Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Department of Mathematics and Physics

Master of Data Science

A Comparative Analysis Of Algorithms To Address The Imbalanced Dataset Problem In Federated Learning

---

THESIS to obtain the **DEGREE** of
MASTER OF DATA SCIENCE

A thesis presented by:
Erika Susana Durán González

Thesis Advisors:
Dr. Gema Berenice Gudiño Mendoza

Tlaquepaque, Jalisco, May, 2025

Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Department of Mathematics and Physics

Master of Data Science Approval Form

Thesis Title: **A Comparative Analysis Of Algorithms To Address The Imbalanced Dataset Problem In Federated Learning**

Author: **Erika Susana Durán González**

Thesis Approved to complete all degree requirements for the Master of Science Degree in Data Science.

Thesis Advisor, **Dr. Gema Berenice Gudiño Mendoza**

Thesis Reader, **Dr. Iván Esteban Villalón Turrubiates**

Thesis Reader, **Dr. Edgar Alejandro Guerrero Arroyo**

Academic Advisor, **Dr. Rocio Carrasco Navarro**

Tlaquepaque, Jalisco, May, 2025

A Comparative Analysis Of Algorithms To Address The Imbalanced Dataset Problem In Federated Learning

Erika Susana Durán González

Abstract

Traditional training in Machine Learning (ML) algorithms requires data collected from various devices to be transferred to a central server, which poses potential security and data-privacy risks. An additional critical aspect of machine learning is class imbalance, which arises when certain classes are underrepresented, potentially leading to suboptimal performance, particularly for minority class data. Different approaches such as oversampling, undersampling, and synthetic data creation have been developed for machine learning to overcome this problem.

Federated Learning (FL) is a promising privacy-preserving Artificial Intelligence (AI) framework that addresses the challenges presented in traditional machine learning training. In federated learning, class imbalance may also occur, but the previously mentioned approaches in machine learning are not directly applicable. In federated learning, the class distribution is unknown to protect privacy. Several federated learning algorithms have been developed to address this problem.

This thesis aims to implement and compare three federating learning algorithms designed to address the class imbalance problem: Combinatorial Upper Confidence Bounds (CUCB), CLass IMBalance Federated Learning (CLIMB), and Federated Feature Distillation (FedFed). Three different data distributions were tested: label imbalance, quantitative imbalance, and double imbalance. To provide common ground for algorithm comparison, the implementation considers the same dataset and data pre-processing, the same neural network model, and hype-parameter training. After implementation, the results showed that CUCB had the best convergence rate, which is due to the algorithm inferring the data distribution from the test dataset. CLIMB addresses the local and global mismatch imbalance type, making the algorithm more robust and exhibiting the best performance in all data distributions. The FedFed does not perform as anticipated, despite utilizing the latest advancements in generative AI. Further exploration needs to be done in this implementation, where a complex environment is tested, such as increasing the number of clients.

Análsis Comparativo De Algoritmos Que Abordan El Problema De Datos Desbalanceados En El Aprendizaje Federado

Erika Susana Durán González

Resumen

Tradicionalmente, el entrenamiento de algoritmos de aprendizaje automático requiere de la transferencia de datos recopilados desde diversos dispositivos a un servidor central, lo cual puede generar riesgos de seguridad y privacidad de datos. Otro aspecto importante en el aprendizaje automático es el desbalanceo de clases, que se produce cuando algunas clases están sub-representadas, lo cual puede provocar un rendimiento deficiente, especialmente para los datos de la clase minoritaria. Se han desarrollado diferentes enfoques en aprendizaje automático para solucionar este problema, como el sobremuestreo, el submuestreo o la creación de datos sintéticos.

El aprendizaje federado proporciona un marco de referencia prometedor en el área de IA, que preserva la privacidad y que busca abordar los desafíos que presenta el entrenamiento tradicional en aprendizaje automático. En el aprendizaje federado también puede producirse desequilibrio de clases, pero los enfoques mencionados anteriormente no son directamente aplicables. En el aprendizaje federado, se desconoce la distribución de clases para proteger la privacidad. Se han desarrollado algunos algoritmos de aprendizaje federado para abordar este problema.

Esta tesis trata sobre la implementación y comparación de tres algoritmos de aprendizaje federado diferentes, diseñados para abordar el problema del desequilibrio de clases: CUCB, CLIMB y FedFed. También se probaron tres distribuciones de datos diferentes: desbalance de etiquetas, desbalance cuantitativo y doble desbalance. Para proporcionar una base común para la comparación de algoritmos, la implementación considera el mismo conjunto de datos y preprocesamiento de datos; el mismo modelo de red neuronal y el mismo entrenamiento con parámetros de hiperactividad.

Tras la implementación, los resultados mostraron que CUCB obtuvo la mejor tasa de convergencia, debido a que el algoritmo infiere la distribución de datos a partir del conjunto de datos de prueba. CLIMB aborda el desequilibrio de desajuste local y global, lo que aumenta la robustez del algoritmo y muestra el mejor rendimiento en todas las distribuciones de datos. FedFed no tuvo el rendimiento esperado, a pesar de que utiliza las técnicas más recientes de IA generativa. Es necesario profundizar en esta implementación, donde se prueba un entorno complejo, como el aumento del número de clientes.

Contents

List of Figures

List of Tables

Page

Acronyms

AdaBoost	adaptive boosting. 57
AFL	Asynchronous Federated Learning. 53
AI	Artificial Intelligence. 5, 34, 47
AIoD	AI-on-Demand. 55
AWS	Amazon Web Services. 55
BCFL	Blockchain-based Federated Learning. 52
CADx	Computer-Aided Diagnosis System. 36
CAGR	Compound Annual Growth Rate. 54
CLIMB	CLass IMBalance Federated Learning. 5, 7, 61, 63, 68, 75–81
CMAB	Combinatorial Multi-Armed Bandit. 60
CNN	Convolutional Neural Network. 36, 37, 44, 48, 68–70, 73
CPU	Central Processing Unit. 27, 47, 48, 53
CSL	cost-sensitive learning. 29
CUCB	Combinatorial Upper Confidence Bounds. 5, 7, 59–61, 68, 75–82
CUDA	Compute Unified Device Architecture. 47, 48, 73
DBN	Deep Belief Network. 36
DIHS	Digital Innovation Hubs. 55
DNN	Deep Neural Network. 35
EEG	electroencephalogram. 36
FedABC	Federated Averaging via Binary Classification. 58
FedAvg	Federated Averaging. 47, 63, 75–80
FedFed	Federated Feature Distillation. 5, 7, 64, 65, 68, 75–80, 82

FL	Federated Learning. 5
FRL	Federated Reinforcement Learning. 57
GDDR6	Graphics Double Data Rate 6. 48, 73
GDPR	General Data Protection Regulation. 27
GPGPU	General-Purpose GPU (Graphics Processing Unit). 47
GPU	Graphics Processing Units. 47, 48
HDP-FL	Hybrid Differential Privacy with Federated Learning. 53
HE	Homomorphic Encryption. 52, 53
HIPAA	Health Insurance Portability and Accountability Act. 27
IMU	Inertial Measurement Unit. 78
IoMT	Internet of Medical Things. 27, 57
IoT	Internet of Things. 28, 49–52, 55
IoUT	Internet of Underwater Things. 50
k-NN	k-Nearest Neighbors. 35
LSFL	Lightweight and Secure Federated Learning. 53
LSTM	Long Short-Term Memory. 36, 37
MCC	Matthews Correlation Coefficient. 78
MELLODDY	Machine Learning Ledger Orchestration for Drug Discovery. 54, 55
ML	Machine Learning. 5
MLP	Multi Layer Perceptron. 37
NN	Neural Network. 35, 43, 44
non-IID	Non Independent and Identically Distributed. 28–30, 42, 51, 52, 59, 60, 65
pFedCSPC	personalized federated learning via cross silo prototypical calibration. 58
px	protected features. 64
RAG	Retrieval-Augmented Generation. 79
ReLU	Rectified Linear Unit. 69
RL	Reinforcement Learning. 57, 58
RNN	Recurrent Neural Networks. 36

SDAE	Stacked Denoising Autoencoder. 36
SFL	Secure Federated Learning. 53
SME	Small and Medium-sized Enterprise. 55, 56
SVM	Support Vector Machine. 35, 36
xr	performance-robust features. 64
xs	performance-sensitive features. 64

Dedicated to:

To my beloved parents, for their unconditional love, endless support, and the values they instilled in me that guided me through every step of this journey.

To my husband, For his unwavering encouragement, constant support, and enduring belief in my potential, which has uplifted me every step of the way.

This work is dedicated to all of you, with deep love and appreciation.

Dedicado a:

A mis queridos padres, por su amor incondicional, su apoyo infinito y los valores que me inculcaron y que me han guiado a través de este viaje.

A mi esposo, por su aliento inquebrantable, su constante apoyo y su perseverante fe en mi potencial, que me ha inspirado en cada paso del camino.

Este trabajo está dedicado a todos ustedes, con profundo amor y aprecio.

Aknowledgments

I would like to express my sincere gratitude to my thesis advisor, Dr. Gema Berenice Gudiño Mendoza, for her invaluable guidance, continuous support, and encouragement throughout the development of this research. Her expertise and insightful feedback was essential to the completion of this work.

With heartfelt gratitude to the ITESO and SCIyT that generously provided me with a scholarship, enabling me to pursue this academic path and turn a dream into reality.

Reconocimientos

Deseo expresar mi más sincero agradecimiento a mi directora de tesis, Dr. Gema Berenice Gudiño Mendoza, por su valiosa orientación, apoyo continuo y motivación a lo largo del desarrollo de esta investigación. Su experiencia y comentarios constructivos fueron fundamentales para la realización de este trabajo.

Con sincera gratitud al ITESO y al SCIyT, que generosamente me otorgaron una beca, permitiéndome estudiar este posgrado y convertir un sueño en realidad.

1 Introduction

Contents

Typical centralized machine learning approaches, where data is collected and uploaded to a central server for model training, raise significant security and privacy concerns. This traditional method exposes sensitive information to potential breaches and unauthorized access, compromising user privacy [1]. The centralization of data creates a single point of failure, making it an attractive target for cyberattacks. This approach also conflicts with data protection regulations like Health Insurance Portability and Accountability Act (HIPAA) and General Data Protection Regulation (GDPR), especially in healthcare settings where patient confidentiality is paramount [2]. Moreover, the collection and storage of large amounts of personal data, such as location information, browsing history, and text messages, on central servers increases the risk of data breaches and misuse [3].

To address these issues, alternative approaches like federated learning and local differential privacy have emerged. Federated learning keeps users' data on their own devices, training models locally and only sharing model parameters, thus preserving privacy.

Local differential privacy, on the other hand, involves perturbing data locally before submission to the server [4]. These methods offer a better balance between privacy protection and model performance, with federated learning generally performing better at the cost of higher client Central Processing Unit (CPU) usage [5].

The federated learning decentralized approach also reduces network resource utilization, making it particularly suitable for resource-constrained environments like smart grids and Internet of Medical

[1] S. Singhal. Data privacy, compliance, and security including ai ml. In *IGI Global*, pages 111–126. 2024. DOI: 10.4018/979-8-3693-2909-2.ch009

[2] S. Singhal. Data privacy, compliance, and security including ai ml. In *IGI Global*, pages 111–126. 2024. DOI: 10.4018/979-8-3693-2909-2.ch009

[3] H. Zheng, Z. Han, and H. Hu. Preserving user privacy for machine learning: Local differential privacy or federated machine learning? *IEEE Intelligent Systems*, 35(4):5–14, Jul 2020. DOI: 10.1109/mis.2020.3010335

[4] H. Zheng, Z. Han, and H. Hu. Preserving user privacy for machine learning: Local differential privacy or federated machine learning? *IEEE Intelligent Systems*, 35(4):5–14, Jul 2020. DOI: 10.1109/mis.2020.3010335

[5] G. Drainakis, A. Amditis, P. Pantazopoulos, V. Sourlas, and K. V. Katsaros. Federated vs. centralized machine learning under privacy-elastic users: A comparative analysis. In *Proceedings of the 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, pages 1–8, Nov 2020. DOI: 10.1109/nca51143.2020.9306745

Things (IoMT) [6].

Despite the privacy benefits of federated learning, researchers have identified various vulnerabilities that adversaries can exploit to compromise the trained model or extract private information. These include poisoning attacks, backdoor attacks, and inference-based attacks [7]. Federated learning faces several challenges beyond security risks: Communication efficiency is a significant concern in federated learning. The process involves frequent exchanges of model updates between clients and the central server, which can consume considerable communication resources [8]. Data heterogeneity poses another challenge. In federated learning, data is distributed across different clients. This data may vary significantly in terms of quality, quantity, and distribution. This heterogeneous nature of data can lead to model performance discrepancies across different clients and affect overall model fairness [9].

This research deals with the data heterogeneity challenge in federated learning environments.

[6] M. Aljanabi. Safeguarding connected health: Leveraging trustworthy ai techniques to harden intrusion detection systems against data poisoning threats in iomt environments. *Babylonian Journal of Internet of Things*, 2023:31–37, May 2023. DOI: 10.58496/bjiot/2023/005

[7] N. Bouacida and P. Mohapatra. Vulnerabilities in federated learning. *IEEE Access*, 9:63229–63249, Jan 2021. DOI: 10.1109/access.2021.3075203

[8] M. Asad, A. Moustafa, and T. Ito. Fedopt: Towards communication efficiency and privacy preservation in federated learning. *Applied Sciences*, 10(8):2864, Apr 2020. DOI: 10.3390/app10082864

[9] F. Liu, M. Li, J. Ren, T. Xue, X. Liu, and C. Zhang. A review of federated meta-learning and its application in cyberspace security. *Electronics*, 12(15):3295, Jul 2023. DOI: 10.3390/electronics12153295

1.1 Background

Federated learning is a privacy-preserving framework that facilitates the training of models across decentralized devices without necessitating the exchange of raw data.

The term Non Independent and Identically Distributed (non-IID) refers to data distributions that are not uniform or consistent across different sources or clients in a distributed learning environment. In recommendation systems, non-IID characteristics of users and items can lead to irrelevant, duplicate, or uninteresting recommendations[10].

Some issues arising from non-IID data include:

- The distribution of data classes among different clients is not uniform or consistent
- Insufficient utilization of hierarchical information
- Optimization inconsistency and feature divergence
- Negative impacts on some clients' personal model performance [11].

Real world examples of non-IID scenarios include:

- In Internet of Things (IoT) applications, where mobile clients have diverse data distributions [12]
- In wireless networks, where high dynamics of wireless circumstances and user behavior lead to non-IID data collection [13]
- In image classification tasks, where datasets exhibit highly non-IID characteristics [14].

[10] Longbing Cao. Non-iid recommender systems: A review and framework of recommendation paradigm shifting. *Engineering*, 2(2):212–224, June 2016. DOI: 10.1016/j.eng.2016.02.013

[11] Liang Zhang, Bo Du, Ling-Yu Duan, Yihang Luo, and Yihui Bai. Federated learning for non-iid data via unified feature learning and optimization objective alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4400–4408, October 2021b

[12] Weiqiang Zhang, Peng Zhou, Xiang Zhang, Wenbo Wu, and Xin Wang. Client selection for federated learning with non-iid data in mobile edge computing. *IEEE Access*, 9:24462–24474, January 2021c

[13] Zhengming Zhao et al. Federated learning with non-iid data in wireless networks. *IEEE Transactions on Wireless Communications*, 21(3):1927–1942, March 2022. DOI: 10.1109/twc.2021.3108197

[14] Xiaofeng Shang, Yiu ming Cheung, Hao Wang, and Yuxuan Lu. Fedic: Federated learning on non-iid and long-tailed data via calibrated distillation. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, July 2022.

The issue of class imbalance, characterized by the under-representation of certain classes within the training data, can substantially affect model performance. In the context of federated learning, this challenge is exacerbated by the heterogeneous nature of client data distributions.

Traditional methods for addressing class imbalance, such as oversampling or under sampling, are not directly applicable in federated learning due to the decentralized nature of the data. These methods typically require access to the entire dataset, which contradicts federated learning's privacy-preserving principle.

Researches have suggested a variety of innovative techniques to tackle class imbalance in federated learning.

For example, some federated learning approaches attempt to infer the composition of training data for each round to address imbalance, but this can potentially leak information about the local data distribution[15].

Other techniques such as *personalization* consist of adapting the global model to better suit individual clients' data distributions and requirements.

The global model may not perform optimally for all clients due to non-IID data across participants[16]. Personalization aims to improve model accuracy and performance for each client by tailoring the global model to their specific needs.

This can be achieved through various approaches; a popular one is mixing global and local models [17].

Another method to deal with class imbalance in federated learning is through algorithms. Algorithmic approaches to class imbalance, such as cost-sensitive learning (CSL), can be more effective in addressing the problem directly at the model level. Although they may require more complex implementations and careful consideration of potential overfitting. Algorithms can incorporate dimensionality reduction techniques, which offer significant advantages in reducing communication overhead and improving computational efficiency. These approaches are particularly valuable in distributed learning platforms and resource-constrained environments like wireless sensor networks[18]. Federated learning algorithms designed to handle non-IID data offer several advantages over traditional methods like oversampling or undersampling, because algorithms can simultaneously address global imbalance across the entire federation and local imbalance within individual clients. They achieve this by using optimization techniques and intelligent client selection strategies. These approaches often result in faster convergence, improved accuracy, and reduced communication overhead compared to conventional federated learning methods, making them particularly

[15] L. Wang, S. Xu, X. Wang, and Q. Zhu. Addressing class imbalance in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10165–10173, May 2021. DOI: 10.1609/aaai.v35i11.17219

[16] Xiao Chen Li, Baoyuan Li, Yiqing Shao, Shuaiqiang Song, and De-Chuan Zhan. Fedphp: Federated personalization with inherited private models. In *Advances in Knowledge Discovery and Data Mining (PAKDD 2021)*, Lecture Notes in Computer Science, pages 587–602. Springer, 2021. DOI: 10.1007/978-3-030-86486-6$_{36}$

[17] F. Yu et al. Communication efficient personalized federated meta learning in edge networks. *IEEE Transactions on Network and Service Management*, 20(2):1558–1571, 2023b. DOI: 10.1109/tnsm.2023.3263831

[18] M. A. Attia and R. Tandon. On the worst-case communication overhead for distributed data shuffling. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 961–968, Sep. 2016. DOI: 10.1109/allerton.2016.7852338

suitable for real-world federated learning scenarios with heterogeneous data distributions.

1.2 Justification

Addressing class imbalance in federated learning is crucial for improving model performance and generalization in real-world applications. Since algorithmic approach is highly efficient, comparing different algorithms under various data distribution scenarios will provide insights into their effectiveness and guide future research in this field.

1.3 Problem Statement

The literature on algorithms to deal with class imbalance in federated learning, appears to lack a comprehensive and systematic comparison of multiple algorithms across diverse distribution scenarios. Instead, researchers typically focus on presenting a new algorithm they have developed, comparing it selectively against existing algorithms under conditions that are most advantageous for their proposed solution. This approach is often limited to a narrow range of distribution scenarios, rather than exploring a broad spectrum of possible conditions. This prevalent methodology has some limitations. It may lead to an overly optimistic assessment of the proposed algorithm's performance, lack generalization to real-world scenarios that differ from the tested conditions, fail to provide a comprehensive view of how different algorithms perform across various distribution scenarios, and make it difficult for users to choose the most appropriate algorithm for their specific needs due to the lack of comparative data. To address these limitations, a more robust approach would involve evaluating several algorithms simultaneously across a wide range of distribution scenarios. This would ensure all algorithms are tested under the same conditions for fair comparison. Such a systematic approach would provide a more thorough understanding of algorithm performance and applicability in various contexts.

1.4 Hypothesis

Unbalanced federated learning algorithms can be tested in various data distribution scenarios to achieve multiple objectives. These include evaluating model performance in diverse environments, assessing algorithmic robustness, improving fairness, accelerating convergence rates, and identifying optimal aggregation strategies for non-IID.

1.5 General Objective

Conduct a quantitative and qualitative assessment and comparison of federated learning algorithms' performance in a case study involving various types of class imbalance.

1.6 Specific Objectives

1. Set up a baseline and the same conditions to guarantee fairness during the comparative between the selected algorithms. These conditions include utilizing the same dataset, controlling randomness by setting an identical random seed for all experiments, adjusting hyperparameters such as learning rate and number of iterations, applying uniform preprocessing to input data across all models, employing the same performance metric for evaluation, and using the same model.
2. Implement and test the three selected algorithms in a Federated Learning framework.
3. Compare the results across different data distribution types to identify the strengths and weaknesses of each algorithm.
4. Provide recommendations for algorithm selection based on specific data distribution characteristics in federated learning environments.

1.7 Summary

The Introduction chapter sets the stage for the study on federated learning algorithms in class imbalance scenarios by introducing the background, justification, problem statement, hypothesis, and objectives. The chapter emphasizes the following key points: Traditional centralized machine learning approaches pose security and privacy concerns, prompting the development of federated learning as a privacy-preserving framework. The study tackles the issue of class imbalance in federated learning, which can significantly impact model performance. The research also seeks to quantitatively and qualitatively evaluate and compare the performance of federated learning algorithms across various class imbalance types. Specific objectives include establishing a fair baseline for comparison, implementing and testing selected algorithms, comparing results across different data distribution types, and offering recommendations for algorithm selection.

2 Theoretical Framework

Contents

This chapter delves into significant theories and models relevant to federated learning in scenarios with class imbalance. Initially, it outlines the fundamentals of machine learning. The next part discusses standard performance metrics used in machine learning. Following this, the chapter examines imbalanced data distributions and the mathematical theories that facilitate the simulation of a controlled imbalance within datasets. The subsequent section elucidates the functioning of a convolutional neural network and its effectiveness in addressing classification problems within the realm of federated learning. The next section explains the concept of federated learning and the operation of the aggregation model. Lastly, the chapter describes the hardware architecture that underpins machine learning experiments.

2.1 Machine Learning and Deep learning Overview

2.1.1 Machine Learning

As Raschka et al.[1] describe, in this age of modern technology, there is a large amount of structured and unstructured data. In the second

[1] Vahid Mirjalili Sebastian Raschka. *Python Machine Learning*. Packt, third edition, 2019. ISBN 9781789955750

half of the 20th century, machine learning evolved as a subfield of AI involving self-learning algorithms that derive knowledge from data in order to make predictions. Instead of requiring humans to manually derive rules and build models from analyzing large amounts of data, machine learning offers a more efficient alternative for capturing the knowledge in data to gradually improve the performance of predictive models and make data-driven decisions.

Commonly used terms in machine learning include:

- Feature (x): A column in a data table or data matrix. Synonymous with predictor, variable, input, attribute or co-variate.
- Target (y): Synonymous with outcome, output, response, variable, dependent variable, class, label, and ground truth.
- Loss function: learning objective, which has to be a function that accepts two arguments — the network's output and the desired output. Its responsibility is to return to us a single number — how close the network's prediction is from the desired result. This function is called the loss function, and its output is the loss value.[2]

[2] Maxim Lapan. *Deep Reinforcement Learning Hands-On*. Packt, third edition, Nov 2024

[3] Yuxi (Hayden) Liu. *Python Machine Learning By Example*. Packt, fourth edition, July 2024. ISBN 9781835085622

Depending on the nature of the learning data, Liu[3] classifies machine learning tasks into the following three categories:

Unsupervised learning: When the learning data only contains indicative signals without any description attached (we call this unlabeled data), it's up to us to find the structure of the data underneath, discover hidden information, or determine how to describe the data. Unsupervised learning can be used to detect anomalies, such as fraud or defective equipment, or group customers with similar online behaviors for a marketing campaign. Data visualization that makes data more digestible, as well as dimensionality reduction that distills relevant information from noisy data, are also in the family of unsupervised learning.

Supervised learning: When learning data comes with a description, targets, or desired output besides indicative signals (we call this labeled data), the learning goal is to find a general rule that maps input to output. The learned rule is then used to label new data with unknown output. The labels are usually provided by event-logging systems or evaluated by human experts. Also, if feasible, they may be produced by human raters, through crowd-sourcing, for instance. Supervised learning is commonly used in daily applications, such as face and speech recognition, product or movie recommendations, sales forecasting, and spam email detection.

Reinforcement learning: Learning data provides feedback so that a system adapts to dynamic conditions in order to ultimately achieve a certain goal. The system evaluates its performance based

on the feedback responses and reacts accordingly. The best-known instances include robotics for industrial automation, self-driving cars, and the chess master AlphaGo. The key difference between reinforcement learning and supervised learning is the interaction with the environment.

In supervised learning, there are two main types of problems[4]: classification and regression. Classification problems involve categorizing data into predefined classes or labels, such as "fraud" or "non-fraud" and "spam" or "non-spam." On the other hand, regression problems aim to predict a continuous variable, such as the price of a house.

Machine learning algorithms have gained significant popularity for classification problems across various domains. Support Vector Machine (SVM) and k-Nearest Neighbors (k-NN) are two of the most frequently used algorithms for classification tasks [5].

These methods have demonstrated high accuracy and versatility in handling diverse datasets. In addition to SVM and k-NN, other popular machine learning algorithms for classification include Decision Trees, Random Forests, Logistic Regression, and Neural Networks [6]. Each of these algorithms has its strengths and is suited for different types of classification problems.

For instance, Random Forests have shown excellent performance in diabetes prediction, achieving 80% accuracy [7], while Gradient Boosting achieves the highest overall accuracy of 88.80% in mental health problem classification [8].

Interestingly, some studies have found that ensemble techniques, which combine multiple algorithms, can outperform individual classifiers. For example, a fusion of neural networks and oblique decision trees demonstrated superior performance compared to other state-of-the-art classifiers in multi-class datasets [9].

2.1.2 *Deep Learning*

Chin[10] describes deep learning as a subset of the larger category of machine learning. The main characteristic that separates it from other machine learning algorithms is the foundational building block called Deep Neural Network (DNN) – that is, Neural Network (NN) with multiple hidden layers. As deep learning has advanced tremendously since the early 2000s, it has made many previously unachievable feats possible through its machine learning counterparts. Deep learning has made breakthroughs in recognizing complex patterns that exist in complex, non-linear, and unstructured data such as text, images, videos, and audio.

Deep learning algorithms have become increasingly popular for

[4] Dr. Mounir Abdelaziz Kumar Abhishek. *Machine Learning for Imbalanced Data*. Packt, first edition, Nov 2023. ISBN 9781801070881

[5] Beyza Akbugday. Classification of breast cancer data using machine learning algorithms. In *2019 Medical Technologies Congress (TIPTEKNO)*, pages 1–4, October 2019. DOI: 10.1109/tiptekno.2019.8895222

[6] Jihye Chung and Jason Teo. Single classifier vs. ensemble machine learning approaches for mental health prediction. *Brain Informatics*, 10(1), January 2023. DOI: 10.1186/s40708-022-00180-6

[7] Ayhan Göde and Abdullah Kalkan. Performance comparison machine learning algorithms in diabetes disease prediction. *European Mechanical Science*, 7(3):178–183, September 2023. DOI: 10.26701/ems.1335503

[8] Jihye Chung and Jason Teo. Single classifier vs. ensemble machine learning approaches for mental health prediction. *Brain Informatics*, 10(1), January 2023. DOI: 10.1186/s40708-022-00180-6

[9] Rudra Katuwal and Ponnuthurai Nagaratnam Suganthan. Enhancing multi-class classification of random forest using random vector functional neural network and oblique decision surfaces. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2018. DOI: 10.1109/ijcnn.2018.8489738

[10] Ee Kin Chin. *The Deep Learning Architect's Handbook*. Packt, first edition, 2023. ISBN 9781803235349

classification tasks across various domains. Several algorithms have shown remarkable performance in different classification scenarios:

Convolutional Neural Network (CNN) have demonstrated exceptional results in image classification tasks. They have been successfully applied to lung cancer diagnosis using computed tomography scan images[11], malware classification by converting binaries to grayscale images[12], and hyperspectral image classification [13]. CNN have also shown promise in speech emotion recognition [14] and electroencephalogram (EEG) decoding for brain-computer interfaces [15].

Recurrent Neural Networks (RNN), particularly those with Long Short-Term Memory (LSTM) architecture, have proven effective for sequential data processing tasks. They have been used in speech emotion recognition, EEG decoding, and speech recognition when combined with CNNs [16].

Other popular deep learning algorithms include Deep Belief Network (DBN), Stacked Denoising Autoencoder (SDAE), and various hybrid approaches.

For instance, [17] compares the performance of CNN, DBN, and SDAE in lung nodule classification, while [18] proposes combining deep neural networks with SVM for improved scalability and performance.

Some studies have found that deep learning algorithms can outperform traditional machine learning approaches. For example, [19] reports that DBN and CNN achieved higher accuracy than a traditional Computer-Aided Diagnosis System (CADx) system using SVM.

2.1.3 *Machine Learning Life Cycle*

Machine learning life cycle plays a crucial role in centralized environments, encompassing various stages from data collection to model deployment and maintenance.

ModelHub, for example, is an approach to address life cycle management issues by incorporating a model versioning system, a domain-specific language for model space exploration, and a hosted service[20]. This integrated approach streamlines the process of building, training, and managing machine learning/ deep learning models, enhancing efficiency and reproducibility.

The machine learning life cycle establishes a framework that supports the thorough management of artifacts and tasks related to model development.

This is important because distributed learning models, such as federated learning, which build upon traditional centralized machine learning, adopt many elements of the machine learning life cycle while also presenting their own distinct challenges and considerations.

[11] Wenhao Sun, Bin Zheng, and Wei Qian. Computer aided lung cancer diagnosis with deep learning algorithms. In *Medical Imaging 2016: Computer-Aided Diagnosis*, volume 9785, page 97850Z, March 2016

[12] Mohamed Kalash, Fahad Iqbal, Neil D. B. Bruce, Noman Mohammed, Yang Wang, and Morteza Rochan. Malware classification with deep convolutional neural networks. In *2018 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5, February 2018

[13] Bing Liu, Xiaoyuan Yu, Anlin Yu, and Guodong Wan. Deep convolutional recurrent neural network with transfer learning for hyperspectral image classification. *Journal of Applied Remote Sensing*, 12(02): 026028, June 2018

[14] Wonjoon Lim, Taesu Lee, and Dongsuk Jang. Speech emotion recognition using convolutional and recurrent neural networks. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, December 2016

[15] Joseph Thomas, Justin Dauwels, Nitesh Sinha, Thomas Kluge, and Tomasz Maszczyk. Deep learning-based classification for brain-computer interfaces. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 234–239, October 2017

[16] George Saon and Michael Picheny. Recent advances in conversational speech recognition using convolutional and recurrent neural networks. *IBM Journal of Research and Development*, 61(4/5):1:1–1:10, July 2017

[17] Wenhao Sun, Bin Zheng, and Wei Qian. Computer aided lung cancer diagnosis with deep learning algorithms. In *Medical Imaging 2016: Computer-Aided Diagnosis*, volume 9785, page 97850Z, March 2016

[18] David Díaz-Vico, Anas Omari, José Dorronsoro, and Javier Prada. Deep support vector neural networks. *Integrated Computer-Aided Engineering*, 27(4): 389–402, September 2020

[19] Wenhao Sun, Bin Zheng, and Wei Qian. Computer aided lung cancer diagnosis with deep learning algorithms. In *Medical Imaging 2016: Computer-Aided Diagnosis*, volume 9785, page 97850Z, March 2016

[20] Hao Miao, Amol Deshpande, Larry S. Davis, and Abhishek Li. Modelhub: Deep learning lifecycle management. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 1393–1394, April 2017

[21] describes the traditional machine learning life cycle in Figure 2.1. The same life cycle is also applicable for deep learning.

[21] Ee Kin Chin. *The Deep Learning Architect's Handbook*. Packt, first edition, 2023. ISBN 9781803235349

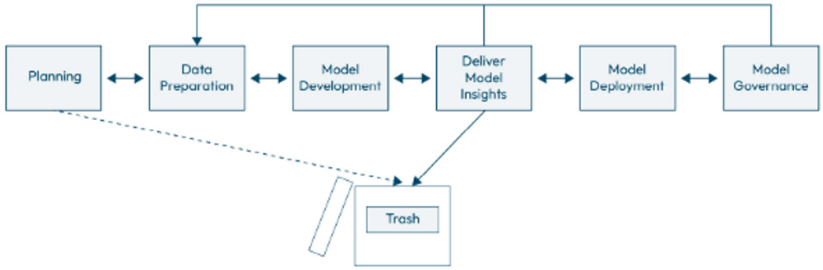


Figure 2.1: Machine learning life cycle Chin, 2023, 1

The six stages consist on the following actions:

1. **Planning**
2. **Data Preparation**
3. **Model Development**
4. **Deliver Model Insights**
5. **Model Deployment**
6. **Model Governance**

Planning involves evaluating machine learning or deep learning worthiness without over complicate problems just to apply advanced algorithms. This stage also includes defining success according to the business goals and planning resources.

Data preparation includes acquisition of raw input and targeted output data, exploratory data analysis of the acquired data and data-preprocessing.

Model development is about the implementation of a machine learning model, for example, a neural network. If the net has three or more neural network layers, then is called a deep learning model. Deep learning models are architectures capable of learning patterns from the data through a loss function and an optimizer algorithm that optimizes the loss function. A loss function defines the error made by the model so that its memory can be updated to perform better in the next iteration. Some deep learning architectures are the Multi Layer Perceptron (MLP), the CNN, Autoencoders, LSTM and Transformers.

Regarding **delivering model insights**, when people care about the decisions that can potentially be made by the machine learning model, they typically require more information to put their trust in the ability of the model to make decisions. Building trust in a model involves

ensuring accurate, reliable and unbiased predictions that align with the domain expertise and business objectives, while providing stakeholders with insight the model's performance metrics and rational behind on its predictions. The process of inducing trust in a model continues when the model is deployed.

Model deployment in the machine learning life cycle refers to the process of integrating a trained machine learning model into a production environment where it can be used to make predictions or decisions on new data. This stage involves transitioning the model from development to operational use, ensuring it performs reliably and efficiently in real-world scenarios[22].

Model governance, on the other hand, encompasses the practices and processes used to manage, monitor, and maintain machine learning models throughout their life cycle, including after deployment. It involves ensuring model quality, performance, and reliability over time, as well as addressing issues such as data drift, concept drift, and bias[23].

While model deployment focuses on the technical aspects of putting a model into production, model governance extends beyond deployment to include ongoing monitoring and maintenance.

[22] Diego Nigenda, Sridhar Alla, Niyati Mehta, Dipayan Banerjee, Shenghui Cheng, Shiva Mandala, Senthil Nathan, Amogh Tiwari, Alexandra Theresia, Ramesh Subramonian, Sunil Mallya, and Subbarao Kambhampati. Amazon sagemaker model monitor: A system for real-time insights into deployed machine learning models. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*, volume 104, pages 3671–3681, August 2022. DOI: 10.1145/3534678.3539145

[23] Diego Nigenda, Sridhar Alla, Niyati Mehta, Dipayan Banerjee, Shenghui Cheng, Shiva Mandala, Senthil Nathan, Amogh Tiwari, Alexandra Theresia, Ramesh Subramonian, Sunil Mallya, and Subbarao Kambhampati. Amazon sagemaker model monitor: A system for real-time insights into deployed machine learning models. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*, volume 104, pages 3671–3681, August 2022. DOI: 10.1145/3534678.3539145

2.2 *Performance and metrics*

Performance metrics are critical for evaluating the effectiveness of machine learning and deep learning models. Several machine learning and deep learning metrics are used to evaluate the performance of classification models.

Given a model that tries to classify an example as belonging to the positive or negative class, there are four possibilities. These are summarized in the **confusion matrix** [24]:

True Positive (TP): This occurs when the model correctly predicts a sample as part of the positive class, which is its actual classification.

False Negative (FN): This happens when the model incorrectly classifies a sample from the positive class as belonging to the negative class.

True Negative (TN): This refers to instances where the model correctly identifies a sample as part of the negative class, which is its actual classification.

False Positive (FP): This occurs when the model incorrectly predicts a sample from the negative class as belonging to the positive class.

[24] Dr. Mounir Abdelaziz Kumar Abhishek. *Machine Learning for Imbalanced Data*. Packt, first edition, Nov 2023. ISBN 9781801070881

Key classification metrics quantify model performance by comparing predicted versus actual labels. These formulas are derived from the components of the confusion matrix.[25]

[25] Shervine Amidi. Machine learning tips and tricks cheatsheet, 2021. URL https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks. Accessed: March 30, 2025

Accuracy

Definition: Overall correctness of predictions across all classes.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

Limitation: Misleading for imbalanced datasets.

Precision

Definition: Accuracy of positive predictions.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.2}$$

Use case: Prioritizing false positive reduction (e.g., spam detection).

Recall (Sensitivity)

Definition: Coverage of actual positives.

$$\text{Recall (Sensitivity)} = \frac{TP}{TP + FN} \tag{2.3}$$

Use case: Critical for medical diagnoses where missing positives is costly.

Specificity

Definition: Coverage of actual negatives.

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{2.4}$$

As Grandini et al.[26] explains, in despite that the accuracy is one of the most popular metrics in multi-class classification, it is not the best metric for imbalanced datasets. This is because the accuracy is the probability that the model prediction is correct.

The accuracy returns an overall measure of how much the model is correctly predicting on the entire set of data. The basic element of the metric are the single individuals in the dataset: each unit has the same weight and they contribute equally to the accuracy value. When we think about classes instead of individuals, there will be classes with a high number of units and others with just few ones. In this situation, highly populated classes will have higher weight compared to the smallest ones, and the accuracy might rate high, even if the minority classes are misclassified.[27]

[26] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview, 2020. URL https://arxiv.org/abs/2008.05756

[27] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview, 2020. URL https://arxiv.org/abs/2008.05756

Balance accuracy

The Balance accuracy is a best-suited metric for imbalanced datasets.

Balanced Accuracy Balanced Accuracy is the average of recalls for each class, giving equal weight to all classes regardless of their size.

$$\text{Balanced Accuracy} = \frac{\sum_{k=1}^{K} \frac{\text{TP}_k}{\text{Total}_{\text{row}_k}}}{K} \tag{2.5}$$

[28] provides an example with calculations based on confusion matrix from Figure 2.2

[28] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview, 2020. URL https://arxiv.org/abs/2008.05756

		PREDICTED classification				
	Classes	a	b	c	d	Total
ACTUAL classification	a	6	0	1	2	9
	b	3	9	1	1	14
	c	1	0	10	2	13
	d	1	2	1	12	16
	Total	11	11	13	17	**52**

Figure 2.2: Multi-class confusion matrix example Grandini et al., 2020, 1

$$\text{Accuracy} = \frac{6+9+10+12}{52} = 0.7115 \tag{2.6}$$

$$\text{Balanced Accuracy} = \frac{\frac{6}{9}+\frac{9}{14}+\frac{10}{13}+\frac{12}{16}}{4} = 0.7072 \tag{2.7}$$

As noticed in the example, overall accuracy is higher than balanced accuracy. This is due to the class imbalance.

While overall accuracy gives more weight to classes with more samples, such as class *d*; balanced accuracy treats each class equally, so the lower recall of classes like *a* and *b*, which are misclassified more often, pulls the average down.

2.3 *Data distribution effects in machine learning*

Abhishek[29] introduces the effects of imbalanced datasets on machine learning. Machine learning algorithms learn from collections of examples that we call datasets. These datasets contain multiple data samples or points, which we may refer to as samples or instances.

[29] Dr. Mounir Abdelaziz Kumar Abhishek. *Machine Learning for Imbalanced Data*. Packt, first edition, Nov 2023. ISBN 9781801070881

A dataset can be said to have a balanced distribution when all the target classes have a similar number of examples.

In general, imbalanced data sets or skewed datasets are those that have some target classes that outnumber the rest of the classes. For instance, refer to Figure 2.3.

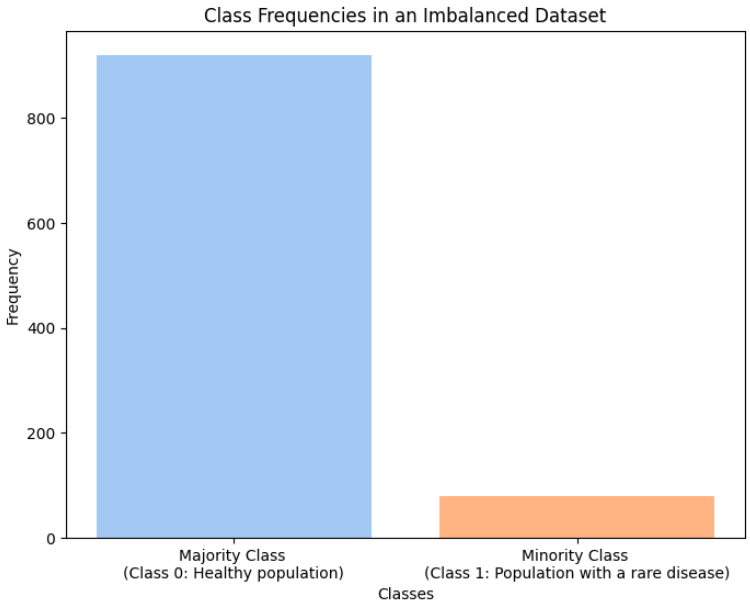


Figure 2.3: Representation of binary class frequencies in an imbalanced dataset

The imbalance scenarios can be categorized as depicted in Figure 2.4 by [30].

Label imbalance: The majority of clients have part labels of a whole, and the client's labels are mostly different from others, while the quantity of each label in the client is the same. For example, client 1 owns labels 4, 7, 9, but client 2 owns labels 0, 2, 8.[31] Refer to Figure 2.4 (a).

Label imbalance distributions are prevalent in various real-world applications. One notable example is natural disaster prediction, such as flood forecasting, where positive instances (actual floods) are rare compared to the abundance of negative instances (non-flood events). "Near-miss" situations—events where floods nearly occurred—offer additional opportunities for enhancing predictive accuracy [32].

Quantity imbalance: Each client owns an entire set of labels, but the quantity of each label in the client varies, that is, client 1 has 10 labels and the number of label 9 is approximately 450, but the number of labels 0 and 1 is approximately 0[33]. See Figure 2.4 (b).

Quantity imbalance in supply chains is a common issue that can occur in various real-world scenarios. For example, the COVID-19 pandemic has highlighted significant quantity imbalances in the pharmaceutical industry. For instance, during the early stages of the pandemic, there was a severe shortage of personal protective equipment and medical supplies, while manufacturers struggled to ramp up production to meet the sudden surge in demand [34]. This imbalance led

[30] Shuo Guo et al. Fedgr: Federated learning with gravitation regulation for double imbalance distribution. In *Database Systems for Advanced Applications. DASFAA 2023*, volume 13943 of *Lecture Notes in Computer Science*. Springer, Cham, 2023

[31] Shuo Guo et al. Fedgr: Federated learning with gravitation regulation for double imbalance distribution. In *Database Systems for Advanced Applications. DASFAA 2023*, volume 13943 of *Lecture Notes in Computer Science*. Springer, Cham, 2023

[32] A. Tanimoto, S. Yamada, T. Takenouchi, M. Sugiyama, and H. Kashima. Improving imbalanced classification using near-miss instances. *Expert Systems with Applications*, 201:117130, Apr. 2022. DOI: 10.1016/j.eswa.2022.117130

[33] Shuo Guo et al. Fedgr: Federated learning with gravitation regulation for double imbalance distribution. In *Database Systems for Advanced Applications. DASFAA 2023*, volume 13943 of *Lecture Notes in Computer Science*. Springer, Cham, 2023

[34] M. Riaz, M. Riaz, Z. Zararsiz, and N. Jamil. Distance and similarity measures for bipolar fuzzy soft sets with application to pharmaceutical logistics and supply chain management. *Journal of Intelligent & Fuzzy Systems*, 42(4):3169–3188, Mar. 2022. DOI: 10.3233/jifs-210873

to supply chain disruptions and challenges in healthcare logistics.

Double imbalance: Each client owns a partition of all labels, and the quantity of each class in the client varies. e.g., client 1 possesses labels 4,7,9, and each class's sample number is imbalanced[35]. As shown in Figure 2.4 (c).

[35] Shuo Guo et al. Fedgr: Federated learning with gravitation regulation for double imbalance distribution. In *Database Systems for Advanced Applications. DASFAA 2023*, volume 13943 of *Lecture Notes in Computer Science*. Springer, Cham, 2023

Real-world examples where both quantity and label imbalance occur can be found in medical domains. For instance, in cancer screening, the number of non-cancerous cases significantly outweighs the number of cancerous cases. This creates both a quantity imbalance in the dataset and a label imbalance in the classification task.

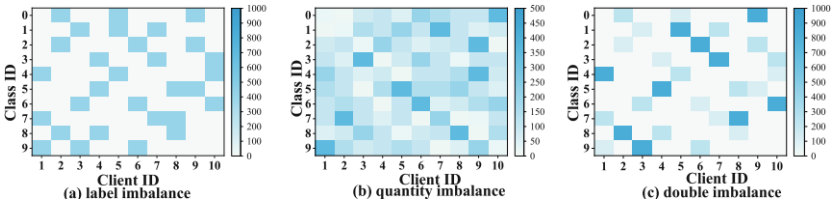


Figure 2.4: Different imbalance distribution scenarios on a dataset with 10 classes and 10 clients Guo et al., 2023, 1

Classical machine learning solves problems with models seen in Section 2.1.1 and predicts the target value. These models typically work with tabular data. By carefully engineering features, classical machine learning models can better handle class imbalance, outliers, or other issues; leading to improved performance and more reliable predictions across all classes.

Susarla[36] defines feature engineering as the process of transforming data into features that better represent the underlying problem, resulting in improved machine learning performance.

[36] Susarla Sinan Ozdemir. *Feature Engineering Made Easy*. Packt, first edition, Jan 2018. ISBN 9781787286474

Feature engineering techniques for tabular data typically include missing data imputation, categorical encoding, variable transformation, discretization, scaling, normalization, dimensionality reduction, and handling outliers.

2.4 *Dirichlet*

The Dirichlet distribution is an important multivariate continuous probability distribution that generalizes the Beta distribution to higher dimensions. It is defined over a probability simplex, making it useful for modeling proportions or probabilities that sum to 1.[37]

[37] Jiayu Lin. On the dirichlet distribution. Master's thesis, Queen's University, Kingston, Ontario, Canada, September 2016. A report submitted to the Department of Mathematics and Statistics in conformity with the requirements for the degree of Master of Science

The Dirichlet distribution is commonly used for simulating non-IID data in machine learning experiments, as it enables control over the variability and imbalance in class distributions across different subsets of data.

Formally, a random vector $Y_k = [Y_1, \ldots, Y_k]$ is said to have a Dirichlet distribution with parameters $\alpha_k = [\alpha_1, \alpha_2, \ldots, \alpha_k]$ if its probability density function is:

$$f(y_k) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} y_i^{\alpha_i - 1} \tag{2.8}$$

where $\alpha_0 = \sum_{i=1}^{k} \alpha_i$, $y_i > 0$, $\sum_{i=1}^{k} y_i = 1$, and $\alpha_i > 0$ for all i.

The Dirichlet distribution has several important properties that make it useful for simulating imbalanced classes:

1. The mean of each component is given by $E[Y_i] = \frac{\alpha_i}{\alpha_0}$
2. The variance of each component is $VAR(Y_i) = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)}$
3. The covariance between components is $COV(Y_i, Y_j) = \frac{-\alpha_i \alpha_j}{\alpha_0^2(\alpha_0 + 1)}$ for $i \neq j$

To simulate imbalanced classes using the Dirichlet distribution, one can adjust the α_i parameters to control the expected proportions and variability of each class. Larger values of α_i relative to the other parameters will result in a higher expected proportion for that class, while smaller values will lead to lower proportions.

An example is shown in Figure 2.5. To simulate a 10-class problem with highly imbalanced classes, one might use parameters $\alpha_3 = (0.1, 0.1, 0.1)$. While using parameters $\alpha_3 = (5, 5, 5)$ would result in balanced class proportions.

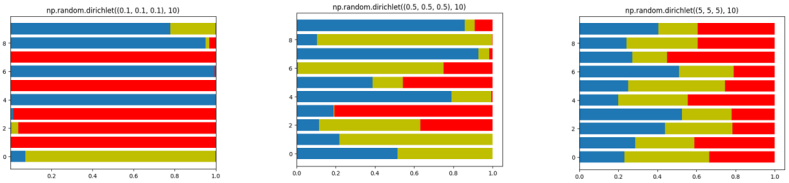


Figure 2.5: Dirichlet distribution to produce imbalance with different values for alpha parameter.

The Dirichlet distribution's flexibility in modeling proportions makes it a valuable tool for generating synthetic datasets with controlled class imbalances, allowing researchers to test and validate machine learning algorithms under various imbalanced scenarios.

2.5 *Convolutional Neural Network*

A **Convolutional neural network (CNN)** is a kind of neural network (NN) with several types of special layers [38]. NN are machine learning

[38] Ivan Vasilev. *Python Deep Learning*. Packt, third edition, Nov 2023. ISBN 9781835085622

models inspired by the structure and function of the human brain [39]. They consist of interconnected processing elements called neurons or nodes, working together to solve specific problems such as pattern recognition, data classification, and prediction [40]. NNs are composed of layers of neurons, including input, hidden, and output layers [41]. They learn from examples through a process of adjusting the connections (synapses) between neurons [42].

CNN are a specialized type of deep learning model ideal for processing structured grid-like data, such as images. They operate by applying convolutional filters across input data to automatically extract hierarchical features, which are then used for tasks like image classification. CNN have demonstrated state-of-the-art performance in a range of computer vision applications, including object recognition, image processing, and even disease diagnosis in crops [43].

CNN are particularly well-suited for image classification tasks within Federated Learning environments for several key reasons. First, they excel at learning complex patterns from image data without the need for manual feature engineering, reducing the effort and expertise required for preprocessing [44]. Second, CNN are inherently parallelizable, enabling efficient distributed training across multiple devices or edge nodes. Lastly, their architecture can be effectively adapted to work with limited or decentralized datasets, which is critical in federated learning scenarios where data privacy, heterogeneity, and communication constraints are prominent challenges [45].

2.6 *Federated learning architecture and model aggregation*

Generating big amounts of data, enables the use of machine learning and deep learning algorithms, but also raises important challenges on how the data is handled, such as data privacy, and data security.

Nakayama [46] defines data privacy, as the right of individuals to control how their personal information is used, which mandates third parties to handle, process, store and use such information properly in accordance with the law. Data security, refers to ensure that data is accurate, reliable, and accessible only to authorized users. The concepts of data privacy and security can be likened to a window representing security and a curtain symbolizing privacy, illustrating how these two elements together form the foundation of data protection.

The privacy compliance has been a bottleneck for enterprises that want to leverage the power of big data to train models, these enterprises might prefer to limit their usage of data, to avoid risks of being fined for violating data protection regulations.

For AI solution providers, that are struggling with public concerns and data privacy, it becomes an issue when a third-party entity such as

[39] M. Kim et al. Deep learning in medical imaging. *Neurospine*, 17(2):471–472, Jun 2020. DOI: 10.14245/ns.1938396.198.c1

[40] R. Yogitha and G. Mathivanan. Performance analysis of transfer functions in an artificial neural network. In *2018 International Conference on Communication and Signal Processing (ICCSP)*, volume 15, pages 393–397, Apr 2018. DOI: 10.1109/iccsp.2018.8524387

[41] W. Salah Alaloul and A. Hannan Qureshi. *Data Processing Using Artificial Neural Networks*. IntechOpen, 2020. DOI: 10.5772/intechopen.91935

[42] S. Maind and P. Wankar. Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(1):96–100, Jan 2014. DOI: 10.17762/ijritcc.v2i1.2920

[43] K. M. N. K. Khalif, A. Gegov, N. A. Shahrul, A. S. A. Bakar, and W. Chaw Seng. Integrated generative adversarial networks and deep convolutional neural networks for image data classification: A case study for covid-19. *Information*, 15 (1):58, 2024. DOI: 10.3390/info15010058

[44] C. Cao, Y. Zhang, H. Lu, C. Lan, Y. Zhang, and W. Zeng. Skeleton-based action recognition with gated convolutional neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(11):3247–3257, 2019. DOI: 10.1109/tcsvt.2018.2879913

[45] T. M. Antico, L. F. R. Moreira, and R. Moreira. Evaluating the potential of federated learning for maize leaf disease prediction. Conference Paper, 2022

[46] George Jeno Kiyoshi Nakayama, PhD. *Federated Learning with Python*. Packt, first edition, 2022. ISBN 9781803248752

Google, Amazon AWS or Microsoft, needs access to the private data for improving the performance of machine learning models and their applications. Federated learning can help, as it is a framework for privacy-preserving machine learning.[47]

[47] George Jeno Kiyoshi Nakayama, PhD. *Federated Learning with Python*. Packt, first edition, 2022. ISBN 9781803248752

In the typical process of machine learning, data is gathered and centralized on a server, where it undergoes preparation and the model is subsequently trained. Federated learning, as an extension of conventional machine learning methods, retains many elements of the machine learning life cycle, but its distributed nature presents new challenges.

In a federated learning system, the machine learning training is performed directly at the location of the data. The central server collects no data, but only the resulting trained models. Then aggregation algorithms are used to produce a global model by combining all the collected models. The global model is sent back to the data locations for further training iterations. The two frameworks are compared in Figure 2.6 [48]

[48] George Jeno Kiyoshi Nakayama, PhD. *Federated Learning with Python*. Packt, first edition, 2022. ISBN 9781803248752

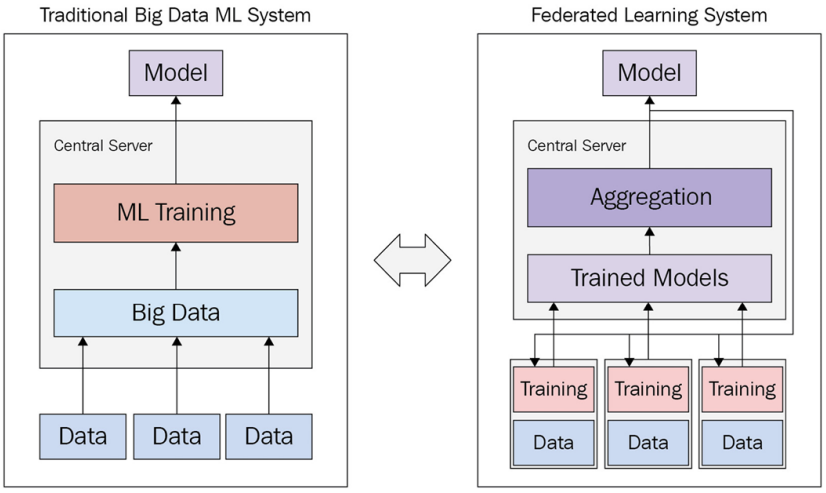


Figure 2.6: Traditional Machine Learning System versus Federated Learning System Kiyoshi Nakayama, 2022, 1

The federated learning systems are distributed systems where clients do not have to send private raw data to the server, specially if owned by third party, they only have to send locally trained models to the central server.

The building blocks of the most standard foundation of federated learning systems are:

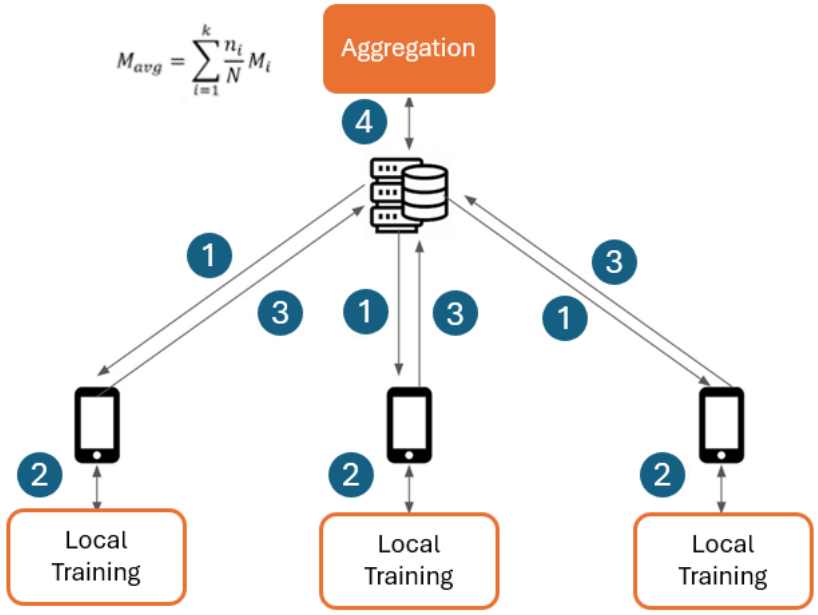
- An aggregator with a federated learning server functions as a program that gathers and combines machine learning models trained by various distributed agents. It then develops global machine learning models, which are returned to these agents. Typically, the

aggregation process involves calculating the average.

- An agent with a federated learning client: A distributed learning environment with a federated learning client such as a local edge device, mobile application, tablet, or any distributed cloud environment where machine learning models are trained in a distributed manner.

- A database: A database and its server to store the data related to the aggregators, agents, and global and local machine learning models and their performance metrics. The database server handles the incoming queries from the aggregators and sends the necessary data back to the aggregators.

The simplest aggregation model is Federated averaging. Its goal is to produce generalized models helpful for all clients collaborating in the federated learning. Nakayama lustrates the sequence in Figure[49] 2.7:



[49] George Jeno Kiyoshi Nakayama, PhD. *Federated Learning with Python*. Packt, first edition, 2022. ISBN 9781803248752

Figure 2.7: Federated Learning process Kiyoshi Nakayama, 2022, 1

1. The aggregate global model parameters are sent to each user's device.

2. The received machine learning models located on the user devices are trained with local data.

3. After a certain amount of training, the local model parameters are sent to the central server.

4. The central server aggregates the local models by applying the Federated Averaging (FedAvg) aggregation function, using a simple arithmetic average of the model parameters [50], producing a new aggregate global model. The aggregate global model parameters are sent to each user's device.

[50] J. Xiao, Z. Duan, C. Du, and W. Guo. A novel server-side aggregation strategy for federated learning in non-iid situations. In *2021 International Symposium on Parallel and Distributed Computing (ISPDC)*, July 2021. DOI: 10.1109/ispdc52870.2021.9521631

2.7 Computing requirements

Graphics Processing Units (GPU) have become a crucial requirement for machine learning due to their ability to significantly accelerate the training and inference processes of neural networks. Initially developed for gaming and 3-D rendering, GPU were recognized as an excellent fit for deep learning tasks due to their parallel processing capabilities [51].

GPU offer massive parallel execution resources and high memory bandwidth, often achieving better performance than CPU for compute-intensive applications like deep learning [52].

The GPU, besides being able to render graphics for video games, also provide a readily accessible means for the general consumer to do massively parallel computing. The applications are many: cryptocurrency miners use GPU to generate digital money such as Bitcoins, geneticists and biologists use GPU for DNA analysis and research, physicists and mathematicians use GPU for large-scale simulations, AI researchers can now program GPU to write plays and compose music, while major internet companies, such as Google and Facebook, use farms of servers with GPUs for large-scale machine learning tasks.[53]

Compute Unified Device Architecture (CUDA), is a framework for General-Purpose GPU (Graphics Processing Unit) (GPGPU) programming from NVIDIA, which was first released back in 2007. It is a mature and stable platform that is relatively easy to use, provides an unmatched set of first-party accelerated mathematical and AI-related libraries, and comes with the minimal hassle when it comes to installation and integration. Moreover, there are readily available and standardized Python libraries, such as PyCUDA and Scikit-CUDA, which make GPGPU programming accessible.

The power of the GPU derives from the fact that there are many, many more cores than in a CPU, which means a huge step forward in throughput. Throughput here refers to the number of computations that can be performed simultaneously. Tuomanen[54] also provides an analogy: A GPU is like a very wide city road that is designed to handle many slower-moving cars at once (high throughput, high latency), whereas a CPU is like a narrow highway that can only admit a few cars at once, but can get each individual car to its destination much quicker (low throughput, low latency).

[51] W. Haensch, R. Puri, and T. Gokmen. The next generation of deep learning hardware: Analog computing. *Proceedings of the IEEE*, 107(1):108–122, January 2019. DOI: 10.1109/jproc.2018.2871057

[52] H. Kimm, H. Kimm, and I. Paik. Performance comparison of tpu, gpu, cpu on google colaboratory over distributed deep learning. In *2021 15th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)*, volume 119, pages 312–319, 2021. DOI: 10.1109/mcsoc51149.2021.00053

[53] Tuomanen. *Hands-On GPU Programming with Python and CUDA*. Packt, first edition, Nov 2018. ISBN 978-1788993913

[54] Tuomanen. *Hands-On GPU Programming with Python and CUDA*. Packt, first edition, Nov 2018. ISBN 978-1788993913

To put it in perspective, the average Intel or AMD CPU has only two to eight cores, while the NVIDIA Tesla T4 GPU available in Google Colab has the following specifications[55]:

- CUDA Cores: 2560
- Tensor Cores: 320
- Memory: 16 GB Graphics Double Data Rate 6 (GDDR6)
- FP32 Performance: 8.1 TFLOPS
- FP16 Performance: 65 TFLOPS

[55] NVIDIA Corporation. Nvidia t4 tensor core gpu, 2018. URL https://www.nvidia.com/en-us/data-center/tesla-t4/. Accessed: 2025-04-02

2.8 *Summary*

This chapter establishes the theoretical foundation for understanding key concepts in federated learning and class imbalance. It begins by introducing core principles of machine learning and deep learning, covering both supervised and unsupervised learning paradigms. The chapter then explores evaluation metrics, with a focus on those suited for imbalanced data, such as balanced accuracy. It discusses the effects of different types of class imbalance — label, quantity, and double imbalance — on model performance. The structure and functionality of CNN are also examined, highlighting their effectiveness in image classification. Additionally, the chapter presents the architecture of federated learning systems, explaining the collaborative roles of clients and the central server during model training. It concludes with an overview of the computational demands of machine learning, emphasizing the importance of GPUs for efficient processing.

3 State of the Art in Federated Learning

Contents

Since its inception by Google in 2016, federated learning has undergone substantial development, establishing itself as a foundational paradigm for privacy-preserving and distributed machine learning.

The core principle of federated learning —enabling model training across decentralized data sources without exposing raw data— has attracted considerable academic and industrial attention. As a result, federated learning has been increasingly adopted in domains such as healthcare, finance, mobile computing, and IoT applications.

However, one of the key challenges that persists in real-world deployments is class imbalance across client datasets. This phenomenon, where the distribution of labels varies significantly among clients, can adversely affect both the convergence and performance of the global model.

This section presents a detailed overview of the current state of the art on federated learning under class imbalance. It begins with a chronological survey of major milestones in federated learning research from 2016 to 2024, followed by an examination of industry adoption trends. Subsequently, it categorizes the different forms of class imbalance observed in federated settings and reviews existing approaches aimed at improving model robustness and accuracy in such scenarios. The section concludes with an in-depth analysis of three prominent algorithms specifically designed to address class imbalance in federated learning, highlighting their underlying methodologies and imbalance conditions where it performs best.

3.1 *Federated Learning evolution*

Federated learning has evolved significantly since its inception, with key developments shaping its trajectory. The concept was first introduced by Google in 2016 as a way to train machine learning models on distributed datasets without compromising user privacy [1]. This marked the beginning of a new paradigm in distributed machine learning. As the field progressed, researchers began exploring various applications, particularly in medical image analysis, where privacy concerns are paramount [2].

The application of federated learning to the Internet of Underwater Things (IoUT) during 2018–2019 marked a significant advancement in addressing the challenges of data privacy and security in underwater environments.

Federated learning was introduced into IoUT systems to enhance reliability, efficiency, and timeliness in various mission-critical applications[3]. IoUT systems have gained momentum over the past decade, with applications spanning environmental monitoring, underwater exploration, and defense.

Traditional IoUT systems rely on conventional machine-learning approaches, which face limitations in terms of data privacy and security. The integration of federated learning addressed these concerns by allowing distributed learning without centralizing sensitive data, making it particularly suitable for mission-critical scenarios [4].

In general, federated learning has emerged as a promising solution to address privacy and security challenges in IoT environments. Researchers have proposed innovative strategies for optimizing

[1] E. Hernandez and J. Smith. Federated learning: A privacy-preserving approach to distributed ai. *Journal of Distributed AI Research*, 12(3):101–120, 2024

[2] E. Hernandez and J. Smith. Federated learning: A privacy-preserving approach to distributed ai. *Journal of Distributed AI Research*, 12(3):101–120, 2024

[3] J. Victor and L. Chen. Federated learning for the internet of underwater things. *IEEE Internet of Things Journal*, 9 (5):3500–3512, 2022

[4] J. Victor and L. Chen. Federated learning for the internet of underwater things. *IEEE Internet of Things Journal*, 9 (5):3500–3512, 2022

federated learning in heterogeneous IoT environments. Additionally, personalized federated learning frameworks have been introduced to cope with device, statistical, and model heterogeneity inherent in complex IoT environments [5].

The year 2020 marked a significant turning point in the field of Federated Learning, with increased recognition of security and privacy threats leading to intensified research on vulnerabilities and the development of advanced defense solutions.

Researchers have identified various security vulnerabilities in federated learning architectures, including poisoning and inference attacks [6]. These vulnerabilities could be exploited by adversaries to compromise the trained model or infer private data of participants [7].

The decentralized nature of federated learning, while designed to protect privacy, introduced new challenges in detecting and defending against malicious model updates [8]. contrastingly, although federated learning was initially proposed as a privacy-preserving solution, it became evident that existing federated learning protocols did not always provide sufficient security[9].

This realization led to a surge in research focusing on comprehensive threat identification and classification, as well as the development of robust defense mechanisms[10]

In response to these challenges, researchers began exploring advanced defense solutions. These include the integration of blockchain technology for enhanced security[11], the use of homomorphic encryption and differential privacy to safeguard sensitive information [12], and the application of Trusted Execution Environments [13]. The development of these defense strategies aimed to create a more secure and privacy-preserving learning environment for federated learning applications [14].

In 2021, secure aggregation emerged as a critical component in federated learning, addressing privacy concerns in collaborative model training. Researchers have provided formal definitions and systematically categorized existing solutions to tackle specific challenges in federated settings [15].

This development was crucial, as it allowed multiple parties to compute the sum of their data without disclosing individual inputs and enhancing privacy in applications such as electronic voting and smart grid measurements.

As federated learning matured, focus shifted towards addressing specific challenges such as non-IID data, communication efficiency, and model aggregation techniques [16].

The non-IID nature of data across clients poses a major challenge in

[5] H. Wu and Y. Zhang. Personalized federated learning for heterogeneous iot systems. *ACM Transactions on Internet Technology*, 20(4):1–22, 2020

[6] L. Lyu and H. Yu. Threats to federated learning: A survey. *IEEE Access*, 8:173532–173550, 2020

[7] F. Neto and A. Singh. Inference attacks in federated learning: A survey. *Journal of Information Security and Applications*, 68: 103310, 2023

[8] R. Kalapaaking and M. Nguyen. Blockchain for securing federated learning: A survey. *Future Generation Computer Systems*, 145:303–320, 2024

[9] C. Zhang and W. Xie. Security challenges in federated learning: A survey. *ACM Computing Surveys*, 54(5): 1–36, 2021

[10] H. U. Manzoor, A. Zoha, A. Shabbir, D. Flynn, and A. Chen. A survey of security strategies in federated learning: Defending models, data, and privacy. *Future Internet*, 16(10):374, Oct 2024. DOI: 10.3390/fi16100374

[11] R. Kalapaaking and M. Nguyen. Blockchain for securing federated learning: A survey. *Future Generation Computer Systems*, 145:303–320, 2024

[12] R. Aziz, T. Le Vinh, S. Bouzefrane, and S. Banerjee. Exploring homomorphic encryption and differential privacy techniques towards secure federated learning paradigm. *Future Internet*, 15 (9):310, 2023. DOI: 10.3390/fi15090310

[13] F. Fang and X. Qian. Enhancing federated learning with trusted execution environments. *IEEE Transactions on Dependable and Secure Computing*, 18(4): 1671–1685, 2021

[14] C. Zhang and W. Xie. Security challenges in federated learning: A survey. *ACM Computing Surveys*, 54(5): 1–36, 2021

[15] K. Bonawitz and V. Ivanov. Secure aggregation for federated learning. *Proceedings of the ACM on Privacy Enhancing Technologies*, 2021(1):1–25, 2021

[16] T. Li and A. K. Sahu. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning Research*, 139:4294–4304, 2021

federated learning, as it can significantly affect model convergence and accuracy.

Several studies have demonstrated the negative effects of non-IID data on federated learning performance. For instance, [17] reported a notable decline in performance with increased data heterogeneity in medical imaging applications. Similarly, [18] found that non-IID data bring significant challenges to the learning accuracy of federated learning algorithms. Various strategies have been proposed to address this issue.

Communication efficiency is another critical aspect of federated learning because the traditional approach of sharing model parameters can lead to significant overhead in IoT environments [19]. Model aggregation techniques have also been a focus of research to improve federated learning performance.

The year 2022 saw the development of vertical federated learning, a specialized form of federated learning for distributed features. This approach incorporates Homomorphic Encryption (HE) to enable encrypted operations without decryption, thereby further enhancing privacy protection [20].

However, while HE provides robust security guarantees, it also introduces additional communication and computational overhead. Researchers have proposed straggler-resilient and computation-efficient accelerating systems to reduce these overheads, thereby improving the efficiency of vertical federated learning frameworks without compromising security.

While federated learning offers the benefit of circumventing data silos, it also introduces new issues. Particularly, the role of the aggregator in federated learning raises concerns about potential single points of failure and communication bottlenecks.[21].

This prompted research into a variety of federated learning approaches, such as Chain federated learning, which utilizes blockchain technology to distribute model storage across network nodes [22]. Additionally, the need for model ownership verification led to the development of FedIPR, a scheme for embedding watermarks into federated learning models to protect intellectual property rights [23].

The combination of federated learning and blockchain technology has gained significant attention in recent years. This integration, often referred to as Blockchain-based Federated Learning (BCFL), addresses several challenges faced by traditional federated learning systems [24]. BCFL leverages the decentralized nature of the blockchain to improve federated learning's security, performance, and application scope. For instance, it can enhance the reliability of smart public transportation systems by protecting against poisoning or DDoS attacks [25].

[17] M. J. Sheller and B. Edwards. Federated learning in medical imaging: Enabling privacy-preserving collaboration. *Medical Image Analysis*, 65:101765, 2020

[18] Y. Zhao and M. Li. Impact of data heterogeneity on federated learning. *arXiv preprint arXiv:1806.00582*, 2020

[19] P. Kairouz and B. McMahan. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021

[20] Q. Yang and Y. Liu. Vertical federated learning with encrypted computation. *IEEE Transactions on Knowledge and Data Engineering*, 34(2):321–334, 2022

[21] C. Korkmaz, A. Uysal, A. Masry, O. Ozkasap, H. E. Kocas, and B. Akgun. Chain fl: Decentralized federated machine learning via blockchain. In *2020 International Conference on Blockchain Computing and Applications (BCCA)*, volume abs/1905.6731, pages 140–146, November 2020. DOI: 10.1109/bcca50787.2020.9274451

[22] J. Wang and X. Chen. Chainfl: Blockchain-based decentralized federated learning. *IEEE Transactions on Network and Service Management*, 19(3):2765–2778, 2022

[23] X. Zhang and Y. Shen. Fedipr: Intellectual property protection for federated learning. *Journal of Systems Architecture*, 127:102399, 2022

[24] W. Li and H. Yu. Blockchain-based federated learning: A survey. *IEEE Transactions on Industrial Informatics*, 19 (2):1432–1445, 2023

[25] S. Park and D. Kim. Bcfl for secure smart transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 24(4):3421–3430, 2023

By 2023, the integration of soft computing techniques with federated learning addressed drawbacks such as high communication costs, privacy concerns, and data management issues. For instance, [26] proposed a Lightweight and Secure Federated Learning (LSFL) scheme that combines privacy-preserving features with Byzantine robustness to address the challenges in edge-computing scenarios.

Similarly, [27] introduced FedNIC, leveraging smart network interface cards to offload compute-intensive HE operations, reducing CPU resource usage, and improving overall system security.

The evolution of federated learning has also seen the development of hybrid approaches, such as the combination of Hybrid Differential Privacy with Federated Learning (HDP-FL), which demonstrated significant improvements in model accuracy while maintaining privacy [28].

Asynchronous Federated Learning (AFL) was introduced to solve synchronization issues and improve the training efficiency of federated learning systems. This approach reduces the aggregation latency, allowing for more efficient model updates. However, AFL can face challenges in terms of learning performance stability owing to unreasonably weighted local models. To address this, innovations such as the Dynamic Scaling Factor have been proposed to assign reasonable weights to stable local models.

The combination of federated learning with blockchain and the introduction of asynchronous learning represent significant advancements in addressing federated learning challenges. These innovations have shown promise in improving security, efficiency, and reliability in various applications, from smart transportation to credit card fraud detection [29].

However, more research is needed to fully explore the integration of federated learning with other learning paradigms and the potential of federated neural architecture searches.

As shown in the figure 3.1, the period between 2016 and 2024 witnessed a rapid evolution of federated learning, with researchers and practitioners addressing various challenges and expanding its applications [30].

During this time, federated learning evolved from a basic concept to a comprehensive framework encompassing horizontal federated learning, vertical federated learning, and federated transfer learning [31]. The focus shifted towards addressing data privacy and security concerns, leading to the emergence of Secure Federated Learning (SFL) [32].

The development of federated learning has been accompanied by

[26] F. Tang and J. Liu. Lsfl: A lightweight and secure federated learning scheme for iot. *IEEE Internet of Things Journal*, 10(1): 334–345, 2023

[27] Q. Zhou and B. He. Fednic: Enabling federated learning with smart nics. *ACM Transactions on Computer Systems*, 41(1): 1–27, 2023

[28] J. Kang and Q. Yang. Hdp-fl: Hybrid differential privacy federated learning. *Neurocomputing*, 512:155–168, 2023

[29] J. Chen and Y. Lin. Asynchronous federated learning with dynamic scaling factors. *Information Sciences*, 634:19–34, 2023

[30] M. Aziz and D. Patel. Privacy-preserving machine learning with homomorphic encryption in federated learning. *Journal of Privacy and Confidentiality*, 15(2): 1–18, 2023

[31] Q. Yang, Y. Tong, T. Chen, and Y. Liu. Federated machine learning. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–19, 2019. DOI: 10.1145/3298981

[32] L. Yang, J. Cao, J. Huang, and W. Lin. Personalized federated learning on non-iid data via group-based meta-learning. *ACM Transactions on Knowledge Discovery from Data*, 17(4):1–20, 2023a. DOI: 10.1145/3558005

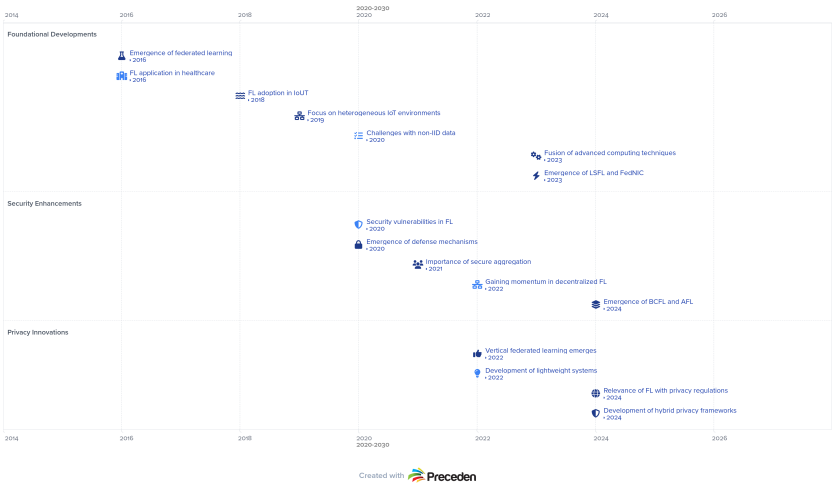


Figure 3.1: Brief evolution of Federated Learning

the integration of various privacy-preserving techniques, such as homomorphic encryption, differential privacy, and secure multi-party computation [33].

As federated learning continued to mature, researchers began exploring its applications in diverse fields, including healthcare, edge computing, communication, and attack detection [34].

The ongoing research in federated learning aims to overcome challenges such as communication costs, system heterogeneity, and unreliable model uploads while maintaining a balance between privacy protection and model performance [35].

[33] M. Aziz and D. Patel. Privacy-preserving machine learning with homomorphic encryption in federated learning. *Journal of Privacy and Confidentiality*, 15(2): 1–18, 2023

[34] J. Qi, Q. Zhou, L. Lei, and K. Zheng. Federated reinforcement learning: techniques, applications, and open challenges. *Intelligence & Robotics*, 2021. DOI: 10.20517/ir.2021.02

[35] S. Bharati, M. R. H. Mondal, V. B. S. Prasath, and P. Podder. Federated learning: Applications, challenges and future directions. *International Journal of Hybrid Intelligent Systems*, 18(1–2):19–35, 2022. DOI: 10.3233/his-220006

3.2 *Industry adoption of Federated Learning*

Federated learning is experiencing significant growth and adoption across various industries, driven by the increasing need for privacy-preserving machine learning techniques. According to market research, the global federated learning solutions market is projected to grow from USD 117 million in 2023 to USD 226 million by 2028, at a Compound Annual Growth Rate (CAGR) of 14.1% during the forecast period [36].

The growth of federated learning is particularly notable in sectors such as healthcare, finance, and smart cities, where data privacy and security are paramount concerns [37].

This technology allows organizations to collaboratively train machine learning models without sharing raw data, addressing the challenges of data islands and privacy regulations [38].

The healthcare and drug discovery industry is leading the federated learning adoption. One representative business case is the Machine

[36] G. et al. Long. Privacy-preserving machine learning: Threats and solutions. *Journal of Computer Science*, 2020; and

[37] B. Gecer and B. Garbinato. Applications of federated learning in urban and smart systems. *Journal of Smart City Research*, 2024; and

[38] A. Alferaidi, G. Dhiman, Y. Alharbi, W. Viriyasitavat, K. Yadav, and S. Kautish. Federated learning algorithms to optimize the client and cost selections. *Mathematical Problems in Engineering*, 2022:1–9, 2022. DOI: 10.1155/2022/8514562

Learning Ledger Orchestration for Drug Discovery (MELLODDY) consortium, comprised of 10 pharmaceutical companies, academic research labs, large industrial companies, and startups. MELLODDY developed a novel platform for applying federated learning to drug discovery [39].

This platform enabled the creation of a global federated model without sharing confidential datasets of individual partners. The MELLODDY project involved an unprecedented cross-pharma dataset of 2.6+ billion confidential experimental activity data points, documenting 21+ million physical small molecules and 40+ thousand assays [40].

The MELLODDY platform was deployed on an Amazon Web Services (AWS) multi-account architecture running Kubernetes clusters in private subnets. This implementation demonstrates AWS's involvement in supporting federated learning solutions. [41]

Federated learning facilitates data sharing and collaboration while preserving privacy, which aligns with the goals of many large tech companies and research institutions [42].

The adoption of federated learning is also being driven by its potential applications in IoT devices and edge computing, where it can enable intelligent decision-making while keeping sensitive data on local devices [43].

Several major technology companies and startups are already participating in the federated learning ecosystem. Major vendors in the global federated learning solutions market include NVIDIA (US), Cloudera (US), IBM (US), Microsoft (US), Google (US), Intel (US), Owkin (US), Intellegens (UK), Edge Delta (US), Enveil (US), Lifebit (UK), DataFleets (US), Secure AI Labs (US), and Sherpa.AI (Spain) [44].

Europe is actively promoting AI adoption for Small and Medium-sized Enterprise (SME)s through various initiatives and projects. The European Union has funded several Horizon 2020 projects, collectively known as the ICT49 cluster, aimed at enhancing the AI-on-Demand (AIoD) platform and supporting businesses in accessing AI expertise, knowledge, algorithms, and tools.

One of these projects is DIH4AI, which focuses on building and connecting regional platforms driven by Digital Innovation Hubs (DIHS) and artificial intelligence to create a pan-European AI toolbox and experimental facility. DIH4AI aims to provide ecosystem-business-technology-transformation services to local SMEs and GovTech companies, addressing the challenges faced by SMEs in the AI landscape.

As part of its technological experiments, DIH4AI has developed PIANAI, a solution that uses federated learning techniques. The

[39] M. Oldenhof et al. Industry-scale orchestrated federated learning for drug discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15576–15584, 2023c. DOI: 10.1609/aaai.v37i13.26847

[40] W. Heyndrickx et al. Melloddy: Cross-pharma federated learning at unprecedented scale unlocks benefits in qsar without compromising proprietary information. *Journal of Chemical Information and Modeling*, 64(7):2331–2344, 2023e. DOI: 10.1021/acs.jcim.3c00799

[41] M. Oldenhof et al. Industry-scale orchestrated federated learning for drug discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15576–15584, 2023c. DOI: 10.1609/aaai.v37i13.26847

[42] Q. et al. Yang. Federated machine learning: Concept and applications. *Nature Machine Intelligence*, 2024

[43] P. et al. Chaves. Federated learning for iot: A privacy-preserving approach. *Sensors*, 2024; and

[44] Aashish Mehra. Federated learning market press release. https://www.marketsandmarkets.com/PressReleases/federated-learning-solutions.asp, 2024. Accessed: 2025-04-12

federated learning approach is particularly valuable for SMEs that may have concerns about data sharing and privacy when adopting AI technologies [45].

[45] O. Markaki et al. Encouraging ai adoption by smes: Opportunities and contributions by the ict49 project cluster. In *2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pages 1–8, 2023d. DOI: 10.1109/IISA59645.2023.10345867

3.3 Class Imbalance in Federated Learning

Besides the label, quantity or double imbalance explored in Section 2.3 that can happen in individual clients, federated learning might face additional types of class imbalance challenges. This is due to its distributed nature:

1. **Local class imbalance** occurs when individual clients have imbalanced data distributions across different classes. This can lead to poor model accuracy on rare but vital classes, especially in health and autonomous driving applications [46]. Local imbalance can also cause unfair competition between classes within each client's dataset [47].

2. **Global class imbalance** arises when the overall distribution of classes across all clients is skewed [48]. This can result in a long-tailed data distribution, where some classes are significantly underrepresented in the federated dataset [49]. Global imbalance can lead to poor model performance on rare classes and inconsistent performance across different clients [50].

3. **Local-global class imbalance inconsistency** is a unique characteristic of federated learning, where the class distribution at individual clients differs from the global distribution [51]. This inconsistency can cause difficulties in detecting and addressing imbalance issues, as neither the participants nor the server have complete visibility of the entire dataset [52]. An example of a local-global mismatch can be seen in a fitness app that monitors user activities. On a global scale, the app may have extensive data for typical activities like walking and running. A professional marathon runner's device aligns well with the global model's abundant data. However, a patient undergoing therapy to recover from an injury has a substantial amount of data on rare movements, leading to a local-global class imbalance. In this scenario, the majority class on the local device becomes the minority class in the global model.

[46] X. Shuai, Z. Yan, Z. Zhao, S. Jiang, G. Xing, and Y. Shen. BalanceFL: Addressing Class Imbalance in Long-Tail Federated Learning. In *Proceedings of the IEEE/ACM IPSN*, 2022. DOI: 10.1109/ipsn54338.2022.00029

[47] D. Wang et al. FedABC: Targeting Fair Competition in Personalized Federated Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10095–10103, 2023. DOI: 10.1609/aaai.v37i8.26203

[48] J. Tian et al. Synergetic focal loss for imbalanced classification in federated xgboost. *IEEE Transactions on Artificial Intelligence*, 5(2):647–660, February 2024a. DOI: 10.1109/tai.2023.3254519

[49] X. Shuai, Z. Yan, Z. Zhao, S. Jiang, G. Xing, and Y. Shen. BalanceFL: Addressing Class Imbalance in Long-Tail Federated Learning. In *Proceedings of the IEEE/ACM IPSN*, 2022. DOI: 10.1109/ipsn54338.2022.00029

[50] X.-X. Wei and H. Huang. Balanced Federated Semisupervised Learning With Fairness-Aware Pseudo-Labeling. *IEEE Transactions on Neural Networks and Learning Systems*, 35(7):9395–9407, 2024. DOI: 10.1109/tnnls.2022.3233093

[51] J. Tian et al. Synergetic focal loss for imbalanced classification in federated xgboost. *IEEE Transactions on Artificial Intelligence*, 5(2):647–660, February 2024a. DOI: 10.1109/tai.2023.3254519

[52] J. Tian et al. Synergetic Focal Loss for Imbalanced Classification in Federated XGBoost. *IEEE Transactions on Artificial Intelligence*, 5(2):647–660, 2024b. DOI: 10.1109/tai.2023.3254519

3.4 Approaches to improve performance under class imbalance federated learning

Researchers have proposed several approaches to mitigate Class imbalance challenges. [53] provides a comprehensive overview of the literature concerning class imbalance estimation methods in Federated Learning, categorized as follows:

[53] Jianzgong Qi Jiayuan He Jing Zhang, Chuanwen Li. A Survey on Class Imbalance in Federated Learning. *Journal of latex class files, VOL. 14, NO. 8*, 2023. https://doi.org/10.48550/arXiv.2303.11673

3.4.1 *Sampling techniques*

- **Data sampling** refers to over-sampling, under-sampling, or hybrid sampling on client-side or server-side data.
- **Client sampling** is the technique of selectively choosing clients to participate in training iterations.
- **Hybrid data and client sampling** combines both approaches.

3.4.2 *Algorithm-centered techniques*

- **Cost-sensitive learning** is specifically designed to address class imbalance problems in classification tasks, it introduces misclassification costs to minimize conditional risk and improve the importance of certain classes during classifier training [54]. This approach pushes decision boundaries away from instances of important classes, leading to improved generalization for those classes.
- **Algorithmic classifier modification** is widely used in various applications, including pattern recognition, intrusion detection, and medical diagnosis.[55] . These classifiers can be modified or optimized to improve their performance and address specific challenges in different domains. One approach to classifier modification is the use of ensemble learning techniques. For example, [56] discusses an adaptive boosting (AdaBoost) classification mechanism that combines a strong machine learning classifier with decision stumps to improve spectrum sensing in cognitive radio networks. This hybrid approach aims to achieve higher detection probability compared to conventional methods. Similarly, in medical contexts, classifier modifications are employed to enhance the resilience of intrusion detection systems against data poisoning threats in IoMT environments[57].
- **Reinforcement Learning** is a framework for decision-making in unknown environments based on trial-and-error interactions. It aims to generate optimal behavior in sequential decision-making environments by learning from the consequences of actions and maximizing cumulative rewards[58].

  Federated Reinforcement Learning (FRL) is an emerging field that combines Federated Learning and Reinforcement Learning (RL) to address privacy concerns while optimizing decision-making processes in distributed systems. Several examples of RL applied in federated learning algorithms have been reported in the literature. In edge computing systems, FRL has been used to handle sequential

[54] A. P. Kalapaaking, I. Khalil, and X. Yi. Blockchain-based federated learning with smpc model verification against poisoning attack for healthcare systems. *IEEE Transactions on Emerging Topics in Computing*, 12(1):269–280, Jan 2024. DOI: 10.1109/tetc.2023.3268186

[55] B. Biggio, G. Fumera, and F. Roli. Multiple classifier systems for robust classifier design in adversarial environments. *International Journal of Machine Learning and Cybernetics*, 1(1–4):27–41, October 2010. DOI: 10.1007/s13042-010-0007-7

[56] J. Zhang, B. Xie, M. Li, D. Zhao, and S. Zeng. A survey on security and privacy threats to federated learning, Oct 2021a

[57] M. Aljanabi. Safeguarding connected health: Leveraging trustworthy ai techniques to harden intrusion detection systems against data poisoning threats in iomt environments. *Babylonian Journal of Internet of Things*, 2023:31–37, May 2023. DOI: 10.58496/bjiot/2023/005

[58] J. Zhang, Q. Xu, F. Wang, J. Zhao, H. Li, and H. Zhu. Security and privacy threats to federated learning: Issues, methods, and challenges. *Security and Communication Networks*, 2022:1–24, Sep 2022. DOI: 10.1155/2022/2886795

decision-making problems [59]. This approach incorporates model-based RL and ensemble knowledge distillation into federated learning, creating an ensemble of dynamics models for clients and training the policy using only the ensemble model.

[59] M. Mansouri, M. Conti, W. Ben Jaballah, and M. Önen. Sok: Secure aggregation based on cryptographic schemes for federated learning. *Proceedings on Privacy Enhancing Technologies*, 2023(1):140–157, Jan 2023. DOI: 10.56553/popets-2023-0009

3.4.3 System-centered techniques

- **Aggregation methods** help mitigate class imbalance by improving the way client models are combined. For instance, the personalized federated learning via cross silo prototypical calibration (pFedCSPC) method [60] employs an adaptive aggregation approach to offer personalized initial models to each client, enabling rapid adaptation to personalized tasks. This method also uses global prototypes to guide local representation learning, which helps mitigate data imbalance problems. Similarly, the multi-arm bandit-based algorithm [61] selects client sets with minimal class imbalance, significantly improving the convergence performance of the global model.

[60] L. Meng et al. Improving global generalization and local personalization for federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, PP(1):76–87, 2025. DOI: 10.1109/tnnls.2024.3417452

[61] M. Yang, H. Zhu, H. Wang, H. Qian, and X. Wang. Federated learning with class imbalance reduction. In *EUSIPCO*, 2021. DOI: 10.23919/eusipco54536.2021.9616052

- **Personalization** techniques are effective in dealing with class imbalance by tailoring models to individual clients' data distributions. The Federated Averaging via Binary Classification (FedABC) framework [62] adopts a *one-vs-all* training strategy in each client, constructing personalized binary classification problems for each class to alleviate unfair competition between classes. Additionally, the Communication-Efficient Personalized Federated Meta-Learning algorithm [63] introduces personalization parameters to improve model accuracy and accelerate convergence.

[62] D. Wang et al. Fedabc: Targeting fair competition in personalized federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10095–10103, 2023a. DOI: 10.1609/aaai.v37i8.26203

[63] F. Yu et al. Communication efficient personalized federated meta learning in edge networks. *IEEE Transactions on Network and Service Management*, 20(2):1558–1571, 2023b. DOI: 10.1109/tnsm.2023.3263831

Meta-learning and system modification methods also contribute to addressing class imbalance.

- **Meta-learning** in federated learning aims to optimize the learning process itself, enabling faster adaptation to new tasks or clients. It focuses on learning how to learn efficiently across different data distributions [64]. For example, Federated Averaging can be interpreted as a meta-learning algorithm that learns a global model that can be quickly personalized for individual clients [65]. Meta-learning approaches can also be used to improve the convergence speed and performance of federated learning in heterogeneous environments [66].

[64] Y. Jiang, K. Rush, J. Konečný, and S. Kannan. Improving federated learning personalization via model agnostic meta learning, 2019. URL https://arxiv.org/abs/1909.12488

[65] Y. Jiang, K. Rush, J. Konečný, and S. Kannan. Improving federated learning personalization via model agnostic meta learning, 2019. URL https://arxiv.org/abs/1909.12488

[66] L. Yang, J. Cao, J. Huang, and W. Lin. Personalized federated learning on non-iid data via group-based meta-learning. *ACM Transactions on Knowledge Discovery from Data*, 17(4):1–20, 2023a. DOI: 10.1145/3558005

- **System modifications** in federated learning involve changes to the underlying architecture, algorithms, or protocols to address specific challenges. These modifications can include adaptive client selection, dynamic sample selection, or the integration of blockchain

technology for improved security [67]. System modifications aim to optimize various aspects of federated learning, such as communication efficiency, privacy preservation, or robustness to non-IID data.

While meta-learning and system modifications are different approaches, they can be complementary. For instance, some research combines meta-learning with other techniques like parallel-ensemble learning [68] or integrates it with blockchain-based frameworks [69] to achieve better performance in federated learning scenarios.

The choice between meta-learning and system modifications depends on the specific challenges and goals of the federated learning application.

The methods for dealing with class imbalance in federated learning scenarios are represented in Figure 3.2

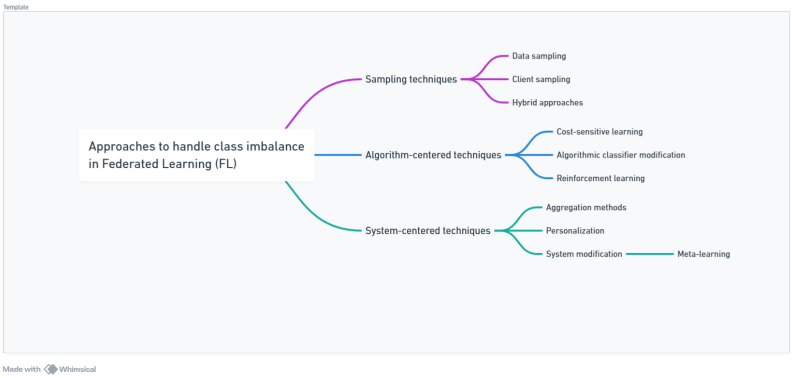


Figure 3.2: Approaches to improve performance under class imbalance federated learning

In the subsequent sections, we will examine three cost-sensitive algorithmic-centered techniques specifically designed to address federated learning in the context of class imbalance. It is rare for an algorithm to be classified into a single category. Recent algorithms often integrate multiple techniques to enhance performance. For instance, system modification for client selection may be combined with a cost-sensitive algorithm during client-side training.

3.5 *Algorithm 1. CUCB (Yang, 2020)*

3.5.1 *Key principles and motivations*

The CUCB[70] is designed to address the client selection problem in

[67] Y. Li, Q. Yan, N. Liu, Z. Zheng, C. Chen, and H. Huang. A blockchain-based decentralized federated learning framework with committee consensus. *IEEE Network*, 35(1):234–241, 2020. DOI: 10.1109/mnet.011.2000263

[68] H. Yu, C. Wu, H. Yu, X. Wei, S. Liu, and Y. Zhang. A federated learning algorithm using parallel-ensemble method on non-iid datasets. *Complex & Intelligent Systems*, 9(6):6891–6903, 2023. DOI: 10.1007/s40747-023-01110-7

[69] W. Sun, B. Guo, L. Xu, T. Q. Duong, Y. Zhao, and W. Ma. Accelerating convergence of federated learning in mec with dynamic community. *IEEE Transactions on Mobile Computing*, pages 1–17, 2023. DOI: 10.1109/tmc.2023.3241770

[70] Miao Yang, Akitanoshou Wong, Hongbin Zhu, Haifeng Wang, and Hua Qian. Federated learning with class imbalance reduction. *CoRR*, abs/2011.11266, 2020. URL https://arxiv.org/abs/2011.11266

federated learning with class imbalance reduction. Key principles include:

- **Privacy preservation**: Address class imbalance in federated learning without accessing raw client data.
- **Handling heterogeneity**: Addresses class imbalance and non-IID data distributions across clients.
- **Principled formulation**: Optimize client selection to minimize class imbalance and improves convergence performance of the global model.

3.5.2 *How the algorithm works*

The functionality of the algorithm is described in below steps and represented in Figure 3.3.

1. The algorithm begins with a class estimation scheme that reveals the class distribution of client datasets without accessing raw data by analyzing the gradients of an auxiliary balanced dataset. The scheme utilizes the relationship between gradient magnitudes and class sample sizes to estimate the class composition vector for each client. Building upon this estimation, the algorithm employs a Combinatorial Multi-Armed Bandit (CMAB) framework to select clients iteratively. Each client is treated as an arm, and the selected set of clients represents a super arm. The reward for each client is calculated based on the Kullback-Leibler divergence [71] between its estimated class distribution and a uniform distribution, with a lower divergence indicating better balance. Then the aggregate global model parameters are initialized with random values and sent to each selected client. The Figure 3.4 shows the algorithm where clients 1 and 3 were selected.
2. The received machine learning models located on the user devices are trained with local data. The core of the algorithm is the Combinatorial Upper Confidence Bounds CUCB approach, which balances exploration and exploitation in client selection. It maintains estimates of individual client rewards and uses these estimates, along with an exploration factor, to make selection decisions. The algorithm artificially increases the reward estimates for less frequently selected clients, promoting exploration of potentially beneficial client combinations.
3. Once the training is completed, the model parameters are sent to the central server.
4. The central server aggregates the local models by applying the FedAvg aggregation function, producing a new aggregate global

[71] The Kullback-Leibler divergence (KL divergence) is a statistical measure that quantifies the difference between two probability distributions, assessing the amount of information lost when one distribution approximates another. It is widely used in various fields, including information theory, statistics, and machine learning, for measuring the fit of two distributions and understanding how well a model represents underlying data.

V. Nawa and S. Nadarajah. Exact expressions for kullback-leibler divergence for univariate distributions. *Entropy (Basel, Switzerland)*, 26(11), 2024. DOI: 10.3390/e26110959; and

model. The aggregate global model parameters are sent to each user's device. In each round, the algorithm selects a set of clients that minimizes overall class imbalance. It does this by iteratively adding clients to the selection set, choosing each subsequent client to minimize the Kullback-Leibler divergence of the combined class distribution. This process continues until the desired number of clients is selected.

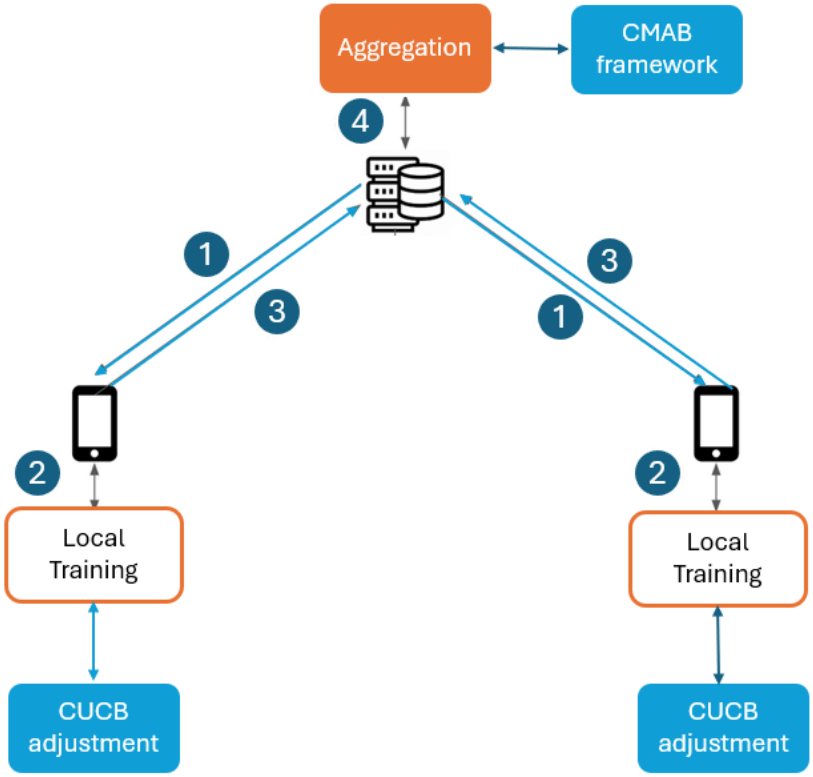


Figure 3.3: CUCB algorithm

3.5.3 *Imbalance Conditions Where It Performs Best*

CUCB method shows improved convergence speed and higher test accuracy compared to random and greedy selection schemes in such imbalanced scenarios.

3.6 *Algorithm 2. CLIMB (Shen et al.,2022)*

3.6.1 *Key principles and motivations*

The CLIMB[72] algorithm is motivated by addressing class imbalance

[72] Zebang Shen, Juan Cervino, Hamed Hassani, and Alejandro Ribeiro. An agnostic approach to federated learning with class imbalance. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Xo0lbDt975

in federated learning settings, particularly when there is significant mismatch between local and global imbalance. Key principles include:

- **Privacy preservation**: The algorithm does not rely on or require knowledge of how data is distributed across different classes, and does not require identifying minority classes.
- **Handling heterogeneity**: It aims to work well when local and global class imbalances differ significantly.
- **Principled formulation**: Uses a mathematical technique to find the optimal solution to a problem, while satisfying certain constraints instead of experience-based techniques that may work well in many cases but lack theoretical guarantees.

3.6.2 *How the algorithm works*

The process is described in the steps below and illustrated in Figure 3.4

1. The aggregate global model parameters initialized with random values and are sent to each user's device.
2. The received machine learning models located on the user devices are trained with local data. The algoritm formulates a constrained federated learning problem to enforce similar loss across clients. The objective function is to minimize the average loss across all clients. The constraint ensures client's performance remains within a certain range closed as possible to the average performance across all clients. The primal-dual approach converts the constrained problem into an unconstrained one using Lagrangian relaxation - Introduces dual variables associated with the similarity constraints. The primal update is the standard model training based on weighted average of client's performance. It modifies the local objective function for each client by incorporating weights. After each primal update, the server computes the loss for each client by calculating the violation of the similarity constraints. The dual update adjust weights balancing the impact of underrepresented classes by increasing dual variables for clients with higher losses. The local objective must be closed to global objective with an epsilon tolerance.
3. The dual variables directly influence the weights in the next primal update. Higher dual variables lead to higher weithgs, emphasizing those client's objectives. Lower dual variables result in lower weithgs, reducing the impact of those clients. The process continues until a stopping criterion is met (e.g., maximum iterations, convergence threshold). As the algoritm progresses, it balances the overall loss

minimization with the enforcement of similar losses across clients. Once the training is completed, the model parameters are sent to the central server.

4. The central server aggregates the local models by applying the FedAvg aggregation function, producing a new aggregate global model. The aggregate global model parameters are sent to each user's device.

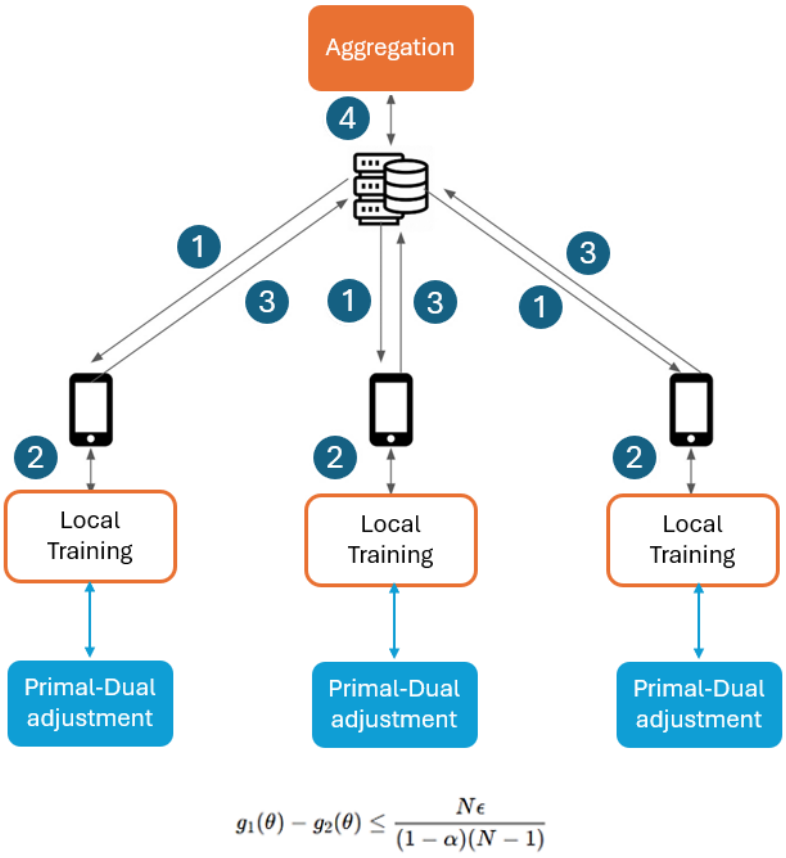


Figure 3.4: CLIMB algorithm

3.6.3 *Imbalance Conditions Where It Performs Best*

The algorithm CLIMB works best for imbalanced cases with:

- Severe heterogeneity across clients
- Significant mismatch between local and global imbalance
- Both moderate (100) and large (500) numbers of clients
- Various imbalance ratios and numbers of minority classes

3.7 Algorithm 3.FedFed (Yang et al. 2023)

3.7.1 Key principles and motivations

The motivation to develop FedFed[73] is addressing data heterogeneity in federated learning by sharing partial features rather than full dataset.

Key principles include:

- **Privacy preservation**: Data is partitioned into performance-sensitive features (greatly contributing to model performance) and performance-robust features (limitedly contributing to model performance). Only performance-sensitive features are shared globally while keeping performance-robust features locally to balance privacy and performance. Differential privacy protects the shared performance-sensitive features.
- **Handling heterogeneity**: The algorithm can deal with highly heterogeneous data across many clients.
- **Principled formulation**: The FedFed algorithm combines ideas from feature distillation[74], information bottleneck[75], and differential privacy[76] to enable sharing of partial features across clients in a privacy-preserving manner.

[73] Zhiqin Yang, Yonggang Zhang, Yu Zheng, Xinmei Tian, Hao Peng, Tongliang Liu, and Bo Han. Fedfed: Feature distillation against data heterogeneity in federated learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b

[74] Feature distillation is a technique that focuses on transferring intermediate-level knowledge from a larger teacher network to a smaller student network.

Minguk Ji, Seungeun Park, and Bohyung Heo. Show, attend and distill: Knowledge distillation via attention-based feature matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7945–7952, May 2021. DOI: 10.1609/aaai.v35i9.16969

[75] Information bottleneck (IB) provides a balance between data compression and information preservation. IB helps in summarizing high-dimensional data into lower-dimensional feature vectors, which is crucial for computationally feasible solutions in statistical learning.

Ziv Goldfeld and Yury Polyanskiy. The information bottleneck problem and its applications in machine learning. *IEEE Journal on Selected Areas in Information Theory*, 1(1):19–38, May 2020. DOI: 10.1109/jsait.2020.2991561

[76] Differential privacy is a framework designed to safeguard sensitive personal data in unstructured formats. It functions by introducing random noise into the data, which helps preserve certain statistical characteristics while ensuring individual privacy is maintained.

Yuzhe Zhao and Jie Chen. A survey on differential privacy for unstructured data content. *ACM Computing Surveys*, 54(10s):1–28, January 2022. DOI: 10.1145/3490237

3.7.2 How the algorithm works

Refer to the steps below in conjunction with Figure 3.5

1. The aggregate global model parameters initialized with random values and are sent to each user's device.
2. The received machine learning models located on the user devices are trained with local data. Each client distills their local private data into performance-sensitive features (xs) and performance-robust features (xr) using a competitive mechanism inspired by the information bottleneck method. This is achieved through a generative model that produces xr, while xs is implicitly defined as the difference between the raw data and xr. Clients apply differential privacy protection to their xs by adding Gaussian noise, creating protected features (px).
3. The protected features are shared with the central server to construct a global dataset.
4. The central server aggregates the local models by applying the FedAvg aggregation function, producing a new aggregate global model. The aggregate global model parameters are sent to each user's device. In the iterative process, next time clients will receive

the globally shared dataset and update their local models using both their private raw data and the shared protected features. This allows the model to learn from a more diverse dataset, mitigating the effects of data heterogeneity.

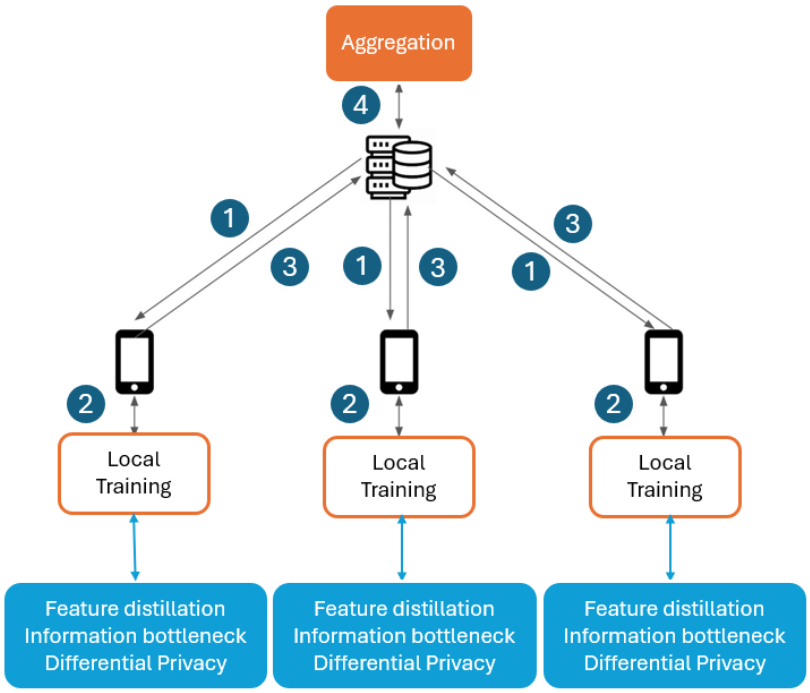


Figure 3.5: FedFed algorithm

3.7.3 *Imbalance Conditions Where It Performs Best*

FedFed is designed to reduce communication costs, making it suitable for scenarios with constrained network resources. FedFed can efficiently manage federated learning with numerous participating clients. It can handle scenarios where clients join or leave the federation during the learning process. The algorithm is robust to clients that are slow to respond or occasionally unavailable. These properties make FedFed suitable for heterogenous data distributions, when client datasets are non-IID, FedFed can effectively handle the data heterogeneity.

3.8 *Summary*

This section provides a comprehensive overview of the current landscape of federated learning, with a focus on the challenges posed by class imbalance. It traces the evolution of federated learning from its inception by Google in 2016 to recent developments as of 2023,

highlighting its growing adoption across sectors such as healthcare and drug discovery. The chapter identifies various forms of class imbalance that commonly emerge in federated settings and reviews state of the art strategies to address them, including sampling-based, algorithm-focused, and system-level approaches. It also presents an in-depth analysis of three specific algorithms — CUCB, CLIMB, and FedFed — detailing their core principles, operational mechanisms, and conditions under which they achieve optimal performance.

4 Methodology

Contents

This chapter presents the methodology employed in the study on federated learning algorithms under different class imbalance scenarios. The discussion begins with an overview of the general approach, followed by detailed descriptions of the dataset, preprocessing steps, types of class imbalance examined, machine learning and deep learning models utilized, training hyperparameters, and system characteristics of the Google Colab environment used for experiments. Through this methodology, the aim is to systematically evaluate federated learning algorithm performance under controlled class imbalance scenarios, providing insights into their robustness and effectiveness for real-world applications with non-uniform data distributions.

4.1 General approach

The approach involves training a fundamental federated learning model for an image classification task, which will serve as a benchmark for comparison with three other algorithms. These additional algorithms are specifically designed to address class imbalance issues. Furthermore, three distinct types of class imbalance have been established to create various federated learning scenarios. The implementation of the four federated learning algorithms (one baseline and three addressing class imbalance) utilizes the same dataset, class imbalance types, machine learning model, and training parameters. Subsequently, the

performance of these federated learning algorithms is quantitatively assessed using the balanced accuracy metric. The balanced accuracy provides a more comprehensive evaluation than the overall accuracy by taking into account the class imbalance present in the dataset.

The implementations were made by copying the repositories of each algorithm, CUCB [1], CLIMB [2] and FedFed [3], then incorporating the following changes:

- Implementation using the same CIFAR-10 data set.
- The addition of a module that generates three types of class imbalances.
- Replace the machine learning model with a CNN model.
- Addition of early stopping.
- Change the training parameters
- Addition of balanced accuracy as measuring metric.

[1] Miao Yang. Federated-Learning-Pytorch, 9 2020. URL https://github.com/ym1231/fl-cir

[2] Zebang Shen. Federated-Learning-Pytorch, 9 2021. URL https://github.com/shenzebang/Federated-Learning-Pytorch

[3] Miao Yang. FedFed: Feature Distillation against Data Heterogeneity in Federated Learning (NeurIPS 2023), 10 2023. URL https://github.com/visitworld123/FedFed

4.2 *Dataset*

The CIFAR-10 dataset [4] consists of 60000, 32 x 32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The size of the dataset is 163 MB. The data set is divided into five training batches and one test batch, each with 10000 images.

[4] Alex Krizhevsky. Learning multiple layers of features from tiny images. https://www.cs.toronto.edu/~kriz/cifar.html, 2009. Technical Report, University of Toronto

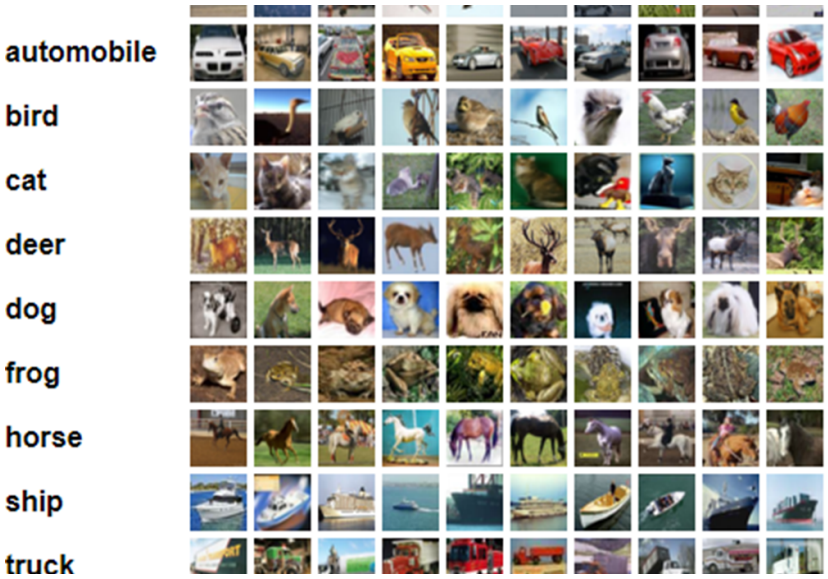


Figure 4.1: CIFAR-10 dataset sample Krizhevsky, 2009, 1

4.3 Machine learning model

The model to train all 4 algorithms is taken from [5]. The CNN architecture consists of two convolutional layers, each with 64 filters of size 5×5, followed by two fully connected layers with 384 and 192 neurons, respectively. Each convolutional layer is followed by a max-pooling operation with a kernel size of 3 and stride 2, which reduces the spatial dimensions of the feature maps and helps to control overfitting by reducing the number of parameters. After the convolutional and pooling layers, the network flattens the output and passes it through the fully connected layers before generating the final class predictions.

The Rectified Linear Unit (ReLU) activation function is applied after each convolution and fully connected layer. ReLU introduces non-linearity into the model, allowing it to learn complex patterns in the data. It is defined as

$$f(x) = \max(0, x)$$

ReLU is widely used in CNNs due to its computational efficiency and effectiveness in preventing the vanishing gradient problem during training.

According to Shen et al., when comparing model architectures, using models with greater capacity can lead to higher testing accuracy on the datasets involved, but this is not the main focus of the research. The primary aim is to ensure fairness in comparisons by training all algorithms with the same model.

[5] Zebang Shen, Juan Cervino, Hamed Hassani, and Alejandro Ribeiro. An agnostic approach to federated learning with class imbalance. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Xo0lbDt975

```
class ConvNet(nn.Module):
    def __init__(self, num_classes=10):
        super(ConvNet, self).__init__()
        self.conv1 = nn.Conv2d(3, 64, kernel_size=5)
        self.conv2 = nn.Conv2d(64, 64, kernel_size=5)
        self.fc1 = nn.Linear(64*4*4, 384)
        self.fc2 = nn.Linear(384, 192)
        self.fc3 = nn.Linear(192, num_classes)
    def forward(self, x):
        x = F.relu(F.max_pool2d(self.conv1(x), 3, stride=2))
        x = F.relu(F.max_pool2d(self.conv2(x), 3, stride=2))
        x = x.view(-1, 64*4*4)
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

4.4 Preprocessing

The preprocessing steps for CNN typically include:

1. Image resizing to standardize input dimensions to match the CNN architecture requirements.

2. Data augmentation, which refers to applying transformations like rotation or flipping, to increase dataset diversity and prevent overfitting.

3. Normalization, which consists on scale pixel values to a consistent range typically between [0,1] or [-1, 1], to improve training stability.

4. Applying techniques to ensure class balance, such as oversampling, undersampling, weighted sampling (during batch creation, assign higher probabilities to samples from minority classes), class weighting (adjust the loss function to give more importance to underrepresented classes) or syntetic data generation to create new samples for minority classes.

The preprocessing steps implemented for the experiments include:

1. **Image resizing**: Not necessary because the pictures on CIFAR-10 dataset are already small and of a consistent size across the complete dataset.

2. **Data augmentation**:

   - Random horizontal flip
   - Random cropping
   - Random rotation
   - Color jitter

3. **Normalization**: Pixel values are normalized to 0.5 of the mean and standard deviation. This normalization technique centers the data around zero by subtracting the mean, scales the data to a consistent range by dividing by the standard deviation, and reduce the range of values by dividing the standard deviation by 2.

4. **Data partitioning** with class imbalance scenario: Typically this section aims to obtain a class balanced dataset. Since the CIFAR-10 dataset is already class balanced, and the thesis is studying the effects of class imbalance, the preprocessing requires a module to create the class imbalance. Three types of class imbalance scenarios at client level have been introduced: quantity imbalance, label imbalance, and double imbalance scenarios.

4.5 Class imbalance samples

Fifty clients participated in the federated learning experiments. Three class imbalance scenarios at client level have been generated: Label imbalance, client imbalance and double imbalance.

The Figure 4.2 shows a sample of the *label imbalance* for the first 10 clients. In label class imbalance, each client has data from only a few classes, and some classes are completely absent.

The Dirichlet distribution can be used to induce a label imbalance when used with very small alpha values, however the Dirichlet distribution was not able to create an extreme label imbalance where each client gets only some classes and none from the others. Hence a function was developed. The label-imbalanced distribution was obtained by assigning each class of the dataset to only a subset of clients rather than all clients. The degree of this imbalance is controlled by the alpha parameter of 0.5

A lower alpha results in each class being distributed to fewer clients, thereby limiting the class diversity within individual clients.

```
Client 1 class distribution: [200   0   0 200 200   0 200 200 200 200]
Client 2 class distribution: [  0 200   0   0 200   0   0   0 200 200]
Client 3 class distribution: [200 200 200 200 200   0 200 200 200 200]
Client 4 class distribution: [200   0   0   0 200 200   0   0 200   0]
Client 5 class distribution: [200 200 200   0 200   0 200   0   0 200]
Client 6 class distribution: [  0 200   0 200 200   0 200   0   0 200]
Client 7 class distribution: [  0 200   0   0   0 200 200 200   0   0]
Client 8 class distribution: [  0 200 200   0 200 200 200   0   0 200]
Client 9 class distribution: [200 200 200   0   0   0 200 200   0 200]
```

Figure 4.2: Sample for the first 10 clients **label imbalance**

The Figure 4.3 shows the *imbalance by quantity* sample of 10 out of 50 clients. Each client has samples from all classes, but the number of samples per class varies.

The quantity imbalance was obtained by applying an imbalance ratio to the same classes. The Dirichlet distribution was not employed to recreate the quantity imbalance, since the imbalance was added on purpose on the first three classes for easier tracking.

```
Client 1 class distribution: [  8   8   8 104  97  94 111 112  88 100]
Client 2 class distribution: [  9  14  12 104 111  86 105 108  84  97]
Client 3 class distribution: [ 10  18  12  99  89  94 108  88 112 100]
Client 4 class distribution: [ 10   9   4  97 109  99 100 108 101  93]
Client 5 class distribution: [  9   9   6  98 103  93  84 109 114 105]
Client 6 class distribution: [  6   9  10  86  93 119 108  93 112  94]
Client 7 class distribution: [  6  16  14  95 100  92  91 102 117  97]
Client 8 class distribution: [  7   8   6  95 104 123  89  92 115  91]
Client 9 class distribution: [ 14   9  13 112 106  84 113 100  95  84]
```

Figure 4.3: Sample for the first 10 clients with **quantity imbalance**

The Figure 4.4 represents a double imbalance (label and quantity) sample of first 10 clients from a total of 50 clients.

In the double imbalance, each client may see different proportions of each class, and the number of total samples per client is different.

The dataset was divided among several clients to mimic a dual imbalance scenario, as follows:

For each dataset class, a Dirichlet distribution with an alpha of 0.5 was used to create a set of proportions that dictate the allocation of that class's data to each client.

These proportions differ for each class and are applied independently, resulting in clients receiving varying amounts of each class, with some classes potentially absent from certain clients.

Given the relatively low alpha value of 0.5, the class distributions are expected to be skewed, with many clients having a concentration of only a few classes, leading to uneven class coverage across clients.

Consequently, some clients will possess significantly more data than others, effectively simulating a realistic scenario of heterogeneous data.

```
Client 1 class distribution: [  6  70  99  17 227  46 257  75  22   9]
Client 2 class distribution: [ 52  38  13   7 143  24  41   2   5   8]
Client 3 class distribution: [  7  60 227 432 322  10 103   8  34   0]
Client 4 class distribution: [202  41  23 214 121  15  25 298 528   2]
Client 5 class distribution: [343   1   0  37  18 144  26  66  88 205]
Client 6 class distribution: [ 16  32 239  25   7  97  49  63   2 203]
Client 7 class distribution: [ 13  46 519   0 160  94 198  33 145 328]
Client 8 class distribution: [ 13  92 501  72 504   7 576 125   0  62]
Client 9 class distribution: [ 70  15  77 199   1 497   4  10  84   0]
```

Figure 4.4: Sample for the first 10 clients **double imbalance**

4.6 *Training hyperparameters*

The machine learning hyperparameters are configuration settings that are not learned from the dataset, but must be set prior to training the model.

The hyperparameters controls how the learning process progress.

The hyperparameters of the federated learning setup for a neural network model are:

- Epochs: 500 The model will iterate 500 times through the entire dataset during training.
- Early stop: patience 10 Early stop is a regularization technique that prevents overfitting. Training will stop if the model's performance doesn't improve for 10 consecutive epochs.
- Learning rate: 0.01 The learning rate controls the step size during the optimization process. 0.01 is a typical starting point.

- Optimizer: Adam. It combines the benefits of adaptive learning rates and momentum, making it well-suited for handling the complex loss landscapes often encountered in image classification tasks. It adapts the learning rate for each parameter individually, allowing for faster convergence on sparse gradients [6]. This adaptive nature helps in dealing with the high dimensionality and variability present in image data. In comparative studies, Adam has shown superior performance in image classification tasks. For instance, it achieved 98.49% accuracy on the MNIST dataset and 75.20% on the CIFAR10 dataset [7].
- Number of clients: 50 In the federated learning scenario, the number of clients represents the number of participating devices or the number of entities. 50 is a moderated scenario.
- Rounds: 500 This refers to the number of federated learning rounds. Each round involves local training on clients and the model aggregation on central server.

[6] Q. Tang, F. Shpilevskiy, and M. Lécuyer. DP-AdamBC: Your DP-Adam Is Actually DP-SGD (Unless You Apply Bias Correction). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15276–15283, March 2024. DOI: 10.1609/aaai.v38i14.29451

[7] H. Sun, Y. Chen, J. Lin, X. Zhou, and Z. Gao. A Linear Interpolation and Curvature-Controlled Gradient Optimization Strategy Based on Adam. *Algorithms*, 17(5):185, April 2024. DOI: 10.3390/a17050185

4.7 System characteristics

All algorithms were trained using the NVIDIA Tesla T4 GPU [8]. This processor is provided as a hardware resource in Google Colab.

As described in section 2.7, it has 2560 CUDA cores and 16 GB GDDR6 of Memory. Moreover, the processor is capable of performing 8.1 trillion 32-bit floating-point operations every second!

[8] NVIDIA Corporation. Nvidia t4 tensor core gpu, 2018. URL https://www.nvidia.com/en-us/data-center/tesla-t4/. Accessed: 2025-04-02

4.8 Summary

The methodology used to evaluate federated learning algorithms under different class imbalance scenarios is thoroughly described in this section. The study centers on training a baseline federated learning model for image classification, which serves as a reference point for comparing three specialized algorithms designed to mitigate class imbalance. The experiments use the CIFAR-10 dataset, consisting of 60,000 color images (32×32 pixels) categorized into 10 classes. A CNN with two convolutional layers followed by two fully connected layers is implemented. The methodology includes preprocessing steps such as data augmentation and normalization, alongside a custom module for generating class imbalance scenarios —specifically label imbalance, quantity imbalance, and double imbalance. Key training parameters, including the number of epochs, learning rate, optimizer, client count, and communication rounds, are detailed. Experiments are run using the NVIDIA Tesla T4 GPU available on Google Colab. Overall, this methodological framework enables a structured assessment of

algorithm performance in federated settings with controlled class imbalance conditions.

5 Results and Discussion

Contents

The Results and Discussion chapter compares three federated learning algorithms and a baseline algorithm under varying class imbalance conditions: label imbalance, quantity imbalance, and double imbalance. The study evaluates performance using the balanced accuracy metric. This analysis aims to provide insights into the robustness and effectiveness of federated learning algorithms when confronted with different imbalance scenarios, common challenges in real-world machine learning applications.

By examining balanced accuracy, the study offers a nuanced understanding of each algorithm's performance in handling imbalanced class distributions.

The following sections present detailed results for each algorithm across various imbalance types, highlighting the key findings.

5.1 Results

This section provides a comparative analysis of the performance of federated learning algorithms, specifically FedAvg, CUCB, CLIMB, and FedFed.

These algorithms are evaluated under various types of class imbalance, which are presented in separate tables: quantity imbalance,

label imbalance, and double class imbalance.

Each table details the balanced accuracy and the elapsed time for each algorithm. The experiments were conducted using the CIFAR-10 dataset.

The results illustrate how different algorithms manage class-unbalanced distributions and their efficacy in federated learning environments.

5.1.1 *Performance comparative for quantity imbalance type*

The table 5.1 summarizes the performance comparative of four federated learning algorithms: FedAvg, CUCB, CLIMB and FedFed. These algorithms are analyzed under quantity imbalance type, where some classes have significantly fewer samples than others. For example, the classes 0, 1, and 2, have only 10% of the data of other classes. The table shows the balanced accuracy and the elapsed time per algorithm. The recall is used to calculate the balanced accuracy, a more appropriate metric for imbalanced datasets. Elapsed time is provided to evaluate algorithm efficiency.

Test Accuracy by Class	**FedAvg**	**CUCB**	**CLIMB**	**FedFed**
Class 0 (plane)	0%	22.80%	49.46%	6%
Class 1 (car)	0%	20.40%	61.60%	3%
Class 2 (bird)	0%	3.60%	28.50%	0%
Class 3 (cat)	34%	55.60%	60.10%	48%
Class 4 (deer)	52%	65.60%	85.20%	62%
Class 5 (dog)	53%	58.80%	74.70%	53%
Class 6 (frog)	72%	78.30%	88.30%	75%
Class 7 (horse)	60%	72.20%	87.30%	64%
Class 8 (ship)	82%	91.10%	94.30%	87%
Class 9 (truck)	78%	79.40%	91.30%	80%
Balanced Accuracy	43.10%	54.78%	72.08%	47.60%
Elapsed Time (s)	4298.90"	557.67"	6080.38	8072.75"

Table 5.1: Performance Comparative under Quantity Imbalance Type

We can observe that FedAvg performs poorly for underrepresented classes, where these classes have zero accuracy. CUCB improves the accuracy across most classes, but still performs poorly in low-sample categories. CLIMB achieves the highest balanced accuracy. FedFed performs better than FedAvg, but not better than CUCB or CLIMB. CUCB is the fastest algorithm.

5.1.2 *Performance comparative for label imbalance type*

The table 5.2 displays the performance of the four algorithms in a label balance context. The label imbalance is represented by certain labels appearing less frequently across clients, or even can be missing.

Test Accuracy by Class	FedAvg	CUCB	CLIMB	FedFed
Class 0 (plane)	59.30%	69.30%	64.33%	63.60%
Class 1 (car)	62.80%	82.50%	75.20%	69.40%
Class 2 (bird)	39.00%	54.80%	71.50%	45.40%
Class 3 (cat)	35.90%	62.50%	42.77%	38.20%
Class 4 (deer)	38.30%	48.50%	64.63%	41.70%
Class 5 (dog)	55.80%	59.60%	71.43%	57.10%
Class 6 (frog)	55.90%	81.80%	76.83%	54.70%
Class 7 (horse)	57.10%	72.00%	73.82%	63.00%
Class 8 (ship)	60.50%	87.30%	83.11%	71.10%
Class 9 (truck)	59.50%	71.40%	81.95%	64.40%
Balanced Accuracy	52.01%	68.77%	70.96%	56.36%
Elapsed Time (s)	5633.37"	1056.49"	5755.77"	11257.53"

Table 5.2: Performance Comparison under Label imbalance type

As can be noticed, FedAvg performs significantly worse than the other algorithms. Interestingly, the four algorithms cope better with label imbalance than with quantity imbalance. Even CLIMB that presents a stable balanced accuracy, shows improvement in consuming a shorter elapsed time.

5.1.3 *Performance comparative for double imbalance type*

The table 5.3 displays the balanced accuracy of four algorithms in a scenario characterized by double imbalance.

This scenario is not as acute as previous ones, but also present challenges in federated learning environments. CUCB performs better than FedAvg. And CLIMB performs better than CUCB. FedFed performs sligthly better than FedAvg and consumes less time, however its balanced accuracy is still very low.

FedAvg shows lower accuracy, CLIMB outperforms the other algorithms as in the other distribution types, while CUCB and FedFed shows little improvement compared with FedAvg.

5.2 *Discussion*

Although quantity imbalances impact algorithm performance more significantly than label and double imbalances in the experiments, it is difficult to assert that the distribution type of quantity imbalance

Test Accuracy by Class	FedAvg	CUCB	CLIMB	FedFed
Class 0 (plane)	60.80%	66.70%	86.80%	59.30%
Class 1 (car)	70.20%	80.40%	85.40%	69.10%
Class 2 (bird)	38.20%	50.10%	65.30%	42.60%
Class 3 (cat)	22.90%	38.60%	65.20%	28.20%
Class 4 (deer)	35.20%	52.60%	81.50%	41.70%
Class 5 (dog)	59.80%	61.50%	56.70%	57.10%
Class 6 (frog)	68.00%	75.70%	82.90%	67.70%
Class 7 (horse)	61.90%	64.80%	73.20%	67.30%
Class 8 (ship)	63.60%	72.70%	88.50%	67.80%
Class 9 (truck)	54.90%	65.00%	80.00%	58.80%
Balanced Accuracy	53.55%	62.81%	76.52%	55.96%
Elapsed Time (s)	7341.11"	889.20"	6789.05"	11089.38"

Table 5.3: Performance Comparison with Double Class Distribution

alone is the primary reason for this greater degradation compared to other imbalance types. For example, Mahmood *et al.* [1] found that when the minority class makes up less than 20% of the dataset, the average Matthews Correlation Coefficient (MCC) falls to 0.15, whereas it rises to 0.34 when the minority class proportion reaches 20%.

The quantity imbalance results in only 10% of the dataset being represented in the first three classes, likely contributing to the depletion of the MCC[2] and exacerbating the class imbalance issue.

CUCB took less time to run, this is because one important step in CUCB algorithm is to approximate the distribution of customer data from a subset of the test data. However, this is not ideal, it doesn't keep data privacy, which is one important feature for federated learning. FedFed also shares data by using a subset of features from the data set, but this features are encrypted. FedFed obtains slightly better performance than FedAvg, and slightly worst than CUCB. The difference is that FedFed does meet Federated Learning privacy-preserving conditions, since it is sharing only some encrypted features and not the complete test dataset as in CUCB.

When comparing algorithms, we have to take into account the type of distribution they were designed for. FedAvg does not consider unbalanced datasets, whereas CUCB does so, in a general manner by resembling the distribution from the test dataset.

CLIMB is specifically designed to deal with imbalances when the local majority class is a global minority. For instance, in action recognition tasks using Inertial Measurement Unit (IMU) data collected from various mobile devices, the overall dataset may exhibit a long-tailed distribution (global imbalance), while individual devices may have their own unique imbalanced distributions of actions (local imbalance).

[1] Z. Mahmood, P. C. R. Lane, D. Bowes, and T. Hall. What is the impact of imbalance on software defect prediction performance? In *Proceedings of the 2015 International Conference on Predictive Models in Software Engineering (PROMISE)*, volume 40, pages 1–4, 2015. DOI: 10.1145/2810146.2810150

[2] The MCC is a robust and reliable statistical measure used to evaluate the performance of binary classifications and their associated confusion matrices. MCC is considered more informative and trustworthy than other commonly used metrics such as accuracy, F1 score, and confusion-entropy error. It produces a high score only when the prediction obtains good results in all four confusion matrix categories, proportionally to both the size of positive and negative elements in the dataset

Z. Mahmood, P. C. R. Lane, D. Bowes, and T. Hall. What is the impact of imbalance on software defect prediction performance? In *Proceedings of the 2015 International Conference on Predictive Models in Software Engineering (PROMISE)*, volume 40, pages 1–4, 2015. DOI: 10.1145/2810146.2810150

As a result, CLIMB offers a strong solution for handling complex imbalance situations and excels in performance not just for a particular distribution type but also across various other distributions, with CLIMB consistently delivering the best outcomes.

In general, these algorithms emphasize the need for designed approaches in federated learning under class imbalance, since the standard method like FedAvg is insufficient. The results show that in all class-unbalanced distributions, FedAvg always had the lowest accuracy, both by class and overall.

Several generalized concepts from machine learning are being applied in federated learning. For example, the FedFed algorithm leverages the distillation techniques from Retrieval-Augmented Generation (RAG) neural networks and applies those techniques in the federated learning environment.

5·3 *Algorithm Selection Recommendations based on specific data distribution*

In scenarios characterized by significant quantity imbalance, where certain classes possess substantially fewer samples, constituting less than 20% of the dataset, the performance of algorithms may deteriorate irrespective of the specific algorithm employed. In the course of this research, CLIMB exhibits the highest balanced accuracy under the specified conditions.

In situations with label imbalance, all the algorithms tested showed superior performance compared to scenarios with quantity imbalance. CLIMB remains the top choice, offering consistent balanced accuracy and faster processing times.

In scenarios marked by double imbalance, the CLIMB algorithm demonstrated superior performance compared to other algorithms. In contexts where there is a local-global mismatch, CLIMB is specifically engineered to address situations in which local majority classes are global minorities, rendering it particularly effective for such conditions.

Additional Considerations Beyond Data Distribution: Although CUCB demonstrated strong performance, it may not be suitable for environments with strict privacy requirements, as it relies on test data to estimate client data distribution. In such contexts, FedFed presents a more privacy-conscious alternative through its use of encrypted feature sharing. Moreover, in scenarios involving a large number of clients or constrained network conditions, FedFed is particularly advantageous, as it is optimized to minimize communication overhead and manage issues related to slow or intermittently connected clients.

5·4 Summary

The chapter on results and discussion examines the performance of federated learning algorithms when subjected to different class imbalance conditions. It offers a comparative analysis of four methods —FedAvg (baseline), CUCB, CLIMB, and FedFed— evaluated across scenarios involving quantity, label, and double imbalance. Balanced accuracy serves as the main evaluation metric due to its relevance for imbalanced datasets. CLIMB emerges as the top performer in quantity imbalance scenarios, achieving the highest balanced accuracy. CUCB improves upon FedAvg in most cases, while FedFed shows moderate gains. All algorithms perform better under label imbalance than quantity imbalance, with CLIMB also demonstrating greater stability and efficiency. The findings underscore the significant impact of quantity imbalance on performance and highlight CLIMB's effectiveness across varied imbalance challenges, reinforcing the need for specialized approaches in federated learning.

6 Conclusions

Contents

6.1 Conclusions

This study, adopts a comprehensive approach to evaluating federated learning distributed algorithms designed to address class imbalance. The resilience and flexibility of these algorithms are evaluated by exposing them to different scenarios of class imbalance and not only to the scenario for which the algorithm was developed. The process of parameter and model selection involved: Determining suitable hyperparameters, choosing an appropriate dataset, designing a neural network model tailored to the specific task.

Since the selected dataset was perfectly balanced, a controlled imbalanced had to be added. Creating the desired class imbalance scenarios can be achieved using the Dirichlet distribution or heuristics to produce extremely imbalanced datasets. Another critical aspect is selecting the most appropriate performance evaluation metric for the specific experimental conditions. In this case, the chosen metric must account for the imbalanced datasets to prevent convergence issues caused by the under representation of minority classes.

The experiments evaluated quantity imbalance, label imbalance, and double imbalance conditions using balanced accuracy as the primary performance metric. CUCB, which deduces data distribution from the test dataset, showed quicker convergence than other algorithms. This indicates that utilizing test set information can be advantageous for scaling solutions. However, this method is not favored in federated learning as it is perceived that it does not fully comply with the privacy principles. CLIMB, which focuses on local-global mismatch imbalance, achieved strong performance, particularly in scenarios where local majority classes were global minorities, highlighting the importance

of considering not only local class imbalance type within the client, but also the relationship between local and global data distribution mismatches in federated learning.

Interestingly, the extreme imbalances simulated in some cases, likely contributed to lower performance across algorithms, rather than significant differences in how the algorithms handle specific imbalance types.

In this thesis it was shown how these algorithms handle imbalance in federated learning. They incorporate innovations from the machine learning development to tackle class imbalance challenges, such as generative adversarial networks (FedFed). These algorithms also integrate various methods to address class imbalance. For instance, client selection and cost-sensitive algorithms (CUCB). Additionally, also is shown that there is an inclination to merge multi-objective federated learning algorithms that simultaneously resolve multiple issues. For example, an algorithm that not only deals with class imbalance but also incorporates scalability to enhance widespread adoption in scenarios involving numerous edge computing environments (FedFed).

**Provide recommendations for algorithm selection based on specific data distribution characteristics in federated learning environments.

6.2 *Future work*

Future research will focus on examining the potential reasons behind FedFed's low balanced accuracy ranking. Key areas for further investigation include the fact that FedFed is a relatively new algorithm that integrates cutting-edge techniques from machine learning and cybersecurity. FedFed introduces several significant innovations, such as feature distillation, which categorizes data into performance-sensitive features that greatly improve model performance and performance-robust features that have a minimal impact on performance. It selectively shares only the performance-sensitive features globally, while keeping performance-robust features local. This helps address data heterogeneity while maintaining privacy. The algorithm incorporates differential privacy protection and an information bottleneck approach to determine which features to share. Additionally, an AI generative model is used to create performance-robust features. Due to these factors, FedFed is expected to deliver high results.

FedFed is primarily designed to minimize communication costs in environments with limited network resources. This enables the algorithm to manage situations where new clients join or leave the federation during the learning process or where clients are slow to respond. These characteristics help FedFed tackle class imbalance.

Future work will also involve exploring the model's performance under suitable conditions for the algorithm to thrive.

While conducting market research on companies currently utilizing federated learning or offering related solutions, not firms in Mexico specifically focusing on federated learning were found. However, there are applications like https://www.diagnostikare.com/que-es/ that are gathering private data and training models. If they are not already employing federated learning, they likely will in the near future. In the healthcare sector, adhering to data protection standards is crucial. An interesting future work would be to determine the specific constraints these models face and select a suitable FL algorithm to handle the data -which probably is imbalanced, while maintaining privacy.

Bibliography

Beyza Akbugday. Classification of breast cancer data using machine learning algorithms. In *2019 Medical Technologies Congress (TIPTEKNO)*, pages 1–4, October 2019. DOI: 10.1109/tiptekno.2019.8895222.

W. Salah Alaloul and A. Hannan Qureshi. *Data Processing Using Artificial Neural Networks*. IntechOpen, 2020. DOI: 10.5772/intechopen.91935.

A. Alferaidi, G. Dhiman, Y. Alharbi, W. Viriyasitavat, K. Yadav, and S. Kautish. Federated learning algorithms to optimize the client and cost selections. *Mathematical Problems in Engineering*, 2022:1–9, 2022. DOI: 10.1155/2022/8514562.

M. Aljanabi. Safeguarding connected health: Leveraging trustworthy ai techniques to harden intrusion detection systems against data poisoning threats in iomt environments. *Babylonian Journal of Internet of Things*, 2023:31–37, May 2023. DOI: 10.58496/bjiot/2023/005.

Shervine Amidi. Machine learning tips and tricks cheatsheet, 2021. URL https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks. Accessed: March 30, 2025.

T. M. Antico, L. F. R. Moreira, and R. Moreira. Evaluating the potential of federated learning for maize leaf disease prediction. Conference Paper, 2022.

M. Asad, A. Moustafa, and T. Ito. Fedopt: Towards communication efficiency and privacy preservation in federated learning. *Applied Sciences*, 10(8):2864, Apr 2020. DOI: 10.3390/app10082864.

M. A. Attia and R. Tandon. On the worst-case communication overhead for distributed data shuffling. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 961–968, Sep. 2016. DOI: 10.1109/allerton.2016.7852338.

M. Aziz and D. Patel. Privacy-preserving machine learning with homomorphic encryption in federated learning. *Journal of Privacy and Confidentiality*, 15(2):1–18, 2023.

R. Aziz, T. Le Vinh, S. Bouzefrane, and S. Banerjee. Exploring homomorphic encryption and differential privacy techniques towards secure federated learning paradigm. *Future Internet*, 15(9):310, 2023. DOI: 10.3390/fi15090310.

S. Bharati, M. R. H. Mondal, V. B. S. Prasath, and P. Podder. Federated learning: Applications, challenges and future directions. *International Journal of Hybrid Intelligent Systems*, 18(1–2):19–35, 2022. DOI: 10.3233/his-220006.

B. Biggio, G. Fumera, and F. Roli. Multiple classifier systems for robust classifier design in adversarial environments. *International Journal of Machine Learning and Cybernetics*, 1(1–4):27–41, October 2010. DOI: 10.1007/s13042-010-0007-7.

K. Bonawitz and V. Ivanov. Secure aggregation for federated learning. *Proceedings of the ACM on Privacy Enhancing Technologies*, 2021(1):1–25, 2021.

N. Bouacida and P. Mohapatra. Vulnerabilities in federated learning. *IEEE Access*, 9:63229–63249, Jan 2021. DOI: 10.1109/access.2021.3075203.

C. Cao, Y. Zhang, H. Lu, C. Lan, Y. Zhang, and W. Zeng. Skeleton-based action recognition with gated convolutional neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(11): 3247–3257, 2019. DOI: 10.1109/tcsvt.2018.2879913.

Longbing Cao. Non-iid recommender systems: A review and framework of recommendation paradigm shifting. *Engineering*, 2 (2):212–224, June 2016. DOI: 10.1016/j.eng.2016.02.013.

P. et al. Chaves. Federated learning for iot: A privacy-preserving approach. *Sensors*, 2024.

J. Chen and Y. Lin. Asynchronous federated learning with dynamic scaling factors. *Information Sciences*, 634:19–34, 2023.

Ee Kin Chin. *The Deep Learning Architect's Handbook*. Packt, first edition, 2023. ISBN 9781803235349.

Jihye Chung and Jason Teo. Single classifier vs. ensemble machine learning approaches for mental health prediction. *Brain Informatics*, 10 (1), January 2023. DOI: 10.1186/s40708-022-00180-6.

NVIDIA Corporation. Nvidia t4 tensor core gpu, 2018. URL https://www.nvidia.com/en-us/data-center/tesla-t4/. Accessed: 2025-04-02.

G. Drainakis, A. Amditis, P. Pantazopoulos, V. Sourlas, and K. V. Katsaros. Federated vs. centralized machine learning under privacy-elastic users: A comparative analysis. In *Proceedings of the 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, pages 1–8, Nov 2020. DOI: 10.1109/nca51143.2020.9306745.

David Díaz-Vico, Anas Omari, José Dorronsoro, and Javier Prada. Deep support vector neural networks. *Integrated Computer-Aided Engineering*, 27(4):389–402, September 2020.

D. Wang et al. Fedabc: Targeting fair competition in personalized federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10095–10103, 2023a. DOI: 10.1609/aaai.v37i8.26203.

F. Yu et al. Communication efficient personalized federated meta learning in edge networks. *IEEE Transactions on Network and Service Management*, 20(2):1558–1571, 2023b. DOI: 10.1109/tnsm.2023.3263831.

L. Meng et al. Improving global generalization and local personalization for federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, PP(1):76–87, 2025. DOI: 10.1109/tnnls.2024.3417452.

M. Oldenhof et al. Industry-scale orchestrated federated learning for drug discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15576–15584, 2023c. DOI: 10.1609/aaai.v37i13.26847.

O. Markaki et al. Encouraging ai adoption by smes: Opportunities and contributions by the ict49 project cluster. In *2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pages 1–8, 2023d. DOI: 10.1109/IISA59645.2023.10345867.

W. Heyndrickx et al. Melloddy: Cross-pharma federated learning at unprecedented scale unlocks benefits in qsar without compromising proprietary information. *Journal of Chemical Information and Modeling*, 64(7):2331–2344, 2023e. DOI: 10.1021/acs.jcim.3c00799.

F. Fang and X. Qian. Enhancing federated learning with trusted execution environments. *IEEE Transactions on Dependable and Secure Computing*, 18(4):1671–1685, 2021.

B. Gecer and B. Garbinato. Applications of federated learning in urban and smart systems. *Journal of Smart City Research*, 2024.

Ziv Goldfeld and Yury Polyanskiy. The information bottleneck problem and its applications in machine learning. *IEEE Journal*

on Selected Areas in Information Theory, 1(1):19–38, May 2020. DOI: 10.1109/jsait.2020.2991561.

Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview, 2020. URL https://arxiv.org/abs/2008.05756.

Shuo Guo et al. Fedgr: Federated learning with gravitation regulation for double imbalance distribution. In *Database Systems for Advanced Applications. DASFAA 2023*, volume 13943 of *Lecture Notes in Computer Science*. Springer, Cham, 2023.

Ayhan Göde and Abdullah Kalkan. Performance comparison machine learning algorithms in diabetes disease prediction. *European Mechanical Science*, 7(3):178–183, September 2023. DOI: 10.26701/ems.1335503.

W. Haensch, R. Puri, and T. Gokmen. The next generation of deep learning hardware: Analog computing. *Proceedings of the IEEE*, 107(1): 108–122, January 2019. DOI: 10.1109/jproc.2018.2871057.

E. Hernandez and J. Smith. Federated learning: A privacy-preserving approach to distributed ai. *Journal of Distributed AI Research*, 12(3): 101–120, 2024.

Minguk Ji, Seungeun Park, and Bohyung Heo. Show, attend and distill: Knowledge distillation via attention-based feature matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7945–7952, May 2021. DOI: 10.1609/aaai.v35i9.16969.

Y. Jiang, K. Rush, J. Konečný, and S. Kannan. Improving federated learning personalization via model agnostic meta learning, 2019. URL https://arxiv.org/abs/1909.12488.

Jianzgong Qi Jiayuan He Jing Zhang, Chuanwen Li. A Survey on Class Imbalance in Federated Learning. *Journal of latex class files, VOL. 14, NO. 8*, 2023. https://doi.org/10.48550/arXiv.2303.11673.

P. Kairouz and B. McMahan. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14 (1–2):1–210, 2021.

A. P. Kalapaaking, I. Khalil, and X. Yi. Blockchain-based federated learning with smpc model verification against poisoning attack for healthcare systems. *IEEE Transactions on Emerging Topics in Computing*, 12(1):269–280, Jan 2024. DOI: 10.1109/tetc.2023.3268186.

R. Kalapaaking and M. Nguyen. Blockchain for securing federated learning: A survey. *Future Generation Computer Systems*, 145:303–320, 2024.

Mohamed Kalash, Fahad Iqbal, Neil D. B. Bruce, Noman Mohammed, Yang Wang, and Morteza Rochan. Malware classification with deep convolutional neural networks. In *2018 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5, February 2018.

J. Kang and Q. Yang. Hdp-fl: Hybrid differential privacy federated learning. *Neurocomputing*, 512:155–168, 2023.

Rudra Katuwal and Ponnuthurai Nagaratnam Suganthan. Enhancing multi-class classification of random forest using random vector functional neural network and oblique decision surfaces. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2018. DOI: 10.1109/ijcnn.2018.8489738.

K. M. N. K. Khalif, A. Gegov, N. A. Shahrul, A. S. A. Bakar, and W. Chaw Seng. Integrated generative adversarial networks and deep convolutional neural networks for image data classification: A case study for covid-19. *Information*, 15(1):58, 2024. DOI: 10.3390/info15010058.

M. Kim et al. Deep learning in medical imaging. *Neurospine*, 17(2): 471–472, Jun 2020. DOI: 10.14245/ns.1938396.198.c1.

H. Kimm, H. Kimm, and I. Paik. Performance comparison of tpu, gpu, cpu on google colaboratory over distributed deep learning. In *2021 15th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)*, volume 119, pages 312–319, 2021. DOI: 10.1109/mcsoc51149.2021.00053.

George Jeno Kiyoshi Nakayama, PhD. *Federated Learning with Python*. Packt, first edition, 2022. ISBN 9781803248752.

C. Korkmaz, A. Uysal, A. Masry, O. Ozkasap, H. E. Kocas, and B. Akgun. Chain fl: Decentralized federated machine learning via blockchain. In *2020 International Conference on Blockchain Computing and Applications (BCCA)*, volume abs/1905.6731, pages 140–146, November 2020. DOI: 10.1109/bcca50787.2020.9274451.

Alex Krizhevsky. Learning multiple layers of features from tiny images. https://www.cs.toronto.edu/~kriz/cifar.html, 2009. Technical Report, University of Toronto.

Dr. Mounir Abdelaziz Kumar Abhishek. *Machine Learning for Imbalanced Data*. Packt, first edition, Nov 2023. ISBN 9781801070881.

Maxim Lapan. *Deep Reinforcement Learning Hands-On*. Packt, third edition, Nov 2024.

T. Li and A. K. Sahu. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning Research*, 139:4294–4304, 2021.

W. Li and H. Yu. Blockchain-based federated learning: A survey. *IEEE Transactions on Industrial Informatics*, 19(2):1432–1445, 2023.

Xiao Chen Li, Baoyuan Li, Yiqing Shao, Shuaiqiang Song, and De-Chuan Zhan. Fedphp: Federated personalization with inherited private models. In *Advances in Knowledge Discovery and Data Mining (PAKDD 2021)*, Lecture Notes in Computer Science, pages 587–602. Springer, 2021. DOI: 10.1007/978-3-030-86486-636.

Y. Li, Q. Yan, N. Liu, Z. Zheng, C. Chen, and H. Huang. A blockchain-based decentralized federated learning framework with committee consensus. *IEEE Network*, 35(1):234–241, 2020. DOI: 10.1109/mnet.011.2000263.

Wonjoon Lim, Taesu Lee, and Dongsuk Jang. Speech emotion recognition using convolutional and recurrent neural networks. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, December 2016.

Jiayu Lin. On the dirichlet distribution. Master's thesis, Queen's University, Kingston, Ontario, Canada, September 2016. A report submitted to the Department of Mathematics and Statistics in conformity with the requirements for the degree of Master of Science.

Bing Liu, Xiaoyuan Yu, Anlin Yu, and Guodong Wan. Deep convolutional recurrent neural network with transfer learning for hyperspectral image classification. *Journal of Applied Remote Sensing*, 12 (02):026028, June 2018.

F. Liu, M. Li, J. Ren, T. Xue, X. Liu, and C. Zhang. A review of federated meta-learning and its application in cyberspace security. *Electronics*, 12 (15):3295, Jul 2023. DOI: 10.3390/electronics12153295.

Yuxi (Hayden) Liu. *Python Machine Learning By Example*. Packt, fourth edition, July 2024. ISBN 9781835085622.

G. et al. Long. Privacy-preserving machine learning: Threats and solutions. *Journal of Computer Science*, 2020.

L. Lyu and H. Yu. Threats to federated learning: A survey. *IEEE Access*, 8:173532–173550, 2020.

Z. Mahmood, P. C. R. Lane, D. Bowes, and T. Hall. What is the impact of imbalance on software defect prediction performance? In *Proceedings of the 2015 International Conference on Predictive Models in*

Software Engineering (PROMISE), volume 40, pages 1–4, 2015. DOI: 10.1145/2810146.2810150.

S. Maind and P. Wankar. Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(1):96–100, Jan 2014. DOI: 10.17762/ijritcc.v2i1.2920.

M. Mansouri, M. Conti, W. Ben Jaballah, and M. Önen. Sok: Secure aggregation based on cryptographic schemes for federated learning. *Proceedings on Privacy Enhancing Technologies*, 2023(1):140–157, Jan 2023. DOI: 10.56553/popets-2023-0009.

H. U. Manzoor, A. Zoha, A. Shabbir, D. Flynn, and A. Chen. A survey of security strategies in federated learning: Defending models, data, and privacy. *Future Internet*, 16(10):374, Oct 2024. DOI: 10.3390/fi16100374.

Aashish Mehra. Federated learning market press release. https://www.marketsandmarkets.com/ PressReleases/federated-learning-solutions.asp, 2024. Accessed: 2025-04-12.

Hao Miao, Amol Deshpande, Larry S. Davis, and Abhishek Li. Modelhub: Deep learning lifecycle management. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 1393–1394, April 2017.

V. Nawa and S. Nadarajah. Exact expressions for kullback-leibler divergence for univariate distributions. *Entropy (Basel, Switzerland)*, 26 (11), 2024. DOI: 10.3390/e26110959.

F. Neto and A. Singh. Inference attacks in federated learning: A survey. *Journal of Information Security and Applications*, 68:103310, 2023.

Diego Nigenda, Sridhar Alla, Niyati Mehta, Dipayan Banerjee, Shenghui Cheng, Shiva Mandala, Senthil Nathan, Amogh Tiwari, Alexandra Theresia, Ramesh Subramonian, Sunil Mallya, and Subbarao Kambhampati. Amazon sagemaker model monitor: A system for real-time insights into deployed machine learning models. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*, volume 104, pages 3671–3681, August 2022. DOI: 10.1145/3534678.3539145.

S. Park and D. Kim. Bcfl for secure smart transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 24(4):3421–3430, 2023.

J. Qi, Q. Zhou, L. Lei, and K. Zheng. Federated reinforcement learning: techniques, applications, and open challenges. *Intelligence & Robotics*, 2021. DOI: 10.20517/ir.2021.02.

M. Riaz, M. Riaz, Z. Zararsiz, and N. Jamil. Distance and similarity measures for bipolar fuzzy soft sets with application to pharmaceutical logistics and supply chain management. *Journal of Intelligent & Fuzzy Systems*, 42(4):3169–3188, Mar. 2022. DOI: 10.3233/jifs-210873.

George Saon and Michael Picheny. Recent advances in conversational speech recognition using convolutional and recurrent neural networks. *IBM Journal of Research and Development*, 61(4/5):1:1–1:10, July 2017.

Vahid Mirjalili Sebastian Raschka. *Python Machine Learning*. Packt, third edition, 2019. ISBN 9781789955750.

Xiaofeng Shang, Yiu ming Cheung, Hao Wang, and Yuxuan Lu. Fedic: Federated learning on non-iid and long-tailed data via calibrated distillation. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, July 2022. DOI: 10.1109/icme52920.2022.9860009.

M. J. Sheller and B. Edwards. Federated learning in medical imaging: Enabling privacy-preserving collaboration. *Medical Image Analysis*, 65: 101765, 2020.

Zebang Shen. Federated-Learning-Pytorch, 9 2021. URL https://github.com/shenzebang/Federated-Learning-Pytorch.

Zebang Shen, Juan Cervino, Hamed Hassani, and Alejandro Ribeiro. An agnostic approach to federated learning with class imbalance. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Xo0lbDt975.

X. Shuai, Z. Yan, Z. Zhao, S. Jiang, G. Xing, and Y. Shen. BalanceFL: Addressing Class Imbalance in Long-Tail Federated Learning. In *Proceedings of the IEEE/ACM IPSN*, 2022. DOI: 10.1109/ipsn54338.2022.00029.

Susarla Sinan Ozdemir. *Feature Engineering Made Easy*. Packt, first edition, Jan 2018. ISBN 9781787286474.

S. Singhal. Data privacy, compliance, and security including ai ml. In *IGI Global*, pages 111–126. 2024. DOI: 10.4018/979-8-3693-2909-2.ch009.

H. Sun, Y. Chen, J. Lin, X. Zhou, and Z. Gao. A Linear Interpolation and Curvature-Controlled Gradient Optimization Strategy Based on Adam. *Algorithms*, 17(5):185, April 2024. DOI: 10.3390/a17050185.

W. Sun, B. Guo, L. Xu, T. Q. Duong, Y. Zhao, and W. Ma. Accelerating convergence of federated learning in mec with dynamic community. *IEEE Transactions on Mobile Computing*, pages 1–17, 2023. DOI: 10.1109/tmc.2023.3241770.

Wenhao Sun, Bin Zheng, and Wei Qian. Computer aided lung cancer diagnosis with deep learning algorithms. In *Medical Imaging 2016: Computer-Aided Diagnosis*, volume 9785, page 97850Z, March 2016.

F. Tang and J. Liu. Lsfl: A lightweight and secure federated learning scheme for iot. *IEEE Internet of Things Journal*, 10(1):334–345, 2023.

Q. Tang, F. Shpilevskiy, and M. Lécuyer. DP-AdamBC: Your DP-Adam Is Actually DP-SGD (Unless You Apply Bias Correction). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15276–15283, March 2024. DOI: 10.1609/aaai.v38i14.29451.

A. Tanimoto, S. Yamada, T. Takenouchi, M. Sugiyama, and H. Kashima. Improving imbalanced classification using near-miss instances. *Expert Systems with Applications*, 201:117130, Apr. 2022. DOI: 10.1016/j.eswa.2022.117130.

Joseph Thomas, Justin Dauwels, Nitesh Sinha, Thomas Kluge, and Tomasz Maszczyk. Deep learning-based classification for brain-computer interfaces. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 234–239, October 2017.

J. Tian et al. Synergetic focal loss for imbalanced classification in federated xgboost. *IEEE Transactions on Artificial Intelligence*, 5(2):647–660, February 2024a. DOI: 10.1109/tai.2023.3254519.

J. Tian et al. Synergetic Focal Loss for Imbalanced Classification in Federated XGBoost. *IEEE Transactions on Artificial Intelligence*, 5(2): 647–660, 2024b. DOI: 10.1109/tai.2023.3254519.

Tuomanen. *Hands-On GPU Programming with Python and CUDA*. Packt, first edition, Nov 2018. ISBN 978-1788993913.

Ivan Vasilev. *Python Deep Learning*. Packt, third edition, Nov 2023. ISBN 9781835085622.

J. Victor and L. Chen. Federated learning for the internet of underwater things. *IEEE Internet of Things Journal*, 9(5):3500–3512, 2022.

D. Wang et al. FedABC: Targeting Fair Competition in Personalized Federated Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10095–10103, 2023. DOI: 10.1609/aaai.v37i8.26203.

J. Wang and X. Chen. Chainfl: Blockchain-based decentralized federated learning. *IEEE Transactions on Network and Service Management*, 19(3): 2765–2778, 2022.

L. Wang, S. Xu, X. Wang, and Q. Zhu. Addressing class imbalance in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10165–10173, May 2021. DOI: 10.1609/aaai.v35i11.17219.

X.-X. Wei and H. Huang. Balanced Federated Semisupervised Learning With Fairness-Aware Pseudo-Labeling. *IEEE Transactions on Neural Networks and Learning Systems*, 35(7):9395–9407, 2024. DOI: 10.1109/tnnls.2022.3233093.

H. Wu and Y. Zhang. Personalized federated learning for heterogeneous iot systems. *ACM Transactions on Internet Technology*, 20(4):1–22, 2020.

J. Xiao, Z. Duan, C. Du, and W. Guo. A novel server-side aggregation strategy for federated learning in non-iid situations. In *2021 International Symposium on Parallel and Distributed Computing (ISPDC)*, July 2021. DOI: 10.1109/ispdc52870.2021.9521631.

L. Yang, J. Cao, J. Huang, and W. Lin. Personalized federated learning on non-iid data via group-based meta-learning. *ACM Transactions on Knowledge Discovery from Data*, 17(4):1–20, 2023a. DOI: 10.1145/3558005.

M. Yang, H. Zhu, H. Wang, H. Qian, and X. Wang. Federated learning with class imbalance reduction. In *EUSIPCO*, 2021. DOI: 10.23919/eusipco54536.2021.9616052.

Miao Yang. Federated-Learning-Pytorch, 9 2020. URL https://github.com/ym1231/fl-cir.

Miao Yang. FedFed: Feature Distillation against Data Heterogeneity in Federated Learning (NeurIPS 2023), 10 2023. URL https://github.com/visitworld123/FedFed.

Miao Yang, Akitanoshou Wong, Hongbin Zhu, Haifeng Wang, and Hua Qian. Federated learning with class imbalance reduction. *CoRR*, abs/2011.11266, 2020. URL https://arxiv.org/abs/2011.11266.

Q. Yang and Y. Liu. Vertical federated learning with encrypted computation. *IEEE Transactions on Knowledge and Data Engineering*, 34(2):321–334, 2022.

Q. Yang, Y. Tong, T. Chen, and Y. Liu. Federated machine learning. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–19, 2019. DOI: 10.1145/3298981.

Q. et al. Yang. Federated machine learning: Concept and applications. *Nature Machine Intelligence*, 2024.

Zhiqin Yang, Yonggang Zhang, Yu Zheng, Xinmei Tian, Hao Peng, Tongliang Liu, and Bo Han. Fedfed: Feature distillation against data heterogeneity in federated learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b.

R. Yogitha and G. Mathivanan. Performance analysis of transfer functions in an artificial neural network. In *2018 International Conference on Communication and Signal Processing (ICCSP)*, volume 15, pages 393–397, Apr 2018. DOI: 10.1109/iccsp.2018.8524387.

H. Yu, C. Wu, H. Yu, X. Wei, S. Liu, and Y. Zhang. A federated learning algorithm using parallel-ensemble method on non-iid datasets. *Complex & Intelligent Systems*, 9(6):6891–6903, 2023. DOI: 10.1007/s40747-023-01110-7.

C. Zhang and W. Xie. Security challenges in federated learning: A survey. *ACM Computing Surveys*, 54(5):1–36, 2021.

J. Zhang, B. Xie, M. Li, D. Zhao, and S. Zeng. A survey on security and privacy threats to federated learning, Oct 2021a.

J. Zhang, Q. Xu, F. Wang, J. Zhao, H. Li, and H. Zhu. Security and privacy threats to federated learning: Issues, methods, and challenges. *Security and Communication Networks*, 2022:1–24, Sep 2022. DOI: 10.1155/2022/2886795.

Liang Zhang, Bo Du, Ling-Yu Duan, Yihang Luo, and Yihui Bai. Federated learning for non-iid data via unified feature learning and optimization objective alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4400–4408, October 2021b.

Weiqiang Zhang, Peng Zhou, Xiang Zhang, Wenbo Wu, and Xin Wang. Client selection for federated learning with non-iid data in mobile edge computing. *IEEE Access*, 9:24462–24474, January 2021c.

X. Zhang and Y. Shen. Fedipr: Intellectual property protection for federated learning. *Journal of Systems Architecture*, 127:102399, 2022.

Y. Zhao and M. Li. Impact of data heterogeneity on federated learning. *arXiv preprint arXiv:1806.00582*, 2020.

Yuzhe Zhao and Jie Chen. A survey on differential privacy for unstructured data content. *ACM Computing Surveys*, 54(10s):1–28, January 2022. DOI: 10.1145/3490237.

Zhengming Zhao et al. Federated learning with non-iid data in wireless networks. *IEEE Transactions on Wireless Communications*, 21(3):1927–1942, March 2022. DOI: 10.1109/twc.2021.3108197.

H. Zheng, Z. Han, and H. Hu. Preserving user privacy for machine learning: Local differential privacy or federated machine learning? *IEEE Intelligent Systems*, 35(4):5–14, Jul 2020. DOI: 10.1109/mis.2020.3010335.

Q. Zhou and B. He. Fednic: Enabling federated learning with smart nics. *ACM Transactions on Computer Systems*, 41(1):1–27, 2023.

Index