UNIVERSITY OF TORINO

Ph.D. in Modeling and Data Science

XXXV cycle

Final dissertation

Machine Learning and Federated Learning for Medical Applications

Supervisor: Marco Aldinucci
Co-supervisor: Roberto Esposito

Candidate: Yasir Arfat

ACADEMIC YEAR 2021/2022

Summary

Machine learning (ML) is a rapidly growing field that has made significant strides in many research fields, including medical research and, especially in cardiology. The most prominent developments include improving diagnostic accuracy, risk stratification, and treatment optimization in various cardiovascular diseases. Specifically, ML is becoming a central tool for predicting the likelihood of adverse cardiovascular events identifying the risks of heart failure, and arrhythmia detection. These tasks are challenging due to the nature of data, which is collected in large amounts and from heterogeneous data sources, including electronic health records (EHRs), imaging studies, and wearable devices. While the data is plentiful, it is usually scattered between multiple entities that are legally bound to protect patient privacy. This poses a significant challenge for ML researchers, as the data available at a single site is usually not sufficient to train a robust model, but the aforementioned privacy concerns prevent the sharing of data across sites. This challenge can be addressed by federated learning, a promising approach that allows multiple actors to collaborate in building a shared model without compromising patient privacy. In federated learning, data remains stored locally at each site, and ML models are trained using shared model parameters from all sites rather than sharing the underlying data itself.

This thesis explores different tools for federated learning and their specific applications in medical domains. We conducted a comprehensive literature review on recent techniques used in federated learning and commonly used machine learning techniques for cardiology risk score assessment. We discuss these tools, analyzing the usage of novel tools as well as those frequently used in the cardiology field. From the literature analysis, we learn that simple tools based on ensemble methods (specifically Random Forests) and Logistic Regression are the most commonly employed techniques. Indeed, while neural networks are currently at the forefront of machine learning

advancements, they are rarely used in this field. The reason is likely multifaceted, but we argue that this may be potentially due to interpretation and training challenges. We suggest that adopting large, federated datasets and semi-supervised techniques could significantly improve the current performance of predictions based on machine learning techniques and pave the way for more sophisticated approaches.

Another contribution of this thesis is the development of the PRAISE score calculator. The PRAISE score calculator is an application used in the field of cardiology. It is a tool specifically designed to assess the risk of adverse cardiovascular events in patients who undergo percutaneous coronary intervention (PCI), which is a common procedure to treat coronary artery disease. The PRAISE score considers various clinical factors, such as age, sex, comorbidities, and procedural characteristics, to provide an estimation of the patient's risk profile. It assists healthcare professionals in making informed decisions regarding the management and treatment of patients undergoing PCI. It is risk analysis tool based on the PRAISE dataset having AdaBoost and Random Forest machine learning techniques. We finely tuned the parameters of the algorithm, and we were able to achieve good predictive results. We were then able to build a web-based tool (the PRAISE score calculator), which is freely available online for anyone to use. Finally, we also analyzed the possibility of developing similar tools on the PRAISE dataset using federated learning techniques. We compared the performance of two methods FedAvg and `AdaBoost.F`, from the point of view of prediction performances, computation requirements, and communication costs. We achieved $F1$ and $F2$ scores consistently comparable to the PRAISE score study using a 16-parties federation but within an order of magnitude less time and ensuring that the privacy of the involved patients was not compromised.

Acknowledgements

Enrolling in a Ph.D. program is comparable to embarking on a lengthy and meandering route that gradually moves towards scientific, professional, and personal growth. It involves traversing a prolonged expedition, where the elation and gratification of mastering new concepts interweave with despondency and disappointment at yet-to-be-mastered ones. While an individual's thirst for knowledge is undoubtedly crucial in reaching the ultimate goal of this expedition, I have found that travelling alongside the appropriate companions is equally significant in my personal experience. This section is a tribute to all the members who have been a part of my journey.

Firstly, I want to express my gratitude and appreciation to Prof. Marco Aldinucci, my supervisor during my PhD degree, for his unwavering support. I could not have asked for a better advisor and mentor for my PhD degree than him. His expertise, motivation, and mentorship have been invaluable in helping me develop my problem-solving and research skills. Whenever I faced research problems, his patient guidance and immense knowledge were instrumental in helping me overcome them.

Throughout my research, Prof. Marco Aldinucci advice and constructive suggestions kept me motivated and focused on my goals. His unwavering support and belief in me gave me the confidence to achieve my objectives. I am profoundly grateful to him for his indispensable help in completing this thesis within the required time frame.

I also extend my heartfelt thanks to Prof. Roberto Esposito for his unwavering support and encouragement throughout this thesis. His moral support played a vital role in motivating me to complete this research within the deadline. He also provided invaluable assistance in resolving problems related to the research problems. I cannot overstate the role that he played in making this research possible, and I will always be grateful for his unwavering commitment to my success.

I would also like to express my sincere gratitude to my colleagues, including Barbara Cantalupo, Robert Birke, Iacopo Colonnelli, Gianluca Mittone, Doriana Medic, Bruno Casella, Alberto Riccardo Martinelli, Alberto Mulone and Giulio Malenza. Together, we have experienced the highs and lows of research, and even a pandemic did not hinder our collaboration, idea-sharing, and enjoyment of good company. I want to thank my friend Thaha Muhammad for his invaluable support throughout my PhD journey. He generously dedicated his time to helping me resolve countless technical issues, enhancing my research's value. I will always cherish his significant contributions, which will be remembered for years.

I would like to express my profound gratitude and affection to my family, including my parents, wife & children, brother and sisters, for their unwavering support and prayers. Their assistance has been invaluable, and I could not have completed this project without them. I would also like to extend my heartfelt thanks to everyone who supported me and provided me with the strength to persevere through the challenges, particularly during the difficult times.

I am also grateful to all my teachers in the Department of Computer Science and Modeling and Data Science Department particular, and in University of Turin(UniTo) in general. I am so proud to be a student of UniTo.

Contents

List of Tables 9

List of Figures 10

1 Introduction 11
- 1.1 Main contributions . . . 13
- 1.2 List of publications . . . 14
  - 1.2.1 Publications organised by venue . . . 14
  - 1.2.2 Developed Tools . . . 16

2 Background on Federated Learning 17
- 2.1 Federated Learning . . . 17
- 2.2 Distribution types of Federated Learning . . . 18
  - 2.2.1 Datacenter distributed learning . . . 18
  - 2.2.2 Cross-silo federated learning . . . 20
  - 2.2.3 Cross-device federated learning . . . 21
- 2.3 Categorization of federated learning . . . 22
  - 2.3.1 Horizontal federated learning . . . 22
  - 2.3.2 Vertical federated learning . . . 23
  - 2.3.3 Federated transfer learning . . . 24
- 2.4 Federated Learning Tools . . . 24
  - 2.4.1 The intel OpenFL® framework . . . 24
  - 2.4.2 TensorFlow Federated . . . 26
  - 2.4.3 PySyft . . . 26
  - 2.4.4 IBM® federated Learning . . . 27
  - 2.4.5 HPE Swarm Learning . . . 28
  - 2.4.6 FATE . . . 28
  - 2.4.7 FedML . . . 29
  - 2.4.8 Flower . . . 30

2.4.9 Fed-BioMed . . . 31
2.4.10 FederatedScope . . . 31
2.4.11 FLUTE . . . 31
2.4.12 FLARE . . . 32
2.5 Literature on Federated Learning . . . 32
2.5.1 Strong scalability in federated learning . . . 34
2.5.2 Weak scalability in federated learning . . . 34

3 Machine learning for Cardiology 35
3.1 Machine Learning: A background and importance . . . 36
3.2 Data handling . . . 38
3.2.1 Features types . . . 38
3.2.2 Missing values . . . 39
3.2.3 Feature selection . . . 39
3.2.4 Class imbalances . . . 40
3.2.5 Feature normalization . . . 40
3.2.6 Dataset splitting . . . 41
3.2.7 Dataset size . . . 41
3.2.8 Follow-up time . . . 42
3.2.9 Privacy, security, and features . . . 42
3.3 Machine Learning techniques . . . 42
3.3.1 Logistic Regression . . . 45
3.3.2 K-Nearest Neighbors . . . 47
3.3.3 Decision Trees . . . 47
3.3.4 Random Forest . . . 48
3.3.5 Boosting . . . 48
3.3.6 Neural Networks . . . 49
3.4 Statistical approaches . . . 50
3.4.1 The Cox Proportional Hazards Model . . . 51
3.4.2 Heart Failure Survival Score . . . 52
3.4.3 Seattle Heart Failure Model . . . 53
3.4.4 ORBIT . . . 53
3.4.5 PARIS . . . 54
3.4.6 PRECISE-DAPT . . . 54
3.5 Discussion . . . 55

4 Traditional Machine Learning and Federated Learning on critical Datasets 59
4.1 Traditional Machine Learning . . . 60

4.1.1 PRAISE Score: A main motivation towards the Federated Learning 60
4.2 Federated Learning on critical Datasets 62
4.2.1 Why Federated Learning on critical Datasets 63
4.2.2 Methods 66
4.2.3 FL with gradient descent 66
4.2.4 FL without gradient descent 67
4.2.5 Experiments 69
4.2.6 Results 70

5 Conclusion 75

6 Future work and directions 79

List of Tables

3.1 Summary of research goals in different research papers. . . . 42
4.1 Summary of statistics used to evaluate prediction performances. 70
4.2 Prediction performance of the FNN with FedAvg. Values reported are the average ± stdev of 5 runs. The first run in the strong scaling setting is equivalent to the non-federated case. 71
4.3 Prediction performance of `AdaBoost.F`. Values reported are the average ± stdev of 5 runs. The first run in the strong scaling setting is equivalent to the non-federated case. 72

List of Figures

2.1 A lifecycle of Federated Learning 19
2.2 A horizontal federated learning system 23
2.3 A vertical federated learning system 24
2.4 A federated transfer learning 25
2.5 The intel OpenFL software stack. 25
3.1 A machine learning process 37
3.2 Machine learning techniques discussed in this chapter 43
3.3 Examples of (a) bias, (b) variance, (c) noise decomposition, and (d) aggregated bias, variance, and noise. The red curved line is the true concept to be approximated, the blue line is the average regressor, the gray lines are individual regressors, and the black dots are noisy observations. As can be seen, these three error components have a massive effect on approximation performance. 44
3.4 The logistic function. 46
3.5 Number of appearances of Machine Learning techniques in the reviewed literature 55
3.6 Number of times each Machine Learning technique ranked first in the reviewed literature. Papers where only a single technique was presented are not included. 56
4.1 Calibration plots of AdaBoost (a) and RF (b) 60
4.2 Praise Score 62
4.3 `AdaBoost.F` and FedAvg training time for 100 rounds executed on the C3S machines in the strong scaling setting. . . 73
4.4 `AdaBoost.F` and FedAvg training time for 100 rounds executed on the C3S machines in the weak scaling setting. . . . 73

Chapter 1

Introduction

Machine learning is a rapidly growing field of computer science focused on developing algorithms and techniques that allow machines to learn from data and make predictions or decisions without being explicitly programmed. ML is a subfield of artificial intelligence(AI) that focuses on developing algorithms and statistical models that enable machines to learn from data without being explicitly programmed. ML is a powerful technology that is being applied across a wide range of industries to solve complex problems, automate processes, and make data-driven decisions[1]. As more data becomes available and ML algorithms continue to improve, the possibilities for this technology are endless. With the increasing amount of data available, ML has become a powerful tool for solving many real-world problems[2, 3]. This technology is increasingly used in various applications such as healthcare, finance, transportation, e-commerce, speech recognition, security, and natural language processing.

Another area where ML is gaining popularity is in natural language processing (NLP)[4].With the help of ML algorithms, machines can understand and generate human language, enabling them to perform tasks such as sentiment analysis, language translation, and chatbot development[5, 6].ML algorithms train the NLP models, allowing them to analyze text, identify patterns, and extract meaning from written or spoken language[7].This technology is used in customer service, marketing, and education industries. ML is also being used to improve the efficiency of manufacturing processes[8]. With the help of predictive analytics, machines can learn to predict equipment failures and optimize production schedules, leading to significant cost savings. ML is used in finance for fraud detection[9], investment management, risk management[10]. By analyzing large amounts of data, machines

can identify fraudulent activity patterns, predict market trends, and identify investment opportunities. ML is also being used to improve transportation systems[11]. ML is also used to optimize city traffic flow, which can help reduce congestion and improve air quality. Self-driving cars rely on ML algorithms to analyze data from sensors and cameras to decide how to navigate roads and avoid obstacles[12]. In e-commerce, ML analyzes customer data to make personalized product recommendations and enhance the shopping experience[13]. Machine learning algorithms can analyze a customer's past purchases and browsing history.

ML has a wide range of applications in the field of medicine. It can diagnose diseases, predict patient outcomes, and identify treatment plans. One of the challenges in using machine learning for medical applications is the need to work with large amounts of sensitive patient data, which can create privacy and security concerns[14, 15]. Federated learning is a technique that addresses these concerns by allowing machine learning models to be trained on decentralized data sources without the need to transfer sensitive data to a central location[16, 17]. In federated learning, the machine learning model is trained locally on each data source, and then the updates to the model are aggregated to create a global model. This approach can help to ensure patient privacy while allowing for effective machine learning in medical applications. The FL healthcare project illustrates the implementation of federated learning in medicine to create a platform for tabular and medical image analysis using federated learning techniques[15]. The initiative investigates the potential of federated learning to evaluate medical information, including forecasting stroke and diabetes risk and examining CT scans and X-rays for detecting illnesses such as COVID-19 and lung cancer[18–20]. Using federated learning, the platform would enable medical organizations to partner and exchange data to train a ML model without compromising patient privacy. Personalized medicine is another instance where ML is utilized in the field of medicine. FL can analyze large amounts of patient data to identify patterns and predict individual patient outcomes. This approach can help doctors to tailor treatment plans for individual patients based on their specific needs and risk factors. Additionally, FL has the potential to transform field of medicine, enabling new insights and personalized treatment plans while preserving patient privacy and security.

The contribution of this dissertation is threefold. First, we studied different federated learning techniques in the literature and tools to better understand the federated learning methodology. Second, we investigated

the cardiology literature for machine learning. We also applied the machine learning techniques to the PRAISE dataset to predict a risk score and achieved excellent results. We are able to develop a tool called PRAISE Score. At last, we used the PRAISE dataset the federated learning using FedAvg and `AdaBoost.F` and We achieved F_1 and F_2 scores which are consistently comparable to the PRAISE score study of a 16-parties federation but within an order of magnitude less time. A comprehensive summary of these contributions can be found in Section 1.1.

This dissertation is the final result of a three-year Ph.D. program, and the author produced 6 research publications in journals and conferences, which form the foundation of this dissertation. The articles cover a variety of subjects, primarily federated learning, machine learning, deep learning (DL), and cardiology. Some of the articles are directly related to the thesis topic. The author's main contribution is detailed in Section 1.1, and a comprehensive list of the author's publications is provided in Section 1.2.

1.1 Main contributions

The first focus of our thesis was to explore the various tools for federated learning. Through our investigation, we discovered that these tools are designed to serve a specific purpose. Furthermore, we conducted a review of recent literature regarding techniques used in federated learning, which enhanced our understanding of the most up-to-date approaches in this field.

Second, we reviewed the traditional machine learning approaches for medicine more. Specifically, we focused on the cardiology risk score assessment. This work found different widely used machine learning and statistical techniques. In our observation, we have discovered that ensemble methods, particularly Random Forests and logistic regression, are the most frequently utilized machine learning tools in recent cardiovascular research studies. Despite being at the forefront of machine learning advancements, neural networks are not as prevalent in this field as expected. We speculate that this may be due, in part, to the challenges in interpreting them and their difficulty in training, requiring larger datasets if not appropriately tuned. Large (federated) datasets and unsupervised techniques are still not used much. Adopting them would significantly improve the current performances of predictions based on ML techniques and pave the way to broader adoption of more sophisticated ML techniques.

Third, we used a cardiology dataset called PRAISE datasets for machine

learning techniques AdaBoost and Random Forest. We achieved good results after tuning the hyperparameter. We built a web-based tool called the PRAISE score calculator, which is online and accessible to everybody. At last, we used the same PRAISE dataset for federated learning in which we compared the two methods `AdaBoost.F` and FedAvg. We study the performance of test accuracy of `AdaBoost.F` (accuracy, F_1, F_2, precision, recall) and scalability in two different execution environments: a cluster of Virtual Machines on an OpenStack cloud and an HPC cluster. We achieved F_1 and F_2 scores which are consistently comparable to the PRAISE score study of a 16-parties federation but within an order of magnitude less time.

1.2 List of publications

This section lists all the author's publications in reverse chronological order. Research work organised in Section 1.2.1 categorises it based on the venue. Among them, J1, J3, C1 and a PRAISE tool T1 are direct results of this dissertation, but all cover topics and use-cases that served as inspirations for the main contributions.

1.2.1 Publications organised by venue

Journal papers

J1 **Y. Arfat**, G. Mittone, R. Esposito, B. Cantalupo, G. M. de Ferrari, and M. Aldinucci, "Machine learning for cardiology," Minerva Cardiology and Angiology 2022 February; 70 (1): 75-91.

J2 G. Agosta, M. Aldinucci, C. Alvarez, R. Ammendola, **Y. Arfat**, O. Beaumont, M. Bernaschi, A. Biagioni, T. Boccali, B. Bramas, C. Brandolese, B. Cantalupo, M. Carrozzo, D. Cattaneo, A. Celestini, M. Celino, I. Colonnelli, P. Cretaro, P. D'Ambra, M. Danelutto, R. Esposito, L. Eyraud-Dubois, A. Filgueras, W. Fornaciari, O. Frezza, A. Galimberti, F. Giacomini, B. Goglin, D. Gregori, A. Guermouche, F. Iannone, M. Kulczewski, F. Lo Cicero, A. Lonardo, A. R. Martinelli, M. Martinelli, X. Martorell, G. Massari, S. Montangero, G. Mittone, R. Namyst, A. Oleksiak, P. Palazzari, F. Reghenzani, C. Rossi, S. Saponara, F. Simula, F. Terraneo, S. Thibault, M. Turisini, P. Vicini, M. Vidal, D. Zoni, and G. Zummo, "Towards extreme scale technologies and accelerators for eurohpc hw/sw supercomputing applications

for exascale: the textarossa approach," Microprocessors and microsystems, vol. 95, p. 104679, 2022. doi:10.1016/j.micpro.2022.104679

J3 F. D'Ascenzo, O. De Filippo, G. Gallone, G. Mittone, M. A. Deriu, M. Iannaccone,A. Ariza-Solé, C. Liebetrau, S. Manzano-Fernández, G. Quadri, T.Kinnaird, G. Campo, J. P. Simao Henriques, J. M. Hughes, A. Dominguez-Rodriguez, M. Aldinucci, U. Morbiducci, G. Patti, S. Raposeiras-Roubin, E.Abu-Assi, G. M. De Ferrari, F. Piroli, A. Saglietto, F. Conrotto, P. Omedé,A. Montefusco, M. Pennone, F. Bruno, P. P. Bocchino, G. Boccuzzi, E. Cerrato,F. Varbella, M. Sperti, S. B. Wilton, L. Velicki, I. Xanthopoulou, A. Cequier,A. Iniguez-Romo, I. Munoz Pousa, M. Cespon Fernandez, B. Caneiro Queija, R.Cobas-Paz, A. Lopez-Cuenca, A. Garay, P. F. Blanco, A. Rognoni, G. Biondi-Zoccai, S. Biscaglia, I. Nunez-Gil, T. Fujii, A. Durante, X. Song, T. Kawaji, D.Alexopoulos, Z. Huczek, J. R. Gonzalez Juanatey, S.-P. Nie, M.-a. Kawashiri,I. Colonnelli, B. Cantalupo, R. Esposito, S. Leonardi, W. Grosso Marra, A.Chieffo, U. Michelucci, D. Piga, M. Malavolta, S. Gili, M. Mennuni, C. Montalto,L. Oltrona Visconti, and **Y. Arfat**, «Machine learning-based prediction ofadverse events following an acute coronary syndrome (PRAISE): a modellingstudy of pooled datasets», The Lancet, vol. 397, no. 10270, pp. 199–207, 2021,issn: 0140-6736. doi: 10.1016/S0140-6736(20)32519-8

Conference papers

C1 **Y. Arfat**, G. Mittone, I. Colonnelli, F. D'Ascenzo, R. Esposito, and M. Aldinucci, "Pooling critical datasets with federated learning," in Proc. of 31st euromicro intl. conference on parallel distributed and network-based processing (pdp), Napoli, Italy, 2023.

C2 I. Colonnelli, B. Casella, G. Mittone, **Y. Arfat**, B. Cantalupo, R. Esposito, A. R. Martinelli, D. Medić, and M. Aldinucci, "Federated learning meets HPC and cloud," in Astrophysics and space science proceedings, Catania, Italy, 2022.

C3 G. Agosta, W. Fornaciari, A. Galimberti, G. Massari, F. Reghenzani, F. Terraneo, D. Zoni, C. Brandolese, M. Celino, F. Iannone, P. Palazzari, G. Zummo, M. Bernaschi, P. D'Ambra, S. Saponara, M. Danelutto, M. Torquati, M. Aldinucci, **Y. Arfat**, B. Cantalupo, I. Colonnelli, R. Esposito, A. R. Martinelli, G. Mittone, O. Beaumont, B. Bramas, L. Eyraud-Dubois, B. Goglin, A. Guermouche, R. Namyst, S. Thibault,

A. Filgueras, M. Vidal, C. Alvarez, X. Martorell, A. Oleksiak, M. Kulczewski, A. Lonardo, P. Vicini, F. L. Cicero, F. Simula, A. Biagioni, P. Cretaro, O. Frezza, P. S. Paolucci, M. Turisini, F. Giacomini, T. Boccali, S. Montangero, and R. Ammendola, "TEXTAROSSA: towards extreme scale technologies and accelerators for eurohpc hw/sw supercomputing applications for exascale," in Proc. of the 24th euromicro conference on digital system design (DSD), Palermo, Italy, 2021. doi:10.1109/DSD53832.2021.00051

1.2.2 Developed Tools

The PRAISE score calculator is an application used in the field of cardiology. It is a tool specifically designed to assess the risk of adverse cardiovascular events in patients who undergo percutaneous coronary intervention (PCI), which is a common procedure to treat coronary artery disease. The PRAISE score considers various clinical factors, such as age, sex, comorbidities, and procedural characteristics, to provide an estimation of the patient's risk profile. It assists healthcare professionals in making informed decisions regarding the management and treatment of patients undergoing PCI.

T1 Praise Score: https://praise.hpc4ai.it/

Chapter 2

Background on Federated Learning

The most well-known federated learning tools will be covered in this chapter, along with the most recent federated learning methodology research. However, in our thesis, we will leverage OpenFL [21] as the base framework for FL, but over the past few years, there have been numerous attempts to apply federated learning (FL) technology to healthcare and other industries. These efforts, which range from open-source projects like TensorFlow Federated [22] and PySyft [23] to commercial offerings like IBM® Federated Learning [24] and HP Swarm Learning [25], have aimed to address the needs of researchers and practitioners in a variety of settings.

However, some of these efforts have focused on simulated environments for research purposes, while others have been designed specifically for production use cases. Other notable FL projects in the healthcare and other industries include FedML [26], FATE [27], Flower [28], Fed-BioMed [29], FederatedScope [30], FLUTE [31], and FLARE [32]. Each tool has unique qualities depending on how it will be used. The section 2.1 covers these tools in more detail.

2.1 Federated Learning

Google introduced the phrase "Federated Learning" in 2016 to describe a machine learning environment in which numerous entities known as clients (such as mobile devices or entire enterprises) work together to jointly train a model while maintaining different machines for training data. Each client's raw data is kept locally rather than being transmitted or transported to

fulfill the learning aim; instead, focused updates designed for quick aggregation are employed. Introducing an imbalanced and non-IID (identically and independently distributed) data partitioning over a sizable number of unreliable devices with constrained connection bandwidth served as the defining set of challenges. The machine learning process is distributed to the edge using federated learning.

While the training data is kept on the device, it enables clients to learn a shared model collectively. It distinguishes between the requirements for machine learning and cloud data storage. A client downloads the common model in a federated learning computing system. The model is then trained locally, and it is subsequently improved by using locally stored data for learning. The changes are then briefly updated, often with the model parameters and associated weights. Since the term "federated learning" was first used to describe applications for mobile and edge devices, interest in adapting FL to other applications, including ones that may only involve a limited number of consistently dependable clients, such as numerous businesses cooperating to train a model, has significantly expanded. Figure 2.1 shows a typically federated learning process, for example, if a user's phone personalizes the model locally based on her usage (A). Many users' updates are then aggregated (B). The global update is shared with the clients (C).

2.2 Distribution types of Federated Learning

In the first section 2.1, we established a precise definition of federated learning and examined its connection to the current challenges in the field of learning. Distinctions between data center distributed learning, cross-silo federated learning, and cross-device federated learning were presented with regard to network topology and entities. Each type of federated learning entails a unique approach and set of conditions that influence task coordination. Moving forward, we will delve into the specifics of each federated learning category.

2.2.1 Datacenter distributed learning

Datacenter distributed learning is a promising approach for training machine learning models in distributed environments. It can be applied to various use cases, such as healthcare, finance, and IoT. In this approach, the training of machine learning models using data from multiple sources, often located in different locations or even different organizations. In this

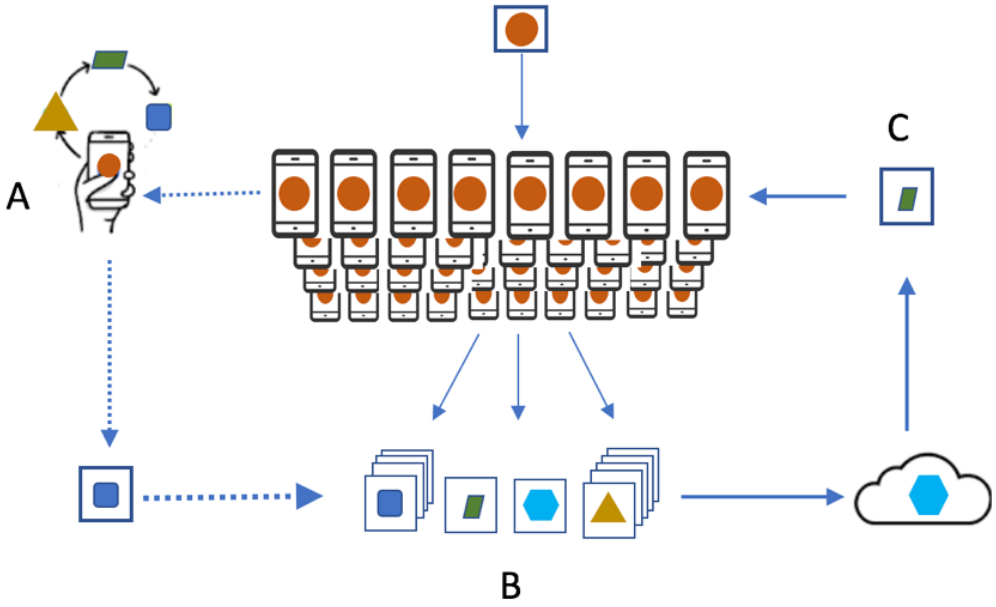


Figure 2.1: A lifecycle of Federated Learning

technique, each data source (or "node") trains its model using its data and then shares its model updates with other nodes. These updates are then combined to create a global model that is more accurate and robust than any single node's model.

This approach has several advantages over traditional centralized learning methods, where all data is collected and processed in a single location. For example, datacenter distributed learning can:

- Improve privacy and security, as data never needs to leave the control of its original owner
- Handle large and diverse data sets, as each node can have its own data and model
- Reduce communication and storage costs, as only model updates need to be shared, not the entire dataset

However, datacenter distributed learning also poses some challenges, such as:

- Handling data heterogeneity and bias across nodes
- Ensuring data security and privacy during model updates

- Managing the coordination and communication between nodes

2.2.2 Cross-silo federated learning

A model is trained using siloed data in a cross-silo federated learning environment. Clients can be various businesses (such as financial or medical institutions) or geographically dispersed data centers. Locally generated data is still decentralized. Each client keeps its own data in storage; other clients' data cannot be read. Data is not delivered uniformly or independently. The training is organized by a central server or service that never sees the raw data. Hub-and-spoke structure, with the spokes connecting to the clients and the hub indicating a coordinating service provider (usually without data). From 2 to 100 clients, all are almost always accessible. Communication or computation could be a bottleneck. Every client has a name or identity that the system can access. Since clients carry state from round to round and are stateful, they can participate in each compute round with minimal failures. The data division has been rectified. This approach is beneficial for the following reasons:

- **Improved data privacy**: Federated learning allows for training models on decentralized data, reducing the need for data to be centralized and shared among multiple organizations.
- **Increased data diversity**: The resulting models can be more robust and better generalized to diverse populations by training models on data from multiple organizations.
- **Reduced data labeling costs**: Federated learning allows for the sharing of labeled data among organizations, reducing the need for each organization to label its data individually.

There are many situations when this approach is not useful as:

- **Complex coordination**: Coordinating the training of models across multiple organizations can be complex and time-consuming.
- **Data distribution differences**: The data distribution across different organizations may not be the same, leading to potential bias in the resulting models.
- **Limited scalability**: Federated learning may not be practical for large-scale datasets or organizations with limited computational resources.

2.2.3 Cross-device federated learning

Many mobile or IoT devices have clients for cross-device federated learning. Locally generated data is still decentralized. Each client stores its own data; other clients' data cannot be read. Data is not delivered uniformly or independently. The training is organized by a central server or service that never sees the raw data. A hub-and-spoke structure, where the spokes link to clients and the hub represents a coordinating service provider (usually without data). At any given time, only a small portion of clients are accessible, frequently with daily or other fluctuations. Depending on the task, communication is typically the main bottleneck.

Cross-device federated computations typically employ slower connections or wi-fi. Direct client indexing is impossible (i.e., no use of client identifiers). Because clients are stateless and are expected to participate in a job just once, it is commonly assumed that each round of computation will use a new sample of clients who have never been seen before. This approach is extremely unreliable; it is estimated that 5 percent or more of the clients taking part in a computing round may fail or drop out (e.g., because the device becomes ineligible when the battery, network, or idleness requirements are violated). The data partition cannot be changed and can only be partitioned as an example (horizontal). This technique is beneficial:

- **Better model performance**: By combining data from multiple devices, cross-device federated learning can improve the performance of machine learning models by providing a more extensive and diverse dataset to train on.
- **Cost savings**: Cross-device federated learning eliminates the need for centralized data storage, which can save costs associated with server infrastructure and data storage.
- **Improved data security**: By not sharing personal data, cross-device federated learning reduces the risk of data breaches and other security threats.
- **Real-time updates**: With cross-device federated learning, models can be updated in real-time, allowing for more accurate predictions and better decision-making.
- **Improved user engagement**: By allowing users to contribute their data to the training process, cross-device federated learning can increase user engagement and participation.

However, cross-device federated learning also poses some challenges. As these challenges are:

- **Device compatibility issues**: Not all devices may be compatible with the federated learning system, which can limit the number of devices that can participate in the learning process.
- **Network connectivity issues**: Cross-device federated learning requires a stable and fast internet connection, which can be problematic in areas with the poor network coverage.
- **Complex implementation**: Implementing a cross-device federated learning system requires significant technical expertise and resources.
- **Limited scalability**: Cross-device federated learning may not be able to scale to large numbers of devices, making it less suitable for specific use cases.
- **Limited control over data**: Cross-device federated learning relies on data from other devices, making it difficult to control the quality and accuracy of the data used in the learning process.
- **Security risks**: Cross-device federated learning can pose security risks, as sensitive data may be shared between devices, which hackers or other malicious actors could compromise.

2.3 Categorization of federated learning

We can classify the federated learning techniques into different categories based on the data partitioning features. We categorize federated learning as horizontal federated learning, vertical federated learning, and federated transfer learning depending on how data is dispersed across multiple parties in the feature and sample ID space. The features and sample space of the data parties may not be identical.

2.3.1 Horizontal federated learning

In situations where datasets have the same feature space but vary in samples, horizontal federated learning (also called sample-based federated learning) is employed. This approach is illustrated in Figure 2.2 and was first introduced by [14]. An example scenario could involve two regional banks with different user groups from their respective regions but with comparable

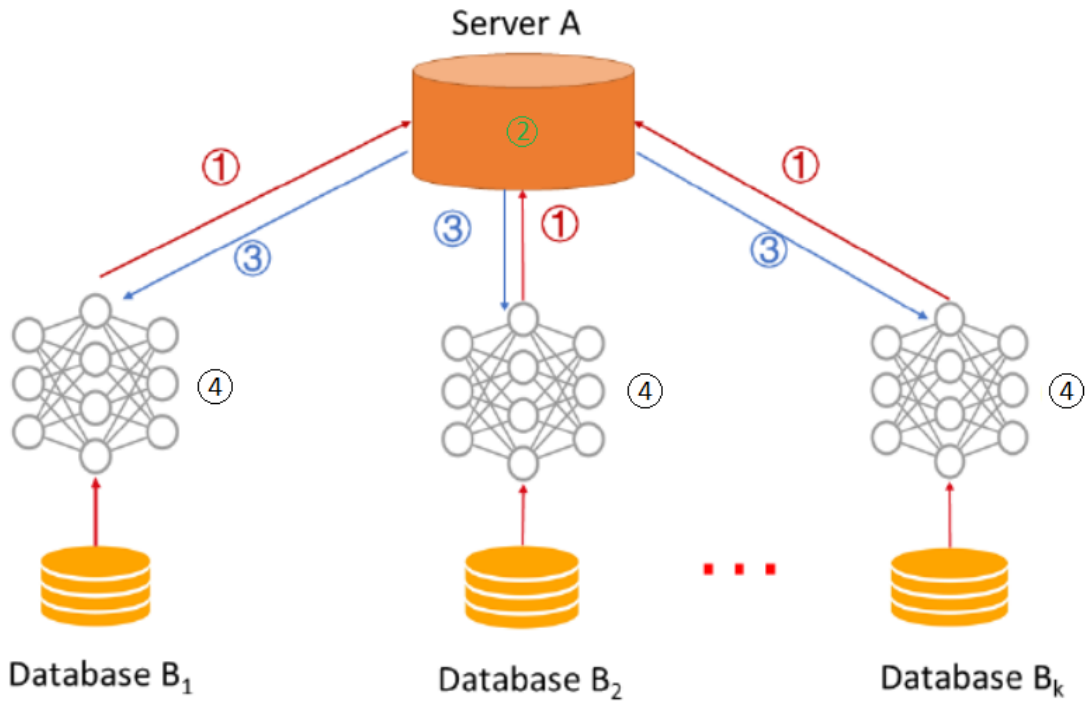


Figure 2.2: A horizontal federated learning system

business operations and feature spaces. In this system, a group of k participants with identical data structures collaboratively learn a machine learning model utilizing the aid of a parameter or cloud server. It is generally assumed that the participants are trustworthy, while the server is honest but inquisitive, which means that no data leakage from any participants to the server is permissible.

2.3.2 Vertical federated learning

Vertical federated learning, referred to as feature-based federated learning, is a technique used when datasets share the same sample ID space but differ in feature space. As illustrated in Figure 2.3 and described by [14], this approach can be applied to scenarios such as two different companies in the same city, one being a bank and the other an e-commerce company. Their user bases are likely to include most of the residents of the area, resulting in a large intersection of their user space. However, since the bank records the user's revenue, expenditure behavior, and credit rating, while the e-commerce company retains the user's browsing and purchasing

history, their feature spaces are considerably different.

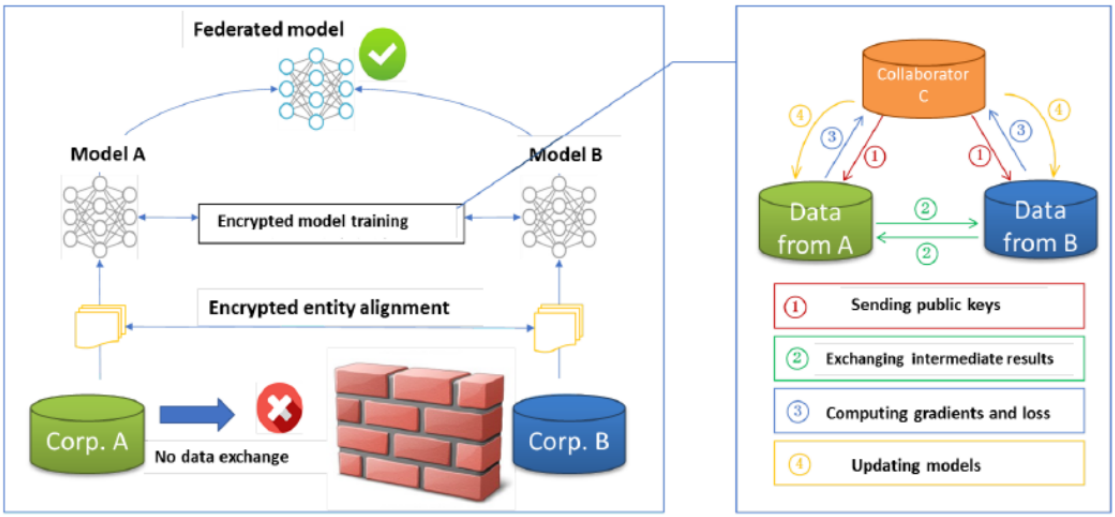


Figure 2.3: A vertical federated learning system

2.3.3 Federated transfer learning

Federated transfer learning is applicable when datasets differ not only in their samples but also in their feature spaces. Figure 2.4 depicts an example of federated transfer learning [14]. Suppose there are two institutions: a bank in China and an e-commerce company in the United States. The user groups of the two institutions have limited intersections due to geographical restrictions. Additionally, only a small fraction of the feature space overlaps because of their distinct businesses.

2.4 Federated Learning Tools

In this section, we will enumerate and examine all the tools for federated learning.

2.4.1 The intel OpenFL® framework

OpenFL® [21] is an open-source tool for cross-silo FL based on Python 3, designed to be flexible, extensible, community-driven, and easy to learn for data scientists. The potential of OpenFL has already been showcased in [33], where the tool was used to build the world's largest federation to date. Figure 2.5 shows an overview of the OpenFL architecture. Note

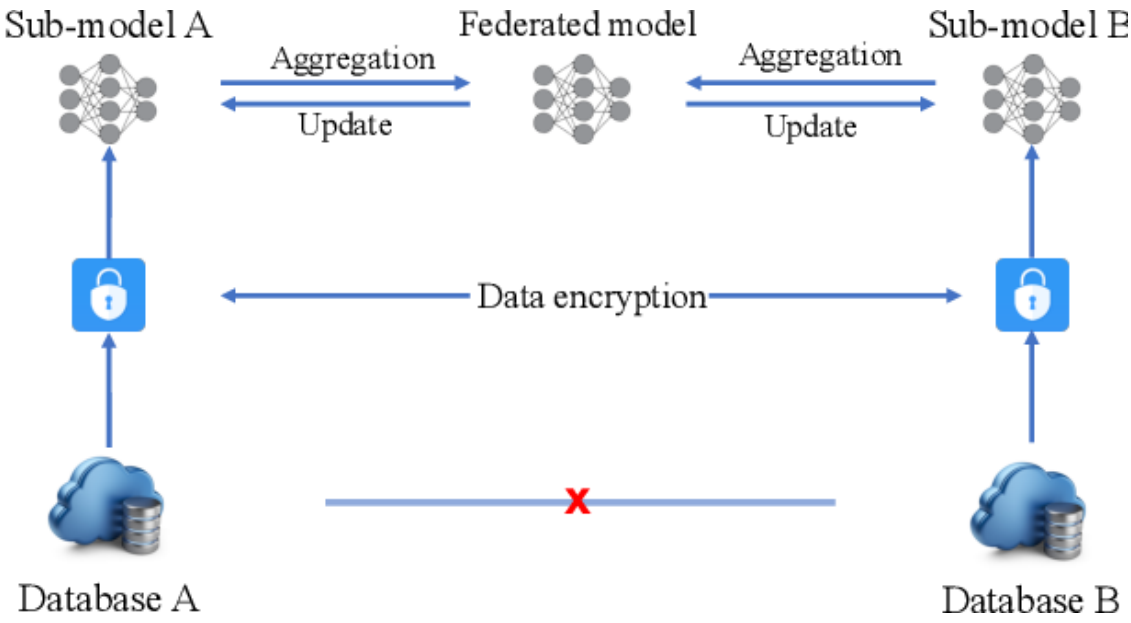


Figure 2.4: A federated transfer learning

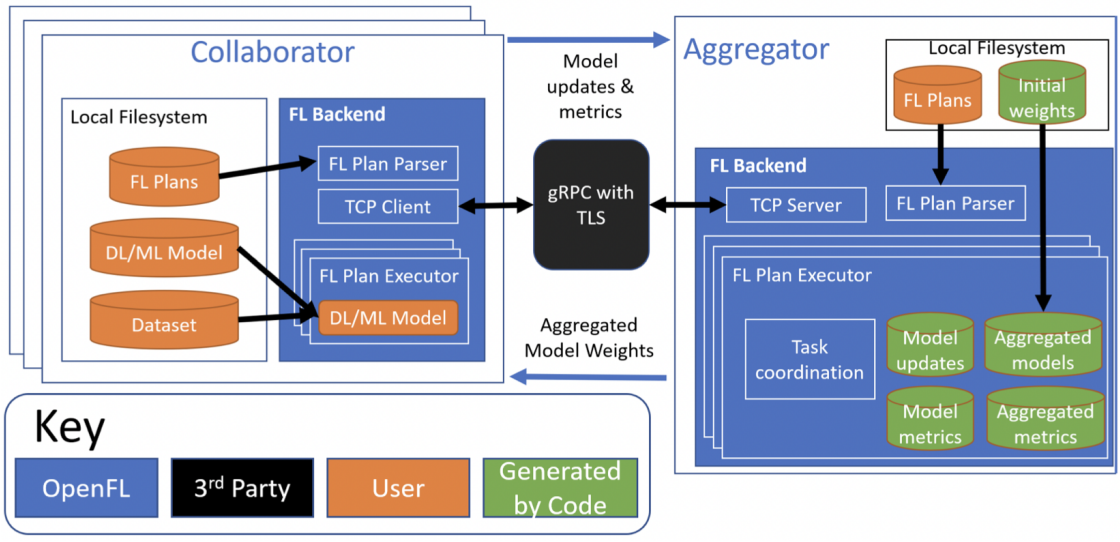


Figure 2.5: The intel OpenFL software stack.

that the vanilla OpenFL only supports neural networks, as most of the FL frameworks in the market. Our re-engineering effort to support `AdaBoost.F` targeted specifically the software component coloured in blue and marginally the orange ones. This way, the user interaction with the framework is only marginally affected. A detailed description of the re-engineering effort, including design choices and implementation details is provided in [136].

There are two key actors in a federation. Each *collaborator* accesses its own data to train a replica of the ML model. A central *aggregator* collects and merges the updates produced by each collaborator. While collaborators live at the edges of the federation, the aggregator usually runs on a data centre or in the cloud. This is necessary due to its high network usage and its reachability and reliability requirements.

2.4.2 TensorFlow Federated

TensorFlow Federated (TFF)[22] is an open-source framework developed by Google that enables implementing of federated learning algorithms. It provides tools and APIs for writing distributed machine learning programs that can run on various devices, such as smartphones, IoT devices, and other edge devices. TFF includes a set of libraries and tools for defining and training federated models and evaluating their performance. It allows developers to build and deploy federated learning models using TensorFlow, a popular machine learning library. TFF also enables customized federated learning algorithms and offers an execution platform for simulating and deploying federated learning systems. It also supports secure aggregation, differential privacy, and other advanced techniques for federated learning.

2.4.3 PySyft

PySyft[23] is a Python library for secure, privacy-preserving ML. It is an open-source framework built on top of PyTorch, a popular deep learning framework. PySyft provides tools and abstractions for building and training ML models using distributed computing while maintaining privacy and security. The main goal of PySyft is to enable machine learning researchers and practitioners to collaborate on private and sensitive data while ensuring that data remains secure and confidential. PySyft supports privacy-preserving techniques such as federated learning, homomorphic encryption, and differential privacy.

PySyft is a powerful tool for building privacy-preserving ML applications in various domains, including healthcare, finance, and government. It is an active research area, and the community continuously develops new techniques and applications for secure ML. PySyft features the Paillier cryptosystem for partially homomorphic encryption and noise addition tools to achieve differential privacy. Its user-friendly design includes a simple API, similar to PyTorch, and a variety of tutorials and examples to facilitate the use of federated learning, homomorphic encryption, and differential privacy.

2.4.4 IBM® federated Learning

IBM® Federated Learning[24] is a framework developed by IBM that allows multiple parties to collaborate on a machine learning project without sharing their data, using a central server to coordinate the training process. In contrast, each party keeps its data on their own systems. The server sends updates to each party, which they use to update their models. The models are then combined to create a final model for all parties. The IBM Federated Learning framework provides a secure and privacy-preserving approach to training machine learning models by keeping the data local and using encryption and other security measures to protect the privacy of the data. The framework provides the following features:

- **Privacy-preserving data aggregation** In federated learning, the model updates from each device are aggregated to create a final, centralized model. The IBM Federated Learning framework uses advanced cryptography techniques, such as differential privacy and secure multiparty computation, to ensure that the aggregated model updates do not reveal sensitive information about each device's local data.

- **Distributed model training** The IBM Federated Learning framework enables organizations to train machine learning models on data distributed across multiple devices or locations. This reduces the need for data transfer and enables organizations to take advantage of the benefits of distributed data while also ensuring that sensitive data is not exposed.

- **Client-side model deployment** Once the model has been trained, the IBM Federated Learning framework enables organizations to deploy the model directly to the client devices. This reduces the need for data transfer and enables organizations to take advantage of the benefits of distributed data while also ensuring that sensitive data is not exposed.

- **Customizable model training** The IBM Federated Learning framework enables organizations to customize the training process by specifying the type of model to be trained, the data to be used for training, and the hyperparameters of the model.

The IBM Federated Learning framework consists of several components, including a federated learning server, client SDKs for different platforms,

and tools for managing and monitoring the training process. The server coordinates the training process and aggregates the model updates from the client devices. In contrast, the client SDKs provide the necessary libraries and APIs for integrating the framework into various applications. IBM Federated Learning framework offers a powerful solution for organizations that need to train machine learning models on distributed data while maintaining data privacy and security. By enabling organizations to keep their data local and secure, the framework makes it possible to take advantage of the benefits of distributed data without compromising data privacy and security.

2.4.5 HPE Swarm Learning

HPE Swarm Learning[25] is a decentralized, privacy-preserving framework for performing machine learning model training at the data source. It allows multiple devices to collaborate and learn from each other to improve the accuracy of predictions and enhance the overall efficiency of the learning process. Swarm Learning distributes the learning process across multiple devices, such as edge devices or IoT devices, where each device has its own data set. The devices collaborate by sharing model updates and communicating with each other to improve the accuracy of predictions. One of the critical advantages of Swarm Learning is that it enables privacy-preserving machine learning. Instead of collecting and transferring large amounts of data to a centralized location, the learning algorithm is executed locally on the devices. This approach protects the privacy of the data and reduces the risk of data breaches.

HPE Swarm Learning uses a combination of federated learning, differential privacy, and blockchain technology to ensure the security and privacy of the learning process. Federated learning allows devices to collaborate without sharing data directly, differential privacy provides a way to add noise to data to prevent the reconstruction of sensitive information, and blockchain technology is used to ensure the integrity of the model updates and prevent unauthorized modifications. HPE Swarm Learning presents a compelling method for decentralized machine learning that can improve the accuracy of predictions while preserving privacy and security.

2.4.6 FATE

Federated AI Technology Enabler(FATE)[27] Framework is an open-source platform that enables the development, deployment, and management of

federated learning (FL) models. It provides a set of libraries, tools, and services that allow data scientists and engineers to build FL models securely and efficiently. The FATE Framework consists of several components, including:

- **Data Management**: This component allows for data management, storage, and privacy. It supports various data sources like databases, files, and streams.
- **Federated Learning**: This component provides a set of libraries and tools to implement FL algorithms, such as federated averaging and federated optimization.
- **Model Management**: This component allows for model management, including model training, deployment, and versioning.
- **Security and Privacy**: This component provides data and model encryption, secure communication, and data privacy.
- **Monitoring and Auditing**: This component allows for monitoring and auditing of FL models, including data quality and model performance.

  In summary, the FATE Framework simplifies the process of building FL models and allows for secure and efficient deployment and management of these models.

2.4.7 FedML

Federated Machine Learning Framework(FedML)[26] is an open-source machine learning framework designed to support federated learning. It provides a unified interface for federated learning on various data types and architectures. In FedML, the data is kept on the individual parties' systems and is not shared with others. Instead, the parties use a central server to coordinate the training process, sending updates to each other and combining their models to create a final model that all parties can use. FedML has the potential to enable companies to work together on machine learning projects without compromising the privacy of their data. FedML provides the following features:

- **A flexible data pipeline**: FedML allows users to define their own data pipeline, which enables using various data types, such as images, text, and structured data.

- **Support for various models**: FedML supports many models, including deep learning models, linear models, and decision trees.
- **Support for various federated learning protocols**: FedML supports a wide range of federated learning protocols, such as federated averaging and federated stochastic gradient descent.
- **Easy integration**: FedML is designed to be easily integrated with other machine learning frameworks, such as TensorFlow and PyTorch.
- **Scalability**: FedML is designed to scale to large numbers of devices, which allows for large-scale federated learning experiments.
- **Security**: FedML provides security features, such as secure aggregation and differential privacy, to protect user privacy.

2.4.8 Flower

The Flower [34] framework is an open-source platform for federated learning that enables the training of machine learning models on decentralized data. It provides a high-level programming interface and a set of tools to simplify the process of building and deploying federated learning models. One of the key features of Flower is its ability to support a variety of FL architectures, including both homogeneous and heterogeneous FL. Homogeneous FL involves training the same model across multiple devices with similar data distributions, while heterogeneous FL involves training different models on different devices with varying data distributions.

Flower also supports various optimization algorithms, including stochastic gradient descent (SGD), a popular optimization algorithm used in machine learning. Another feature of Flower is its ability to provide a decentralized approach to training models. Rather than relying on a centralized server for training, Flower allows for the distributed training of models across multiple devices. This approach helps to preserve data privacy by keeping it on the device where it was generated. Flower also offers security features to protect the privacy and confidentiality of data. For example, it uses encryption and authentication techniques to ensure data is transmitted securely and only authorized parties can access it. The Flower framework provides a user-friendly and secure platform for federated learning that developers can use to quickly build decentralized machine learning models.

2.4.9 Fed-BioMed

Federated Biomedical Learning(Fed-BioMed)[35] refers to the use of federated learning techniques in the field of biomedical research. Federated learning involves the collaboration of multiple parties, each with their own data, to train a shared model without sharing the data itself. It can be helpful in the biomedical field, where data privacy is a significant concern and data sharing is often limited due to ethical, legal, and regulatory considerations. Fed-BioMed can enable researchers to work together on machine learning projects and analyze large datasets without compromising their data privacy. It has the potential to improve the accuracy and efficiency of biomedical research and enable the development of new treatments and therapies.

2.4.10 FederatedScope

FederatedScope[30] is a tool for federated learning developed by the Data Science team at WeBank, the digital banking arm of Tencent. It is an open-source platform that allows multiple parties to collaborate on a machine learning project without sharing their data. FederatedScope allows parties to train a shared model using their own data and send updates to a central server, which combines the updates to create a final model. It is designed to be easy to use, with a simple API and a web-based GUI for monitoring and managing the federated learning process. FederatedScope can be used for a variety of machine learning tasks, including classification, regression, and recommendation systems. It is designed to work with both PyTorch, TensorFlow and supports various algorithms and architectures.

2.4.11 FLUTE

Federated Learning Under Threat of Eviction(FLUTE)[31] is a federated learning framework developed by researchers at Carnegie Mellon University. It is designed to resist the attacks that attempt to manipulate the model or prevent certain parties from participating in the federated learning process. FLUTE achieves this by using an optimization algorithm that maximizes the model's performance while ensuring that all parties can contribute to the training process. It also includes a "threat of eviction" mechanism that prevents malicious parties from disrupting the training process by threatening to remove them from the federated learning network. FLUTE effectively improves the resilience of federated learning systems to attacks and improves

the final model's performance.

2.4.12 FLARE

Federated Learning with Adaptive and Resilient Execution(FLARE)[32] is a federated learning framework developed by researchers at Carnegie Mellon University. It is designed to be resilient to attacks that attempt to manipulate the model or prevent certain parties from participating in the federated learning process. FLARE achieves this by using an adaptive optimization algorithm that adjusts the training process in response to network changes and malicious parties' presence. It also includes a mechanism for detecting and mitigating attacks and a "threat of eviction" mechanism that prevents malicious parties from disrupting the training process by threatening to remove them from the federated learning network. FLARE is effective at improving the resilience and performance of federated learning systems.

2.5 Literature on Federated Learning

Federated Learning (FL) has been proposed by McMahan et al. [16] as a way to develop better AI systems without compromising the privacy of final users and the legitimate interests of private companies. Initially deployed by Google for predicting text input on mobile devices, FL has been adopted by many other industries, such as mechanical engineering and health care [36]. Since then, FL has seen a growing interest from the research community, which has identified a few different and interesting settings.

In cross-device FL, the parties can be edge devices (e.g., smart devices and laptops); they can be numerous (order of thousands or even millions). Parties are considered not reliable and with limited computational power. To name a few examples, cross-device FL setting has been adopted in [37] for training a language model for next-word prediction in a virtual keyboard for smartphones; in [38] it has been used to predict emojis (again on a mobile keyboard), or combined with [39] for learning models to be used with IoT devices.

In the cross-silo FL setting, the involved parties are instead organizations; the number of parties is limited, usually in the range [2, 100]. Given the nature of the parties, it can also be assumed that communication and computation are no real bottlenecks. Cross-silo FL settings have been adopted for [40] investigating brain structural relationships across diseases and clinical cohorts; it has also been used for optimizing production through soybean

yield prediction [41] or, combined with differential privacy and secure multi-party computation, for attacking important financial tasks such as optimal trade execution, credit origination, or fraud detection [42].

Another important distinction in FL is the way data are distributed between clients. Based on this assumption, it is customary to distinguish between *horizontal* and *vertical* FL. The most used assumption is the *horizontal* data distribution. In this setting, the data is partitioned horizontally, i.e., each client owns a subset of the rows of the total dataset. In contrast, in vertical FL [43], one assumes that the rows are shared between the parties, but each client has a different view of the data (i.e., a different set of features for describing the rows). Vertical FL is very appealing in cases where the objects in the datasets overlap a lot, but their descriptions have little overlap. It has been applied to several interesting tasks ranging from 5G communications [44] and proposed as a way to improve small and medium enterprises' credit ratings [43].

The Artificial neural network(ANNs) and Deep neural network(DNNs) are rarely used for prediction tasks involving health care represented in tabular form, and in the few cases where they are applied, they do not show performances that are better than traditional ML approaches [45, 46]. Even though they have the potential to perform as well if not better than other approaches, they are hard to tune, and this makes it difficult for their usage by healthcare institutions. An alternative to ANN/DNN-based FL has been recently proposed in [47]. This alternative, based on the Extreme Gradient Boosting (XGB) ensemble algorithm, is still based on gradient descent, but it allows one to train decision tree models locally. Interestingly, XGB has been used several times in the medical literature on tabular health care datasets [48–51] showing promising results. While XGB-based techniques address some of the problems outlined above, they require the clients to adopt specific learning algorithms (usually decision trees) and are thus inflexible. The work presented in [52] introduces three FL adaptations of the AdaBoost ensemble algorithm that work without exchanging gradients between the aggregator and the clients. The approach allows parties to train any kind of model locally (in principle, even different models on each site), thus overcoming the main inflexibility of the XGB approach.

2.5.1 Strong scalability in federated learning

Strong scaling[53] also known as speedup scaling(Amdahl's law), is a measure of the improvement in execution time when the problem size is fixed, and the number of processing units (such as CPUs or GPUs) is increased. In strong scaling, the goal is to determine how much faster a parallel algorithm or system can solve a fixed-size problem as more processors are added.

Strong scaling in federated learning refers to the scenario where the number of client devices or participants in the federated learning system increases while keeping the total amount of data fixed. The goal of strong scaling in federated learning is to reduce the training time or increase the model's accuracy while maintaining the same amount of data per client[54][16]. With strong scaling in federated learning, the challenge lies in effectively coordinating and aggregating the local model updates from the increased number of client devices. Communication overhead and network bandwidth limitations can become significant factors affecting strong scaling performance[16]. The system should efficiently handle the increased volume of updates, synchronize the models, and ensure convergence while maintaining data privacy and security[54][22].

2.5.2 Weak scalability in federated learning

In weak scaling[53] both the problem size and the number of processing units are increased in proportion. Gustafson's law governs weak scaling and states that as the problem size grows, the relative time spent on parallelizable portions of the computation increases, leading to higher overall performance with more processors.

Weak scaling in federated learning involves increasing both the number of client devices and the total amount of data, while maintaining a fixed data distribution per client. In this scenario, the federated learning system aims to handle larger-scale datasets while preserving the same per-client data distribution and ensuring comparable training quality[54]. In weak scaling, the federated learning system needs to handle increased data volumes and computational requirements. It should scale horizontally to accommodate more client devices, distribute the computational workload effectively, and maintain the same data distribution across clients. Handling the increased data volume without overwhelming the network bandwidth and computational resources becomes crucial in weak scaling[54].

Chapter 3

Machine learning for Cardiology

This chapter reviews recent cardiology literature and reports how Artificial Intelligence (AI) Tools (specifically, Machine Learning techniques) are being used by physicians in the field. Each technique is introduced with enough details to allow the understanding of how it works and its intent, but without delving into details that do not add immediate benefits and require expertise in the field. We specifically focus on the principal Machine Learning based risk scores used in cardiovascular research. After introducing them and summarizing their assumptions and biases, we discuss their merits and shortcomings. Based on our expertise in Machine Learning, we report on how frequently they are adopted in the field and suggest why this is the case. We complete the analysis by reviewing how corresponding statistical approaches compare with them.

Finally, we discuss the main open issues in applying Machine Learning tools to cardiology tasks, also drafting possible future directions. Despite the growing interest in these tools, we argue that there are many still underutilized techniques: while Neural Networks are slowly being incorporated into cardiovascular research, other important techniques, such as Semi-Supervised Learning and Federated Learning are still underutilized. The former would allow practitioners to harness the information contained in large datasets that are only partially labeled, while the latter would foster collaboration between institutions allowing building larger and better models.

Figure 3.1 gives an overview of the chapter structure, describing in particular Sections 3.2 and 3.3, summarising the usual process followed by

ML practitioners: the data are first preprocessed to improve their quality (removing missing values, performing feature selection, ...), then the ML algorithm is trained on them. The model obtained as output is iteratively refined by searching for optimal (hyper) parameters by performing experiments on the training or the validation data. Finally, the model is evaluated on the test data and deployed for usage. After that, Section 3.4 deals with statistical methods, and Section 3.5 discusses our findings.

3.1 Machine Learning: A background and importance

Recent years have witnessed a Cambrian explosion of tools and techniques able to tackle problems that were only solvable by humans up to a few years ago; collectively, we refer to these computer science methods as AI. AI is accumulating astounding successes at a breakneck pace in both research and applications: from helping in recovering photos by their descriptions[55] on devices used by billions of people to providing tools for investigating the depths of the visible universe[56], AI has never been as capable and popular as today. AI encompasses a vast variety of different techniques: intelligent agents[57], symbolic and subsymbolic reasonings[58], planning[59], case-based reasoning[60], fuzzy systems[61], and expert systems[62] are just a few of them. Despite this diversity, one sub-field in AI single-handedly provided the tools that allowed most of the mentioned successes to be achieved: Machine Learning (ML).

In this chapter, we review some of the recent cardiology literature and report on how ML tools are being used by medical doctors and scientists in the complex tasks of understanding and predicting patients' clinical situations. AI, specifically ML, can provide clinicians with powerful tools supporting and helping everyday crucial clinical decisions[63–65]. For this, the exploitation of AI in medicine is a research direction actively endorsed by national and European funding bodies. The 15M€ EU IA "DeepHealth"[66] (Deep-Learning and HPC to Boost Biomedical Applications for Health, 2019-22) and 6M€ EU RIA "Brainteaser" (BRinging Artificial INTelligencE home for a better cAre of amyotrophic lateral sclerosis and multiple SclERosis, 2021-24) projects are just two recent examples of multi-disciplinary projects directly addressing the development of novel ML tools for AI-assisted diagnosis through medical imaging. With such a great deal of investments and with the renewed interest in the field, there are good chances that AI techniques could become crucial tools to assist clinicians in accurately assessing

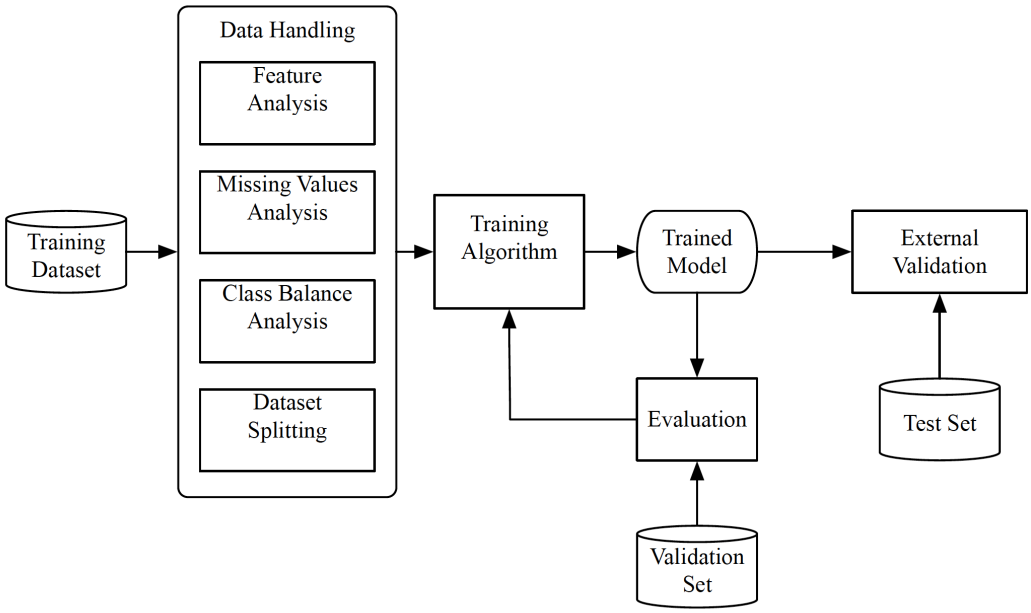


Figure 3.1: A machine learning process

all the relevant factors leading to a diagnosis and the actions that follow. In this context, physicians will remain central to all decisions but supported by tools tailored to ease some of the burdens they face when dealing with the complexity of their work.

ML encompasses all AI approaches specifically oriented towards building models that improve their performances on a given task with experience, that is, data; it is a vast research field able to tackle many tasks and includes many techniques. One broad way to categorize such techniques is by looking at the kind of supervision the learning algorithm receives with the learning examples. The main distinction here is between supervised learning (where all examples are associated with a label), unsupervised learning (where none of the examples is associated with a label), and semi-supervised learning (where only a few of the examples are labeled).

The topic is addressed from a technical perspective, introducing criteria to compare the different techniques, explaining them, and critically reviewing their pros and cons. We describe and review the most crucial risk scores based on ML techniques, giving the reader a comprehensive perspective on the AI applications currently available in the cardiovascular (CV) field. We

also briefly analyze the main statistical approaches, comparing them with ML methods.

In this chapter, our primary goals are to address the current knowledge in the field of AI-based cardiological risk scores and offer new perspectives on their development. The latest papers were collected reporting reviews and comparisons of current methodologies from Google Scholar, covering a wide range of works in computer science and cardiology communities. From there, we proceeded backward, exploring the main literature strands focusing specifically on supervised learning. As the reader shall see, only supervised techniques are reviewed. While we initially set out to include unsupervised and semi-supervised methods in our review, we realized that researchers in the CV field are not currently exploiting these techniques. We comment on this aspect in the final discussion (see section 3.5).

3.2 Data handling

In the supervised learning scenario, data comes from a labeled dataset X=(x_i,y_i) where examples (a.k.a. samples) x_i are associated with labels y_i and are assumed to be i.i.d. (independent and identically distributed). This section discusses widespread data preprocessing techniques to overcome common issues like outliers, missing values, noisy readings, and many others that often affect learning performance.

3.2.1 Features types

In the medical scenario, the samples x_i usually describe patients' data and are structured into several fields known as features in the ML community. Features can take many forms, but as far as most learning algorithms are concerned, they can be subdivided into three different categories[67]:

- **Quantitative**: those with a meaningful numerical scale;
- **Ordinal**: ordered features without a scale;
- **Categorical**: those without an ordering or scale.

The feature type is crucial to ML algorithms since not all algorithms can deal with all kinds of features, and even when they can, they usually handle them differently. Some feature types are more informative than others: quantitative features contain more details than ordinal ones, and the same relationship holds between ordinal and categorical features. The empirical

impact of this statement is present in many of the papers included in this review: risk scores obtained by reducing the number of used features often end up using more quantitative and ordinal features than categorical ones[68–78].

3.2.2 Missing values

An aspect that is important to discuss further is the handling of missing values. Many different approaches exist to deal with this problem, relying on and exploiting different assumptions on the meaning of a missing value. In the medical field, the absence of a value can have a significant clinical meaning; if some values are not collected, there could be some specific reason[79] (e.g., the medical treatment prevents data from being collected, or some values are derived from others). In those cases, expert intervention is needed to understand how to handle the issue correctly. For some models, like Decision Trees, a correct approach to address this issue can be creating a specific value for missing data, signifying that data could not be collected, and giving more information to the model than the simple missing value. If data is not missing for a specific reason, imputation can be exploited to guess its value; this is a powerful technique capable of enhancing the richness of information in a dataset, but it should be carefully handled since it can drastically reduce the data variance. Imputation is frequently exploited in the field[68–70] [75],[80–82], especially employing Monte Carlo or regression methods. Some works explicitly targeting the imputation of medical data are also available[83, 84].

3.2.3 Feature selection

While it is intuitive that the more features are available, the more precise the prediction will be, this is not always the case. On the one hand, by adding more features to the training process, the ones related to the target will more likely be available to the ML algorithm. On the other hand, the risk of capturing random regularities grows exponentially with the number of added features. A high number of features makes the predictive process also less interpretable.

In addition, from a medical perspective, it is not useful to introduce multiple features referring to the same medical parameter: these will be highly correlated and will not add any relevant information to the process. Feature selection can overcome these problems and can be achieved in various ways. The most frequently used approach for feature selection that

we found in the reviewed literature is the forward selection/backward elimination process (as, for instance, in papers: [68–71],[85–88], in which the model is trained multiple times using different sets of features. Features are added/eliminated at each iteration according to a greedy strategy. Other strategies are available[89], each of them addressing specific situations.

3.2.4 Class imbalances

A common issue in medical datasets is the balance between the investigated classes of patients[90]. Since ML models try to optimize some misclassification loss, when the classes are very imbalanced, the algorithm may decide that it is better to disregard (or to focus less of its efforts on) the minority class since errors on that class do not contribute too much to the classification error anyway.

This is a problem, and it should be taken into account when working with imbalanced datasets (this also holds for other ML tasks like regression and clustering). In this scenario, it is appropriate to counter the problem to ensure that the algorithm reaches its full potential in terms of generalization capabilities[71],[74],[78],[91]. There are two standard techniques for achieving this: oversampling (duplicate samples from minority class) or undersampling (removal of samples from the majority class). At the same time, the first approach can lead to some bias if data duplication is not correctly applied, and the second approach inevitably leads to loss of information.

3.2.5 Feature normalization

One more technique that can be exploited to obtain better performance with some models like K-Nearest Neighbors, Support Vector Machines (SVMs), Naive Bayes, and Neural Networks (see section 3.3 for an introduction to these models) is feature normalization. It consists of rescaling all the numerical features to have them on the same scale, thus allowing the ML algorithms that exploit numerical methods (e.g., gradient descent, distance-based algorithm) to work better way[72, 73],[81],[48, 51, 92].

We can apply feature normalization in several ways. In cases where the feature values are all positive, one can scale them to the [0,1] range by dividing each value by their maximum; otherwise, it is common to scale so that the values have zero mean and unit variance by subtracting the mean and dividing the result by the standard deviation of the feature being normalized.

3.2.6 Dataset splitting

After data have been pre-processed to enhance their quality, they should be prepared for the learning process. The whole dataset is typically divided into two or three smaller sets. The ML names for these sets are:

- **Training set**: data used to train the models;
- **Validation set**: data used to tune the hyperparameters of the model;
- **Test set**: data used to assess the generalization capability of the model.

In medical literature, these terms are sometimes different:

- Derivation cohort/set corresponds to the training set;
- No specific cohort/set is identified explicitly for hyperparameters tuning;
- Validation cohort/set corresponds to the test set.

Following the three splits schema allows us to correctly train, tune and evaluate the ML model without compromising the rigorousness of the results. Many of the reviewed paper authors do not use a validation set[75],[81],[82],[50, 92–97] tuning the hyperparameters of their models on the training set or the test set. It is worth emphasizing that using only two splits is not considered a best practice because one is likely to overfit either the training set or (worse) the test set.

3.2.7 Dataset size

The majority of the datasets used in the current studies span from few thousands[68–73] [49, 98, 99] to hundreds of thousands of patients[74], [81], [50], [97] while it is unusual to find smaller ones[78], [87], [92], [94]. The general trend is to use ever larger and larger datasets over time: this is positive since the dataset size requirements grow with the "complexity" of the concepts to be learned. Conducting an ML study only on a few hundred patients would severely limit the scope of possible applications. In this regard, it is worth mentioning that, in particular cases, useful knowledge can be extracted from a limited amount of data by exploiting a deep understanding of the specific phenomena to be analyzed[73],[74],[77],[95]. The data sources can be either a local trial[85] or a shared resource like a national or international registry[68–70],[86].

3.2.8 Follow-up time

A fixed follow-up time for the investigations makes the learning process more effective, resulting in less data variance and more interpretable results. It is also possible to incorporate time in the predictive process, but this type of analysis is more complex and delicate. Some works exploit this technique to obtain survival time predictions or time-to-event analysis; for this kind of analysis, statistical methods are typically more effective than ML ones. For instance, the SHFM risk score[86] is based on the Cox Proportional Hazard Model, explored in section 3.4 and takes time into account for his inference. The most well-known counterpart in ML is DeepHit[100], a Deep Neural Network-based survival analysis tool; its first application to medical data appears now in some preprints.

3.2.9 Privacy, security, and features

Typically medical datasets include a list of clinical features like age, sex, type of diabetes, etc.; it is not unusual to include patients' habits like smoking or drinking. These are all sensitive information that shall be manipulated according to privacy policies: many techniques allow handling sensitive data without the need to share it or move it physically (e.g., edge computing[101] and federated learning[14]). Still, we are unaware of any study on exploiting such technologies in the CV field.

Table 3.1: Summary of research goals in different research papers.

Research papers	**Major objectives**
[67–81]	Predict all-cause mortality (ACM)
[69, 71, 74, 79, 82, 83]	Predict heart failure (HF)
[84, 85]	Improving clinical decision
[86, 87]	Classify different disease
[88]	Predict risk of stroke
[80, 89]	Predict coronary artery disease

3.3 Machine Learning techniques

In this section, the most common supervised techniques are introduced. A summary of them is shown in Figure 3.2, and references to relevant literature are provided in Table 3.1. Table 3.1 also provides a list of references organized according to these study objectives. We shall explain the differences between different techniques and contextualize their usage in the

current literature. To do that, we need a few tools to make high-level but grounded claims about the techniques themselves. Specifically, we need to discuss two essential ML concepts: the "bias/variance decomposition of the error" and the "learning bias" of learning algorithms. Since the term "bias" is used with slightly different meanings in these two topics, we shall use the "bias" or "bias component of the error" to denote the former sense (the one used in the bias/variance decomposition) and always use the term "learning bias" for the latter.

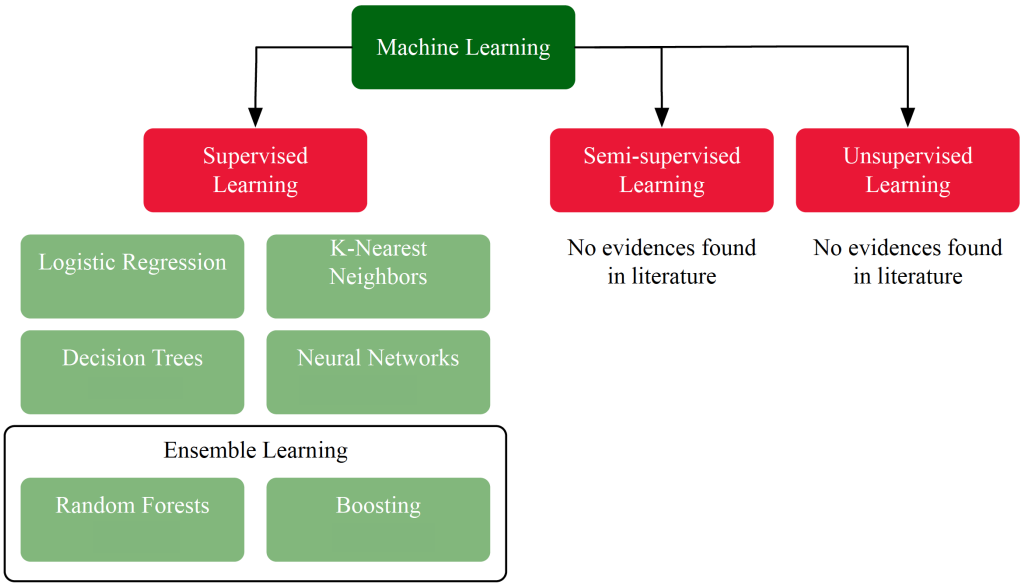


Figure 3.2: Machine learning techniques discussed in this chapter

Let us start from the bias/variance decomposition of the error[102, 103]. In a nutshell: the error made, on average, by a learning algorithm can always be decomposed into three components: bias, variance, and noise (see Figure 3.3). The bias component of the error measures how much the average decision surface differs from the true concept. This difference usually correlates with the concept space size searched by the algorithm: if the algorithm searches between all linear concepts and the true concept is a cubic polynomial, then the bias component of the error will be significant. The variance part of the error measures how much, on average, a concept learned by the algorithm differs from the average concept.

The variance component of the error usually correlates inversely with

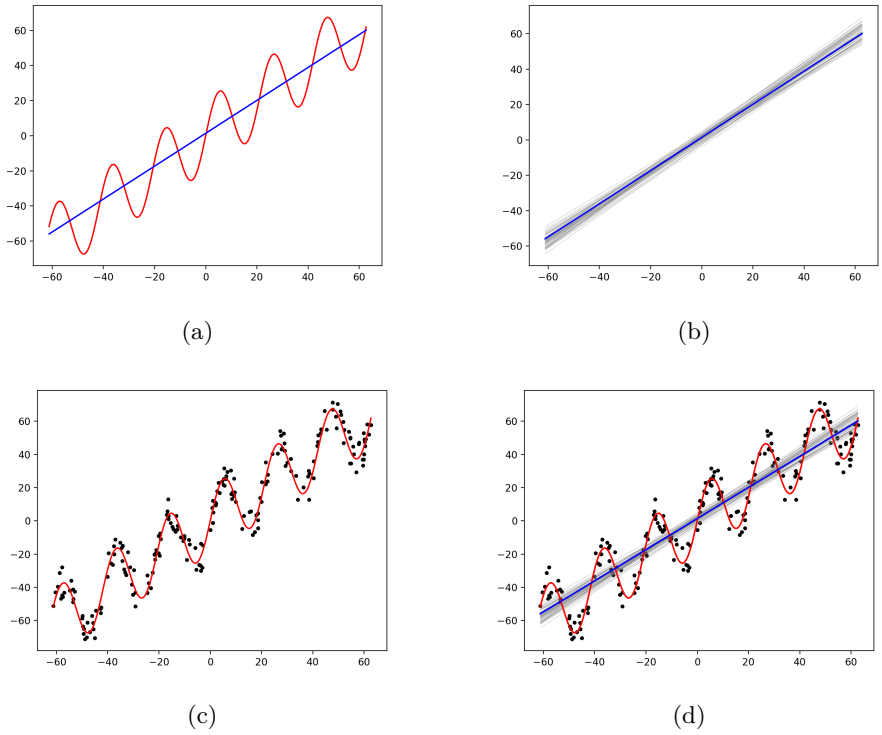


Figure 3.3: Examples of (a) bias, (b) variance, (c) noise decomposition, and (d) aggregated bias, variance, and noise. The red curved line is the true concept to be approximated, the blue line is the average regressor, the gray lines are individual regressors, and the black dots are noisy observations. As can be seen, these three error components have a massive effect on approximation performance.

the size of the concept space searched by the algorithm: in our previous example, a learning algorithm exploring the space of all cubic polynomial will have a larger error variance than that of an algorithm searching a linear concept space (having more degree of freedom, it will more easily adapt to variations in the dataset, thus producing more diversified results). Noise is just the error component due to errors in acquiring the example's features or labels. Bias and variance are usually competing forces, and decreasing one often causes the other to increase. Knowing if an algorithm has high/low bias/variance allows one to understand which are the best possible actions to improve results and enable to compare algorithms based on how brittle they are (how much the variance component of their error is high) and how well they would work when combined with other methods (e.g., ensemble techniques).

The learning bias of an algorithm refers to the heuristic that the learning

algorithm adopts to choose between different concepts. This broad definition encompasses many algorithm details, such as the space it searches and how it selects between equally good concepts on the training set. Understanding the algorithm's learning bias is important because it is the key to knowing how it suits the problem at hand. Indeed the no-free-lunch theorem[104] implies that, without further assumptions, all learning algorithms are created equal and perform equally (bad/well) on a random problem. In other terms, there is no better/best algorithm in absolute terms: an algorithm is only as good as the fitness of its learning bias on the problem at hand.

3.3.1 Logistic Regression

Logistic Regression (LR) is a supervised algorithm that can induce models for classification tasks[1]. The primary assumption made by the algorithm (i.e., its learning bias) is that the logarithm of the odds $\log\left(\frac{P(y=1)}{P(y=0)}\right)$ is a linear function of the input $\mathbf{x}$. The implication, which gives the name to the technique, is that the probability of the positive class has the form of the logistic function $f(x) = \frac{e^x}{1+e^x}$ as explained in Figure 3.4 Formally:

$$\begin{aligned}
&\log\left(\frac{P(y=1)}{P(y=0)}\right) = \mathbf{w}\cdot\mathbf{x} \\
\Rightarrow &\log\left(\frac{P(y=1)}{1-P(y=1)}\right) = \mathbf{w}\cdot\mathbf{x} \\
\Rightarrow &\frac{P(y=1)}{1-P(y=1)} = e^{\mathbf{w}\cdot\mathbf{x}} \\
\Rightarrow &P(y=1) = e^{\mathbf{w}\cdot\mathbf{x}} - P(y=1)e^{\mathbf{w}\cdot\mathbf{x}} \\
\Rightarrow &P(y=1) = \frac{e^{\mathbf{w}\cdot\mathbf{x}}}{1+e^{\mathbf{w}\cdot\mathbf{x}}}
\end{aligned} \tag{3.1}$$

The main benefits of logistic regression are that being a linear model, it tends to have low variance and requires small sample sizes to perform well. For the same reasons, it does not usually work well when the relationships to be modeled are not linear (in that case, the higher bias component of the error tends to be not compensated enough by the low variance).

[1]Please note that the "regression" part in the name of the technique can be misleading. The name is due to the fact that the main idea is to predict the probability of the classes (which is a numeric value that justifies the regression name), but it is then almost always used to solve classification problems

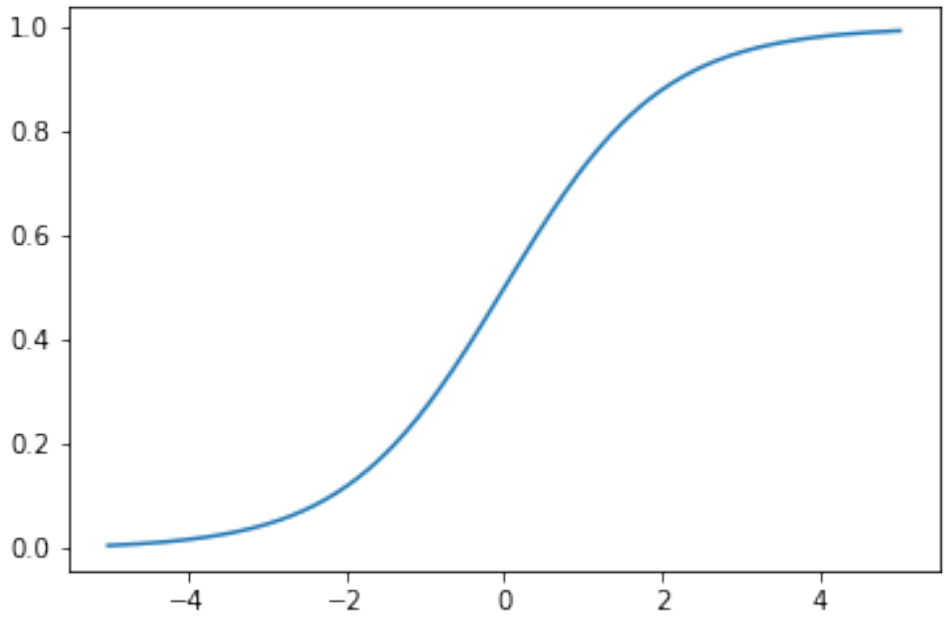


Figure 3.4: The logistic function.

Among the technologies we found in the papers we reviewed, LR ranks very high in popularity and performance. HAS-BLED[105] aims to provide a simple score for major bleeding risk in patients with atrial fibrillation. In this case, LR has been combined with univariate statistical analysis to iteratively select a subset of features highly correlated with the risk of major bleeding. EuroSCORE II[106] was built exploiting univariate LR, likelihood ratios, and Akaike's Information Criterion to select a subset of features highly correlated with the investigated endpoint (cardiac surgical mortality). These features are then inserted in a logistic equation that gives the final predicted mortality. This study is an update of the original 1999 EuroSCORE[107] and exploits a broad international database of patients.

More advanced techniques are used in [108] where LR is exploited in univariate and multivariate fashion and in a hierarchical way to study data heterogeneity across different medical centers. In SPRM[109], a multinomial LR is used on a selected set of features (obtained by univariate LR, variance, χ^2 analysis, and backward elimination) to directly compare the proportion of mortality attributable to two distinct causes: information that other models can hardly provide. ScREEN[93] exploits LR for feature

selection and the Youden index to establish cut-offs for quantitative variables. Each selected feature was assigned a single risk point, and the total risk of collateral events after Cardiac Resynchronization Therapy is the sum of these points. Many other papers exploring this topic exist[81, 96], but they apply the above LR techniques to other pathologies to the best of our knowledge.

3.3.2 K-Nearest Neighbors

Despite its age and simplicity, K-Nearest Neighbors (KNN) is still used in some of the works we reviewed. The main idea of the KNN algorithm is to store the dataset in memory and then compute the predictions for a new example $\mathbf{x}$ by recovering the K examples nearest to $\mathbf{x}$ and averaging the results (for regression) or deciding the class by a majority vote (for classification). Here the learning bias is in the assumption that *similar* examples (as estimated by the distance measure adopted) should have similar labels. From the bias/variance decomposition point of view, this algorithm has some flexibility since the K parameter controls the trade-off (the lower the K parameter, the lower the bias component, and the higher the variance component). In [92], KNN has been used to detect the early risk of coronary artery diseases.

3.3.3 Decision Trees

Decision Trees(DTs)[110] are tried and tested ML tools that can be easily applied to a wide variety of problems and provide very interpretable results. DTs' flexibility derives from their capability to do both classification and regression and their ability to work well with a wide variety of feature kinds (numerical, categorical, ordinal, ...). DTs are very interpretable models since they can be interpreted as a list of nested if-then-else clauses. They are very brittle models, meaning that they are low bias, high variance models. They also tend to overfit data unless countermeasures are taken; for these reasons, DTs are usually pruned (making the trees smaller, lowering their variance at the expense of a higher bias) and averaged using some form of ensemble algorithm. Pruning introduces an additional learning bias preferring shorter trees over taller ones.

While DTs are still very popular when used as pieces of an ensemble model (e.g., as components of Random Forests (3.3.4) or used as weak learners in Boosting procedures (3.3.5)), they are not very popular as standalone models. In [94], long-time prediction for atrial fibrillation has been

performed using DTs: the researchers produced a risk stratification system using a classification and regression tree, categorizing the patients in low, medium, and high risk.

3.3.4 Random Forest

Random Forest (RF) is an ensemble learning supervised machine learning algorithm. Its flexibility, performance, and ease of use make it a popular choice for regression and classification tasks in many application contexts. Ensemble learning encompasses different techniques, combining many models to build a more robust one; in RF, this technique is Bagging[111]. This algorithm creates many copies of a model, each one of them being trained on a different subset of the available data, and combines them through a simple majority vote or averaging the predictions. Bagging's main strength is the ability to reduce the variance of the combined models, i.e., it works best with low-bias, high-variance algorithms[112, 113]. RF builds on these strengths by bagging models obtained by a slightly updated version of the DTs learning algorithm that exacerbates these traits of DT models.

Our literature review found that RF appeared second-highest in the papers we reviewed, ranging from classification tasks[80, 87, 97] to regression tasks. Many authors use RFs to predict all-cause mortality, and others apply RF to particular cardiovascular diseases[49, 91] or for risk assessment of heart failure[87] and venous thromboembolism[76]. In some studies, authors claim that the inferred RF models are helpful for clinical decisions, allowing to estimate whether a patient is suffering from heart failure with preserved ejection fraction or not[87] and that their risk score assessment performs better than the state of the art ones[76, 80, 97].

3.3.5 Boosting

Boosting algorithms are particular kinds of ensemble algorithms. Boosting algorithms encompass those ensemble techniques that guarantee a decrease in the training error (usually by descending the gradient of some loss suffered by the ensemble). These models are very popular since they are typically easy to use, have very few parameters, can be applied to both classification and regression tasks, and tend not to overfit the data. Several boosting algorithms have proved to be particularly popular in our literature review: AdaBoost, LogitBoost, Gradient Boosting, Light Gradient Boosting, and eXtreme Gradient Boosting.

Adaboost[114] is the original boosting algorithm, and it can be shown to implicitly optimize the exponential loss suffered by the ensemble. LogitBoost[115] is a variant of AdaBoost derived by casting AdaBoost as a generalized additive model and substituting the cost function with the logistic loss $\sum_i \log(1+e^{y_i f(\mathbf{x}_i)})$ (where i sums over all examples $(\mathbf{x}_i, y_i)$). Gradient Boosting is a variant of boosting where the loss function is explicitly optimized via gradient descent, and eXtreme Gradient Boosting is a refined version of the Gradient Boosting approach[116].

Recently, AdaBoost has been used to predict all-cause mortality[49, 72] and report accuracies on par or better than the state-of-the-art based on LR[93] and provide clues useful for the clinical decision-making process. LogitBoost has been instrumental in developing cardiovascular risk predictions[71, 98], outperforming established risk scores such as the Framingham Risk Score and the Segment Stenosis Score.

eXtreme Gradient Boosting has been applied in predicting mortality[50, 51] and predicting the risk of cardiovascular disease Coronary Artery Calcium Score[99]. [50]introduced a risk assessment tool based on this latter technique that provides early prediction of older people's mortality using Electronic Health Records. In [117], authors also used eXtreme Gradient Boosting to enhance the risk stratification to maximize coronary CTA usage derived from plaque information. In all these works, the authors reported that using eXtreme Gradient Boosting improved their overall accuracy with respect to the competing approaches.

In [88], they applied Light Gradient Boosting to Intensive Care Unit patients on data collected from three hospitals. The authors claim their approach helps make better clinical decisions, and the model performs well for predictions.

Gradient Boosting of DTs has succeeded in several interesting works trying to predict mortality[78] and heart failures[48]. The latter work also provides additional information for medical staff to understand complications after heart failure.

3.3.6 Neural Networks

Neural Networks (NNs) are connectionist models with roots in cybernetics and the attempt to model the human brain[118]; since then, the models evolved into practical tools with only a faint resemblance to the first ones developed. After an exciting start in the '60s and a resurgence in the '80s, NNs did not progress for almost two decades. Many tasks required too

much data and computational power, and the research focused on simpler models that were easier to train. Thanks to a breakthrough in the training algorithms, progresses in CPUs and GPUs, the creation of big computational farms[119], and the advent of internet tools allowing the collection of huge labeled datasets, NNs have seen new interest from the research community and are nowadays at the forefront of the research in many critical applicative fields.

NNs are built from basic units called neurons, which can be easily arranged in layers. Layers are easy to connect, and the whole network can be trained end-to-end using the Stochastic Gradient Descent algorithm[120]. While a single-layer network can approximate any function to arbitrary precision (as implied by the Universal Approximation Theorem[121]), the real power of these models is in the automatic abstractions provided by stacking multiple layers into a Deep Neural Network (DNN). Each layer abstracts its inputs providing the following layer with a data representation that is easier to work within the context of the task being solved. State-of-the-art NNs models are DNNs models and have been shown to provide super-human performances[122, 123] on many tasks involving hard-to-abstract data, such as those involved with image and audio processing.

Shallow NNs have been used to predict mortality due to heart failure[74, 77], showing performances outperforming other learning methods despite being trained on unbalanced datasets in recent literature. DNNs have been exploited for predicting the risk of mortality[73, 124] or heart failure and acute heart failure[95]. The authors of these two works compared DNNs with other ML techniques showing performance improvements.

Another interesting recent application of NNs in this field is to exploit their ability to work with data correlated very complicatedly. This is the case of the Deep Cox Mixtures[125], in which a NN assists a Cox Regression Model (section 3.4.1) to fit the hazard ratios of the regression. This work is based on a sound statistical and ML background, is comprehensively exposed, and offers state-of-the-art performance when working with different groups of individuals.

3.4 Statistical approaches

Alongside the ML approaches discussed in section 3.3, it is worth briefly describing the main statistical techniques currently used in the prediction of cardiovascular events since they still cover an essential role in the field[126,

127]. As in the previous section, every major technique implied in the field will be reviewed and explained, together with a brief comment on the top risk scores that exploit them.

3.4.1 The Cox Proportional Hazards Model

Survival Analysis is a broad branch of statistics that studies how much time it takes for an event to occur or, in the specific case of medical applications, how much time would likely pass before an event affects a given individual. Given this brief description of survival analysis, it is clear that its tools are well-fit for predicting medical events. In this scenario, the Cox Proportional Hazards (PH) Model[128] takes a predominant place as being, by far, the most used statistical technique for the prediction of cardiovascular events[68, 69, 75, 82, 85, 86, 129]. The typical analysis of the relation between a single risk factor and an event is carried on by evaluating the Instantaneous Hazard Rate λ(t). This measure is defined as the rate at which events occur, given the total number of individuals at risk. Formally:

$$\lambda(t) = \lim_{\delta t \to 0} \frac{Ev(t, t+\delta t)/N(t)}{\delta t} \tag{3.2}$$

where $Ev(t, t+\delta t)$ is the number of events occurred between times t and $t+\delta t$ and $N(t)$ is the number of individuals at risk at time t.

Typically, in medical studies, one is interested in comparing different populations, each with different characteristics, like the assumption (or the lack thereof) of a drug. Then, one tries to model each group's survival possibilities to assess the effects of the given drug on the population. In these cases, it is convenient to model the variations in risk hazards of the different populations by mean of the Hazard Ratio (HR), which is the ratio of the two different Instantaneous Hazard Rates:

$$HR = \frac{\lambda_1(t)}{\lambda_0(t)} \tag{3.3}$$

where $\lambda_1(t)$ and $\lambda_1(t)$ are the Instantaneous Hazar Rate for the populations 1 and 2 at time t. With an HR ratio above 1, the events are more likely to occur in population 1 and vice-versa, with the magnitude of HR indicating the difference's strength.

The kind of analysis shown before, although useful, can be applied only to investigate the impact of a single risk factor on the survival possibilities of a population. In order to assess the simultaneous impact of multiple risk

factors, more complex tools are needed. In this context, the Cox PH model finds its bases, allowing to assess of the impact of multiple risk factors on the survival time of a population under three assumptions [130]:

- the survival capability of an individual is independent of the other individuals in the population;
- the risk factors and the hazard are multiplicatively related (i.e., incrementing one of the risks multiplies the hazard);
- the hazard ratio over time is constant;

these assumptions make the Cox PH model semi-parametric since it makes assumptions on the relationship between risk factors and hazard but not on the hazard function itself. This approach is justified by a conditional argument by Cox that here will not be presented. In general, a Cox PH model can be written as:

$$\lambda(t|X_i) = \lambda_0(t) \exp(X_i \beta) \tag{3.4}$$

where X_i is the vector of the risk factor values for the i-th individual (usually called covariates in this context), β is the vector of regression coefficients, and $\lambda_0(t)$ is the baseline hazard when all the risk factors are zero. The values of β assess every risk factor's impact on the population's survival; positive values of β will proportionally increase the hazard risk and vice-versa. Of course, it is possible to calculate the HR of two hazard rates calculated with a Cox PH model; thus, it is possible to investigate the survival capabilities of different populations based on multiple risk factors.

3.4.2 Heart Failure Survival Score

One of the most well-known risk scores exploiting a Cox PH model is the Heart Failure Survival Score (HFSS)[85]. The first step for the derivation of this score has been the clinical features selection by mean of univariate statistical analysis methods like Kaplan-Meier method[131] and log-rank tests[132]; in this way, the researchers successfully reduced the analysis on a set of forty features, against the eighty available. The Cox PH model has been applied to these features, but with two additional strategies: a stepwise forward-entry/backward-elimination selection based on the p-value, and the best-subset discovery, based on a χ^2 test. In this way, a subset of only eleven features has been selected as the best trade-off between feature number and predictive power.

The HFSS is then defined as the absolute value of the sum of the products of the Cox PH model coefficients and the respective risk factor value ($|\beta_0 x_0 + \beta_1 x_1 + ... + \beta_n x_n|$ where $x_1, x_2, ..., x_n$ are the actual variable values and $\beta_1, \beta_2, ..., \beta n$ are the computed coefficient). This risk score achieved good results for its time, but it lacked generalization capabilities: performance was limited when applied to other datasets than the one on which has been derived due to the low number of patients involved in the study and the specific requirements that they had to match. A positive aspect of this work, though, is that not only one model has been developed: two of them have been derived, one exploiting an invasive medical feature (mean PCWP) and the other one not; despite that the two models reached similar performance, thus raising attention on the real necessity of doing or not invasive procedures.

3.4.3 Seattle Heart Failure Model

Another risk score exploiting the Cox PH model is the Seattle Heart Failure Model (SHFM)[86]. In this case, the feature selection has been made by means only of the Cox PH model, with a stepwise forward-entry/backward-elimination, partially from the derivation dataset, partially from large published trials (for the features not exhaustively described by the derivation cohort). Once the model has been derived, the SHFM score is defined as the sum of the products of the β-coefficients with the value of the corresponding parameter ($SHFM\ score = |\beta_0 x_0 + \beta_1 x_1 + ... + \beta_n x_n|$). The survival value at time t for a patient is then defined as $survival(t) = e^{(-\lambda t) e^{(SHFM\ Score)}}$, where $e^{(-\lambda t)}$ is the baseline survival (survival at time t when all risk factors are zero) and λ the slope/year derived from the dataset. For how it is constructed, this risk score allows a per-patient analysis, and his reliability is well-documented since it has been tested on five different datasets; it is also an example of a score in which some risk factors (like age and sex), have been forced into the model, thus merging the statistical approach and the medical knowledge.

3.4.4 ORBIT

In ORBIT[68], the Cox PH model is used, together with a feature selection step based on the backward selection process, to create the best performing model based on a pool of medically relevant risk factors for major bleeding. The derivation dataset is large, counting more than ten thousand patients,

and the missing data were imputed only once through Markov Chain Monte Carlo or regression methods. From the full final model, only five risk factors have been retained, the ones with the highest χ^2 statistic, and to each one of them, an integer score is assigned based on the strength of their correlation with major bleeding.

The result is a simple risk score, easily computable in a real-world situation. This risk score is another example of how a limited group of risk factors can be exploited for good performance. To assess the technique's performance, the paper reports an evaluation on an external dataset where ORBIT is compared against HAS-BLED[133] and ATRIA[134]. The GISSI[129] risk score exploits a similar approach, but in this case, there are no predefined integer points assigned to each final feature, but a nomogram is provided for bedside application; in this way, it is possible also to take into account the value of the risk predictors.

3.4.5 PARIS

PARIS[69] is another risk score based on the Cox PH model: differently from the other approaches presented above, it exploits data imputation in the derivation process. Employing a multivariate normal regression, specific missing values of decisive risk factors have been imputed multiple times and, for each set of imputed data, a Cox PH model with backward selection has been fitted to the data. These different models are used to obtain a fully calibrated final Cox PH model. From that model, the β-coefficient is used to obtain integer values for the risk factors. This approach has been repeated two times, one for the derivation of the major bleeding model and the other for the coronary thrombotic event one, allowing physicians to evaluate the risk of these two events through two integer risk scores.

3.4.6 PRECISE-DAPT

The PRECISE-DAPT[70] score exploits the Cox PH model in two univariate and multivariate flavors, with backward elimination, to assess the potential predictors of major and minor TIMI bleeding. The result is an integer risk score computed on five clinical variables, and each variable is associated with an integer score based on its value and β-coefficient. The paper also offers a nomogram for bedside single-patient evaluation. This score was derived from a broad dataset and validated on two external cohorts. It has also been compared to the PARIS score during the evaluation to assess these two approaches' differences.

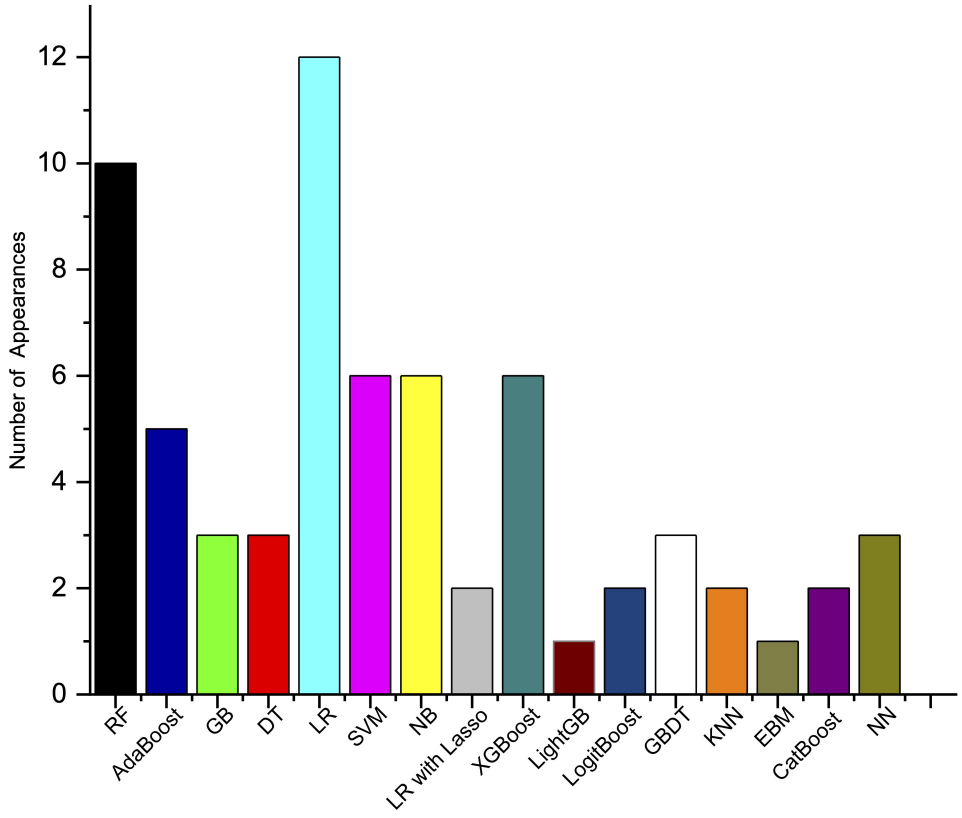


Figure 3.5: Number of appearances of Machine Learning techniques in the reviewed literature

3.5 Discussion

Figure 3.5 reports the histogram of the frequencies with which ML techniques have been used in the papers we reviewed. While the figure shows quite a number of different approaches, we observe that most of the experimentation happens with ensemble techniques. In fact, 9 techniques (RF, AdaBoost, Gradient Boosting, eXtreme Gradient Boosting, Light Gradient Boosting, LogitBoost, Gradient Boosting of DTs, Explainable Boosting Machines, and CatBoost) out of 16 are ensemble algorithms. It also happens that while Random Forests is the most popular ensemble approach, most of the others are some variants of Gradient Boosting.

The preference for ensemble learning, in general, and Boosting, in particular, is highly understandable since Boosting usually gets very accurate models without requiring much tuning of the parameters. Also, Boosting

naturally counterbalances overfitting[135] by increasing the "margin" of the classification even when the training error stops decreasing. If we remove ensemble learning from the picture, we see that Logistic Regression is used almost as much as all the remaining approaches cumulatively (14 times versus 17). Again, this technique is straightforward to apply and very robust to overfitting, so it can be easily applied to smaller datasets.

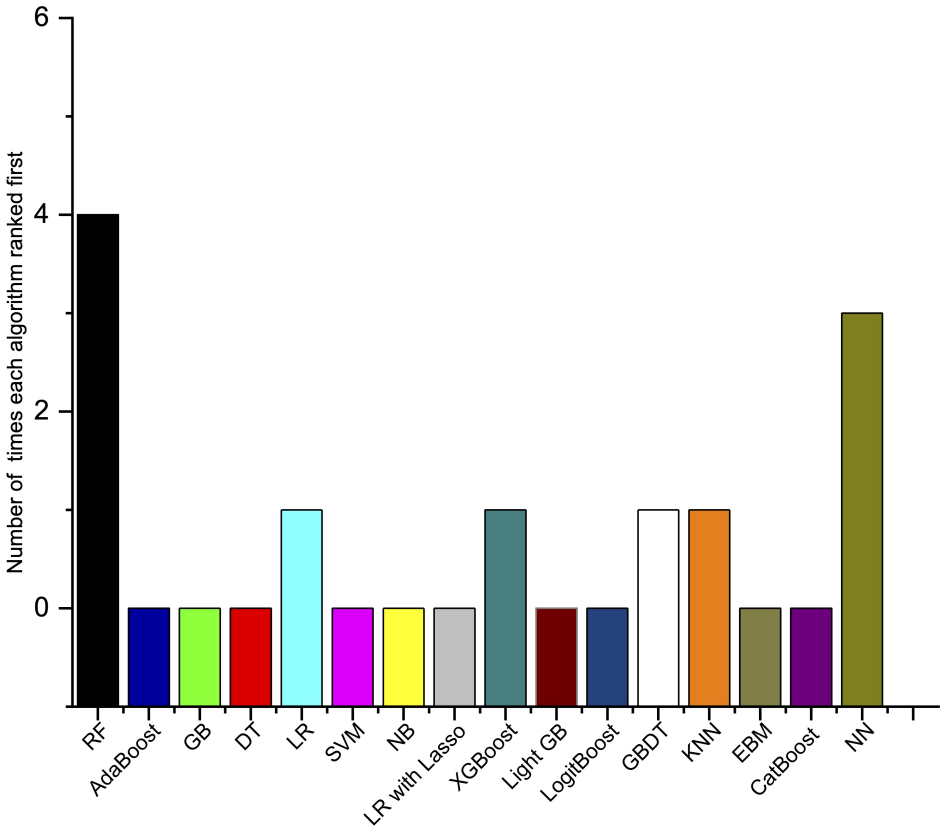


Figure 3.6: Number of times each Machine Learning technique ranked first in the reviewed literature. Papers where only a single technique was presented are not included.

The remaining models (in order of popularity) are SVMs, Naive Bayes, and KNN. SVMs, despite being very popular in our sample of papers, never achieve the highest ranking, often being outperformed by Logistic Regression. In theory, SVMs should be able to match or outperform logistic regression when properly configured and trained. Unfortunately, there is often no information[76, 95] or very little information[48, 73, 76, 87] in the papers we

reviewed about how SVMs are configured (which kernels are used in the experiments and how other hyperparameters have been chosen), so it is hard to tell how much effort was devoted to tuning these tools to the problem at hand. Naive Bayes and KNN are seldom used and do not seem to perform well anyway; KNN ranks first in one case[92], but in that case, the study has only 60 patients and compares it only with one other approach(Random Forest).

Figure 3.6 reports, for each technique, the number of times it ranked first in the papers where it competed. In that figure, we omit to count an algorithm as ranking first if the paper did not compare it with other methods (explaining why some of the methods appear with a count of 0 and the sum of the counting is shorter than the list of papers we reviewed). Despite not being a very popular method, they perform best in the three occasions where NNs are used. This is not a surprising result: NNs are notoriously hard to train and bring the necessity of selecting many hyperparameters, which explains why they are not the preferred choice in many works. However, when they are properly configured and when data is abundant, they usually perform very well.

Key Messages

- Ensemble Methods (especially Random Forests) and Logistic Regression are the ML tools most used in recent CV studies;
- Neural Networks, despite being at the forefront of recent ML developments, are under-represented in this field. We speculate that this is in part because of the difficulty in interpreting them and in part because Neural Networks are harder to train and require larger datasets (if not tuned appropriately).
- Large (federated) datasets and unsupervised techniques are still not much used. Adopting them would significantly improve the current performances of predictions based on ML techniques and pave the way to broader adoption of more sophisticated ML techniques.

Chapter 4

Traditional Machine Learning and Federated Learning on critical Datasets

In this chapter, our objective was to investigate the ML techniques for the PRAISE dataset and the development of the PRAISE tool. Initially, we focused on traditional ML methods, as discussed in section 4.1 . We thoroughly examined various ML techniques applied to the PRAISE dataset and discussed their implications and outcomes. We aimed to gain insights into the dataset and create the PRAISE tool, which would be a valuable resource for further analysis.

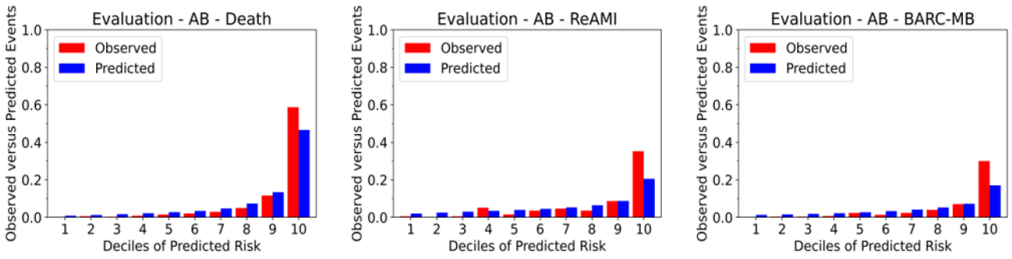
Subsequently, we expanded our exploration to include federated learning (FL) techniques on the same PRAISE dataset. Section 4.2 of this chapter focuses explicitly on FL techniques, where we applied them and analyzed their impact. We conducted a rigorous assessment of the FL techniques using diverse evaluation metrics. Through this comparative analysis, we aimed to develop a comprehensive understanding of the strengths and weaknesses of FL techniques in the context of the PRAISE dataset. By devoting separate sections to traditional ML methods (section 4.1) and FL techniques (section 4.2), we provided an extensive exploration of their applications and performances on the PRAISE dataset. This chapter is a comprehensive guide, offering valuable insights into the ML and FL approaches employed and highlighting their respective contributions to the field.

4.1 Traditional Machine Learning

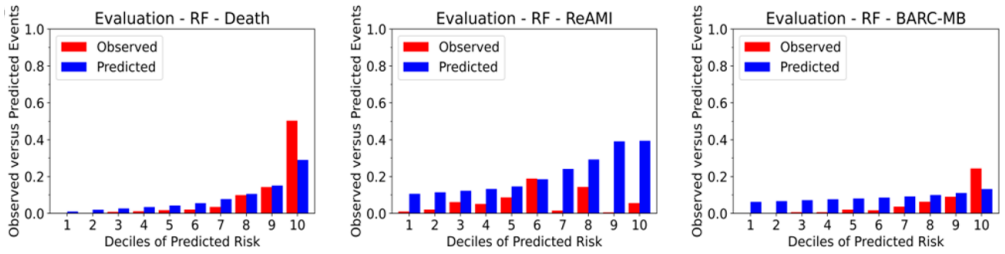
This section will comprehensively discuss the machine learning techniques applied to the highly critical PRAISE dataset. Specifically, we will explore AdaBoost, and Random Forest algorithms in this dataset, as well as their respective accuracies and areas under the curve(AUC) achieved through hyperparameter optimization. Furthermore, we will address the tool that was developed utilizing these models.

4.1.1 PRAISE Score: A main motivation towards the Federated Learning

This work compares various machine learning algorithms, including *AdaBoost* and Random Forest, for predicting risk scores using target variables such as All-cause death, BARCMB, and RENAMI. In particular, my contribution to this work involved conducting hyperparameter tuning for ML techniques, including *AdaBoost*, Random Forest, Decision Trees, and Support Vector Machines. I worked alongside the other authors to improve the performance of these ML techniques, and we ultimately found that only *AdaBoost* and Random Forest performed well after parameter tuning. Therefore, these two algorithms were selected for the PRAISE score study.



(a) AdaBoost



(b) RF

Figure 4.1: Calibration plots of AdaBoost (a) and RF (b)

The PRAISE score for 1-year all-cause death showed an AUC of 0.93 (95 % CI 0.91-0.94) and 0.94 (95%CI 0.92-0.96) in the training and the validation cohort, respectively. The PRAISE score for major bleeding showed an AUC of 0.90 (95%CI 0.89-0.91) in the training cohort and 0.87 (95%CI 0.83-0.94) in the validation cohort. The PRAISE score for MI showed an AUC of 0.89 (95%CI 0.88-0.90) and 0.90 (95%CI 0.85-0.94) in the training and the validation cohort, respectively. In Figure 4.1, we have shown AdaBoost and RF calibration plots with observed and predicted events.

The major contribution from my side to this work was a web-based tool called the PRAISE Score, which is accessible online[1] and shown its interface in the Figure 4.2. To create this tool, we utilized the previously trained models from our research paper[72], namely All-cause death, BARCMB, and RENAMI. The PRAISE calculator comprises three key components: a front-end, back-end, and Rest API for facilitating communication between the two. The front-end is designed using simple HTML, CSS, and PHP and allows users to input data for a single patient or upload a file containing multiple patient data. The back end employs our death, BARCMB, and RENAMI models. To establish communication between the front-end and back-end, we utilized an H2O REST API[2], which enabled us to perform create, read, and delete operations.

Moreover, in today's healthcare systems, an enormous amount of data can be leveraged to create statistical models based on machine learning using medical data. However, access to medical data is restricted due to privacy concerns, leading to underutilizing of existing data in the health sector. Federated learning is one of the viable solutions in this scenario. The key motivation behind federated learning is privacy and avoiding data sharing. In our experience of working in machine learning for medicine, we have observed that not all medical datasets are publicly accessible. Therefore, federated learning presents the best option when dealing with sensitive medical data.

As mentioned earlier in this chapter 3, it would be beneficial to apply FL techniques in the field of cardiology, as this area remains largely unexplored and requires attention. We have previously noted that traditional machine learning has yielded fruitful results. However, given the sensitive nature of

[1] Available at: https://praise.hpc4ai.it/

[2] An open source API of H2O framework available at: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/rest-api-reference.html

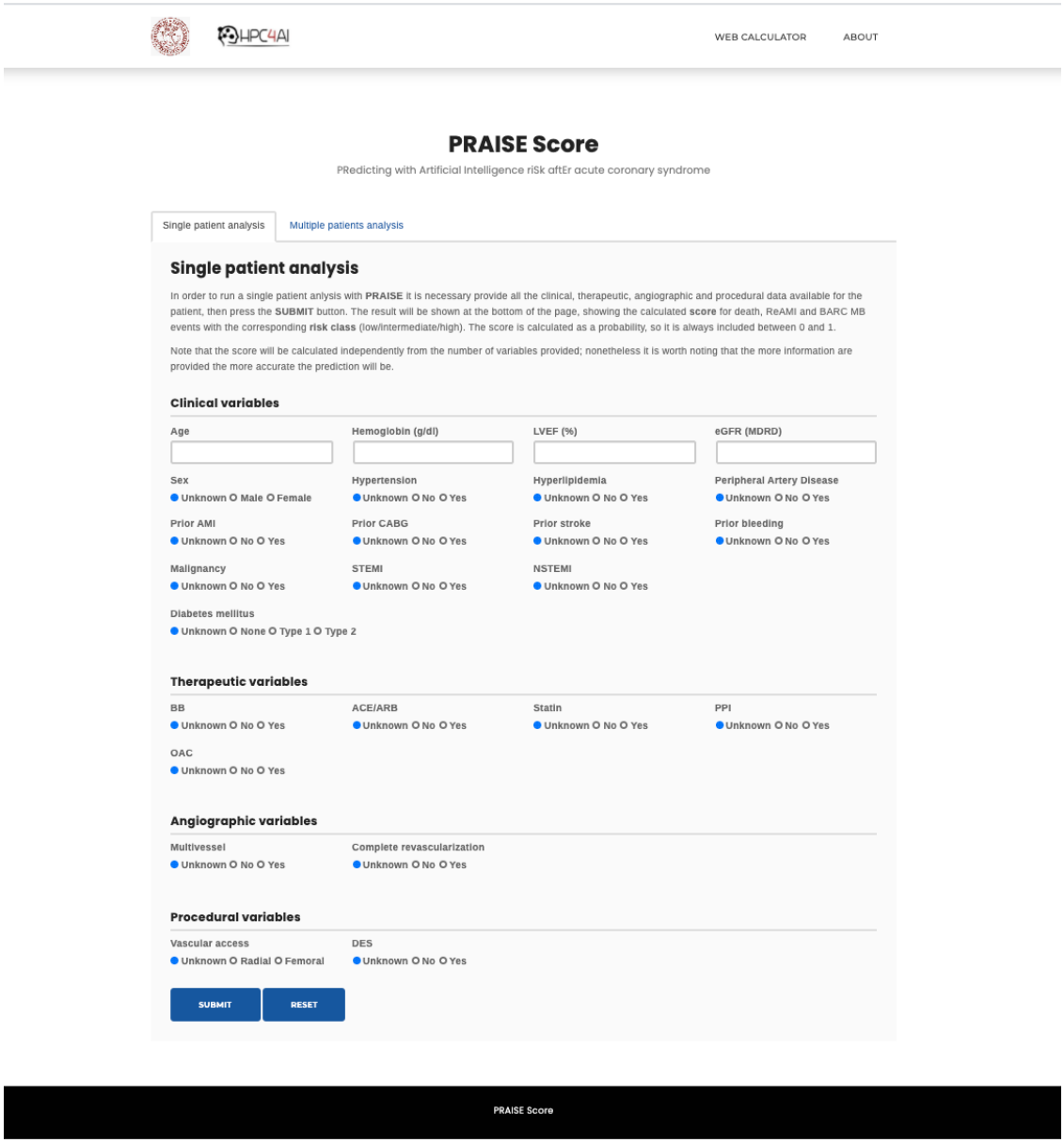


Figure 4.2: Praise Score

the data in this domain, it is imperative to identify the best approach for sharing and accessing the data, and in this regard, FL is the only solution.

4.2 Federated Learning on critical Datasets

We started from lancet work as discussed in the section 4.1 that is successful because we succeeded in putting together a very large data set (pooled), and

we found an ML method that works quite well. We imagined FL as a way to work on a large dataset without putting the data together (moving models instead of data). Therefore we applied this technique to the original lancet dataset split into parts, comparing the loss due to collaboration (as opposite to put together data).

4.2.1 Why Federated Learning on critical Datasets

Recent years have been characterized by crucial advances in AI systems. The ubiquitous availability of data sets and processing elements supported these advances. The consequent deployment of ML methods throughout many industries has been a welcome innovation, albeit one that generated newfound concerns across multiple dimensions, such as performance, energy efficiency, privacy, criticality, and security. Concerns about data access and movement are particularly felt by industrial sectors such as healthcare, defense, finance, et cetera.

This work will mainly focus on healthcare and Federated Learning (FL), a learning paradigm where multiple parties *(clients)* collaborate in solving a learning task using their private data. Importantly, each client's local data is not exchanged or transferred to any participant since, in its most common configuration, clients collaborate by exchanging local models instead of moving the data. The *aggregator* collects the local models and aggregates them to produce a global model. The global model is then sent back to the clients, who use it to update their local models. Then, using their private data, they further update the local model. This process is repeated until the global model converges to a satisfactory solution or a maximum number of rounds is reached.

Thanks to its capability to transform inherently distributed and segregated data into shared knowledge, FL is becoming popular in healthcare; As we already surveyed, the main related work in the Sec. 2.5. Until recently, the usage of FL was restricted to a specific paradigm of ML, namely Artificial Neural Networks (ANN), often in the Deep Neural Network (DNN) variant, in which models can be easily aggregated using simple associative operators, such as the average of the weights of the DNN[16, 136]. Unfortunately, ANN/DNN is not always the best tool to analyze healthcare data, which is most often given in the form of tabular data. A tabular dataset is a type of data structure that organizes information into a table format with rows and columns. Each row represents a single data point or record, and each column represents a specific attribute or variable. The data in

a tabular dataset is usually numerical or categorical in nature and can be easily analyzed and visualized using tools such as spreadsheets, databases, or data visualization software. Examples of tabular datasets include financial data and demographic data. Several factors may undermine the use of ANN/DNN for (some) healthcare tasks:

- ANN/DNN usually require very large data sets, which are often not easy to collect under the strict privacy laws that regulate health institutions;
- ANN/DNN is hardly explainable, and this is often not acceptable in this field;
- while ANN/DNN performance superiority is undisputed for several tasks, such as image classification, voice recognition, and natural language models, they are not particularly suited to tabular data[137];
- representation learning, which is one of the main driving forces underlying the success of DNN, is much less useful for tabular data since very often the features have been already engineered with great care and carefully tuned;

Recently Polato et al.[52] proposed several novel Federated Learning algorithms not relying on ANN/DNN; they instead extend distributed AdaBoost techniques to the FL case. Among other algorithms, the work introduces `AdaBoost.F`, a federated variant of the Samme algorithm[3] [138], which can be coupled with a *weak learner* to build a global model on the federated dataset. A weak learner is a learning algorithm, such as decision trees or a logistic regression, which is not required to return very good models (instead, they are required to return models that are better than the random guess). It is worth noting that the very weak assumption on the kind of models built by the clients of the federation allows using the ML model that is best suited for the tabular healthcare data at hand. The mentioned work analyzes the performance of `AdaBoost.F` on several standard datasets in a simulated distributed environment. To our knowledge, the proposed methodology has neither been tested on a real-world dataset in the healthcare domain nor in a distributed execution environment.

As a real-world example, we replicated a notable ML-based risk stratification model in cardiology that recently appeared in the Lancet [139]. The

[3] A multi-class variant of AdaBoost.

study proposed PRAISE[4] an ML-based score to predict all-cause death, recurrent acute myocardial infarction, and major bleeding after Acute Coronary Syndrome (ACS). Several (non-ANN) ML models have been trained in a cohort of 19826 adult patients with ACS, including patients across several hospitals. The cohort has been split into a training cohort (80%) and an internal validation cohort (20%). The PRAISE score is the best-performing model tested in an external validation cohort of 3444 patients with ACS pooled from a randomized controlled trial. The PRAISE score showed a performance across all possible classification thresholds (Area Under the ROC Curve or AUC) far better than the previously known scores for the same classification task: 0.82 in the internal validation cohort and 0.92 in the external validation cohort for 1-year all-cause death; an AUC of 0.74 in the internal validation cohort and 0.81 in the external validation cohort for 1-year myocardial infarction; and an AUC of 0.70 in the internal validation cohort and 0.86 in the external validation cohort for 1-year major bleeding.

The PRAISE score authors concur in claiming that one of the ingredients making it possible to define a high-quality ML-based score has been gathering one of the most extensive data sets on ACS ever built. The cohort of 19826 adult patients has been manually collected from different hospitals in different countries (see [139]). Despite being made on anonymized patients, the gathering itself is reported as a very complex process for the privacy and secrecy concerns related to managing critical data from different hospitals in different countries. The current paper attempts to use the dataset of the PRAISE score to be analyzed through federated learning techniques expecting it will provide better results, evidenced by the discussion in Section 4.2.5. Following are the contributions made by the authors:

- This work aims to simplify future studies, enabling running ML processes on a virtually pooled dataset using a privacy-preserving FL approach.
- Specifically, we aim to demonstrate that FL `AdaBoost.F` can build an almost equally good ML-based model for the PRAISE score while maintaining the data from different sites distributed and mutually secret to the parties running the FL process.
- We also show that the non-ANN models still have a performance edge

[4]PRAISE: PRedicting with AI riSk aftEr acute coronary syndrome. Available as Software-as-a-Service at https://praise.hpc4ai.it

over more popular ANN/DNN federated models on this specific data set (and, we argue, on many tabular data sets).

- We study the performance of test accuracy of `AdaBoost.F` (accuracy, F1, F2, precision, recall) and scalability in two different execution environments: a cluster of Virtual Machines on an OpenStack cloud and an HPC cluster.

4.2.2 Methods

This section describes the methods and the tools used for the experiments we report in Section 4.2.5. As mentioned, FL has been traditionally based on some variation of the gradient descent algorithm, whereas the PRAISE score model has been built on AdaBoost derived models, i.e., a non-gradient descent algorithm [5]. We frame both gradient descent and other approaches in the FL paradigm; then, we describe a recent extension of the OpenFL framework supporting both approaches. Eventually, we argue on the potentiality of the FL as a general privacy-preserving paradigm to extract shared knowledge from datasets from different organizations, i.e. to define a novel methodology to manage data distributed at the edge.

4.2.3 FL with gradient descent

FedAvg is an iterative algorithm where a central node (the *aggregator*) collaborates with the other parties (which are termed *collaborators* or *clients*) to develop a shared model. The aggregator starts the process by sharing a randomly initialized neural network with the collaborators. At each round, all collaborators perform one or more training epochs on the given network using their local datasets. The updated model is then shared with the aggregator. The aggregator averages then the contributions using the weighted average:

$$\mathbf{w}^{t+1} = \sum_{c=1}^{C} \frac{n_c}{n} \mathbf{w}_c^{t+1} \tag{4.1}$$

where $\mathbf{w}$ is a vector containing the weights of the neural networks, t denotes the current round, n_c is the size of the local dataset of client c, and $n =$

[5] Technically AdaBoost can be explained as an additive algorithm performing a coordinate-wise gradient descent of an exponential loss. The gradient descent is, however implicit and does not require gradients to be exchanged nor calculated.

$\sum_c n_c$. The $\mathbf{w}$ vector is then redistributed to all clients, and a new round can start.

One of the difficulties in FL is dealing with clients having examples from different distributions (i.e., they cannot be assumed to be independent and identically distributed (IID)). While we are not addressing this issue in this paper, it is worth mentioning that a few algorithms have been proposed to better cope with these cases. Among them, we can cite FedCurv [140], FedProx'[141], FedNova [142], and SCAFFOLD [143], which substantially increase the complexity of the federation protocol, but fall short in equally increasing the prediction performance of the resulting models. We refer to the literature for further details and comparative studies [144].

Another difficulty in FL is handling a large number of parties in the federation, which can be in the hundreds or even more. These cases can be handled by including in a round a random selection of clients.

4.2.4 FL without gradient descent

In this chapter, we experiment with the `AdaBoost.F` algorithm, first introduced in [145]. We report the pseudo-code of the algorithms using the same notation as in the original paper in Algorithms 1 and 2. These notations are: to identify a message that carries x from a client to the aggregator, we will use the function $\mathbf{send}_x(\text{aggregator}, x)$ (client-side). The function $\mathbf{broadcast}_x(x)$, which transmits x to all clients, is used on the aggregator side. There is a $\mathbf{receive}_x(s)$ in the receiver for every $\mathbf{send}_x(\text{aggregator}, x)$ or $\mathbf{broadcast}_x(x)$ from a sender, where s denotes the sender. Other notations represent weighted error (ϵ^t), weight of the weak hypothesis (α^t), distribution $\mathbf{d}$ and $\mathbf{h}^{t\star}$ represents "global" weak classifier.

The training phase of `AdaBoost.F` is similar to the one of AdaBoost, but it happens in a distributed manner. At each iteration, a new weak hypothesis is learned from each client and sent to the aggregator. The aggregator collects the weak hypotheses and broadcasts them all to all clients. The clients evaluate the received hypotheses on the local dataset and send the weighted errors ϵ to the aggregator, which is then able to aggregate these values into a matrix $\mathbf{E}^t$. Values in $\mathbf{E}^t$ are then used to find the best hypothesis for the current round c^{t*} and to compute the current α^t term. By propagating these pieces of information to the clients, they are then able to update their local copy of the ensemble and to update the local examples' weights $\mathbf{d}$. Overall, the algorithm has strong resemblances with the original

Algorithm 1: `AdaBoost.F` (aggregator)

Input: C: number of clients
T: dimension of the ensamble
K: number of classes
Output: $\text{ens}(\mathbf{x}) \triangleq \text{vote}([h^{t\star}]_{t=1}^{T}, [\alpha^t]_{t=1}^{T}, \mathbf{x})$

for $t \in \{1 \dots T\}$ **do**
  $Z \leftarrow \| [\mathbf{receive}_Z(c)]_{c=1}^{C} \|_1$
  $\mathbf{h}^t \leftarrow [\mathbf{receive}_h(c)]_{c=1}^{C}$
  $\mathbf{broadcast}_{\mathbf{h}}(\mathbf{h}^t)$
  $\mathbf{E}^t \leftarrow \frac{1}{Z} [\mathbf{receive}_{\boldsymbol{\epsilon}}(c)]_{c=1}^{C}$ ▷ $C \times C$ `errors matrix`
  $c^{t\star} \leftarrow \arg\min_c \sum_{c'=1}^{C} \mathbf{E}_{cc'}^t$
  $\epsilon^{t\star} \leftarrow \sum_{c=1}^{C} \mathbf{E}_{cc^{t\star}}^t$
  $\alpha^t \leftarrow \log\left(\frac{1-\epsilon^{t\star}}{\epsilon^{t\star}}\right) + \log(K-1)$
  $\mathbf{broadcast}_{\alpha}(\alpha^t)$
  $\mathbf{broadcast}_{c}(c^{t\star})$
$\mathbf{broadcast}_{\text{stop}}(\text{stop})$

Algorithm 2: `AdaBoost.F` (client)

Input: $\mathcal{A}$: weak learner
$\mathbf{X} \in \mathbb{R}^{n \times m}$: training data
$\mathbf{y} \in \{1, \dots, K\}^n$: training labels

$\mathbf{d} \leftarrow \mathbf{1}$
while *not stop* **do**
  $\mathbf{send}_Z(\text{aggregator}, \|\mathbf{d}\|_1)$
  $h \leftarrow \mathcal{A}\left(\mathbf{X}, \mathbf{y}, \frac{\mathbf{d}}{\|\mathbf{d}\|_1}\right)$
  $\mathbf{send}_h(\text{aggregator}, h)$
  $\mathbf{h} \leftarrow \mathbf{receive}_{\mathbf{h}}(\text{aggregator})$
  $\boldsymbol{\epsilon} \leftarrow [\mathbf{d}^\top [\![\mathbf{y} \neq h_c(\mathbf{X})]\!]]_{c=1}^{|\mathbf{h}|}$
  $\mathbf{send}_{\boldsymbol{\epsilon}}(\text{aggregator}, \boldsymbol{\epsilon})$
  $\alpha^\star \leftarrow \mathbf{receive}_{\alpha}(\text{aggregator})$
  $c^\star \leftarrow \mathbf{receive}_{c}(\text{aggregator})$
  $\mathbf{d} \leftarrow [d_i \exp(-\alpha^\star [\![h_{c^\star}(x_i) \neq y_i]\!])]_{i=1}^{n}$

AdaBoost algorithm; one interesting difference is the fact that $\mathbf{d}$ is kept unnormalized in the client. This is important to make it possible to compute a global normalization factor in the aggregator.

4.2.5 Experiments

This section compares boosting algorithms and neural network models on a real-world binary classification problem, assessing their prediction accuracy and training time performance. We also compare federated algorithms against their non-federated counterpart, which serves as a baseline for the prediction performance.

In particular, we trained a simple Feed-forward Neural Network (FNN) model, and an AdaBoost ensemble on the PRAISE dataset [139], containing 19826 adult patients suffering from Acute Coronary Syndrome (ACS) with one year of follow-up. This dataset is the union of two previously existing registries, BleeMACS (NCT02466854) and RENAMI [147] and contains 25 features categorical and ordinal, and three categorical outcomes: all-cause death, recurrent acute myocardial infarction, and major bleeding one year after discharge. We only trained the models for this study to predict the all-cause death outcome.

As a first step, we pre-processed the dataset to mitigate the high imbalance in the outcomes (using SMOTE [148]) and to handle missing data in the features (using the median value along each column). Then we split the dataset to use 80% of the rows for training, leaving the remaining ones for validation. The FNN model is a two-layer perceptron with 35 inputs, 35 hidden units, a single output, and a binary cross-entropy loss. We trained it for 100 rounds of one epoch each, using Adam [149] ($lr = 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-7}$) as the optimizer and FedAvg [16] as the aggregation strategy. The federated AdaBoost ensemble is built by running 100 rounds of the `AdaBoost.F` algorithm [52], using a decision tree with at most 10 leaves as the weak learner.

All FL runs have been orchestrated using the intel OpenFL framework [21] with a single aggregator and up to 16 collaborators. We tested two different configurations to assess both the strong and weak scaling of the training processes. strong scaling, where we increase the collaborators while keeping the same problem size by spitting the dataset samples in uniform chunks across collaborators; and weak scaling, where we scale the problem size with the number collaborators by assigning each collaborator the entire dataset. Moreover, to test the strong scaling, we divided the entire dataset into n i.i.d. subsets without replacement and assigned a subset to each of the n collaborators involved in the federation. This configuration keeps the same amount of total rows for each experimental configuration. Conversely, to test the weak scaling, we sampled 16 subsets of the complete datasets

Table 4.1: Summary of statistics used to evaluate prediction performances.

Measure	Definition	Description
Accuracy	$\frac{\text{TP+TN}}{\text{TOT}}$	Fraction of the examples correctly classified.
Precision	$\frac{\text{TP}}{\text{TP+FP}}$	Fraction of examples predicted as positive that are actually positive.
Recall	$\frac{\text{TP}}{\text{POS}}$	Fraction of positive examples that are correctly predicted as positive.
F_1 score	$2\frac{\text{precision}\cdot\text{recall}}{\text{precision+recall}}$	Harmonic mean of precision and recall.
F_2 score	$5\frac{\text{precision}\cdot\text{recall}}{\text{4precision+recall}}$	Weighted harmonic mean of precision and recall (giving precision twice the importance of recall).

and assigned one of them to each collaborator. In this setting, the size of the problem increases linearly with the number of collaborators involved in the federation.

We took both learning performances and training times for each setting. Learning performances have been measured on a virtualized environment on top of the OpenStack-based HPC4AI cloud infrastructure [150], with the aggregator running into a VM with 4 cores and 8GB RAM and up to 16 collaborators hosted in VMs with 8 cores and 8GB RAM each. Conversely, times have been measured on the C3S HPC facility [151], allocating an entire bare metal node with 2 intel Xeon E5-2697 sockets (18 cores, 2.30GHz) and 128GB RAM to each component of the OpenFL deployment.

4.2.6 Results

We analyze the performance of the algorithms in terms of prediction quality and computational/communication times. To assess the prediction performances, we used the five metrics reported in Table 4.1: accuracy, F_1 score, F_2 score, precision, and recall. Despite being the most known and used metric, accuracy is a very bad indicator of a model's performance when the dataset is not balanced, as in this case. The accuracy of a constant model always predicting *false* would score 97% on the PRAISE dataset. F_1 and F_2 scores are better candidates in these cases since, by averaging precision and recall, they require the classifier to recover most of the positive cases (to score a high recall) and to be correct on them (to score a high precision).

Table 4.2: Prediction performance of the FNN with FedAvg. Values reported are the average ± stdev of 5 runs. The first run in the strong scaling setting is equivalent to the non-federated case.

Clients	Accuracy	F1 Score	F2 Score	Precision	Recall
Strong scaling setting					
1	.39 ± .47	.14 ± .08	.22 ± .04	.17 ± .09	.72 ± .39
2	.56 ± .47	.19 ± .09	.26 ± .06	.15 ± .09	.61 ± .36
4	.88 ± .01	.23 ± .01	.30 ± .01	.17 ± .01	.39 ± .02
8	.72 ± .38	.20 ± .06	.27 ± .04	.16 ± .06	.48 ± .29
16	.90 ± .01	.24 ± .01	.29 ± .01	.12 ± .01	.35 ± .02
Weak scaling setting					
1	.56 ± .47	.16 ± .07	.22 ± .03	.12 ± .07	.56 ± .40
2	.69 ± .37	.17 ± .05	.25 ± .04	.12 ± .06	.49 ± .30
4	.72 ± .38	.20 ± .07	.27 ± .05	.15 ± .06	.49 ± .29
8	.90 ± .04	.18 ± .10	.24 ± .13	.13 ± .08	.30 ± .17
16	.55 ± .46	.17 ± .08	.26 ± .06	.11 ± .06	.63 ± .34

The main difference between these two metrics is about the relative importance of precision and recall: F_1 gives them the same importance, while F_2 is better for cases where precision is to be considered twice as important as recall. Table 4.2 reports the results obtained with the federated FNN with FedAvg model, while Table 4.3 refers to the `AdaBoost.F` ensemble.

We start by noticing that, from the point of view of accuracy, `AdaBoost.F` dominates by reaching 95% accuracy in most experiments[6]. As mentioned, however, accuracy is a bad metric in this particular case, and the high accuracy values suggest that the ensemble model probably categorises most of the examples as negatives. The rest of the metrics confirm this intuition: recall values are much lower in the case of `AdaBoost.F` than in the case of the FNN with FedAvg model. Accuracies, in the case of the FNN with FedAvg model, are much more erratic, showing both a higher variance as the number of collaborators grows and a higher variance in the 5 experiment repetitions. However, the important metrics (F_1 and F_2) show much better

[6]In Table 4.3 the accuracy column shows the average ± stdev values zero due to too low values as we rounded the values to up two digits

Table 4.3: Prediction performance of `AdaBoost.F`. Values reported are the average ± stdev of 5 runs. The first run in the strong scaling setting is equivalent to the non-federated case.

Clients	Accuracy	F1 Score	F2 Score	Precision	Recall
Strong scaling setting					
1	.95 ± .00	.19 ± .07	.15 ± .06	.35 ± .10	.13 ± .05
2	.95 ± .00	.23 ± .03	.19 ± .03	.36 ± .04	.17 ± .03
4	.94 ± .00	.19 ± .02	.16 ± .02	.26 ± .04	.15 ± .02
8	.94 ± .00	.20 ± .04	.17 ± .03	.28 ± .06	.16 ± .03
16	.94 ± .00	.19 ± .03	.17 ± .03	.25 ± .04	.16 ± .03
Weak scaling setting					
1	.95 ± .00	.09 ± .02	.06 ± .01	.33 ± .05	.05 ± .01
2	.95 ± .00	.10 ± .02	.07 ± .01	.45 ± .05	.05 ± .01
4	.95 ± .00	.15 ± .04	.12 ± .04	.32 ± .06	.10 ± .10
8	.95 ± .00	.17 ± .02	.14 ± .01	.28 ± .04	.13 ± .01
16	.94 ± .00	.20 ± .03	.18 ± .02	.27 ± .04	.16 ± .02

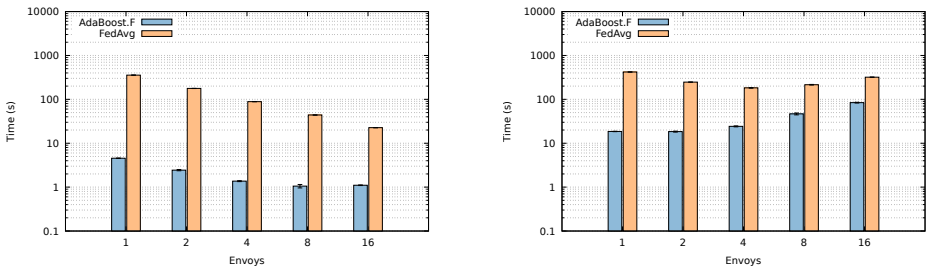
performances of the FNN with FedAvg models w.r.t. the ensemble model. Indeed, as far as we can tell, results reported in table 4.2 are even better than those shown in the state-of-the-art technique [139] (the paper that introduced the dataset and the PRAISE score).

While these are indeed good news, we refrain from calling these models better because the models in [139] have been thoroughly evaluated under a multitude of aspects, not just F_1 and F_2. Nonetheless, we plan to investigate these models in the future further. Another interesting facet of the reported FNN with FedAvg results is that the F metrics do not follow a growing pattern as the number of collaborators grows. They show an inverted v shape, which is difficult to explain. In fact, at least in the weak scaling setting, we would have expected the performances to continue to grow since adding more collaborators amount, in this latter case, to increasing the dataset size. A possible explanation might be that the FL procedure finds it difficult to leverage all the available data when the number of involved parties grows (to the point of making adding new parties to the federation no longer worthwhile).

In summary, from the point of view of prediction quality, quite unexpectedly, the FNN with FedAvg model appears to be better than the ensemble

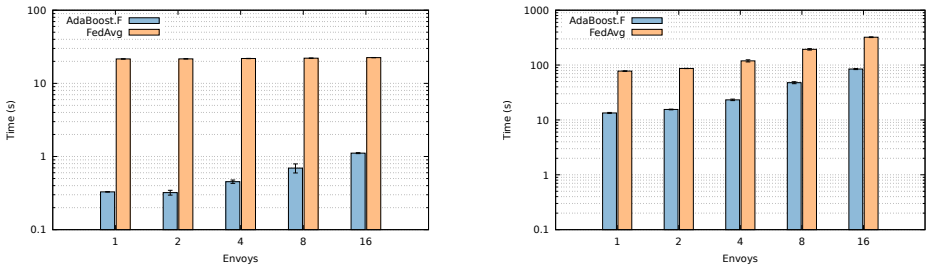
of trees acquired by `AdaBoost.F`. Whether they are overall better models is, however, to be decided since, for the specific case of health institutions, other factors (above all, the interpretability of the results) might prevail in evaluating the possible solutions.

As stated earlier, this dataset is highly imbalanced, so it is tough to get good F_1 and F_2. However, it was an unexpected observation that the best F_1 scores are for models acquired in the federated case, even in the strong scaling setting. Contrary to our expectations, the classical case (corresponding to the first line of the strong scaling experiments) was not the best. From the point of view of training the model, this is the easiest configuration, where all data are located in a single point, and there are no privacy issues. This is a further point to be further investigated, but a possible explanation could be that the FL process acts as a regularization factor preventing the model from overfitting the training set.



(a) Computation time (for a single envoy) (b) Total time

Figure 4.3: `AdaBoost.F` and FedAvg training time for 100 rounds executed on the C3S machines in the strong scaling setting.



(a) Computation time (for a single envoy) (b) Total time

Figure 4.4: `AdaBoost.F` and FedAvg training time for 100 rounds executed on the C3S machines in the weak scaling setting.

Fig. 4.3 and 4.4 report the execution times of 100 training rounds for the FNN model and the `AdaBoost.F` ensemble in the strong and weak scaling

settings, respectively. In the strong scaling setting, the size of the pooled dataset is constant; data is equally partitioned among envoys, whereas in the weak scaling setting, the size of data associated with each envoy is constant; the more envoy, the more data. The baseline is the strong scaling setting with a single envoy, which is identical to a non-federated process.

The most relevant thing to notice is that training an FNN with FedAvg is between 5 and 30 times longer than training an ensemble of decision trees with `AdaBoost.F` in both settings. However, this gap is much more evident when considering only the actual computation time, as the overhead introduced by serialization and communication is much more evident with `AdaBoost.F`. Plus, the communication time appears to be the actual bottleneck in the overall execution, as the total training time increases with the number of federation members. This finding suggests that the benefit of using `AdaBoost.F` will be much more evident with a more efficient FL framework, whose development is already in our future research plan.

Analyzing the two algorithms' strong and weak scaling behaviour, it is worth noting that the FNN with FedAvg model follows a common trend in both settings. Indeed, the total time to solution decreases with more collaborators in the strong scaling setting, while it remains almost constant in the weak scaling one. Conversely, with `AdaBoost.F` the time to solution decreases only up to 8 collaborators in the strong scaling setting, and it linearly grows up in the weak scaling setting. This behaviour is justified by the fact that the second phase of the algorithm requires each collaborator to evaluate n decision trees on the local data to determine the best one, where n is the number of collaborators in the federation. We are planning a more efficient version of the algorithm in the future, aiming to reduce the computation overhead to select the best model at each round.

Chapter 5

Conclusion

AI is a growing force in everyday life, and its usage is slowly but consistently percolating in medical professions. In most cases, ML is the main force driving the adoption of AI. On the other hand, FL is an essential tool for building machine learning models across parties that must maintain their local datasets' privacy. In this thesis, we first provided a comprehensive review of the different types and categories of FL. We also explored the various FL tools that have been utilized and presented an overview of the recent literature in the field of FL.

Second, we presented the main applications of ML in recent cardiology literature. We provided an introduction of the techniques used most often, reviewed competing statistical methods, and critically reviewed these tools' usage in recent applications and research. We found that in most cases, the usage of ML is limited to tools that have been firmly understood for many years. In our opinion, newer and more data-hungry approaches are currently under-represented. Indeed, one of the problems we outline is the paucity of very large datasets. In fact, the efforts to build such datasets are hampered by difficulties in labeling vast amounts of data and privacy concerns that do not allow merging datasets acquired by different institutions. We suggested exploiting semi-supervised techniques to tackle the labeling problem and combine Federated Learning and Differential Privacy to overcome privacy issues. Medical data is challenging to collect, usually noisy, and the involved tasks are hard to solve. ML can be very useful to ease the burden on physicians, and, in part, it is already helping in that area. We hope that, with the improvements in data collection and their sharing, better models will be learned; physicians will be able to work faster and more accurately, and, ultimately, many lives will be saved.

Third, we also applied ML techniques to the cardiology dateset for the prediction of mortality. We achieved good results, and we were also able to develop a web-based PRAISE score tool for predicting Death, BARCMB, and RENAMI. It was pretty successful, and we imagine applying FL to the same datasets will be useful as it was not explored. Then finally, we applied FL techniques to the PRAISE dataset. We also found that the most widely used FL algorithms are based on some variation of the gradient descent algorithm and are used to train neural networks. However, neural networks are not always desirable models and are often harder to train when tabular data is involved. We investigated how the FedAvg algorithm and `AdaBoost.F` compare for the task of predicting all-cause mortality. This exploratory study constitutes a proof-of-concept of a more general paradigm, the *federated pooling*, which allows to compose of complex "virtual" pooled datasets while leaving the actual data undisclosed at the edge.

Edge computing is a distributed computing paradigm that brings computation and data storage closer to the data source[146]. Edge computing developed with the expectation of keeping part of the computation close to the data sources reducing latency and bandwidth requirements in processing inherently distributed data streams (as opposite to BigData batch processing typical of cloud computing). FL addresses the same conceptual architecture, where data is mainly processed near the data sources, where the two presented FL settings (cross-device and cross-site) define their features in terms of scalability, computing power, energy efficiency, security, and reliability. In this work, we mainly focus on cross-site scenarios, which fit a consortium of trusted data owners (e.g., hospitals) connected with a secure and reliable network that do not wish to share their private data. In this respect, FL might be considered an example of edge computing that extends original motivations beyond execution performance.

The PRAISE score offers a prediction performance far better than similar scores [139]. Notice that the pooled dataset used to train the PRAISE score model is orders of magnitude larger than those driving similar studies, typically gathered within a single organization. A basic assumption underneath all FL algorithms is that they can *approximate* a traditional model trained on a pooled dataset, making it possible to train models on an increasingly larger dataset from different organizations.

There are two assumptions subject to empirical verification. First, a consortium of organizations, each of them owning private data, can train a model virtually pooling all datasets via FL, and this model *exhibits a*

comparable prediction performance of the best-known model that can be built from the union of all datasets stored in a single data lake. Second, the kind of models that can be trained is not limited to ANN but encompasses explainable ML models, such as decision trees. Formally proving these assumptions goes far beyond the scope of the present work. Nevertheless, the empirical validation makes it possible to envision entirely novel data operation for edge systems enabling data analysis: *federated pooling*.

Generally speaking, the two basic operations on data are *read* and *write*. Concurrent systems (e.g., parallel, distributed) require an additional *atomic-read-write* primary operation to enforce data integrity on shared data, which is needed to support data sharing primitives among concurrent activities (e.g., transactions). A further step in distributed systems is reaching a *consensus* among parties, i.e., agreeing on some data value needed during computation. A distributed consensus protocol makes it possible to implement a distributed ledger to distinguish a digital object from its copy.

Federated pooling can make a further step in distributed data management since it can virtually pool many datasets for a specific data analytic task. Federated pooling does not subsume data operations (read, write, compare-and-swap); it is stateless because it does not permanently affect data. For this, it can be re-executed many times and with different organizations, thus finely controlled and billed. We envision federated pooling as the basic API of a new kind of service for edge computing: FL-as-a-Service (FLaaS).

Finaly, we compared the performance of these two methods from the point of view of prediction performances as well as from the point of view of computation and communication time. While we expected `AdaBoost.F` to be a better solution for this specific use case, we found mixed results: the decision trees trained by `AdaBoost.F` are faster to train, but the resulting ensemble does not perform as well as the FedAvg model prediction-wise. Computationally `AdaBoost.F` seems to require less resources, but it does not scale well as the number of involved parties grow (a less demanding algorithm is being developed by the original authors, but we couldn't test it as it has not been released yet).

Chapter 6

Future work and directions

In this thesis, we explored the PRAISE dataset for the experiment in federated and none federated settings. For future work, in addition to further investigate the unexpected good performances of the FNN model, we would also like to explore this dataset's two other target variables, such as RENAMI and BARCMB. We would also like to better understand the power consumption of these two techniques. In addition, as previously discussed in chapter 2, numerous tools are currently available. We will perform experiments on various federated learning tools using the PRAISE datasets.

Due to its ability to analyze and interpret vast amounts of patient data, natural language processing (NLP) technology is becoming increasingly prevalent in the healthcare industry. By utilizing advanced algorithms and machine learning, NLP can uncover valuable insights from clinical notes that were previously inaccessible, thereby aiding healthcare providers in comprehending quality, refining methodologies, and achieving better outcomes for patients. As physicians devote considerable time to inputting data into electronic health record systems, NLP can accurately extract unstructured data for further analysis. Ultimately, NLP has the potential to enhance patient care by providing valuable insights into healthcare data. The sensitivity of healthcare data requires that it be handled in a manner that ensures its privacy. While NLP has not been widely used in Federated Learning (FL), there is potential for further exploration in this area, making it an interesting prospect.

Federated learning technique that allows multiple parties to collaboratively train a model without sharing their data with each other. Blockchain, on the other hand, is a distributed ledger technology that allows multiple parties to communicate and update a ledger in a secure and decentralized

manner. Combining federated learning with blockchain technology has the potential to enhance data privacy and security in machine learning applications. By storing the federated learning model on the blockchain, all participants can access and verify the model's integrity without exposing their data to each other. Additionally, smart contracts can be used to govern the federated learning process and ensure that all participants follow the agreed-upon rules. Finally, we will work on developing a Federated Learning as a Service (FLaaS) infrastructure for federated pooling in the continuum, which will serve as a framework to experiment with novel ML/DNN models and datasets coming from a diverse set of use cases.

References

[1] F. J. Provost and T. Fawcett, "Authors' response to gong's, "comment on data science and its relationship to big data and data-driven decision making"," *Big Data*, vol. 2, no. 1, p. 1, 2014.

[2] H. Nozari, J. Ghahremani-Nahr, and A. Szmelter-Jarosz, "Ai and machine learning for real-world problems," ser. Advances in Computers. Elsevier, 2023.

[3] I. Colonnelli, B. Casella, G. Mittone, Y. Arfat, B. Cantalupo, R. Esposito, A. R. Martinelli, D. Medić, and M. Aldinucci, "Federated learning meets HPC and cloud," in *Astrophysics and Space Science Proceedings*. Springer, 2022.

[4] C. Aone, M. E. Okurowski, and J. Gorlinsky, "Trainable, scalable summarization using robust NLP and machine learning," in *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL '98, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference*, C. Boitet and P. Whitelock, Eds. Morgan Kaufmann Publishers / ACL, 1998, pp. 62–66.

[5] S. Kausar, X. Huahu, W. Ahmad, and M. Y. Shabir, "A sentiment polarity categorization technique for online product reviews," *IEEE Access*, vol. 8, pp. 3594–3605, 2019.

[6] E. H. Almansor and F. K. Hussain, "Survey on intelligent chatbots: State-of-the-art and future research directions," in *Complex, Intelligent, and Software Intensive Systems - Proceedings of the 13th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2019, Sydney, NSW, Australia, 3-5 July 2019*, ser. Advances in Intelligent Systems and Computing, L. Barolli, F. K. Hussain, and M. Ikeda, Eds., vol. 993. Springer, 2019, pp. 534–543.

[7] V. Oliinyk, V. Vysotska, Y. Burov, K. Mykich, and V. B. Fernandes, "Propaganda detection in text data based on NLP and machine learning," in *Proceedings of the 2nd International Workshop on Modern Machine Learning Technologies and Data Science (MoMLeT+DS 2020). Volume I: Main Conference, Lviv-Shatsk, Ukraine, June 2-3, 2020*, ser. CEUR Workshop Proceedings, M. Emmerich, V. Lytvyn, V. Vysotska, V. B. Fernandes, and V. Lytvynenko, Eds., vol. 2631. CEUR-WS.org, 2020, pp. 132–144.

[8] Z. Jin, Z. Zhang, and G. X. Gu, "Automated real-time detection and prediction of interlayer imperfections in additive manufacturing processes using artificial intelligence," *Adv. Intell. Syst.*, vol. 2, no. 1, p. 1900130, 2020.

[9] A. Ali, S. Abd Razak, S. H. Othman, T. A. E. Eisa, A. Al-Dhaqm, M. Nasser, T. Elhassan, H. Elshafie, and A. Saif, "Financial fraud detection based on machine learning: A systematic literature review," *Applied Sciences*, vol. 12, no. 19, p. 9637, 2022.

[10] A. Mashrur, W. Luo, N. A. Zaidi, and A. Robles-Kelly, "Machine learning for financial risk management: A survey," *IEEE Access*, vol. 8, pp. 203 203–203 223, 2020.

[11] M. S. Medikonduru, A. Devadari, S. C. Kasaraneni, M. B. Thota, and S. H. Yakkala, "Traffic prediction for an intelligent transportation system using ml," in *2022 International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2022, pp. 1206–1213.

[12] H. Thadeshwar, V. Shah, M. Jain, R. Chaudhari, and V. Badgujar, "Artificial intelligence based self-driving car," in *2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*. IEEE, 2020, pp. 1–5.

[13] K. Singh, P. Booma, and U. Eaganathan, "E-commerce system for sale prediction using machine learning technique," in *Journal of Physics: Conference Series*, vol. 1712, no. 1. IOP Publishing, 2020, p. 012042.

[14] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 12:1–12:19, 2019.

[15] B. Casella, R. Esposito, A. Sciarappa, C. Cavazzoni, and M. Aldinucci, "Experimenting with normalization layers in federated learning

on non-iid scenarios," Computer Science Department, University of Torino, Tech. Rep., 2023.

[16] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. of the 20th Intl. Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, ser. Proc. of Machine Learning Research, A. Singh and X. J. Zhu, Eds., vol. 54. PMLR, 2017, pp. 1273–1282.

[17] G. Mittone, N. Tonci, R. Birke, I. Colonnelli, D. Medić, A. Bartolini, R. Esposito, E. Parisi, F. Beneventi, M. Polato, M. Torquati, L. Benini, and M. Aldinucci, "Experimenting with emerging arm and risc-v systems for decentralised machine learning," Computer Science Department, University of Torino, Tech. Rep., 2023.

[18] I. Feki, S. Ammar, Y. Kessentini, and K. Muhammad, "Federated learning for covid-19 screening from chest x-ray images," *Applied Soft Computing*, vol. 106, p. 107330, 2021.

[19] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *Journal of Healthcare Informatics Research*, vol. 5, pp. 1–19, 2021.

[20] G. Mittone, F. Svoboda, M. Aldinucci, N. D. Lane, and P. Lio, "A federated learning benchmark for drug-target interaction," Computer Science Department, University of Torino, Tech. Rep., 2023.

[21] P. Foley, M. J. Sheller, B. Edwards, S. Pati, W. Riviera, M. Sharma, P. N. Moorthy, S.-h. Wang, J. Martin, P. Mirhaji, P. Shah, and S. Bakas, "OpenFL: the open federated learning library," *Physics in Medicine & Biology*, 2022.

[22] K. Bonawitz, E. Hubert, W. Grieskamp, H. Dzmitry, I. Alex, I. Vladimir, K. Chloe, K. Jakub, M. Stefano, M. Brendan *et al.*, "Towards federated learning at scale: System design," *Proceedings of Machine Learning and Systems*, vol. 1, pp. 374–388, 2019.

[23] Z. Alexander, T. Andrew, L. Antonio, S. Benjamin, W. Bobby, B. Emma, N. Jean-Mickael, P.-P. Jonathan, P. Kritika, R. Nick *et al.*, "Pysyft: A library for easy federated learning," in *Federated Learning Systems*. Springer, 2021, pp. 111–139.

[24] L. Heiko, B. Nathalie, T. Gegi, Z. Yi, A. Ali, R. Shashank, O. Yuya, J. Radhakrishnan, V. Ashish, S. Mathieu *et al.*, "Ibm federated learning: an enterprise framework white paper v0. 1," *arXiv preprint arXiv:2007.10987*, 2020.

[25] S. Warnat-Herresthal, H. Schultze, K. L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N. A. Aziz *et al.*, "Swarm Learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, Jun. 2021.

[26] C. He, S. Li, J. So, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu, L. Shen, P. Zhao, Y. Kang, Y. Liu, R. Raskar, Q. Yang, M. Annavaram, and S. Avestimehr, "Fedml: A research library and benchmark for federated machine learning," *CoRR*, vol. abs/2007.13518, 2020.

[27] Y. Liu, T. Fan, T. Chen, Q. Xu, and Q. Yang, "FATE: an industrial grade platform for collaborative learning with data protection," *J. Mach. Learn. Res.*, vol. 22, pp. 226:1–226:6, 2021.

[28] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, "Flower: A friendly federated learning research framework," *CoRR*, vol. abs/2007.14390, 2020.

[29] S. Silva, A. Altmann, B. Gutman, and M. Lorenzi, "Fed-biomed: A general open-source frontend framework for federated learning in healthcare," in *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning - Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4-8, 2020, Proceedings*, ser. Lecture Notes in Computer Science, vol. 12444. Springer, 2020, pp. 201–210.

[30] Y. Xie, Z. Wang, D. C. an Dawei Gao, L. Yao, W. Kuang, Y. Li, B. Ding, and J. Zhou, "Federatedscope: A comprehensive and flexible federated learning platform via message passing," *CoRR*, vol. abs/2204.05011, 2022.

[31] D. Dimitriadis, M. H. Garcia, D. M. Diaz, A. Manoel, and R. Sim, "FLUTE: A scalable, extensible framework for high-performance federated learning simulations," *CoRR*, vol. abs/2203.13789, 2022.

[32] N. Wang, Y. Xiao, Y. Chen, Y. Hu, W. Lou, and Y. T. Hou, "FLARE: defending federated learning against model poisoning attacks via latent space representations," in *ASIA CCS '22: ACM Asia Conference on Computer and Communications Security, Nagasaki, Japan, 30 May 2022 - 3 June 2022*, Y. Suga, K. Sakurai, X. Ding, and K. Sako, Eds. ACM, 2022, pp. 946–958.

[33] S. Pati, U. Baid, B. Edwards, M. Sheller, S.-H. Wang, G. A. Reina, P. Foley, A. Gruzdev, D. Karkada, C. Davatzikos *et al.*, "Federated learning enables big data for rare cancer boundary detection," *arXiv preprint arXiv:2204.10836*, 2022.

[34] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, "Flower: A friendly federated learning research framework," *CoRR*, vol. abs/2007.14390, 2020.

[35] S. Silva, A. Altmann, B. Gutman, and M. Lorenzi, "Fed-biomed: A general open-source frontend framework for federated learning in healthcare," in *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning - Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4-8, 2020, Proceedings*, ser. Lecture Notes in Computer Science, vol. 12444. Springer, 2020, pp. 201–210.

[36] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[37] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[38] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays, "Federated learning for emoji prediction in a mobile keyboard," *arXiv preprint arXiv:1906.04329*, 2019.

[39] Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Z. Li, L. Lyu, and Y. Liu, "Privacy-preserving blockchain-based federated learning for iot devices," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1817–1829, 2020.

[40] S. Silva, B. A. Gutman, E. Romero, P. M. Thompson, A. Altmann, and M. Lorenzi, "Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, 2019, pp. 270–274.

[41] A. Durrant, M. Markovic, D. Matthews, D. May, J. Enright, and G. Leontidis, "The role of cross-silo federated learning in facilitating data sharing in the agri-food sector," *Computers and Electronics in Agriculture*, vol. 193, p. 106648, 2022.

[42] D. Byrd and A. Polychroniadou, "Differentially private secure multi-party computation for federated learning in financial applications," in *Proceedings of the First ACM International Conference on AI in Finance*, 2020, pp. 1–9.

[43] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[44] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.

[45] D. M. J. Gutierrez, H. M. Hassan, L. Landi, A. Vitaletti, and I. Chatzigiannakis, "Application of federated learning techniques for arrhythmia classification using 12-lead ecg signals," *arXiv preprint arXiv:2208.10993*, 2022.

[46] Y. Arfat, G. Mittone, R. Esposito, B. Cantalupo, G. M. de Ferrari, and M. Aldinucci, "Machine learning for cardiology," *Minerva Cardiology and Angiology 2022 February; 70 (1): 75-91.*

[47] Y. Liu, Z. Ma, X. Liu, S. Ma, S. Nepal, R. H. Deng, and K. Ren, "Boosting privately: Federated extreme gradient boosting for mobile crowdsensing," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, 2020, pp. 1–11.

[48] C. Yuwen and Q. Baolian, "Representation learning in intraoperative vital signs for heart failure risk prediction," *BMC Medical Informatics and Decision Making*, vol. 19, no. 1, pp. 1–15, 2019.

[49] E. D. Adler, A. A. Voors, L. Klein, F. Macheret, O. O. Braun, M. A. Urey, W. Zhu, I. Sama, M. Tadel, C. Campagnari *et al.*, "Improving risk prediction in heart failure using machine learning," *European journal of heart failure*, vol. 22, no. 1, pp. 139–147, 2020.

[50] C. Ye, J. Li, S. Hao, M. Liu, H. Jin, L. Zheng, M. Xia, B. Jin, C. Zhu, S. T. Alfreds *et al.*, "Identification of elders at higher risk for fall with statewide electronic health records and a machine learning algorithm," *International journal of medical informatics*, vol. 137, p. 104105, 2020.

[51] K. Arman, G. Anshul, M. J. K, G. Eva, T. W. Lam, T. G. Gleason, I. Sultan, and A. Dubrawksi, "Predictive utility of a machine learning algorithm in estimating mortality risk in cardiac surgery," *The Annals of Thoracic Surgery*, vol. 109, no. 6, pp. 1811–1819, 2020.

[52] M. Polato, R. Esposito, and M. Aldinucci, "Boosting the federation: Cross-silo federated learning without gradient descent," in *International Joint Conference on Neural Networks, IJCNN 2022, Padua, Italy, July 18-23, 2022*. IEEE, 2022, pp. 1–10.

[53] H. Shoukourian, T. Wilde, A. Auweter, and A. Bode, "Predicting the energy and power consumption of strong and weak scaling hpc applications," *Supercomputing frontiers and innovations*, vol. 1, no. 2, pp. 20–41, 2014.

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

[55] H. Pham, Z. Dai, Q. Xie, and Q. V. Le, "Meta pseudo labels," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 11 557–11 568.

[56] J. Kremer, K. Stensbo-Smidt, F. Gieseke, K. S. Pedersen, and C. Igel, "Big universe, big data: Machine learning and image analysis for astronomy," *IEEE Intell. Syst.*, vol. 32, no. 2, pp. 16–22, 2017.

[57] W. Brenner, R. Zarnekow, and H. Wittig, *Intelligent software agents - foundations and applications*. Springer, 1998.

[58] A. S. d'Avila Garcez, T. R. Besold, L. D. Raedt, P. Földiák, P. Hitzler, T. Icard, K. Kühnberger, L. C. Lamb, R. Miikkulainen, and D. L.

Silver, "Neural-symbolic learning and reasoning: Contributions and challenges," in *2015 AAAI Spring Symposia, Stanford University, Palo Alto, California, USA, March 22-25, 2015*. AAAI Press, 2015.

[59] D. E. Wilkins, *Practical planning - extending the classical AI planning paradigm*, ser. Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, 1988.

[60] B. López, *Case-Based Reasoning: A Concise Introduction*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2013.

[61] Y. R. R and Z. L. A, *An introduction to fuzzy logic applications in intelligent systems*. Springer Science & Business Media, 2012, vol. 165.

[62] L. Jay, *The handbook of applied expert systems*. cRc Press, 2019.

[63] P. I. Dorado-Díaz, J. Sampedro-Gómez, V. Vicente-Palacios, and P. L. Sánchez, "Applications of artificial intelligence in cardiology. the future is already here," *Revista Española de Cardiología (English Edition)*, vol. 72, no. 12, pp. 1065–1075, 2019.

[64] B. Diana, "Artificial intelligence in cardiology," *Wiener Klinische Wochenschrift*, vol. 129, no. 23, pp. 866–868, 2017.

[65] P. Sardar, J. D. Abbott, A. Kundu, H. D. Aronow, J. F. Granada, and J. Giri, "Impact of artificial intelligence on interventional cardiology: from decision-making aid to advanced interventional procedure assistance," *Cardiovascular Interventions*, vol. 12, no. 14, pp. 1293–1303, 2019.

[66] I. Colonnelli, B. Cantalupo, I. Merelli, and M. Aldinucci, "Streamflow: Cross-breeding cloud with HPC," *IEEE Trans. Emerg. Top. Comput.*, vol. 9, no. 4, pp. 1723–1737, 2021.

[67] P. A. Flach, *Machine Learning - The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press, 2012.

[68] E. C. O'Brien1, D. N. Simon, L. E. Thomas, E. M. Hylek, B. J. Gersh, J. E. Ansell, P. R. Kowey, K. W. Mahaffey, P. Chang, G. C. Fonarow, M. J. Pencina, J. P. Piccini, , and E. D. Peterson, "The orbit bleeding score: a simple bedside score to assess bleeding risk in atrial

fibrillation," *European heart journal*, vol. 36, no. 46, pp. 3258–3264, 2015.

[69] U. Baber, R. Mehran, G. Giustino, D. J. Cohen, T. D. Henry, S. Sartori, C. Ariti, C. Litherland, G. Dangas, C. M. Gibson *et al.*, "Coronary thrombosis and major bleeding after pci with drug-eluting stents: risk scores from paris," *Journal of the American College of Cardiology*, vol. 67, no. 19, pp. 2224–2234, 2016.

[70] F. Costa, D. Van Klaveren, S. James, D. Heg, L. Räber, F. Feres, T. Pilgrim, M.-K. Hong, H.-S. Kim, A. Colombo *et al.*, "Derivation and validation of the predicting bleeding complications in patients undergoing stent implantation and subsequent dual antiplatelet therapy (precise-dapt) score: a pooled analysis of individual-patient datasets from clinical trials," *The Lancet*, vol. 389, no. 10073, pp. 1025–1034, 2017.

[71] D. Han, K. K. Kolli, H. Gransar, J. H. Lee, S.-Y. Choi, E. J. Chun, H.-W. Han, S. H. Park, J. Sung, H. O. Jung *et al.*, "Machine learning based risk prediction model for asymptomatic individuals who underwent coronary artery calcium score: Comparison with traditional risk prediction approaches," *Journal of cardiovascular computed tomography*, vol. 14, no. 2, pp. 168–176, 2020.

[72] F. D'Ascenzo, O. De Filippo, G. Gallone, G. Mittone, M. A. Deriu, M. Iannaccone, A. Ariza-Solé, C. Liebetrau, S. Manzano-Fernández, G. Quadri *et al.*, "Machine learning-based prediction of adverse events following an acute coronary syndrome (praise): a modelling study of pooled datasets," *The Lancet*, vol. 397, no. 10270, pp. 199–207, 2021.

[73] P. Guimarães, A. Keller, M. Böhm, L. Lauder, J. L. Ayala, J. R. Banegas, A. de la Sierra, E. Vinyoles, M. Gorostidi, J. Segura *et al.*, "Risk prediction with office and ambulatory blood pressure using artificial intelligence," *medRxiv*, 2020.

[74] S. E. Awan, M. Bennamoun, F. Sohel, F. M. Sanfilippo, and G. Dwivedi, "Machine learning-based prediction of heart failure readmission or death: implications of choosing the right model and the right metrics," *ESC heart failure*, vol. 6, no. 2, pp. 428–435, 2019.

[75] S. J. Pocock, C. A. Ariti, J. J. McMurray, A. Maggioni, L. Køber, I. B. Squire, K. Swedberg, J. Dobson, K. K. Poppe, G. A. Whalley

et al., "Predicting survival in heart failure: a risk score based on 39 372 patients from 30 studies," *European heart journal*, vol. 34, no. 19, pp. 1404–1413, 2013.

[76] X. Wang, Y.-Q. Yang, S.-H. Liu, X.-Y. Hong, X.-F. Sun, and J.-h. Shi, "Comparing different venous thromboembolism risk assessment machine learning models in chinese patients," *Journal of Evaluation in Clinical Practice*, vol. 26, no. 1, pp. 26–34, 2020.

[77] S. E. Awan, M. Bennamoun, F. Sohel, F. M. Sanfilippo, B. J. Chow, and G. Dwivedi, "Feature selection and transformation by machine learning reduce variable numbers and improve prediction for heart failure readmission or death," *PloS one*, vol. 14, no. 6, p. e0218760, 2019.

[78] M. Mamprin, J. M. Zelis, P. A. L. Tonino, S. Zinger, and P. H. N. de With, "Gradient boosting on decision trees for mortality prediction in transcatheter aortic valve implantation," *CoRR*, vol. abs/2001.02431, 2020.

[79] D. Schmidt, M. Niemann, and G. L. von Trzebiatowski, "The handling of missing values in medical domains with respect to pattern mining algorithms," in *Proceedings of the 24th International Workshop on Concurrency, Specification and Programming, Rzeszow, Poland, September 28-30, 2015*, ser. CEUR Workshop Proceedings, Z. Suraj and L. Czaja, Eds., vol. 1492. CEUR-WS.org, 2015, pp. 147–154.

[80] M. Tokodi, W. R. Schwertner, A. Kovács, Z. Tősér, L. Staub, A. Sárkány, B. K. Lakatos, A. Behon, A. M. Boros, P. Perge *et al.*, "Machine learning-based mortality prediction of patients undergoing cardiac resynchronization therapy: the semmelweis-crt score," *European heart journal*, vol. 41, no. 18, pp. 1747–1756, 2020.

[81] J. W. O'Sullivan, A. Shcherbina, J. M. Justesen, M. Turakhia, M. Perez, H. Wand, C. Tcheandjieu, S. L. Clarke, M. A. Rivas, and E. A. Ashley, "Combining clinical and polygenic risk improves stroke prediction among individuals with atrial fibrillation," *Circulation: Genomic and Precision Medicine*, vol. 14, no. 3, p. e003168, 2021.

[82] R. Vazquez, A. Bayes-Genis, I. Cygankiewicz, D. Pascual-Figal, L. Grigorian-Shamagian, R. Pavon, J. R. Gonzalez-Juanatey, J. M. Cubero, L. Pastor, J. Ordonez-Llanos *et al.*, "The music risk score: a

simple method for predicting mortality in ambulatory patients with chronic heart failure," *European heart journal*, vol. 30, no. 9, pp. 1088–1096, 2009.

[83] F. Cismondi, A. S. Fialho, S. M. Vieira, S. R. Reti, J. M. da Costa Sousa, and S. N. Finkelstein, "Missing data in medical databases: Impute, delete or classify?" *Artif. Intell. Medicine*, vol. 58, no. 1, pp. 63–72, 2013.

[84] K. J. Janssen, A. R. T. Donders, F. E. Harrell Jr, Y. Vergouwe, Q. Chen, D. E. Grobbee, and K. G. Moons, "Missing covariate data in medical research: to impute is better than to ignore," *Journal of clinical epidemiology*, vol. 63, no. 7, pp. 721–727, 2010.

[85] K. D. Aaronson, J. S. Schwartz, T.-M. Chen, K.-L. Wong, J. E. Goin, and D. M. Mancini, "Development and prospective validation of a clinical index to predict survival in ambulatory patients referred for cardiac transplant evaluation," *Circulation*, vol. 95, no. 12, pp. 2660–2667, 1997.

[86] W. C. Levy, D. Mozaffarian, D. T. Linker, S. C. Sutradhar, S. D. Anker, A. B. Cropp, I. Anand, A. Maggioni, P. Burton, M. D. Sullivan *et al.*, "The seattle heart failure model: prediction of survival in heart failure," *Circulation*, vol. 113, no. 11, pp. 1424–1433, 2006.

[87] S. Angraal, B. J. Mortazavi, A. Gupta, R. Khera, T. Ahmad, N. R. Desai, D. L. Jacoby, F. A. Masoudi, J. A. Spertus, and H. M. Krumholz, "Machine learning prediction of mortality and hospitalization in heart failure with preserved ejection fraction," *JACC: Heart Failure*, vol. 8, no. 1, pp. 12–21, 2020.

[88] X. Li, X. Xu, F. Xie, X. Xu, Y. Sun, X. Liu, X. Jia, Y. Kang, L. Xie, F. Wang *et al.*, "A time-phased machine learning model for real-time prediction of sepsis in critical care," *Critical Care Medicine*, vol. 48, no. 10, pp. e884–e888, 2020.

[89] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*. Ieee, 2015, pp. 1200–1205.

[90] M. M. Rahman and D. N. Davis, "Addressing the class imbalance problem in medical datasets," *International Journal of Machine Learning and Computing*, vol. 3, no. 2, p. 224, 2013.

[91] C. Ricciardi, K. J. Edmunds, M. Recenti, S. Sigurdsson, V. Gudnason, U. Carraro, and P. Gargiulo, "Assessing cardiovascular risks from a mid-thigh ct image: a tree-based machine learning approach using radiodensitometric distributions," *Scientific reports*, vol. 10, no. 1, pp. 1–13, 2020.

[92] H. A. G. Elsayed and L. Syed, "An automatic early risk classification of hard coronary heart diseases using framingham scoring model," in *Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing*, 2017, pp. 1–8.

[93] R. Providencia, E. Marijon, S. Barra, C. Reitan, A. Breitenstein, P. Defaye, N. Papageorgiou, R. Duehmke, S. Winnik, R. Ang *et al.*, "Usefulness of a clinical risk score to predict the response to cardiac resynchronization therapy," *International Journal of Cardiology*, vol. 260, pp. 82–87, 2018.

[94] K. Furui, I. Morishima, Y. Morita, Y. Kanzaki, K. Takagi, R. Yoshida, H. Nagai, N. Watanabe, N. Yoshioka, R. Yamauchi *et al.*, "Predicting long-term freedom from atrial fibrillation after catheter ablation by a machine learning algorithm: Validation of the caap-af score," *Journal of Arrhythmia*, vol. 36, no. 2, pp. 297–303, 2020.

[95] J.-m. Kwon, K.-H. Kim, K.-H. Jeon, S. E. Lee, H.-Y. Lee, H.-J. Cho, J. O. Choi, E.-S. Jeon, M.-S. Kim, J.-J. Kim *et al.*, "Artificial intelligence algorithm for predicting mortality of patients with acute heart failure," *PloS one*, vol. 14, no. 7, p. e0219302, 2019.

[96] D. F. Hernandez-Suarez, Y. Kim, P. Villablanca, T. Gupta, J. Wiley, B. G. Nieves-Rodriguez, J. Rodriguez-Maldonado, R. Feliu Maldonado, I. da Luz Sant'Ana, C. Sanina *et al.*, "Machine learning prediction models for in-hospital mortality after transcatheter aortic valve replacement," *Cardiovascular Interventions*, vol. 12, no. 14, pp. 1328–1338, 2019.

[97] L. Yang, H. Wu, X. Jin, P. Zheng, S. Hu, X. Xu, W. Yu, and J. Yan, "Study of cardiovascular disease prediction model based on random

forest in eastern china," *Scientific reports*, vol. 10, no. 1, pp. 1–8, 2020.

[98] M. Motwani, D. Dey, D. S. Berman, G. Germano, S. Achenbach, M. H. Al-Mallah, D. Andreini, M. J. Budoff, F. Cademartiri, T. Q. Callister *et al.*, "Machine learning for prediction of all-cause mortality in patients with suspected coronary artery disease: a 5-year multicentre prospective registry analysis," *European heart journal*, vol. 38, no. 7, pp. 500–507, 2017.

[99] J. Lee, J.-S. Lim, Y. Chu, C. H. Lee, O.-H. Ryu, H. H. Choi, Y. S. Park, and C. Kim, "Prediction of coronary artery calcium score using machine learning in a healthy population," *Journal of personalized medicine*, vol. 10, no. 3, p. 96, 2020.

[100] C. Lee, W. Zame, J. Yoon, and M. Van Der Schaar, "Deephit: A deep learning approach to survival analysis with competing risks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[101] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[102] P. M. Domingos, "A unified bias-variance decomposition for zero-one and squared loss," in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA*, H. A. Kautz and B. W. Porter, Eds. AAAI Press / The MIT Press, 2000, pp. 564–569.

[103] R. Kohavi and D. H. Wolpert, "Bias plus variance decomposition for zero-one loss functions," in *Machine Learning, Proceedings of the Thirteenth International Conference (ICML '96), Bari, Italy, July 3-6, 1996*, L. Saitta, Ed. Morgan Kaufmann, 1996, pp. 275–283.

[104] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.

[105] R. Pisters, D. A. Lane, R. Nieuwlaat, C. B. De Vos, H. J. Crijns, and G. Y. Lip, "A novel user-friendly score (has-bled) to assess 1-year risk of major bleeding in patients with atrial fibrillation: the euro heart survey," *Chest*, vol. 138, no. 5, pp. 1093–1100, 2010.

[106] S. A. Nashef, F. Roques, L. D. Sharples, J. Nilsson, C. Smith, A. R. Goldstone, and U. Lockowandt, "Euroscore ii," *European journal of cardio-thoracic surgery*, vol. 41, no. 4, pp. 734–745, 2012.

[107] S. A. Nashef, F. Roques, P. Michel, E. Gauducheau, S. Lemeshow, R. Salamon, and E. S. Group, "European system for cardiac operative risk evaluation (euro score)," *European journal of cardio-thoracic surgery*, vol. 16, no. 1, pp. 9–13, 1999.

[108] B. Iung, C. Laouénan, D. Himbert, H. Eltchaninoff, K. Chevreul, P. Donzeau-Gouge, J. Fajadet, P. Leprince, A. Leguerrier, M. Lièvre *et al.*, "Predictive factors of early mortality after transcatheter aortic valve implantation: individual risk assessment using a simple score," *Heart*, vol. 100, no. 13, pp. 1016–1023, 2014.

[109] R. Shadman, J. E. Poole, T. F. Dardas, D. Mozaffarian, J. G. Cleland, K. Swedberg, A. P. Maggioni, I. S. Anand, P. E. Carson, A. B. Miller *et al.*, "A novel method to predict the proportional risk of sudden cardiac death in heart failure: derivation of the seattle proportional risk model," *Heart Rhythm*, vol. 12, no. 10, pp. 2069–2077, 2015.

[110] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[111] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[112] P. Bühlmann and B. Yu, "Analyzing bagging," *The annals of Statistics*, vol. 30, no. 4, pp. 927–961, 2002.

[113] R. Esposito and L. Saitta, "Monte carlo theory as an explanation of bagging and boosting," in *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, G. Gottlob and T. Walsh, Eds. Morgan Kaufmann, 2003, pp. 499–504.

[114] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Machine Learning, Proceedings of the Thirteenth International Conference (ICML '96), Bari, Italy, July 3-6, 1996*, L. Saitta, Ed. Morgan Kaufmann, 1996, pp. 148–156.

[115] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder

by the authors)," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[116] S. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting algorithms as gradient descent in function," in *12th International Conference on Neural Information Processing Systems, Denver, Colorado, USA*, vol. 29, 1999.

[117] A. R. van Rosendael, G. Maliakal, K. K. Kolli, A. Beecy, S. J. Al'Aref, A. Dwivedi, G. Singh, M. Panday, A. Kumar, X. Ma *et al.*, "Maximization of the usage of coronary cta derived plaque information using a machine learning based algorithm to improve risk stratification; insights from the confirm registry," *Journal of cardiovascular computed tomography*, vol. 12, no. 3, pp. 204–209, 2018.

[118] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[119] M. Aldinucci, S. Rabellino, M. Pironti, F. Spiga, P. Viviani, M. Drocco, M. Guerzoni, G. Boella, M. Mellia, P. Margara, I. Drago, R. Marturano, G. Marchetto, E. Piccolo, S. Bagnasco, S. Lusso, S. Vallero, G. Attardi, A. Barchiesi, A. Colla, and F. Galeazzi, "HPC4AI: an ai-on-demand federated platform endeavour," in *Proceedings of the 15th ACM International Conference on Computing Frontiers, CF 2018, Ischia, Italy, May 08-10, 2018*, D. R. Kaeli and M. Pericàs, Eds. ACM, 2018, pp. 279–286.

[120] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[121] M. R. Baker and R. B. Patil, "Universal approximation theorem for interval neural networks," *Reliab. Comput.*, vol. 4, no. 3, pp. 235–239, 1998.

[122] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. P. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nat.*, vol. 529, no. 7587, pp. 484–489, 2016.

[123] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[124] S. Jallepalli, P. Pathak, P. Gupta, S. Kar, and M. Gupta, "Development and validation of artificial intelligence-based cardiovascular disease (ai-cvd) risk score," *Available at SSRN 3444410*, 2019.

[125] C. Nagpal, S. Yadlowsky, N. Rostamzadeh, and K. A. Heller, "Deep cox mixtures for survival regression," in *Proceedings of the Machine Learning for Healthcare Conference, MLHC 2021, 6-7 August 2021, Virtual Event*, ser. Proceedings of Machine Learning Research, K. Jung, S. Yeung, M. P. Sendak, M. W. Sjoding, and R. Ranganath, Eds., vol. 149. PMLR, 2021, pp. 674–708.

[126] M. Canepa, C. Fonseca, O. Chioncel, C. Laroche, M. G. Crespo-Leiro, A. J. Coats, A. Mebazaa, M. F. Piepoli, L. Tavazzi, A. P. Maggioni *et al.*, "Performance of prognostic risk scores in chronic heart failure patients enrolled in the european society of cardiology heart failure long-term registry," *JACC: Heart Failure*, vol. 6, no. 6, pp. 452–462, 2018.

[127] R. Yoshida, H. Ishii, I. Morishima, A. Tanaka, Y. Morita, K. Takagi, N. Yoshioka, K. Hirayama, N. Iwakawa, H. Tashiro *et al.*, "Performance of has-bled, orbit, precise-dapt, and paris risk score for predicting long-term bleeding events in patients taking an oral anticoagulant undergoing percutaneous coronary intervention," *Journal of cardiology*, vol. 73, no. 6, pp. 479–487, 2019.

[128] D. R. Cox, "Regression models and life-tables," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 34, no. 2, pp. 187–202, 1972.

[129] S. Barlera, L. Tavazzi, M. G. Franzosi, R. Marchioli, E. Raimondi, S. Masson, R. Urso, D. Lucci, G. L. Nicolosi, A. P. Maggioni *et al.*, "Predictors of mortality in 6975 patients with chronic heart failure in the gruppo italiano per lo studio della streptochinasi nell'infarto miocardico-heart failure trial: proposal for a nomogram," *Circulation: Heart Failure*, vol. 6, no. 1, pp. 31–39, 2013.

[130] F. E. Harrell, "Cox proportional hazards regression model," in *Regression modeling strategies*. Springer, 2015, pp. 475–519.

[131] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *Journal of the American statistical association*, vol. 53, no. 282, pp. 457–481, 1958.

[132] R. Peto and J. Peto, "Asymptotically efficient rank invariant test procedures," *Journal of the Royal Statistical Society: Series A (General)*, vol. 135, no. 2, pp. 185–198, 1972.

[133] G. Y. Lip, R. Nieuwlaat, R. Pisters, D. A. Lane, and H. J. Crijns, "Refining clinical risk stratification for predicting stroke and thromboembolism in atrial fibrillation using a novel risk factor-based approach: the euro heart survey on atrial fibrillation," *Chest*, vol. 137, no. 2, pp. 263–272, 2010.

[134] M. C. Fang, A. S. Go, Y. Chang, L. H. Borowsky, N. K. Pomernacki, N. Udaltsova, and D. E. Singer, "A new risk scheme to predict warfarin-associated hemorrhage: The atria (anticoagulation and risk factors in atrial fibrillation) study," *Journal of the American College of Cardiology*, vol. 58, no. 4, pp. 395–401, 2011.

[135] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, 2020.

[136] G. Mittone, W. Riviera, I. Colonnelli, R. Birke, and M. Aldinucci, "Model-agnostic federated learning," Computer Science Department, University of Torino, Tech. Rep., 2023.

[137] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Inf. Fusion*, vol. 81, pp. 84–90, 2022.

[138] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[139] F. D'Ascenzo, O. De Filippo, G. Gallone, G. Mittone, M. A. Deriu, M. Iannaccone, A. Ariza-Solé, C. Liebetrau, S. Manzano-Fernández, G. Quadri *et al.*, "Machine learning-based prediction of adverse events following an acute coronary syndrome (PRAISE): a modelling study of pooled datasets," *The Lancet*, vol. 397, no. 10270, pp. 199–207, 2021.

[140] I. J. Goodfellow, M. Mirza, X. Da, A. C. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgeting in gradient-based neural networks," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2014.

[141] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*, I. S. Dhillon, D. S. Papailiopoulos, and V. Sze, Eds. mlsys.org, 2020.

[142] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020.

[143] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: stochastic controlled averaging for federated learning," in *Proc. of the 37th Intl. Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, ser. Proc. of Machine Learning Research, vol. 119. PMLR, 2020, pp. 5132–5143.

[144] B. Casella, R. Esposito, C. Cavazzoni, and M. Aldinucci, "Benchmarking fedavg and fedcurv for image classification tasks," in *Proceedings of the 1st Italian Conference on Big Data and Data Science, ITADATA 2022, September 20-21, 2022*, ser. CEUR Workshop Proceedings, M. Anisetti, A. Bonifati, N. Bena, C. Ardagna, and D. Malerba, Eds. CEUR-WS.org, 2022.

[145] M. Polato, R. Esposito, and M. Aldinucci, "Boosting the federation: Cross-silo federated learning without gradient descent," in *International Joint Conference on Neural Networks, IJCNN 2022, Padua, Italy, July 18-23, 2022*. IEEE, July 2022.

[146] K. Arabi, "Mobile computing opportunities, challenges and technology drivers," The 51st Annual Design Automation Conference 2014, DAC '14, San Francisco, CA, USA, June 1-5, 2014, Jun. 2014, keynote Talk.

[147] O. De Filippo, F. D'Ascenzo, S. Raposeiras-Roubin, E. Abu-Assi, M. Peyracchia, P. P. Bocchino, T. Kinnaird, A. Ariza-Solé, C. Liebetrau, S. Manzano-Fernández *et al.*, "P2y12 inhibitors in acute coronary syndrome patients with renal dysfunction: an analysis from the renami and bleemacs projects," *European Heart Journal-Cardiovascular Pharmacotherapy*, vol. 6, no. 1, pp. 31–42, 2020.

[148] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[149] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[150] M. Aldinucci, S. Rabellino, M. Pironti, F. Spiga, P. Viviani, M. Drocco, M. Guerzoni, G. Boella, M. Mellia, P. Margara *et al.*, "HPC4AI: an AI-on-demand federated platform endeavour," in *Proceedings of the 15th ACM International Conference on Computing Frontiers*, 2018, pp. 279–286.

[151] M. Aldinucci, S. Bagnasco, S. Lusso, P. Pasteris, S. Rabellino, and S. Vallero, "OCCAM: a flexible, multi-purpose and extendable HPC cluster," in *Journal of Physics: Conference Series*, vol. 898, no. 8. IOP Publishing, 2017, p. 082039.