para 46,5% e 43,8% do original, respectivamente. No entanto, observa-se uma diferença no impacto sobre a acurácia: o EnBaSe mostra uma redução de 5,7%, enquanto a seleção aleatória experimenta uma perda substancial de 30,5%.

Na Figura 45, representamos a performance obtida nos experimentos relacionados ao Fashion-MNIST. Ao analisar essa figura, é evidente que as curvas de convergência do modelo global apontam para desafios significativos enfrentados pelos algoritmos e métodos selecionados, especialmente no que diz respeito à capacidade de generalização do modelo.



(a) Modelo global utilizando o FedProx no Fashion-MNIST com EnBaSe.



(b) Convergência dos clientes do algoritmo FedProx no Fashion-MNIST com EnBaSe.

Figura 45 – Análise do modelo global do Fashion-MNIST e da convergência de clientes do algoritmo FedProx com EnBaSe.

Fonte: Elaboração Própria.

Inicialmente, observa-se uma variação significativa nas distribuições de dados entre diferentes nós (clientes), indicando a necessidade de um ajuste mais refinado dos hiperparâmetros. Experimentos adicionais seriam necessários para ajustar esses hiperparâmetros ou para a criação de uma nova arquitetura de rede neural em busca de alta precisão.

Por fim, os algoritmos de agregação adotados, como o FedAvg e o FedProx, podem não ser totalmente adequados para superar os desafios específicos apresentados pelo problema de dados limitados, como neste experimento, ou pela pouca variabilidade dos dados disponíveis em cada nó (cliente).

A Tabela 22 fornece uma visão detalhada do desempenho dos algoritmos FedAvg

e FedProx aplicados aos conjuntos de dados CIFAR-10, com foco na influência de diferentes métodos de seleção de dados — Execução Normal, EnBaSe e Aleatório.

Tabela 22 – Resultados médios para FedAvg e FedProx no conjunto de dados CIFAR-10.

Conjunto de Dados	Algoritmo	Seleção	Precisão (%)	Recall (%)	F1-Score (%)	Accuracy (%)	Tempo (s)
CIFAR-10	FedAvg	Execução Normal	≈ 47.4	≈ 71.7	≈ 33.9	≈ 42.4	≈ 14941.2
CIFAR-10	**FedAvg**	**EnBaSe**	**≈ 43.8**	**≈ 38.3**	**≈ 32.7**	**≈ 40.3**	**≈ 8259.0**
CIFAR-10	FedAvg	Aleatório	≈ 9.0	≈ 10.0	≈ 2.7	≈ 10.0	≈ 8207.4
CIFAR-10	FedProx	Execução Normal	≈ 46.1	≈ 38.7	≈ 32.3	≈ 39.5	≈ 15662.9
CIFAR-10	**FedProx**	**EnBaSe**	**≈ 45.0**	**≈ 38.1**	**≈ 32.2**	**≈ 39.9**	**≈ 8413.0**
CIFAR-10	FedProx	Aleatório	≈ 8.5	≈ 10.0	≈ 3.1	≈ 10.1	≈ 8392.9

Fonte: Elaboração Própria.

A análise dos resultados anteriores corrobora a escolha do FedProx com EnBaSe como preferencial para o CIFAR-10, conforme ilustrado na Figura 46. Essa preferência pelo FedProx deve-se à sua capacidade de mitigar variações, facilitando o desenvolvimento de um modelo global mais estável.

No contexto do FedAvg, as abordagens EnBaSe e Aleatória reduziram o tempo de treinamento para 44,7% e 45,0% do tempo original, respectivamente. A abordagem do algoritmo EnBaSe registrou uma perda de acurácia de 2,1%, enquanto o método Aleatório apresentou uma diminuição significativa de 32,4% na acurácia.

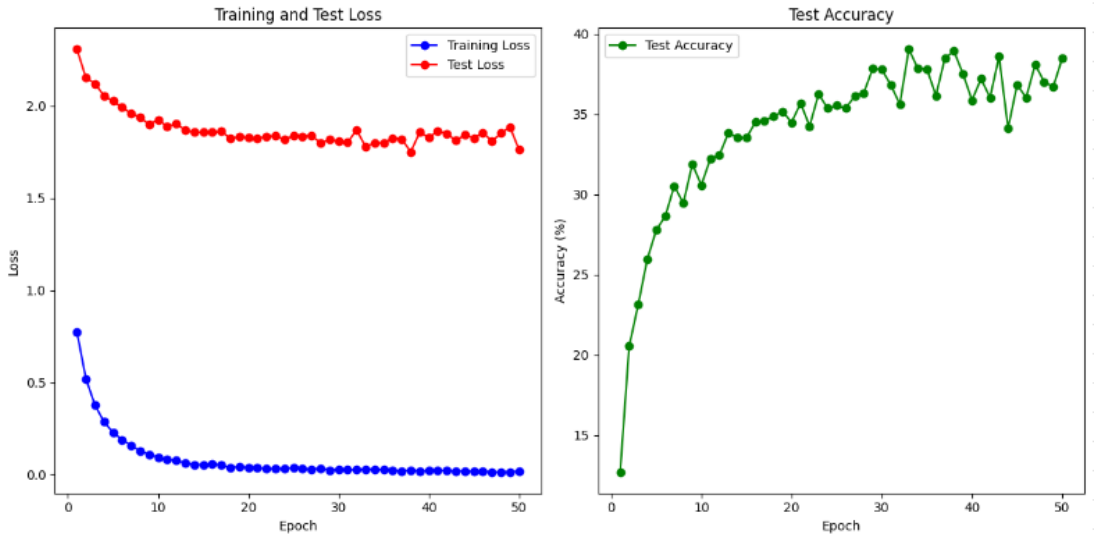
Por outro lado, ao usar o FedProx, os tempos de treinamento com EnBaSe e Aleatório foram reduzidos para 46,2% e 46,4% do tempo original, respectivamente. Interessantemente, a abordagem EnBaSe, nesse cenário, não apenas evitou uma perda de acurácia, mas alcançou um aumento de 0,4%. Em contraste, a seleção aleatória, mesmo com o FedProx, resultou em uma redução de acurácia de 29,4%.

As observações relacionadas à Figura 46 destacam os desafios específicos do aprendizado federado, particularmente em conjuntos de dados complexos como o CIFAR-10. Nesse contexto, o modelo global enfrenta dificuldades em aprender características gerais aplicáveis a todos os participantes (nós/clientes), resultando em flutuações na curva de aprendizado à medida que o modelo tenta ajustar-se às contribuições divergentes dos nós.

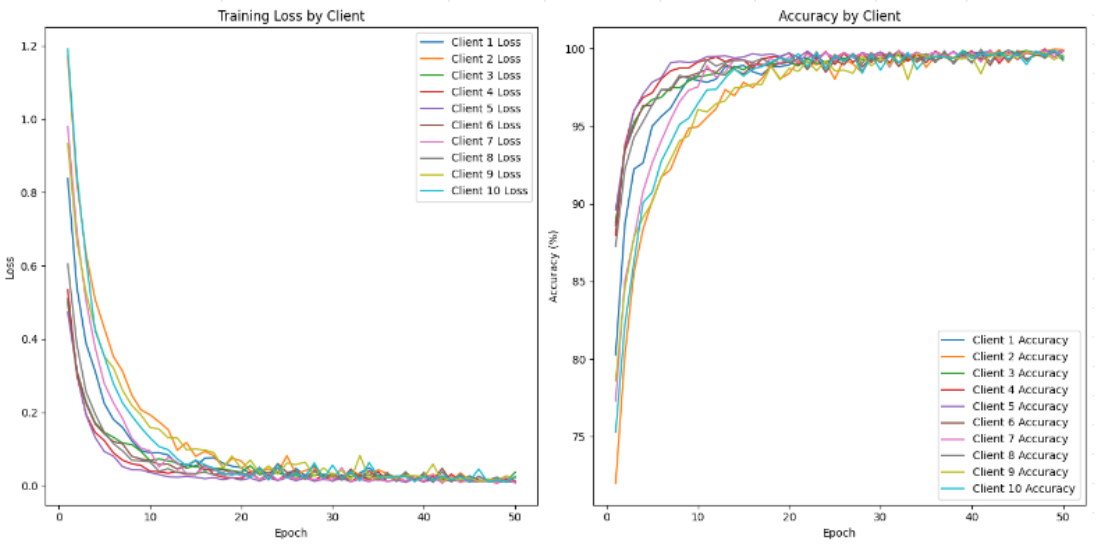
Observa-se também que, individualmente, os modelos de cada dispositivo apresentam alta acurácia e precisão para seus próprios dados. No entanto, o algoritmo utilizado para gerar o cenário non-iid não garante quantidades fixas de classes ou de dados para cada dispositivo.

Assim, uma hipótese seria que, para esse ambiente de FL, talvez sejam necessários ajustes adicionais na arquitetura da rede neural, bem como testes empíricos nos hiperparâmetros, para alcançar uma maior convergência no modelo de agregação. Alternativamente, um maior

número de participantes, que possam apresentar dados relevantes às classes ausentes, poderia contribuir para uma melhora na convergência.



(a) Modelo global utilizando o FedProx no CIFAR-10 com EnBaSe.



(b) Convergência dos clientes do algoritmo FedProx no CIFAR-10 com EnBaSe.

Figura 46 – Análise da convergência do modelo global CIFAR-10 e dos clientes para o algoritmo FedProx com EnBaSe.

Fonte: Elaboração Própria.

A Tabela 23 analisa o desempenho dos algoritmos FedAvg e FedProx aplicados ao conjunto de dados CIFAR-100, notório por sua ampla diversidade de classes e pela complexidade de suas imagens. Esta análise destaca o impacto de diferentes estratégias de seleção de dados — incluindo a Execução Normal, Entropia (EnBaSe) e Aleatória.

Tabela 23 – Resultados médios para FedAvg e FedProx no conjunto de dados CIFAR-100.

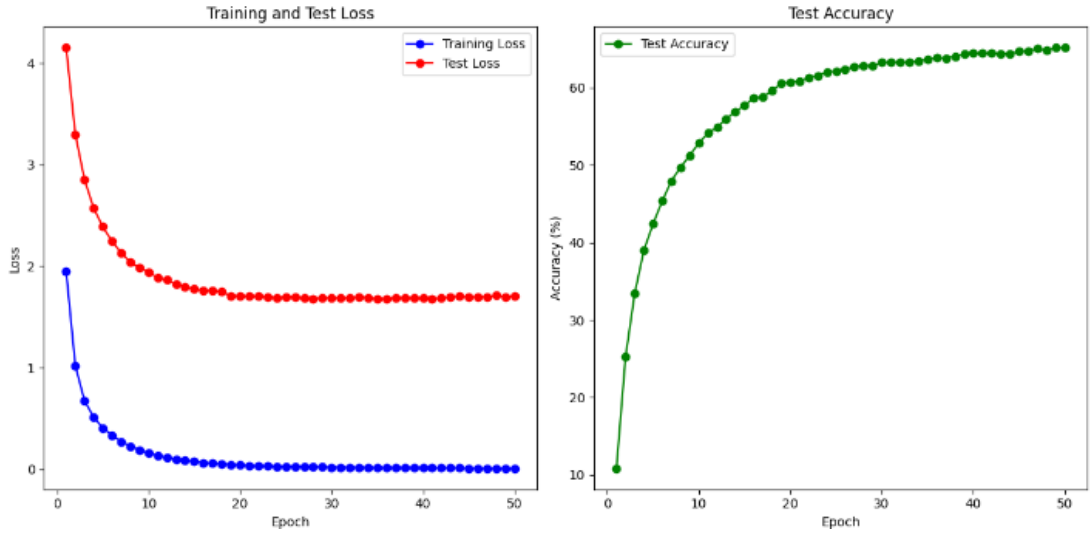
Conjunto de Dados	Algoritmo	Seleção	Precisão (%)	Recall (%)	F1-Score (%)	Accuracy (%)	Tempo (s)
CIFAR-100	FedAvg	Execução Normal	≈ 47.8	≈ 41.2	≈ 34.8	≈ 43.2	≈ 15214.9
CIFAR-100	**FedAvg**	**EnBaSe**	**≈ 44.3**	**≈ 38.6**	**≈ 32.8**	**≈ 41.6**	**≈ 8132.0**
CIFAR-100	FedAvg	Aleatório	≈ 8.3	≈ 10.0	≈ 2.3	≈ 10.0	≈ 8274.2
CIFAR-100	FedProx	Execução Normal	≈ 55.4	≈ 55.9	≈ 51.9	≈ 63.0	≈ 18020.3
CIFAR-100	**FedProx**	**EnBaSe**	**≈ 48.0**	**≈ 46.4**	**≈ 43.0**	**≈ 53.5**	**≈ 8202.7**
CIFAR-100	FedProx	Aleatório	≈ 27.7	≈ 35.7	≈ 28.3	≈ 41.2	≈ 8418.3

Fonte: Elaboração Própria.

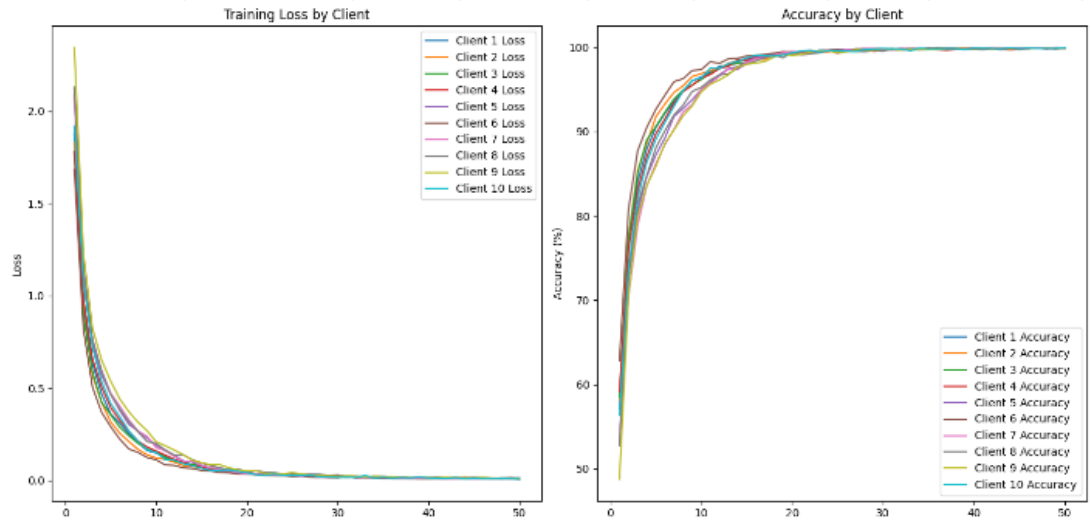
No contexto do FedAvg, o EnBaSe e Aleatória reduziram o tempo de treinamento para 46,5% e 45,6% do original, respectivamente. No entanto, enquanto o EnBaSe registrou uma diminuição na acurácia de 1,6%, a Aleatória resultou em uma queda significativa de 31,7%.

Por outro lado, ao usar o FedProx, o tempo de treinamento foi ainda mais reduzido para 54,4% e 53,2%, respectivamente, para EnBaSe e Aleatório, mostrando uma eficiência superior em comparação ao FedAvg. O EnBaSe, sob o FedProx, apresentou uma redução na acurácia de 9,5%, enquanto o método Aleatório experimentou uma diminuição de 12,3%, indicando um impacto menos severo no desempenho em comparação às reduções observadas com o FedAvg.

A curva de treinamento dos dispositivos locais e do modelo global é representada na Figura 47, para o EnBaSe aplicado ao algoritmo de agregação global FedProx.



(a) Modelo global utilizando o FedProx no CIFAR-100 com EnBaSe.



(b) Convergência dos clientes do algoritmo FedProx no CIFAR-100 com EnBaSe.

Figura 47 – Análise da convergência do modelo global do CIFAR-100 e dos clientes para o algoritmo FedProx com EnBaSe.

Fonte: Elaboração Própria.

Para todos os clientes (nós), o algoritmo FedProx com EnBaSe promoveu uma sincronização eficaz entre os ajustes feitos localmente pelos participantes e a atualização do modelo global. Esse efeito é particularmente relevante no CIFAR-100, que contém 100 classes, onde a diversidade e a complexidade dos dados apresentam desafios notáveis para a generalização eficaz e o aprendizado colaborativo.

5.3.3 Benchmark: Múltiplos Clientes cenário non-iid

Nesta seção, revisaremos o experimento conduzido sob o cenário non-iid, utilizando os mesmos critérios para as métricas de avaliação, conforme especificado na Seção 4.5, onde

foram empregadas as mesmas redes neurais utilizadas no experimento non-iid anterior. Para testar a eficácia do modelo em condições desafiadoras e exigentes, mantivemos fixas a arquitetura e os parâmetros da rede neural. Aumentamos o número de clientes (nós) de 10 para 50 e o número de épocas para 100, avaliando a capacidade do algoritmo sob uma carga de trabalho maior e maior diversidade de clientes (nós).

Na Tabela 24, apresentamos o resultado de um *benchmark*. Dessa forma, podemos observar que, com um grande número de nós (clientes), representando, assim, vários dispositivos e conjuntos de dados, o algoritmo EnBaSe conseguiu se aproximar de uma convergência suave e alta precisão. Além disso, o modelo alcançou o platô de máxima precisão da arquitetura da rede neural muito antes do número de épocas definido para o treinamento. Como o modelo EnBaSe reduz o tempo de processamento para aproximadamente metade, o *benchmark* apresentado na tabela indica que o custo energético e computacional seria consideravelmente maior se o experimento fosse conduzido novamente sem o algoritmo EnBaSe proposto neste trabalho.

A Tabela 24 mostra o desempenho do algoritmo de agregação FedProx nos conjuntos de dados CIFAR-10 e CIFAR-100, utilizando o EnBaSe. A análise dos resultados revela que o aumento no número de épocas e de clientes contribuiu para uma maior diversidade de dados, favorecendo o processo de aprendizagem. Esse aumento resultou em uma melhor convergência e otimização do modelo. Esses resultados são consistentes, de forma proporcional, com aqueles apresentados na Tabela 22.

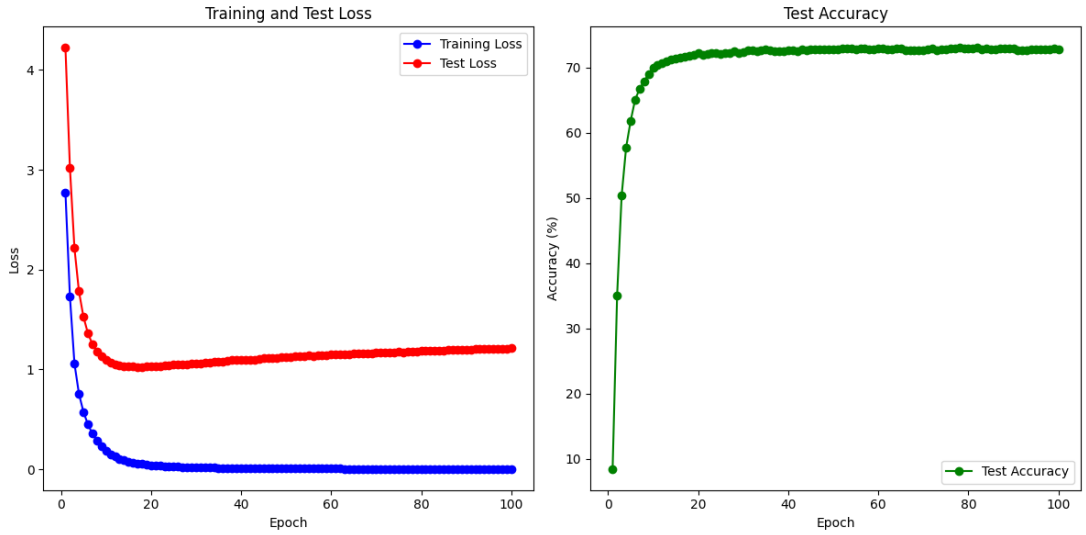
Tabela 24 – Benchmark do EnBaSe nos conjuntos de dados CIFAR-10 e CIFAR-100.

Conjunto de Dados	**Algoritmo**	**Precisão**	**Recall**	**F1-Score**	**Accuracy (%)**	**Loss**	**Tempo (s)**
CIFAR-10	FedProx	82.2	81.5	81.3	84.4	0.515	47011.90
CIFAR-100	FedProx	71.7	70.8	70.6	72.8	1.216	46983.84

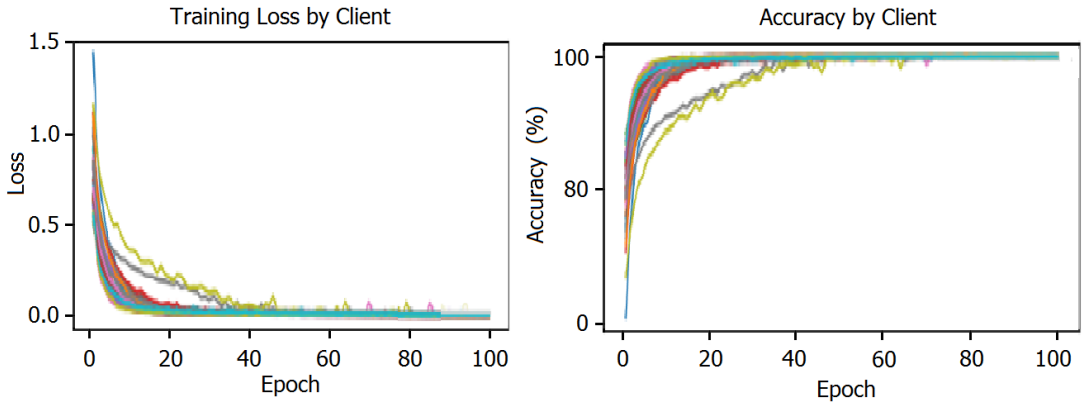
Fonte: Elaboração Própria.

O *benchmark* foi realizado nos *datasets* mais desafiadores encontrados nos experimentos até agora, incluindo especificamente o CIFAR-10 e o CIFAR-100. Esses conjuntos de dados são mais complexos por incluírem canais de cores (RGB), enquanto o MNIST e o Fashion-MNIST apresentam canais em escala de cinza. Além disso, o CIFAR-100 possui uma quantidade de 100 classes, representando, assim, uma visão geral promissora do comportamento do algoritmo quando há maior variabilidade de clientes (nós) para selecionar os dados e maior tempo de treinamento.

A Figura 48 ilustra o comportamento do FedProx com EnBaSe aplicado ao conjunto de dados CIFAR-10.



(a) Convergência global do modelo em 100 épocas para o CIFAR-10 com EnBaSe.



(b) Convergência dos clientes do algoritmo FedProx em 100 épocas para 50 clientes CIFAR-10 com EnBaSe.

Figura 48 – Convergência global do modelo e dos clientes para 100 épocas e 50 clientes CIFAR-10 para o algoritmo FedProx com EnBaSe.

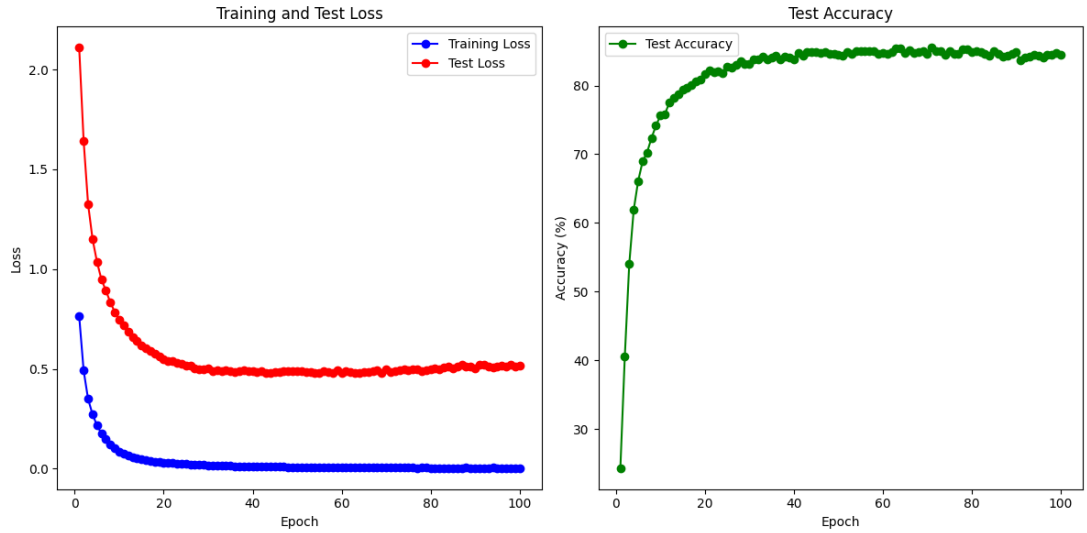
Fonte: Elaboração Própria.

Podemos observar que parte do grupo de clientes (nós) não foi capaz de convergir como a maioria, impactando significativamente o balanceamento dos pesos do modelo global. Esses impactos são naturalmente encontrados em ambientes IoT, devido à alta heterogeneidade dos dados non-iid, onde, nos dispositivos, não há garantias de quantidade de dados disponíveis, quantidade de classes ou qualidade dos dados em cada dispositivo, o que afeta negativamente o processo de treinamento.

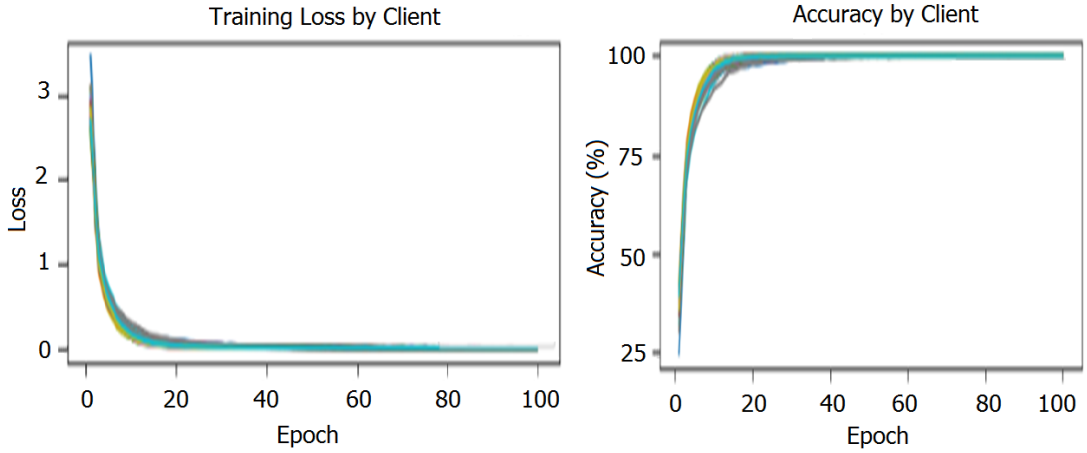
É importante observar o padrão em que o algoritmo EnBaSe foi capaz de manter o balanceamento entre eficiência e capacidade de seleção amostral, controlando a qualidade da entrada de dados para que a rede neural pudesse treinar eficientemente. Essa propriedade não foi demonstrada apenas no *benchmark*, mas também nos experimentos com 10 clientes (nós), sendo

ainda mais evidente com uma maior diversidade de clientes (nós).

Finalmente, seguindo o mesmo padrão de comportamento, o *benchmark* para o CIFAR-100 foi realizado com o algoritmo EnBaSe. Dessa forma, podemos observar, na Figura 49, resultados de convergência do modelo global suaves entre vários clientes (nós).



(a) Convergência global do modelo em 100 épocas para o CIFAR-100 com EnBaSe.



(b) Convergência dos clientes do algoritmo FedProx em 100 épocas para 50 clientes CIFAR-100 com EnBaSe.

Figura 49 – Convergência global do modelo e dos clientes para 100 épocas e 50 clientes CIFAR-100 para o algoritmo FedProx com EnBaSe.

Fonte: Elaboração Própria.

5.3.4 Benchmarking EnBaSe: Comparação com Modelos Recentes

A Tabela 25 apresenta os resultados de *benchmarking* do EnBaSe em comparação com estudos recentes, focando em seu impacto na acurácia. Os resultados de *benchmarking* para os conjuntos de dados CIFAR-10 e CIFAR-100 demonstram o desempenho do FedProx com EnBaSe ao longo de 100 épocas e 50 clientes.

Tabela 25 – Comparação do EnBaSe com modelos recentes do estado da arte.

Conjunto de Dados	Arquitetura	Autor	Modelo	Acurácia (%)
CIFAR-10	ConvNet	(**??**)	FedCOME	75,88
	VGG11	(JU; ZHANG; TOOR; HELLANDER, 2024)	AdaFedAdam	72,77
	ResNet-50	**Nosso Modelo**	**FedProx (EnBaSe)**	**84,46**
	CNN	(**??**)	FedPer++	85,09
	ResNet-50	(**??**)	FedAvg (Adaptado)	90,80
CIFAR-100	ConvNet	(**??**)	FedCOME	37,66
	ResNet-56	(HAMIDI; TAN; YE; YANG, 2024)	Fed-IT	39,29
	CCT-2	(MORAFAH; REISSER; LIN; LOUIZOS, 2024)	FedAvg-Vanilla	40,36
	ResNet-18	(**??**)	FedProx (FedFed)	70,02
	ResNet-50	**Nosso Modelo**	**FedProx (EnBaSe)**	**72,84**

Fonte: Elaboração Própria.

No conjunto CIFAR-10, o modelo FedCOME alcançou uma acurácia de 75,88%, enquanto o AdaFedAdam alcançou 72,77%. Os modelos FedPer++ e FedAvg (Adaptado) obtiveram 85,09% e 90,80%, respectivamente. O EnBaSe combinado com FedProx obteve 84,46%, ficando próximo ao FedPer++. Embora o EnBaSe não tenha superado o FedAvg (Adaptado) em termos de acurácia, destacou-se por sua eficiência computacional, reduzindo os custos de processamento ao selecionar apenas metade dos dados com base na entropia, sem perdas significativas de acurácia.

No conjunto CIFAR-100, os modelos FedCOME, Fed-IT e FedAvg-Vanilla apresentaram acurácias de 37,66%, 39,29% e 40,36%, respectivamente. O FedProx (FedFed) obteve 70,02%, enquanto o EnBaSe, combinado com FedProx, alcançou o melhor desempenho, com 72,84%. Dado o alto número de classes no CIFAR-100, o EnBaSe demonstrou ser capaz de selecionar amostras relevantes enquanto mantinha uma alta acurácia com um volume reduzido de dados.

6 CONCLUSÕES E TRABALHOS FUTUROS

6.1 Discussão

Neste trabalho analisamos a hipótese de que a entropia pode ser utilizada para medir a qualidade dos dados, considerando que ela quantifica a incerteza ou imprevisibilidade em cada cliente (nó) na entrada. Onde, um valor elevado de entropia indicaria que os dados são altamente imprevisíveis ou variam consideravelmente dentro de um cliente (nó). Assim, a hipótese foi testada utilizando a baixa entropia em cada nó, a qual foi utilizada a mediana como um critério de separação para evitar assimetria entre os valores dentro da distribuição.

Uma das dificuldades encontradas na literatura durante o desenvolvimento deste estudo foi a falta de padronização nos experimentos apresentados. Muitos estudos fornecem apenas dados parciais, sem incluir métricas essenciais, como *Validation*, *F1-score*, *Recall*, valores de *loss* ou tempo de treinamento do modelo. Por fim, uma das limitações deste estudo foi o orçamento destinado aos experimentos, o que nos levou a reduzir o número de épocas para 40 e o número de clientes (nós) para 10, devido à grande quantidade de experimentos realizados.

As abordagens recentes do estado da arte em arquiteturas de FL concentram-se predominantemente em melhorar a homogeneidade dos dados e em abordar a heterogeneidade e os desafios associados às distribuições non-iid. Entretanto, tais abordagens frequentemente negligenciam considerações fundamentais sobre a capacidade computacional dos dispositivos e os recursos necessários para que as redes neurais alcancem uma alta capacidade de generalização.

Essa questão é crítica para promover uma maior equidade na integração de sistemas com baixa capacidade computacional e energética, constituindo o foco principal deste estudo. Assim, o algoritmo EnBaSe mostra-se eficiente na otimização dos recursos computacionais alocados à rede neural, alcançando até 50% do custo de computação.

Essa otimização reduz o tempo necessário para o treinamento da rede neural e, consequentemente, diminui os custos associados ao processo, uma vez que os modelos que utilizam o algoritmo EnBaSe concluem o treinamento mais rapidamente. Além disso, esse avanço é particularmente relevante para sistemas com restrições de energia e capacidade computacional, como dispositivos inteligentes que operam com recursos limitados, incluindo drones ou equipamentos médicos baseados em AI com baixa capacidade de processamento.

Por fim, as análises realizadas neste estudo confirmaram a viabilidade do uso de um algoritmo que melhora a qualidade dos dados ao mesmo tempo em que reduz os custos

computacionais diretos e indiretos.

6.2 Conclusão

Nosso trabalho propõe novas direções para a pesquisa e aplicação de FL, contribuindo para o desenvolvimento de soluções de IA mais eficientes. O algoritmo introduzido neste estudo otimiza a seleção de dados utilizando entropia em FL para dados iid e non-iid. Demonstramos como a entropia pode reduzir a quantidade de dados necessários, minimizando os custos e o tempo de processamento.

Este método pode ser particularmente poderoso em aplicações de IoT e saúde móvel, destacando seu potencial para implementações práticas. Pesquisas futuras focarão na adaptação do algoritmo para cenários mais complexos e na sua integração com outras técnicas de otimização. Em resumo, pesquisas futuras devem expandir a aplicabilidade do algoritmo para outros tipos de dados e cenários de FL. Além disso, estudos relacionados à convergência do modelo global buscarão evitar treinamentos excessivos, ao mesmo tempo em que procurarão minimizar ainda mais os custos computacionais e o consumo de energia. Seria útil explorar como o método proposto pode ser combinado com outras técnicas de ML para melhorar a eficiência e a eficácia dos modelos de treinamento de IA.

No entanto, o estudo é limitado pela disponibilidade de dados para testes mais amplos, que necessitam de uma exploração mais extensa. A complexidade da execução de testes mais aprofundados e os requisitos de conjuntos de dados representaram barreiras significativas. Assim, pesquisas futuras devem focar na aplicação do método em domínios diversos e na sua integração com outras técnicas de ML, incluindo dados discretos.

Desta forma, as **principais contribuições** deste estudo incluem:

- Um estudo detalhado sobre o comportamento da entropia em imagens e sua distribuição em CV;
- Análise do impacto de transformações lineares e da normalização na entropia dos dados;
- Redução do custo computacional em dispositivos de borda IoT;
- Apresentação de métricas detalhadas, como acurácia, F1-score, recall, valores de perda e tempo de treinamento dos modelos;
- Organização das métricas para servirem como referência para experimentos futuros;
- Comparação da acurácia em cenários iid e non-iid com outros experimentos;
- Uma revisão abrangente da literatura sobre qualidade de dados e FL; e

- O desenvolvimento do algoritmo EnBaSe, que seleciona eficientemente dados de alta qualidade com base na análise de entropia. Esse método reduz o processamento computacional desnecessário, otimiza a convergência do modelo e melhora a seleção de dados para cenários de (FL) e aprendizado centralizado, especialmente em ambientes com restrições de recursos, como a IoT.

Por fim, **as aplicações incluem**:

- Dispositivos na borda com baixo poder computacional;
- Redução do custo computacional na realização de treinamentos e testes empíricos de arquiteturas de redes neurais;
- Aplicação no campo da CV para reduzir o custo de processamento e acelerar a convergência da rede neural;
- Aplicável em cenários centralizados como uma pré-camada de processamento;
- Aplicável como pipeline em algoritmos de FL; e
- Aplicável na segmentação de dados relevantes em grandes conjuntos de dados;

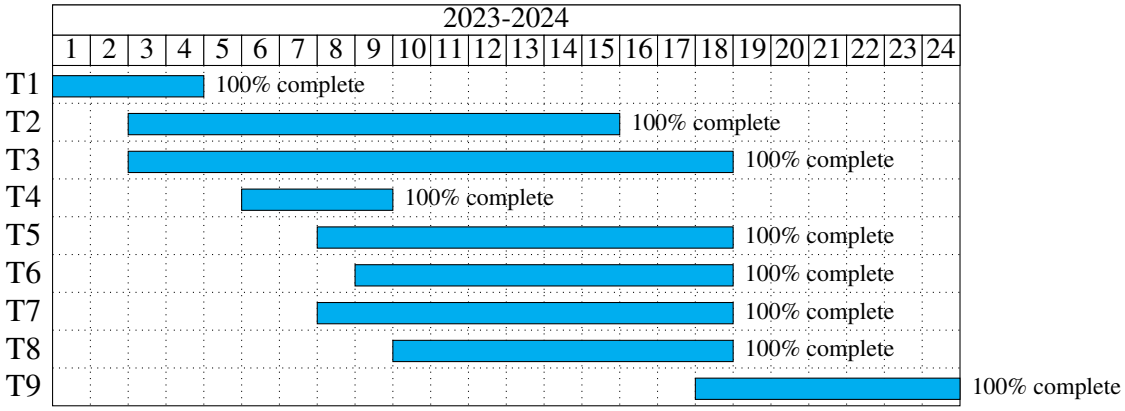
6.3 Trabalhos Futuros e Cronograma

O resumo das tarefas e objetivos do estudo é apresentado na Tabela 26: definição da questão de pesquisa (T1, WP1), revisão de literatura e tecnologia (T2, WP2), coleta de fontes acadêmicas (T3, WP3), seleção de dados (T4, WP4), construção de uma rede neural convolucional (T5, WP5), escrita de dissertações e artigos (T6, WP6), realização de experimentos para validar hipóteses (T7, WP7), escrita dos resultados obtidos nos experimentos (T8, WP8), análise de trabalhos futuros e novas linhas de pesquisa (T9, WP9).

Esta pesquisa identifica várias direções futuras, com base na elaboração das métricas e na análise proposta. Por exemplo, outro domínio de pesquisa futuro inclui a análise da entropia para a construção de modelos de aprendizado de máquina baseados em texto. Considerando que a teoria da informação oferece cálculos para medir a quantidade mínima de informação necessária, essa abordagem poderia quantificar a quantidade mínima e média de bits necessários em uma mensagem de texto para treinar um modelo de AI. Adicionalmente, mais estudos baseados em entropia poderiam ser aplicados ao treinamento de redes neurais voltados à reconstrução de imagens. Especificamente, isso incluiria o uso de entropia em *Support Vector Machines* (SVM), ou *Generative Adversarial Network* (GAN) para esse fim.

Além disso, seria interessante analisar o algoritmo EnBaSe proposto neste trabalho

Tabela 26 – Cronograma de pesquisa.



Tarefas	Objetivos	Descrição
T1	WP1	Definir a questão de pesquisa e o problema que o estudo pretende abordar.
T2	WP2	Revisão da literatura existente, teorias e tecnologias relevantes para o problema definido.
T3	WP3	Coleta sistemática de todas as fontes acadêmicas relevantes que serão referenciadas ao longo do estudo.
T4	WP4	Identificação e seleção dos conjuntos de dados que serão utilizados no estudo.
T5	WP5	Projeto e criação de uma Rede Neural Convolucional, além da análise dos dados para medir e compreender o comportamento do problema.
T6	WP6	Redação da dissertação/artigo detalhando os resultados e a metodologia da pesquisa.
T7	WP7	Execução de experimentos e testes utilizando a rede neural e o algoritmo desenvolvidos para validar a hipótese.
T8	WP8	Redação de um relatório e artigo com base nos resultados dos experimentos.
T9	WP9	Análise de uma nova linha de pesquisa com base nos resultados obtidos.

Fonte: Elaboração Própria.

quanto à sua capacidade de mitigar a influência de dados anômalos, corrompidos ou mal-intencionados inseridos no conjunto de dados. Como, estatisticamente, dados anômalos se desviam do comportamento padrão, a quantificação da incerteza e a variabilidade dos dados por classe também poderiam ser utilizados para identificar ataques, mitigar danos à integridade ou avaliar a qualidade do vídeo.

Os trabalhos futuros também incluem a validação das hipóteses de refinamento para os algoritmos Random e EnBaSe em cenários non-iid, bem como sua comparação com outros algoritmos de agregação global para FL. Além disso, a exploração de uma abordagem híbrida que combine amostragem aleatória e entropia pode ser útil, utilizando a entropia para medir a incerteza ou impureza dos dados, enquanto se adiciona variabilidade estatística.

Esses estudos também indicam a viabilidade de criar um *kernel* para analisar subconjuntos de imagens e calcular seus níveis de entropia, permitindo a segmentação de regiões de interesse com detalhes complexos. Por fim, em estudos futuros, pretendemos incluir TL e DA

nos conjuntos de dados MNIST e Fashion-MNIST, com o objetivo de analisar resultados mais robustos.

REFERÊNCIAS

ABEL, J. D. K.; DHANALAKSHMI, S.; KUMAR, R. **A comprehensive survey on signal processing and machine learning techniques for non-invasive fetal ECG extraction**. *Multimedia Tools and Applications*, Springer, v. 82, n. 1, p. 1373–1400, 2023.

AL-DULAIMY, A. et al. **The computing continuum: From IoT to the cloud**. *Internet of Things*, Amsterdam, v. 27, p. 101272, 2024. DOI https://doi.org/10.1016/j.iot.2024.101272.

AL-SAEDI, A. A.; BOEVA, V.; CASALICCHIO, E. **Fedco: Communication-efficient federated learning via clustering optimization**. *Future Internet*, MDPI, v. 14, n. 12, p. 377, 2022. DOI:https://doi.org/10.3390/fi14120377.

ANJOS, J. C. D. et al. **A survey on collaborative learning for intelligent autonomous systems**. *ACM Computing Surveys*, ACM New York, NY, v. 56, n. 4, p. 1–37, 2023. DOI:https://doi.org/10.1145/3625544.

ANJOS, J. C. D. et al. **Data processing model to perform big data analytics in hybrid infrastructures**. *IEEE Access*, IEEE, v. 8, p. 170281–170294, 2020.

ANJOS, J. C. S. d. et al. **A Survey on Collaborative Learning for Intelligent Autonomous Systems**. *ACM Comput. Surv.*, ACM, v. 56, n. 4, p. 98:1–98:37, nov. 2023. ISSN 0360-0300. DOI:10.1145/3625544.

BASSIOUNI, M. M.; CHAKRABORTTY, R. K.; SALLAM, K. M.; HUSSAIN, O. K. **Deep learning approaches to identify order status in a complex supply chain**. *Expert Systems with Applications*, v. 250, p. 123947, 2024. ISSN 0957-4174. DOI:10.1016/j.eswa.2024.123947.

BUSTINCIO, R. W. C.; SOUZA, A. M. de; COSTA, J. B. D.; BITTENCOURT, L. **EntropicFL: Efficient Federated Learning via Data Entropy and Model Divergence**. In: (ACM), A. for C. M. (Ed.). *Anais do Proceedings of the IEEE/ACM International Conference on Utility and Cloud Computing*. Taormina, 2023. p. 1–6.

CAO, L. **Beyond iid: Non-iid thinking, informatics, and learning**. *IEEE Intelligent Systems*, IEEE, v. 37, n. 4, p. 5–17, 2022. DOI:https://10.1109/TNNLS.2022.3152581.

CHAUDHARY, P. K.; GUPTA, V.; PACHORI, R. B. **Fourier-Bessel representation for signal processing: A review**. *Digital Signal Processing*, Elsevier, p. 103938, 2023. DOI:doi.org/10.1016/j.dsp.2023.103938.

CHEN, H.; VIKALO, H. et al. **The best of both worlds: Accurate global and personalized models through federated learning with data-free hyper-knowledge distillation**. *arXiv preprint arXiv:2301.08968*, 2023. DOI:10.48550/arXiv.2301.08968.

CHEN, J. et al. **Discovery of Stomach Adenocarcinoma Biomarkers by Consensus Scoring of Random Sampling and Machine Learning Modeling**. In: *Anais do International Conference on Bioinformatics and Computational Biology (ICBCB)*. Seoul: IEEE, 2022. p. 112–115.

CHLAP, P. et al. **A review of medical image data augmentation techniques for deep learning applications**. *Journal of Medical Imaging and Radiation Oncology*, Wiley Online Library, v. 65, n. 5, p. 545–563, 2021. DOI:doi.org/10.1111/1754-9485.13261.

COUTINHO-ALMEIDA, J.; CRUZ-CORREIA, R. J.; RODRIGUES, P. P. **Evaluating distributed-learning on real-world obstetrics data: comparing distributed, centralized and local models**. *Scientific Reports*, Nature Publishing Group UK London, London, v. 14, n. 1, p. 11128, 2024. Disponível em: doi.org/10.1038/s41598-024-61371-1. Acesso em: 20 mai. 2024.

CRIADO, M. F. et al. **Non-iid data and continual learning processes in federated learning: A long road ahead**. *Information Fusion*, Elsevier, v. 88, p. 263–280, 2022. DOI:doi.org/10.1016/j.inffus.2022.07.024.

DOLAAT, K. M. M.; ERBAD, A.; IBRAR, M. **Enhancing Global Model Accuracy: Federated Learning for Imbalanced Medical Image Datasets**. In: INTERNATIONAL SYMPOSIUM ON NETWORKS, COMPUTERS AND COMMUNICATIONS (ISNCC), 10TH, 2023, DOHA, QATAR. *Anais do International Symposium on Networks, Computers and Communications (ISNCC)*. Doha, 2023. p. 1–4.

DU, Z. et al. **Federated learning for vehicular internet of things: Recent advances and open issues**. *IEEE Open Journal of the Computer Society*, IEEE, v. 1, p. 45–61, 2020. DOI:10.1109/OJCS.2020.2992630.

FERNANDES, J. et al. **ISABELA–a socially-aware human-in-the-loop advisor system**. *Online Social Networks and Media*, Elsevier, v. 16, p. 100060, 2020. DOI doi.org/10.1016/j.osnem.2020.100060.

GAFNI, T. et al. **Federated learning: A signal processing perspective**. *IEEE Signal Processing Magazine*, IEEE, v. 39, n. 3, p. 14–41, 2022. DOI:10.1109/MSP.2021.3125282.

GAO, D.; YAO, X.; YANG, Q. ***A survey on heterogeneous federated learning***. Preprint arXiv:2210.04505, 2022. Disponível em: https://doi.org/10.48550/arXiv.2210.04505. Acesso em: 21 fev. 2024.

HAMIDI, S. M.; TAN, R.; YE, L.; YANG, E.-H. **Fed-it: Addressing class imbalance in federated learning through an information-theoretic lens**. In: IEEE (Ed.). *Anais do IEEE International Symposium on Information Theory (ISIT)*. Athens, 2024. p. 1848–1853.

HOSSAIN, M. Z.; IMTEAJ, A. **Fedavo: Improving communication efficiency in federated learning with african vultures optimizer**. *arXiv preprint arXiv:2305.01154*, 2023. DOI:https://10.1109/COMPSAC61105.2024.00069.

HUANG, C. et al. ***Neural Collapse Inspired Federated Learning with Non-iid Data***. 2023. DOI:doi.org/10.48550/arXiv.2303.16066.

HUYNH, M.-T.; NIPPA, M.; AICHNER, T. **Big data analytics capabilities: Patchwork or progress? A systematic review of the status quo and implications for future research**. *Technological Forecasting and Social Change*, Elsevier, v. 197, p. 122884, 2023. DOI:https://doi.org/10.1016/j.techfore.2023.122884.

ILIĆ, M.; IVANOVIĆ, M.; KURBALIJA, V.; VALACHIS, A. **Towards optimal learning: Investigating the impact of different model updating strategies in federated learning**. *Expert Systems with Applications*, Elsevier, p. 123553, 2024. DOI:doi.org/10.1016/j.eswa.2024.123553.

ITAHARA, S. et al. **Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data**. *IEEE Transactions on Mobile Computing*, IEEE, v. 22, n. 1, p. 191–205, 2021. DOI:10.1109/TMC.2021.3070013.

IYER, V. N. **A review on different techniques used to combat the non-IID and heterogeneous nature of data in FL**. *arXiv preprint arXiv:2401.00809*, 2024. DOI:doi.org/10.1016/j.neucom.2021.07.098.

JAMALI-RAD, H.; ABDIZADEH, M.; SINGH, A. **Federated learning with taskonomy for non-IID data**. *IEEE transactions on neural networks and learning systems*, IEEE, v. 34, n. 11, p. 8719–8730, 2022. DOI:https://10.1109/TNNLS.2022.3152581.

JU, L.; ZHANG, T.; TOOR, S.; HELLANDER, A. **Accelerating fair federated learning: Adaptive federated adam**. *IEEE Transactions on Machine Learning in Communications and Networking*, IEEE, 2024. DOI:https://10.1109/TMLCN.2024.3423648.

JUAN, A. A. et al. **A review of the role of heuristics in stochastic optimisation: From metaheuristics to learnheuristics**. *Annals of Operations Research*, Springer, v. 320, n. 2, p. 831–861, 2023. DOI:10.1007/s10479-021-04142-9.

KAMM, S. et al. **A survey on machine learning based analysis of heterogeneous data in industrial automation**. *Computers in Industry*, Elsevier, v. 149, p. 103930, 2023. DOI:https://doi.org/10.1016/j.compind.2023.103930 .

KANG, J. et al. **Reliable federated learning for mobile networks**. *IEEE Wireless Communications*, IEEE, v. 27, n. 2, p. 72–80, 2020. DOI:https://10.1109/MWC.001.1900119.

KHAN, L. U. et al. **Federated learning for internet of things: Recent advances, taxonomy, and open challenges**. *IEEE Communications Surveys & Tutorials*, IEEE, v. 23, n. 3, p. 1759–1799, 2021.

LEE, H. **Towards Convergence in Federated Learning via Non-IID Analysis in a Distributed Solar Energy Grid**. *Electronics*, MDPI, v. 12, n. 7, p. 1580, 2023.

LEE, S. X.; MCLACHLAN, G. J. **An overview of skew distributions in model-based clustering**. *Journal of Multivariate Analysis*, Elsevier, v. 188, p. 104853, 2022. DOI:https://doi.org/10.1016/j.jmva.2021.104853.

LI, A. et al. **Lotteryfl: Empower edge intelligence with personalized and communication-efficient federated learning**. In: IEEE (Ed.). *Anais do IEEE/ACM Symposium on Edge Computing (SEC)*. San Jose: [s.n.], 2021. p. 68–79.

LI, B.; CHEN, S.; YU, K. **FeDDkw–Federated Learning with Dynamic Kullback–Leibler-divergence Weight**. *ACM Transactions on Asian and Low-Resource Language Information Processing*, ACM New York, NY, 2023. DOI:https://doi.org/10.1145/3594779.

LI, C.; LI, X.; CHEN, M.; SUN, X. **Deep Learning and Image Recognition**. In: *Anais do IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT)*. Qingdao, China: [s.n.], 2023. p. 557–562.

LI, Q.; DIAO, Y.; CHEN, Q.; HE, B. **Federated learning on non-iid data silos: An experimental study**. In: IEEE (Ed.). *Anais do IEEE 38th international conference on data engineering (ICDE)*. Kuala Lumpur, 2022. p. 965–978.

LI, T. et al. **Federated Optimization in Heterogeneous Networks**. In: DHILLON, I. S.; PAPAILIOPOULOS, D. S.; SZE, V. (Ed.). ***Proceedings of Machine Learning and Systems 2020, MLSys 2020***. Austin,TX, USA: mlsys.org, 2020. v. 2, p. 429–450. DOI: https://doi.org/10.48550/.

LI, Y.; CHAO, X.; ERCISLI, S. ***Disturbed-entropy: a simple data quality assessment approach.*** *ICT Express.* 2022. DOI:https://doi.org/10.1016/j.icte.2022.01.006.

LI, Z.; LIN, T.; SHANG, X.; WU, C. **Revisiting weighted aggregation in federated learning with neural networks**. In: PMLR (Ed.). *Anais do International Conference on Machine Learning*. Honolulu: [s.n.], 2023. p. 19767–19788.

LO, S. K. et al. **Toward Trustworthy AI: Blockchain-Based Architecture Design for Accountability and Fairness of Federated Learning Systems**. *IEEE Internet of Things Journal*, v. 10, p. 3276–3284, 2023. DOI:10.1109/JIOT.2022.3144450.

LU, S. et al. **Edge computing on IoT for machine signal processing and fault diagnosis: A review**. *IEEE Internet of Things Journal*, IEEE, 2023. DOI:10.1109/JIOT.2023.3239944.

MA, X. et al. **A state-of-the-art survey on solving non-iid data in federated learning**. *Future Generation Computer Systems*, Elsevier, v. 135, p. 244–258, 2022. DOI:https://https://doi.org/10.1016/j.future.2022.05.003.

MACNELL, N. et al. **Implementing machine learning methods with complex survey data: Lessons learned on the impacts of accounting sampling weights in gradient boosting**. *Plos one*, Public Library of Science San Francisco, CA USA, v. 18, n. 1, p. e0280387, 2023. DOI:https://doi.org/10.1371/journal.pone.0280387.

MCMAHAN, B. et al. **Communication-Efficient Learning of Deep Networks from Decentralized Data**. In: ***Proceedings of the 20th International Conference on Artificial Intelligence and Statistics***. [S.l.]: PMLR, 2017. (Proceedings of Machine Learning Research, v. 54), p. 1273–1282. DOI:10.48550/arXiv.1602.05629.

MITTAL, S.; SRIVASTAVA, S.; JAYANTH, J. P. **A Survey of Deep Learning Techniques for Underwater Image Classification**. *IEEE Transactions on Neural Networks and Learning Systems*, v. 34, n. 10, p. 6968–6982, 2023.

MORAFAH, M.; REISSER, M.; LIN, B.; LOUIZOS, C. **Stable Diffusion-based Data Augmentation for Federated Learning with Non-IID Data**. *arXiv preprint arXiv:2405.07925*, 2024. DOI:https://doi.org/10.48550/arXiv.2405.07925.

MUNAPPY, A. R. et al. **Data management for production quality deep learning models: Challenges and solutions**. *Journal of Systems and Software*, Elsevier, v. 191, p. 111359, 2022. DOI:https://doi.org/10.1016/j.jss.2022.111359.

MURALIDHARAN, S. et al. **Scalable Prediction Models for Airbnb Listing in Spark Big Data Cluster using GPU-accelerated RAPIDS**. Array, 2022.

NIU, S.; LIU, Y.; WANG, J.; SONG, H. **A decade survey of transfer learning (2010–2020)**. *IEEE Transactions on Artificial Intelligence*, IEEE, v. 1, n. 2, p. 151–166, 2020.

ORLANDI, F. C. et al. **Entropy to mitigate non-IID data problem on Federated Learning for the Edge Intelligence environment**. *IEEE Access*, IEEE Computer Society, v. 11, p. 78845–78857, jul. 2023. ISSN 2169-3536. DOI:https://10.1109/ACCESS.2023.3298704.

OTTONI, A. L. C.; AMORIM, R. M. de; NOVO, M. S.; COSTA, D. B. **Tuning of data augmentation hyperparameters in deep learning to building construction image classification with small datasets**. *International Journal of Machine Learning and Cybernetics*, Springer, v. 14, n. 1, p. 171–186, 2023. DOI:doi.org/10.1007/s13042-022-01555-1.

QIAO, Y.; LE, H. Q.; HONG, C. S. **Boosting federated learning convergence with prototype regularization**. *arXiv preprint arXiv:2307.10575*, 2023. Disponível em: https://doi.org/10.48550/arXiv.2307.10575. Acesso em: 06 jun. 2024.

RAJPUT, D.; WANG, W.-J.; CHEN, C.-C. **Evaluation of a decided sample size in machine learning applications**. *BMC bioinformatics*, Springer, v. 24, n. 1, p. 48, 2023. DOI:https://doi.org/10.1186/s12859-023-05156-9.

RAO, B. et al. **Privacy inference attack and defense in centralized and federated learning: A comprehensive survey**. *IEEE Transactions on Artificial Intelligence*, IEEE, Piscataway, NJ, 2024. Disponível em:10.1109/TAI.2024.3363670 Acesso em: 19 fev. 2024.

RODRÍGUEZ-BARROSO, N.; MARTÍNEZ-CÁMARA, E.; LUZÓN, M. V.; HERRERA, F. **Backdoor attacks-resilient aggregation based on robust filtering of outliers in federated learning for image classification**. *Knowledge-Based Systems*, Elsevier, v. 245, p. 108588, 2022.

ROSENDO, D.; COSTAN, A.; VALDURIEZ, P.; ANTONIU, G. **Distributed intelligence on the Edge-to-Cloud Continuum: A systematic literature review**. *Journal of Parallel and Distributed Computing*, Elsevier, 2022. DOI doi.org/10.1016/j.jpdc.2022.04.004.

SABAH, F. et al. **Model optimization techniques in personalized federated learning: A survey**. *Expert Systems with Applications*, Elsevier, p. 122874, 2023. DOI:10.1016/j.eswa.2023.122874.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. ***Understanding Machine Learning: From Theory to Algorithms***. [S.l.]: Cambridge University Press, 2014.

SHENG, T. et al. **Modeling global distribution for federated learning with label distribution skew**. *Pattern Recognition*, Elsevier, Amsterdam, v. 143, p. 109724, 2023. Disponível em:https://doi.org/10.1016/j.patcog.2023.109724. Acesso em: 05 mar. 2024.

SHORTEN, C.; KHOSHGOFTAAR, T. M.; FURHT, B. **Text data augmentation for deep learning**. *Journal of big Data*, Springer, v. 8, p. 1–34, 2021.

SINGH, D. ***Internet of Things***. [S.l.]: John Wiley Sons, Ltd, 2023. 195-227 p. Disponível em: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119865216.ch9. Acesso em: 04 jun. 2023. ISBN 9781119865216.

SUN, H. et al. **Toward communication-efficient federated learning in the Internet of Things with edge computing**. *IEEE Internet of Things Journal*, IEEE, v. 7, n. 11, p. 11053–11067, 2020. DOI:10.1109/JIOT.2020.2994596.

SUN, Q. et al. **Shapleyfl: Robust federated learning based on shapley value**. In: ACM (Ed.). *Anais do Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Long Beach, 2023. p. 2096–2108.

TAO, Z.; WU, J.; LI, Q. **Preconditioned Federated Learning**. *arXiv preprint arXiv:2309.11378*, 2023. DOI:https://doi.org/10.48550/arXiv.2309.11378.

TAYLOR, P. ***Global big data analytics market size 2021-2029***. 2022. Disponível em: https://www.statista.com/statistics/1336002/big-data-analytics-market-size/. Acesso em: 11 jun. de 2024.

TAYLOR, P. ***Amount of data created, consumed, and stored 2010-2020, with forecasts to 2025***. 2023. Disponível em: https://www.statista.com/statistics/871513/worldwide-data-created/. Acesso em: 12 de ago. de 2024.