DELIVERABLE

Project Acronym: **VALIDATE**

Grant Agreement number: **101057263**
Project Title: **Validation of a Trustworthy AI-based Clinical Decision Support System for Improving Patient Outcome in Acute Stroke Treatment**

D2.1 – Models and report on Iteration 1, transition from TRL 3 to 4

Revision: 0.42

Authors and Contributors	John D. Kelleher (TU Dublin); Adam Hilbert (CUB); Thi Nguyen Que Nguyen (TU Dublin); Jana Rieger (CUB)			
Responsible Author	John D. Kelleher		**Email**	john.d.kelleher@tudublin.ie
	Beneficiary	TU Dublin	**Phone**	+353 87 668 4661

Project co-funded by the European Commission within **HORIZON-HLTH-2021-DISEASE-04-04**		
Dissemination Level		
PU	Public, fully open	xxx
CO	Confidential, restricted under conditions set out in Model Grant Agreement	
CI	Classified, information as referred to in Commission Decision 2001/844/EC	

This project has received funding from the European Union's Horizon Europe research and innovation program under grant agreement No 101057263

Revision History, Status, Abstract, Keywords, Statement of Originality

Revision History

Revision	Date	Author	Organisation	Description
Setup document	13.01.2023	Adam Hilbert	CUB	Adding skeleton
Initial input	12.02.2023	Adam Hilbert	CUB	Initial input from technical workshop nov 11.
	10.03.2023	Adam Hilbert	CUB	Adding discussion notes from meeting
	15.03.2023	Adam Hilbert	CUB	Structuring
	20.4.2023	Thi Nguyet Que Nguyen	TU Dublin	Adding text to Introduction
	24.04.2023	Adam Hilbert	CUB	Refining structure and assign responsible contributors from alignment call
	24.04.2023	Adam Hilbert	CUB	Working on TRL definitions
	25.04.2023	Jana Rieger	CUB	Adding NN experiment details
	25.04.2023	Adam Hilbert	CUB	Working on TRL definitions, structuring, guidelines for content
	26.04.2023	Jana Rieger	CUB	Working on NN experiment Results
	26.04.2023	Adam Hilbert	CUB	Working on TRL definitions, mapping to CRISP-ML
	26.04.2023	John Kelleher	TU Dublin	Drafted introduction and overview
	27.04.2023	John Kelleher	TU Dublin	Revising introduction and overview text, and worked on clinical model refinement and validation framework
	27.04.2023	Jana Rieger	CUB	Working on NN experiment Results
	27.04.2023	Thi Nguyet Que Nguyen	TU Dublin	Adding section of experiment of tree-based modelling
	28.04.2023	Thi Nguyet Que Nguyen	TU Dublin	Adding section Report on T2.3 - tree-based Federated Learning
	28.04.2023	John Kelleher	TU Dublin	Worked on the clinical refinement model and validation framework section
	28.04.2023	Adam Hilbert	CUB	Working on Report on T2.4

	29.04.2023	John Kelleher	TU Dublin	Redrafted section on medical device regulation, reviewed sections on tree-based modelling experiment and section T2.3 tree-based Federate Learning, proofing entire document, prepared bibliography, and formatting of tables, figures and references.
	29.04.2023	Adam Hilbert	CUB	Working on Report on T2.4 (+experiments), Working on Report on T2.3
	30.04.2023	John Kelleher	TU Dublin	Final Review

Date of delivery	Contractual:	30.04.2023	Actual:	30.04.2023
Status	final ☒ /draft □			

Abstract (for dissemination)	This deliverable reports on the work and achievements of the VALIDATE project towards transitioning an AI prognostic model from TRL 3 to TRL 4, and the development of a guideline on model development, validation and lifecycle management of AI models for clinical decision support which will be set out in Validate Deliverable 2.5. The report provides: (a) a review of relevant background literature, including technology readiness levels, best practice in machine for healthcare, and medical device regulations; (b) a review of the state-of-the-art in federated learning; (c) a report on the data understanding and data preparation work carried out on multiple datasets (the German Stroke Registry and MRCLEAN datasets); (d) the development of multiple models to predict the modified Rankin Scale (mRS) of a patient 90 days after stroke onset, using two complementary machine learning approaches, neural networks and tree-based ensemble models; and (e) the in-lab validation of the models on retrospective multicenter data.
Keywords	Technology Readiness Levels, Medical Device Regulation, CRIPS-ML(Q), Machine Learning, stroke, modified Rankin Scale

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation, or both.

Table of Content

Introduction

Figure 1 below (taken from the Validate proposal) provides an overview of the project methodology, and the work reported in this deliverable relates to the very center of this figure where an agile AI model development is used to transition an AI model from TRL 3 to TRL 4.

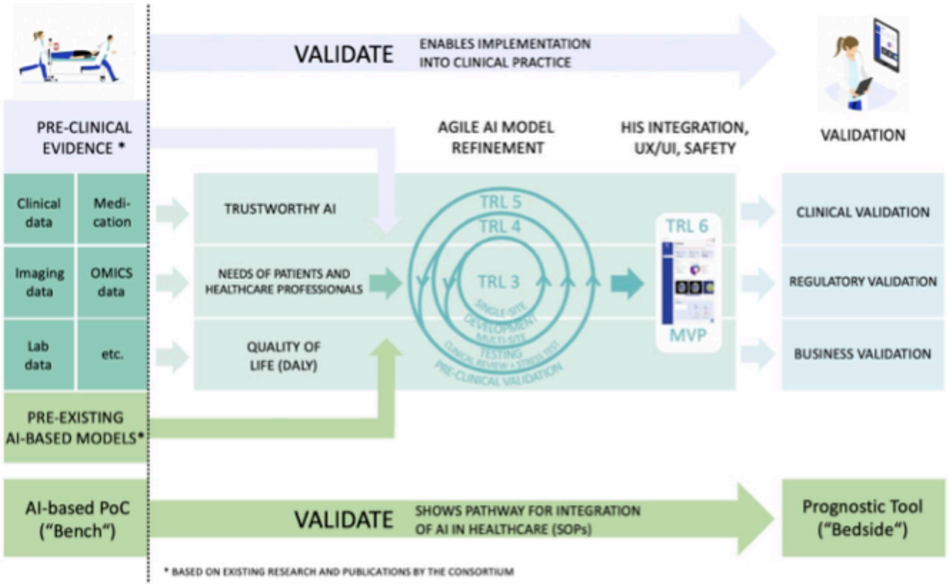
VALIDATE
ENABLES IMPLEMENTATION INTO CLINICAL PRACTICE
PRE-CLINICAL EVIDENCE *
AGILE AI MODEL REFINEMENT
HIS INTEGRATION, UX/UI, SAFETY
VALIDATION
Clinical data
Medi-cation
Imaging data
OMICS data
Lab data
etc.
TRUSTWORTHY AI
NEEDS OF PATIENTS AND HEALTHCARE PROFESSIONALS
QUALITY OF LIFE (DALY)
TRL 5
TRL 4
TRL 3
SINGLE-SITE DEVELOPMENT
MULTI-SITE TESTING
CLINICAL REVIEW + STRESS TEST
PRE-CLINICAL VALIDATION
TRL 6
MVP
CLINICAL VALIDATION
REGULATORY VALIDATION
BUSINESS VALIDATION
PRE-EXISTING AI-BASED MODELS*
AI-based PoC ("Bench")
VALIDATE
SHOWS PATHWAY FOR INTEGRATION OF AI IN HEALTHCARE (SOPs)
Prognostic Tool ("Bedside")
* BASED ON EXISTING RESEARCH AND PUBLICATIONS BY THE CONSORTIUM

Figure 1 Validate Overview

In the VALIDATE proposal the complete agile AI model refinement project is described as *[b]eginning with a pre-existing AI model validated on retrospective data from a single-site (TRL3) the VALIDATE processing iteratively matures the AI through to a clinically validated (TRL6) demonstrator that is ready for regulatory validation, that has a clearly defined pathway to market*. However, this deliverable focuses on reporting on the first iteration of this process, the transition of an AI model from technology readiness levels 3 (TRL3) to a technology validated at TRL4. This work directly contributes to the following objectives of the Validate project:

- Objective 2 Transition an existing AI prognostic model from TRL 3 through to validation at TRL6
- Objective 7 Establish SOPs for the integration of AI in healthcare. Specifically, regarding the development of *a guideline on model development, validation and lifecycle management of AI models for clinical decision support* which will be set out in Validate Deliverable 2.5.

The overall ambition of the VALIDATE project is that the methodology developed for AI model refinement and integration into healthcare would integrate current state-of-the-art machine learning (ML) development cycles, such as CRISP-ML(Q) [1], with a trustworthy AI framework that grounds the EU Trustworthy AI guidelines [2] in the specific needs of AI for healthcare. It was originally envisaged that this integration could be achieved by using a technology readiness level (TRL) framework to track the maturity level of the AI and to tailor the ML development, evaluation and validation methods and the trustworthy AI framework to the considerations relevant to the current TRL. It is hoped that the resulting VALIDATE methodology would establish SOPs for AI validation that would achieve the following:

- ○ Bring AI models from research to practice
- ○ Meet the necessary criteria of clinical practice
- ○ Operationalize "release" and consecutive, multi-tier validation of models
- ○ Define the necessary requirements and infuse co-creation with ethical guidelines and stakeholder requirements

The AI model that is being used as the case study for developing and road-testing this AI refinement process is an AI model built to predict the outcome of stroke patients at 90 days after the onset of stroke symptoms. Although the classification of specific instances of AI technology within the frameworks established by EU Medical Device Regulation (MDR) and the In-vitro Diagnostic Regulation is still under-determined, AI algorithms are generally regarded as Class II (medium risk) [3] . Consequently, MDR regulations are relevant to AI technology and should be factored into the design of the VALIDATE methodology. As a result, in this deliverable, we include a discussion on MDR and how it relates to AI validation. An important aspect of MDR regulation is the definition of the *intended purpose of the tool*, as this is crucial for the assessment of the risk level associated with the tool. As we will discuss this concept provides a point of synergy with the trustworthy AI framework in terms of opening up a discussion in relation to ethical approaches such as *ethics by design* and *defensive design*. It is noteworthy that as part of Task 2.1, the data science teams working on the development of the models have engaged in these discussions with both the researchers in WP1 who are developing the VALIDATE Trustworthy AI framework and the external ethics auditors from the Z-inspection process that the VALIDATE project is engaging. However, these developments will be captured in the deliverables reported through WP1 and so won't be discussed in this document.

Overall, the work carried out in Iteration 1 included:

- a review of relevant background literature, including technology readiness levels, best-practice in machine learning for healthcare, and medical device regulations
- a review and testing of current state-of-the-art federated learning frameworks
- an extensive data understanding and data preparation phase for multiple datasets (the German Stroke Registry and MRCLEAN datasets), involving both the tasks related to the legal and ethical requirements necessary to prepare for data access, as well as work on identifying coming features and implementing feature mappings across the datasets, and data cleaning.
- the development of two complementary machine learning approaches, neural networks, and tree-based ensemble models.
- the in-lab validation of the models on retrospective multicenter data. Models were trained and validated on individual datasets using cross-validation and then validated again on a different dataset to assess each model's robustness when transferred to new distributions. The assessment of the models across different datasets not only has the benefit of assessing the models themselves but also provides insight into the quality and similarity of different datasets, which can be important in a federated learning setting [4].

In summary, the work reported in this deliverable is fundamental work that will enable the next steps of implementing ML models in federated learning (FL) systems, and ultimately to the development of SOPs for the integration of AI in healthcare that are relevant to the state-of-the-art methods and practices in AI model development and EU regulatory and ethical principles.

Report on T2.2: VALIDATE clinical model refinement and validation framework

There is a growing awareness and acceptance of the need for guidance frameworks for the development of data-driven AI predictive models, be they diagnostic or prognostic. Examples of recent scoping reviews and publications on this topic include [5]–[7]. Of these proposals, the scoping review by de Hond et al. [5] is the most directly relevant to this work as the review was deliberately done in a manner that generalized across medical domains. De Hond et al. organised their scoping review using a six-phase AI model development structure, namely: (1) data preparation, (2) AI-based Prediction Model (AIPM) development, (3) AIPM validation, (4) software development, (5) AIPM impact assessment, and (6) AIPM implementation into daily healthcare practice. We take de Hond et al.'s guidance framework as a useful basis for developing our own framework while noting a number of important differences between our approach and that set out by de Hond et al. First, de Hond et al.'s framework does not consider the iterative development of systems, as a project lifecycle it is most naturally understood as a waterfall approach. Consequently, it does not take the iterative nature of agile model development into account, nor the appropriateness of recommendations within each phase with respect to different levels of TRLs. Of the six phases set out the first 3 of de Hond et al.'s phases are most relevant for work at earlier TRL levels, and these are the phases we have focused on in this report.

Key recommendations for the data preparation phase include *clearly specifying the medical problem and context that the model with address and the context that the AIPM will address,* to *describe and define clinical success criteria, to consider trade-offs between predictive performance and privacy, and the identification and inclusion of stakeholders in the different phases of model development*. Within the VALIDATE project these recommendations are primarily addressed through the work done in work package 1 Trustworthy AI, and work package 3 Software Development, Testing and Integration. See for example D1.1 and D3.2 - Integrated requirements report covering technical and user requirements. This works feeds into the work in WP2 most directly through task T2.1 Definition of requirements and review adherence to trustworthy AI framework and the engagement by researchers from across the data science teams in WP3 in the external ethical auditing (Z-inspection) process that VALIDATE is using.

de Hond et al.'s recommendations on *clearly specifying the medical problem and context that the model with address and the context that the AIPM will address* aligns with the requirements within medical device regulations for the determination of a clear *intended purpose,* and so here we borrow from T2.1 to define the intended purpose we have agreed upon for VALIDATE:

> *The medical intended purpose of the VALIDATE software is to provide a tool to enable a prediction about the individual treatment outcome in the treatment of acute ischemic stroke. This is based on the patient's individual initial health status and is geared towards the best treatment outcome applying the Modified Rankin Scale (MRS). It supports the diagnosis as well as the initiation of the appropriate therapy.*

VALIDATE D3.2 Integrated requirements report covering technical and user requirements (p. 32)

A key distinction set out by de Hond et al. between phases 2 and 3 of their framework is the difference between *internal validation* and *external evaluation* of model performance. Phase 2 of their framework *AI-based Prediction Model development* calls for internal validation of the model where the *goal of internal validation is to assess the predictive performance of an AIPM in data that are unseen with respect to model training but come from the same population and setting* [5, p. 5]. By contrast in Phase 3 they recommend *external performance evaluation* which involves *the application of an existing model without modifications to data from a different population or setting compared to model development* [5, p. 6]. Adopting this distinction in our experiment work below we report experiments covering internal validation and external validation, using the German Stroke Registry data to develop models and internal validation, and the MRCLEAN dataset for external evaluation.

Overall, we have identified four different areas of regulation and practice that are relevant to the development of a framework for the development of AI systems for health (see Figure 2). These include technology readiness levels, medical device regulation, trustworthy and ethical AI, technology readiness levels, and best-practice in agile AI model development. The trustworthy AI perspective will be covered in more detail in the deliverables from WP1, so in what follows we will focus on technology readiness levels, medical device regulation and machine learning practice and how (we currently understand) each of these topics as framing or contributing to the TRL3 too TRL4 iteration of AI model development.

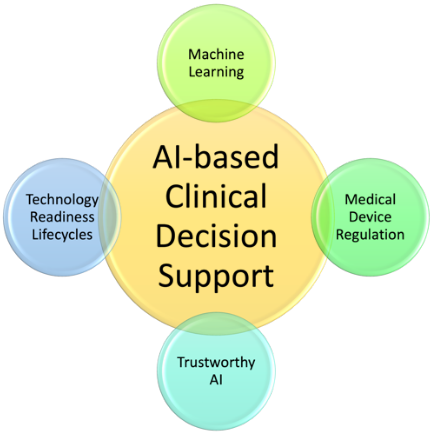
Machine Learning
Technology Readiness Lifecycles
AI-based Clinical Decision Support
Medical Device Regulation
Trustworthy AI

Figure 2 Relevant Perspectives and Regulations for the development of AI-based Clinical Decision Support Systems

Technology Readiness Levels for AI-based Clinical Decision Support systems

Technology Readiness Levels (TRLs) have been used across industries to assess the maturity, the progress of development and with that the risk of the technology. TRL frameworks are generally defined in terms of broad, high-level concepts that allow for adaptation to specific fields of development and understanding for both developers and managers. However, although these general specifications allow for adaptation to specific contexts this does not mean that it is not challenging to satisfactorily define these mappings of TRLS for novel development processes. For example, Medical Device Development (MDD) naturally differs from the original application industries of TRLs (such as space technologies) in many ways, e.g. due to the diverse regulatory pathways across countries. Specific adaptation to MDD and a new mapping have been proposed recently, which provides better categorization and facilitates more precise risk assessment for the various stages of MDD. Table 1 shows a selected overview of references found in the literature, representing a narrowing focus from the original outline by the US Department of Defence (DoD) to a recently proposed translation for MDD.

Table 1 Definitions of TRLs 1-4 from a selection of the literature

	TRL1	TRL2	TRL3	TRL4
	MDLR1		MDLR2	
[8]	Basic principles observed and reported	Technology concept and/or application formulated.	Analytical and experimental critical function and/or characteristic proof of concept.	Component and/or breadboard validation in laboratory environment
[9]	Basic principles observed	Technology concept formulated	Experimental proof of concept	Technology validated in lab
[10]	Scientific literature reviews	Hypothesis(es) is generated.	Initial proof-of-concept for device candidates is	Proof-of-concept and safety of candidate devices/systems

	and initial Market Surveys are initiated and assessed. Potential scientific application to defined problems is articulated.	Research plans and/or protocols are developed, peer reviewed, and approved.	demonstrated in a limited number of laboratory models.	demonstrated in defined laboratory/animal models.
[11]	Basic principles and research data observed and reported. Scientific research findings reviewed and assessed and translation into applied research begun. Potential targets, mechanisms, concepts evaluation.	Technology concept and/or application formulated. Research ideas, hypothesis, experimental designs, potential targets, technologies, solutions (also digital), protocols identified and developed, peer reviewed and approved.	Active R&D, data collection and analysis initiated. First hypothesis testing, target identification, potential candidates characterization, data collection, technological components (also digital) evaluation, alternative concepts exploration carried out. Early proof of concept (PoC)/system application tested in laboratory environment, in a limited number of in vitro & in vivo models.	Preclinical R&D. PoC, safety of potential candidates, device or system demonstrated in a relevant laboratory or animal model (non-GxP). Formulation and manufacturing process development initiated (non-GMP). Identification of relevant parametric data required for technological assessment. System components integrated and tested regarding preliminary efficiency and reliability. Software architecture and other system components development to address reliability, scalability, operability, security etc. Other system components development.
[12]	Needs Assessment. Identification of scientific and design principles to address an existing medical challenge in terms of safety, clinical effectiveness, systems integration, human performance, and satisfaction.	Prototype Development. Development of a working prototype illustrating scientific and design principles to address safety and effectiveness. Potential user performance and system integration issues are identified to improve the design.	Bench Testing. Bench testing to identify mechanical, electrical, and biological engineering performance issues of the device, including ex vivo, in vitro, and in situ animal or human tissue; and animal carcass or human cadaveric testing.	Animal Testing. Initial evidence of MD safety is established, including its performance when used in a living system. Device operator performance issues identified to enhance the design.

In principle, TRLs can also be interpreted as a clustering of requirements specific to consecutive stages of the development process. Reaching a higher level of maturity means a new milestone and fulfilment of another set of various requirements towards the technology. Following this line, we are working on defining a mapping of TRLs to the development life-cycle of AI-based Clinical Decision Support Systems (CDSS). To do this we are building on existing literature and engaging in multidisciplinary co-creation with internal and external industry experts to define all key requirements towards CDSS from relevant sources; namely stakeholder pain points, ethical/trustworthy considerations and Medical Device Regulation (MDR). We follow the process of

1. identifying an exhaustive list of mechanisms, best practices, tests and requirements along the AI life cycle,
2. clustering these into our adapted TRL definition in a way that each level resembles important milestones of AI maturity,
3. iteratively review and discuss our categorization internally and externally.

At this point of reporting, we mostly focused on potential requirements up to TRL4. It is important to mention that we are following a multidisciplinary approach already in the first step of the process and integrated requirements from stakeholder interviews and research (reported on in D3.2 deliverable) as well as – at this point - preliminary requirements from the VALIDATE ethical framework (reported on in D1.1 deliverable). The iterative refinement and development of our TRL categorization will include external assessment as part of the Z-inspection® process and will be extended to further levels of maturity up to TRL6. Later in this document (see Table 3) we set out our current thinking of how TRLs map to different stages in the development life-cycle of a CDSS.

AI Validation and Medical Device Regulation (MDR)

One of the goals of the VALIDATE project is that the process developed for the creation of the VALIDATE demonstrator should align with Medical Device Regulations. Table 2 lists the ISO standards we have identified as being most relevant to our work from a medical device regulation perspective. The most general of these is ISO 60601 Medical Device Equipment, however ISO 62304 deals specifically with medical device software and it calls for both a risk management system and a quality management system. Certification in relation to risk management of medical devices can be obtained by following ISO 14971 and certification in relation quality management is obtained by following ISO 13485.

Table 2 Relevant ISO Standards for Medical Device Regulation and AI Risk Management

ISO Identifier	Description
60601	Medical Device Equipment
62304	Medical Device Software
13484	Medical Devices Quality Management Systems
14971	Medical Devices Application of Risk Management to Medical Devices
23894	Information Technology – Artificial Intelligence – Guidance on risk management

Each of these ISO standards sets out processes for compliance, typically requiring the documentation of processes of the tracking of adherence to these processes. However, none of these ISO standards were designed with AI systems in mind. Indeed, new ISO standards are being developed a published specifically to deal with AI, see for example ‘ISO/IEC 23894:2023 Information technology — Artificial intelligence — Guidance on risk management’ publication date 2023-02. As a result, we are currently working on reviewing these ISO standards and their associated documentation in order to better understand how they should be applied within the context of AI model development, this work is ongoing.

Mapping to the CRISP-ML(Q) process

The potential of AI and applications of Machine Learning (ML) in many industries has been demonstrated in recent years, yet most of the models deployed and used in those solutions are products of a different development process. Standardized development processes are key to meet business and – in areas such as healthcare even more importantly – performance expectations. The CRoss-Industry Standard Process model for the development of Machine Learning applications with

Quality assurance methodology (CRISP-ML(Q)) [1] was proposed to provide such guidance throughout the AI life cycle. The original framework splits the life cycle into six phases: Business & Data Understanding, Data Preparation, Modelling, Evaluation, Deployment, Monitoring & Maintenance. For each of these phases state-of-the-art quality assurance methodologies are proposed to mitigate challenges.

In contrast to other application areas, for MDs and CDSSs require in principle a more rigorous, multi-phase validation as discussed previously. Validation in various clinical environments or study settings (i.e. retrospective, prospective) mean for example re-iterating data verification and quality assessment cycles, thus call for a slightly adjusted process. To this end, we take prescriptions of CRISP-ML(Q) as the basis and propose an adapted framework for developing AI-based CDSS. This enables clear tracking of AI maturity in correspondence to our definition of TRLs for CDSS and provide straightforward guidance to translate the accumulated requirements into action and success while manages the correct expectations on each level of maturity.

First, we decouple Business understanding and consider it as the necessary initial step to create business objectives, Key Performance Indicators and Value Drivers. All these are then translated into key requirements for the system to be achieved at given TRLs. Next, we envision 3 principal phases or in other words, groups of processes, that will follow each other in consecutive iterations. These are Data assessment, Model verification and Performance assessment. One iteration through each should bring the CDSS one level higher in TRLs. Consequently, each of the phases covers different aspects of the umbrella topic in different iterations. On some TRLs, some phases might not prescribe necessary processes and might be skipped. For example, Performance assessment on TRL1. In fact, taking a resource-efficient approach, we argue each iteration shall start with a Performance assessment to determine if the existing solution meets the criteria of a higher TRL. This also enables our framework to be applicable for further development of existing solutions. In case the assessment fails, the iteration starts with the corresponding Data assessment phase of the targeted TRL. Phases are passed once the necessary requirements are achieved and the generated output for the next phase is prepared and approved. In case phases have been passed on a given TRL but the CDSS did not meet the necessary Performance assessment criteria, the unmet requirements should be closely analysed and revisited. Figure 3 depicts the described process.

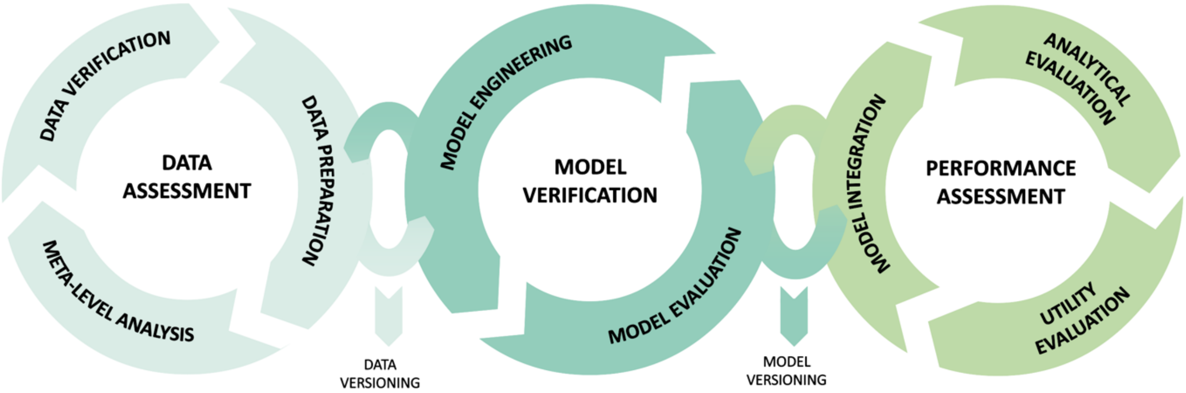

Figure 3 CRISP-ML Process

Let us revisit the challenge of multi-phase validation in MDD. The above-proposed framework allows the definition of specific quality assurance criteria with respect to different validation settings corresponding to the maturity of the MD. For example, in vitro validation and clinical randomized trial are majorly different milestones in the development process and thus would prescribe requirements in the Performance assessment phase of separate TRLs. Our framework aims to provide a standardized way of tracking progress and provides a clearer understanding for all stakeholders where their key points are included in the process.

In the following, we present the resulting list of requirements with an initial implication of categorizations of the levels of maturity and correspondence to the above-mentioned 3 phases. We note however that this has not been reviewed and is a majorly ongoing process.

Table 3 Listing of current mappings from TRLs to requirements

- The clinical need and challenge should be understood - Identification of safety, effectiveness and impact potentials should be initiated - Relevant stakeholder and user groups are identified - Due to the critical application area of healthcare, we suggest inclusion of relevant stakeholders in co-creation as early as TRL1, e.g. in terms of review and early validation of the abovementioned characterisation of the initial idea	Business understanding	
- Feasibility of necessary data collection for modelling should be assessed or existing sources for hypothesis testing and prototyping should be researched	Data assessment	**TRL1 – Model idea**
- The feasibility, acceptability and potential integration (i.e. into current clinical workflow) of a data-driven solution should be assessed by literature review of AI applications in the same medical field and similar data domains	Model verification	
	Performance assessment	
- Necessary data collection is initiated, or existing data sources are pooled	Data assessment	**TRL2 – Model hypothesis**
- Modelling and experimental design necessary for hypothesis testing is created and peer reviewed, including candidate models fitting to the specific data domain and important model and optimization parameters to be tuned - Potential integration challenges are identified, and solutions are researched and reviewed	Model verification	
- Research plans are outlined, hypothesis is formulated around the targeted clinical need with safety and effectiveness measures understood and approved by relevant stakeholders	Performance assessment	
- Modelling data source is prepared and pre-liminary analysis and understanding of the data is performed - Distribution analysis of available predictors as well as outlined prediction targets is performed and reviewed by clinical experts - Relevant study cohort is identified to test hypothesis and selection is reviewed and approved by clinical experts - Data schema for model building is developed in discussion with clinical experts	Data assessment	**TRL3 – Research models, prototype**
- Modelling experiments conducted to test prediction performance, including hyper-parameter tuning and best candidate model is selected.	Model verification	
- Hypothesis tested, first prototype built - Performance is demonstrated on retrospective data adhering to the described cohort from at least one source/clinical center which was not included in the training or parameter selection process - Code framework and machine learning pipeline for pre-processing data, training and evaluating models encapsulated in a formalised/standardized process which have been peer reviewed - Unit tests for important functionalities in the code have been created	Performance assessment	

- Data from multiple sources (e.g. medical diagnostic scanners)/clinical centers/databases is prepared and each source meets established criteria on TRL3	Data assessment	**TRL4** **–** **Analytically validated models**
- Model(s) are prepared for cases of missing information, analysis has been conducted on effects of missing input data - Robustness has been demonstrated with statistical testing using confidence intervals - Overfitting, performance difference in training, validation and held-out test sets has been analyzed and overcome when identified - Model calibration has been analyzed and conducted	Model verification	
- Performance above state-of-the-art is demonstrated using multi-metrics evaluation - Performance is evaluated and retained with negligible loss on retrospective data adhering to the described cohort from multiple sources (e.g. medical diagnostic scanners)/clinical centers/databases - Performance is evaluated and retained with negligible loss on relevant sub-groups of data points of clinical relevance, demographics (age, sex/gender, ethnicity/race, geographic location), edge-cases, random point selection - Potential performance improvement with data augmentation have been analyzed and exploited - A resulting model for further integration can be selected with demonstrated stability and fixed input and output in alignment with the intended clinical decision support	Performance assessment	

Report on T2.3: Federated/Distributed Learning in health

One of the major challenges in developing a data-driven model for healthcare is the tension that exists between the sensitivity and privacy concerns related of personal health data, and the importance of accessing large and representative datasets model training and validation. The VALIDATE project is addressing this challenge by using federated learning. Federated learning is a machine learning technique that enables a decentralised model training. In brief, rather than aggregating data from multiple sites on a central server and training the model on the server, the data stays at the original sites and a central model is distributed to these sites, independently trained at these sites, and the training updates are then shared back to the central server to generate a new updated version of the central model. A major benefit of this process is that the data does not need to move outside of its originating institution. Federated learning does of course assume a commonality across the data schemas for the data set used at the different sites, and so the use of federated learning does involve data harmonization and mapping between different sites. The initial approaches to federated learning were designed to work with gradient-based training algorithms that are most suitable for neural network models. More recently, however, there has been a growing interest in the development of distributed learning methods for other types of models such as tree-based ensembles. This recent development is of particular interest because tree-based ensembles provide a useful contrast to neural models in terms of their interpretability and there is some evidence that tree-based models can outperform neural models on structured tabular data [13]–[15]. Consequently, we have reviewed and experimented with both federated learning methods for neural models and distributed learning frameworks for tree-based ensembles.

Federated Learning with Neural models

The classical implementation of FL relies on a central orchestrating server and distributed clients, where the server's role is to distribute and aggregate and the clients' role is to train on their local data and return results. The server sends an initialized model to all clients, that train the model based on their local data. The trained local models – or necessary updates in model parameters – are sent back to the server, which aggregates all local updates into one global model. In the next round, the newly updated model is distributed, and all the steps are repeated over and over again. A visualization of the process is shown in Figure 4

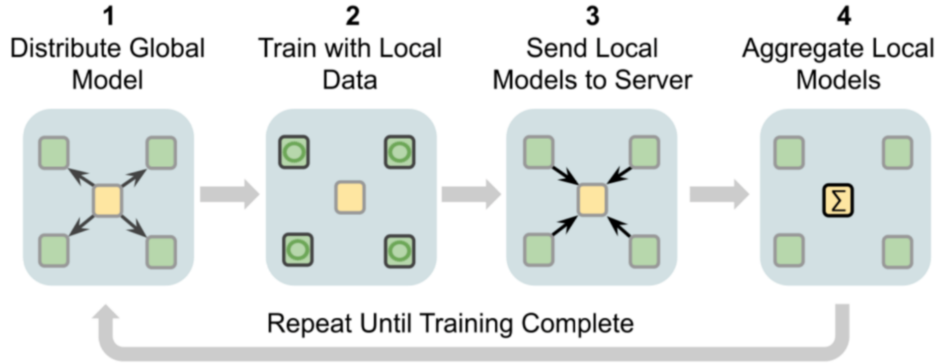

Figure 4 Visualisation of a federated learning process, figure sourced from [16]

Currently, federated averaging is the foremost technique to aggregate local models to create and a global one. This means each client performs one training step and reports new model parameters to the central server. Here, these parameters are averaged based on the ratio of the local sample size and the total sample size [17]. Recent work by [18] shows that Federated Averaging is not only successful in applications but also based on theory.

Thanks to the mathematical formulation of Neural Networks (NN), this classical FL concept became most popular with such models. Practically, different, trained NNs only differ in the actual learned weights (model parameters) while their architecture can remain the same. For example, two NNs with the same number of hidden layers and neurons can perform completely different information processing depending on the learned weights of their neurons. This allows for simple exchange or fusion of different networks with the same capacity (i.e. architecture).

Federated Learning with Tree-based models

An alternative approach for federated learning (FL) is to train a tree-based model using data across different sites. Tree-based models are a well-known machine learning technique that is widely applied in numerous domains of science [19], [20], business [21], and in medical applications [22], [23]. Recently, FL has also been used to train tree-based models [24] and has been applied in medical applications [25].

As mentioned by Ong et al. [24], the use of tree-based models in FL has four main advantages which include:

i) Providing a prediction model with a balance of complexity and interpretability, thus allowing a robust performance with a simple understanding and interpretation in the decision-making process;

ii) Handling categorical and numerical features at the same time;

iii) Tree-based machine learning algorithms intrinsically include a feature selection process that attempts to identify and choose the optimal input features for the modelling across the given data sites. This functionality will strongly support the pre-processing step in FL, cutting down significantly on additional communication costs between data sites.

iv) Providing a robust model performance with respect to handling non-IID (independent and identically distributed) data, especially when using Gradient Boosted Decision Tree ensembles (e.g., XGBoost) [26].

The development of a tree-based model using FL encounters two main challenges: i) How to fuse the data across different sites within tree-based FL model development? and ii) What kind of information can be exchanged between data-sites and the FL server?

To answer the first challenge question, we need to know if the proposed method is a vertical or horizontal FL approach, and what type of tree-based model is begin used for FL implementation. In the literature, most tree-based FL models are horizontal FL, i.e. all the data-site share the same set of input features, and very few applications for vertical FL, i.e. all the data-sites share the same set of data sample identifiers [24]. Besides, the type of tree-based model can be varied from Decision tree [27], to random forest [25] and Gradient Boosting Decision Tree (or XGBoost) [28]. Mostly, Gradient Boosting Decision Tree such as XGBoost or LightGBM are implemented in FL systems [24].

When FL is used to train neural models that are trained using an error gradient learning signal it is relatively straightforward to share information between the data sites and the central server because the error gradients are both (a) not directly interpretable with respect to reconstructing the data that the models are being trained on, and (b) the errors gradients from multiple sites can be integrated by functions such as simply averaging. By contrast, in most tree-based learning algorithms trees are fitted to the data by iteratively growing the tree by adding new nodes to the bottom of the tree, where each node encodes a decision regarding the sorting of examples that have reached that node based on the value of a feature in the dataset. Consequently, sharing a node update involves sending interpretable information such as a threshold being applied to a given feature. The exchange of this type of interpretable and sensitive information may introduce a data security risk for an FL system. Several proposals have been made in the literature to circumvent the leakage of information each data site can exchange a different kind of information; for example, information with introduced noise, gradient values [29].

Recently, Hauschild et al. [25] reported a study of random forest in FL for healthcare applied on different datasets for predicting liver disease, Hepatocellular carcinoma, breast cancer and lung tumors. Another recent tree-based FL framework was proposed by Wassan et al. [29] for a Gradient Boosting Decision Tree FL for a healthcare internet-of-things model. However, beyond these examples, our search of the literature suggests that the application of FL to tree-based models for medical applications is still very limited. Furthermore, studies have shown a potential advantage of the outperformance of tree-based models on tabular data when compared to deep learning approaches [30]. Therefore, the introduction of tree-based FL into this project will help to explore additional advantages of this approach in predicting the outcome of the patient with stroke.

Progress on VALIDATE FL framework

As preparation for multi-centre validation and potential fine-tuning of the developed models, we started setting up the VALIDATE Federated Learning framework. We conducted a thorough literature search of applications of FL, as well as a more technical search on specific Federated algorithms. The results of these are shown in the section below entitled Report on T2.3: Federated/Distributed Learning in health, and also in Appendix C Literature Review on Federated Learning Applications. Next, the development team decided to use the open-source FLOWER FL[1] framework. For integration the following steps have been done:

1. The FLOWER framework has been tested in a single location, by creating simulated clients. The training was successful and achieved comparable performance to the classical global training.
2. During a series of internal technical workshops, the VALIDATE development team decided to create a test setup to pinpoint all challenges before deploying and recreating the setup at the VALIDATE clinical sites. For this HDB (responsible for T2.3) was appointed as server and CUB and TU Dublin as testing client sites.
3. Next, the same scenario and setup from Step 1 were tested with HDB as a server and a remote client on an AWS Virtual Machine. This test did not include any sensitive data and was merely done to test the possibility of communication from the server through the Internet. The test was successful and thus a requirement for client setup could be derived from it.
4. This requirement was applied, and setup was initiated at CUB. An Ubuntu Virtual Machine was allocated inside a DeMilitarized Zone (DMZ) network of CUB, where access to the internet through specific ports can be granted, however, the network is secured and controlled to keep security measures of sensitive data intact.
5. The setup has been successfully tested locally at CUB. Firewall exceptions have to be added by the IT department to allow incoming/outgoing traffic on a specified and controlled port. A request has been filed; the VALIDATE team is waiting for an answer.

In parallel with the FL setup, the development team started to set up remote access to the VALIDATE clinical sites. A request for a Linux-based virtual machine in the DMZ network (with access to the local data) has been filed at the Vall d'Hebron site and communication with the Hadassah site has been initiated, the team is waiting for an answer from the IT department. In HDB, the setup is established and can be used for FL.

The next steps include:

- Adaptation of the VALIDATE code base for FL and integration of the FLOWER framework.
- Test FL setup with CUB and TU Dublin, as well as finalize technical requirements at partner sides so that setup can be initiated at the clinical sites.

[1] https://flower.dev/

- Troubleshoot potential challenges and problems, before setup at the clinical sites.

All these step as well as a comprehensive report on FL experiments will be reported in the next deliverable (D2.2) due in project month 24.

Report on T2.4: stroke outcome prediction

In VALIDATE we demonstrate the use of the researched and developed guidelines for AI-based Clinical Decision Support Systems on the use case of ischemic stroke. Acute ischemic stroke affects more than 1 million people within the EU annually. It occurs when a blood clot blocks a brain-supplying vessel, impedes the supply of oxygen to the brain and subsequently results in damage to brain cells with loss of function such as speech impairment or paralysis. Even though the average treatment effect and outcome benefit across the entire population of patients is proven, outcome still differs significantly for individual patients, where some patients are eligible for treatment but still can show catastrophic outcomes. To this end, we develop multi-variable prognostic AI models to answer the question of: how to improve stroke treatment outcome on an individual patient level?

Definition of prediction paradigm

In order to fulfil our intended purpose and support treatment decisions in stroke we define our prediction paradigm in the following way. Similarly to treatment effectiveness validation, the target of our models will be the prediction of the functional outcome. This is classically measured by the modified Rankin Scale (mRS) at 3 months after the stroke, as shown in Table 4. In previous studies and clinical trials association of certain predictors at the time of hospital admission (baseline) to the mRS score has been analysed in multiple ways. First, clinical experts might look for a simplified dichotomy of the outcome, representing favourable (mRS <= 2) vs. unfavourable (mRS > 3) outcome or trichotomy, describing favourable (mRS <= 2) vs. unfavourable (3 <= mRS < 5) vs. Devastating (mRS >= 5). Second, to show the effectiveness of treatments, the shift in mRS levels between study arms (treated vs. control) has usually been evaluated. The approach resulting in systematically lower mRS levels promotes better outcomes in general. Within our demonstration, we aim to provide users the same information prior to treatment.

Table 4 modified Rankin Scale

mRS 0	No symptoms
mRS 1	No significant disability. Able to carry out all usual activities, despite some symptoms.
mRS 2	Slight disability. Able to look after own affairs without assistance, but unable to carry out all previous activities.
mRS 3	Moderate disability. Requires some help, but able to walk unassisted.
mRS 4	Moderately severe disability. Unable to attend to own bodily needs without assistance, and unable to walk unassisted.
mRS 5	Severe disability. Requires constant nursing care and attention, bedridden, incontinent.
mRS 6	Dead.

In terms of our prediction paradigm, this means we need to build models on the full scale of the mRS score, to be able to show potential shifts with different treatment options. However, from full-scale mRS predictions, one can also derive dichotomized or trichotomized predictions which might have a lower granularity but allows for quicker assessment of the clinical case in an acute scenario. From the translated dichotomized and trichotomized predictions one can calculate the respective performance of the model and compare it with baselines published in clinical literature (e.g. similar AI approaches or statistical approaches/studies showing an association of predictors to outcomes). Important to note, that when developing AI models to support certain decisions, input variables should be restricted to those available up to the point of a certain decision in the clinical workflow to ensure usability and feasibility of integration.

Retrospective data

German Stroke Registry (GSR)

The German Stroke Registry — Endovascular Treatment (GSR, ClinicalTrials.gov Identifier: NCT03356392), is an ongoing, academic, prospective, multicentre registry in Germany [31]. This dataset has been exploited for machine learning outcome prediction, also within the consortium [32]. Due to its extensive sample size, it provides a good basis for initial machine learning models, as it allows for learning possibly many – heterogeneous – variations of the predictive variables. This is why we are basing the initial VALIDATE models on this dataset. Statistics of the dataset can be found in Appendix A in Table 6.

MRCLEAN

MR CLEAN was a pragmatic, phase 3, multicentre clinical trial with randomized treatment-group assignments, open-label treatment, and blinded end-point evaluation. Intraarterial treatment (intraarterial thrombolysis, mechanical treatment, or both) plus usual care (which could include intravenous administration of alteplase) was compared with usual care alone (control group) in patients with acute ischemic stroke and a proximal intracranial arterial occlusion of the anterior circulation that was confirmed on vessel imaging. The study publication can be found here [33]. This study dataset has been a central point of attention and machine learning modelling in the previous years, which have also resulted in the development of the MRPREDICTS outcome prediction tool[2]. Consortium members have worked on this data previously and also have access for further research, thus we included it as one of our external test datasets. Statistics of the dataset can be found in Appendix A in Table 7.

Dataset from Heidelberg University Hospital (HDB)

In a previous project, some consortium members have collaborated in machine learning prediction of stroke outcome on data from the Heidelberg University Hospital [34]. Thus, this – slightly smaller – dataset is also instantly available and was included in the first experiments of modelling. We would like to note, that Heidelberg University Hospital is one of the VALIDATE clinical sites, thus more extensive retrospective data is in progress to be made available. Statistics of the dataset can be found in Appendix A in Table 8.

Data schema definition

In order to develop and validate an AI model across multiple centres (or datasets), one needs to create a common, shared list of input/output variables that are available at each location, and database. This list is called a data schema, which includes the description of variables, the used metrics of representation (e.g., blood pressure – mmHg, age – years) and the definition of any transformations applied to the data (e.g., blood pressure -> [low, mid, high] instead of mmHg). All this information needs to be synchronized across databases (retrospective or prospective) so that during training, testing or deployment models are provided with the same information regardless of the source.

In the case of VALIDATE, this means harmonisation of the available retrospective datasets for our base models, retrospective data at the VALIDATE clinical sites and prospective data to be collected in the planned clinical study. We created a governing task force for data-related decisions, called Data board, including representatives from each VALIDATE clinical centre as well as VALIDATE developers. First, we initiated the harmonization process by looking at similar solutions (e.g., MRPREDICTS[3]) and studies,

[2] https://www.mrclean-trial.org/mr-predicts.html

[3] https://www.mrclean-trial.org/mr-predicts.html

employing machine learning models for the prediction of mRS in literature and previous work within the consortium. We collected an initial set of variables that have been used for this purpose. Next, we organized multiple discussions with the VALIDATE Data board and reviewed the list for completeness and potential extensions. The VALIDATE clinical experts reviewed the list from two main perspectives:

1. are all information captured routinely used and necessary for decision-making in stroke,
2. availability in their retrospective databases.

As a result, we created a so-called target list of variables from the VALIDATE teams perspective, which got compared next with our retrospective databases available from previous research projects described above (MRCLEAN, GSR). We marked variables that the VALIDATE Data board deemed important and were not represented in the retrospective datasets. We will look into certain ways to include these into modelling despite unavailability in the first stages of model training. The resulting Data schema is shown in Appendix A below (see Table 5).

We note that data harmonisation is ongoing, and the overarching process is not finished or finalized at the current time of reporting. Results from modelling experiments will be also considered to finalize the selection of variables since prediction performance and feasibility of handling the inclusion of variables in later fine-tuning stages of the models must be assessed to decide on a final strategy. We will report on the continuation of this process in the next deliverables.

VALIDATE modelling framework and code base

An important aspect of machine learning data analysis is reproducibility. Even more importantly, when purposing models for higher TRLs, experimental robustness and systematic comparison of performance are crucial. To this end, we are developing the VALIDATE modelling framework, which is a maintained code base comprehending the necessary pipeline for training and evaluating models as well as implementations of all necessary functionalities for satisfying the defined requirements. In principle, this framework should give a clear methodological explanation of how a model on a certain TRL level has been trained, evaluated, and released. For robustness, the developed code has a high level of unit test coverage, which ensures the developed functionalities do not produce errors and execute the purposed function. All unit tests are run automatically through Git Actions by every commit and a report is created about passing or failing the tests. The framework is dockerizable and will be distributed to every partner working on model development (WP2), but of course, is also available for review to all consortium members and will be published as open source with relevant project publications. The current development is concerned with integrating the Federated Learning functionalities using the FLOWER framework. This framework has been used to generate the results of Experiment 1 below. Extension with Tree-based modelling is also currently ongoing thus we have not been able to exploit it for Experiment 2 yet.

Conclusion

In the current deliverable we have reported on the progress of the on-going work and specific tasks of Work Package 2. We did not identify any major impediment and all the outlined work seems to progress well. We are at a time point in the project where the major co-creation activities are starting and thus a significant proportion of the reported work so far has been focused on creating the proper pathway for this. Many of the outcomes reported here are either already under review or will be reviewed as a next step by internal and external expert boards.

In particular, WP2 is actively participating in the 2-3 weekly review and planning meetings of WP3 for a regular exchange on progress in model and software development. Together with WP3, we have planned a workshop on Medical Device Regulation for the entire consortium, where we will present the research done so far (also referenced in this document) and we will determine next steps and stakeholders necessary to be involved.

Moreover, as a result of a previous deliverable (D1.1), we are in the phase of establishing a regular exchange and review of ethical requirements with WP1. Additionally, the currently on-going Z-inspection assessment will also greatly contribute to further development of the VALIDATE modelling framework and guidelines on TRL definition for AI-based CDSS.

On a more practical note, we are expecting answers on the remote access and server setups from the clinical sites' IT departments. Once access is granted the experimentation about translation and evaluation of models on the VALIDATE clinical sites will begin. In parallel, we are ensuring the documentation of all experiments and adherence to our own TRL definitions. We expect to be able to report a consistent experiment history corresponding to an established framework of TRL transitions for AI-based CDSS in the next deliverable from WP2.

As a final note, we are planning a few publications from the currently presented works to further increase dissemination of the impactful work in VALIDATE.

Appendix A Data

Harmonised Data schema

Table 5 lists a comparison of the availability of variables considered for the VALIDATE data schema. HDB denotes retrospective data from University Hospital Heidelberg, which is one of the VALIDATE clinical centres. Green rows mark variables considered in the first experiments of modelling reported below. “x” in the cell indicates the availability of the given variable. MRPREDICTS refers to an existing online tool to predict mRS using clinical variables[4]. This tool provided the basis to our harmonisation process, due to the same purpose and the fact that the initial dataset for the development of the tool was the same MRCLEAN dataset also available for the consortium. The other metrics used in the table are: NIHSS: National Institutes of Health Stroke Scale - objectively quantify the impairment caused by a stroke and aid planning post-acute care disposition, ASPECTS: Alberta Stroke Program Early CT score - assess early ischemic changes on non-contrast CT head

Table 5 Current VALIDATE data schema

Group	Variable	MRCLEAN	HDB	German Stroke Registry	MRPREDICTS
Demographic					
	Age	x	x	x	x
	Biological sex	x	x	x	
Examination					
	NIHSS	x	x	x	x
	Systolic blood pressure	x			x
	Pre-stroke mRS	x	x	x	x
Medication					
	Antiplatelet	x		x	
	Oral anticoagulant			x	
	Heparin(oids)	x			
	Statins	x			
Lab parameters					
	INR (prothrombin time test)				
	Creatinine	x			
	Serum glucose	x			
	White-bloodcell count				
	Hemoglobin				

[4] https://www.mrclean-trial.org/mr-predicts.html

	Platelet count				
Medical history					
	Previous stroke	x			x
	Atrial Fibrillation	x		x	
	Diabetes mellitus	x	x	x	x
Treatment					
	Medical				
	Endovascular treatment (EVT)	x	all got EVT	all got EVT	x
	Intravenous thrombolysis	x	x	x	x
Imaging					
	ASPECTS	x		(low, middle, high)	x
	Location of occlusion	x	x	x	x
	Tandem occlusion				
	CTP CVB volume				
	CTP Tmax 6 volume				
	CTP mismatch ratio				
	CTP mismatch volume				
	Collateral score (CTA)	x			x
Workflow					
	Start of thrombolysis	x			
	Onset-to-imaging		x		
	Onset-to-admission				
	Onset-to-groin	x	x		x
	Onset-to-recanalization	x	x		
	Pre-procedure mTICI		x		
	Post-procedure mTICI	x	x	x	

	Procedural complications				
Outcomes					
24hrs					
	NIHSS	x	x	x	
	Symptomatic ICH conversion				
	Groin hematoma				
	Seizure				
	Infection	x			
	Congestive heart failure				
90days					
	mRS	x	x	x	x

Retrospective dataset statistics

Table 6 Summary statistics for the German Stroke Registry Dataset

	count	mean	std	min	25%	50%	75%	max
AGE	5412	73.20048	13.144924	0	65	76	82	100
AGE_MISSING	5412	0.000554	0.02354	0	0	0	0	1
SEX_F	5412	0.507206	0.500733	-1	0	1	1	1
NIHSS_BL	5412	14.295455	7.543217	-1	9	14	19	42
NIHSS_BL_MISSING	5412	0.015706	0.124346	0	0	0	0	1
MRS_PRE	5412	0.676644	1.229235	-1	0	0	1	5
MRS_PRE_MISSING	5412	0.029749	0.169909	0	0	0	0	1
AF	5412	0.401515	0.514888	-1	0	0	1	1
DM	5412	0.208056	0.434114	-1	0	0	0	1
IVT	5412	0.504065	0.509186	-1	0	1	1	1
MRS_90	5412	3.435144	2.159943	0	1	4	6	6
EVT	5412	1	0	1	1	1	1	1
ONSET_TO_GROIN	5412	0	0	0	0	0	0	0
ONSET_TO_GROIN_MISSING	5412	1	0	1	1	1	1	1

Table 7 Summary statistics for the MRCLEAN dataset

	count	mean	std	min	25%	50%	75%	max
AGE	500	64.932	13.76948	23	55.75	66	76	97
AGE_MISSING	500	0	0	0	0	0	0	0
SEX_F	500	0.416	0.493387	0	0	0	1	1
NIHSS_BL	500	17.598	5.572657	3	14	18	22	38
NIHSS_BL_MISSING	500	0	0	0	0	0	0	0
MRS_PRE	500	0.342	0.826076	0	0	0	0	5
MRS_PRE_MISSING	500	0	0	0	0	0	0	0
AF	500	0.27	0.444404	0	0	0	1	1
DM	500	0.136	0.343132	0	0	0	0	1
IVT	500	0.89	0.313203	0	1	1	1	1
EVT	500	0.434	0.496121	0	0	0	1	1
ONSET_TO_GROIN	500	114.99	138.82085	0	0	0	240.75	455
ONSET_TO_GROIN_MISSING	500	0.566	0.496121	0	0	1	1	1
MRS_90	500	3.74	1.615432	0	2	4	5	6

Table 8 Summary statistics for the HBD dataset

	count	mean	std	min	25%	50%	75%	max
AGE	267	67.70412	15.196164	0	59	71	78	95
AGE_MISSING	267	0.003745	0.061199	0	0	0	0	1
SEX_F	267	0.52809	0.500148	0	0	1	1	1
NIHSS_BL	267	15.681648	6.185096	0	12	16	20	33
NIHSS_BL_MISSING	267	0	0	0	0	0	0	0
MRS_PRE	267	0.719101	1.03679	-1	0	0	1	4
MRS_PRE_MISSING	267	0.011236	0.105601	0	0	0	0	1
AF	267	-1	0	-1	-1	-1	-1	-1

DM	267	0.179775	0.403792	-1	0	0	0	1
IVT	267	0.621723	0.485868	0	0	1	1	1
EVT	267	1	0	1	1	1	1	1
ONSET_TO_GROIN	267	598.441948	698.452333	63	252	392	731.5	6374
ONSET_TO_GROIN_MISSING	267	0	0	0	0	0	0	0
MRS_90	267	3.213483	1.794762	0	2	3	4	6

Appendix B Report on experiments in T2.4

Experiment 1: Neural Networks

Rationale

In this section, we present the experiments we have performed using neural network models to predict functional outcome of ischemic stroke patients measured by the mRS score. As discussed above, neural networks have the most straightforward adaptation for federated learning, thus we targeted these models for inclusion within our work. In this experiment, we aimed to train models on a single source of retrospective data (GSR) and test how its performance is retained on other sources of retrospective data originating from other centres (population).

Methods

Data

For training and hyperparameter search we used retrospective data from the German Stroke Registry, statistics can be found in the previous section. As highlighted in the current state of the Data schema (see Table 5) we included the following input variables:

- Age
- Biological sex
- Baseline NIHSS
- Pre-stroke mRS
- Atrial fibrillation
- Diabetes mellitus
- Intravenous thrombolysis
- Endovascular treatment
- (Time from stroke onset to groin) - not present in training data, but used in some experiments in the external evaluation data (MRCLEAN, HDB)

Even though we considered a smaller set of variables for these first experiments than the comprehensive list in our Data schema, there were still cases of variables represented in one data and not in others. Overcoming this issue in a rather flexible way is desirable since it might happen in the actual use of the model that some variables are non-collectable. For the following experiments, we experimented with the following approach:

- For nominal variables, an additional class of “unknown” is introduced (with a value of -1)
- For ordinal variables, an additional indicator variable is introduced, and unknown values are filled with -1
- For continuous variables, an additional indicator variable is introduced, and unknown values are filled with 0

We tested this approach against omitting variables with missing values and did not observe a significant change in prediction performance.

Furthermore, in the GSR data, all patients received endovascular treatment by mechanical thrombectomy (similar to the HDB dataset). This is of course another limitation that we will specifically analyse in the future. Seeing only patients eligible for treatment introduces a bias to the data sample (and hence the model) and means also that the model’s prediction with respect to an outcome might not be precise for those not eligible. To assess this, we include an evaluation of performance on the subgroups of patients from MRCLEAN who did or did not receive (due to randomization) endovascular treatment (referenced as “EVT only” and “without EVT”).

Another detail is that onset-to-groin times were not collected in the GSR database. In contrast to the previous case, this means we will have to omit this variable when evaluating the model since the model has not seen any interaction (or possible distribution) of this variable. Both these limitations are reflected on below in the results section. Lastly, the HDB dataset did not contain information about Atrial fibrillation, thus we encoded all as missing.

As a target variable we used the mRS score in two common ways. First, we employed the classical dichotomization and trained models to predict the binary outcome (referenced as "dichotomized model"):

- mRS 0 - 2 -> good outcome
- mRS 3 - 6 -> bad outcome

Second, we took the full-scale mRS score and trained a multi-class model to deliver a probability score for each of the 7 classes of the mRS scale, these probabilities add up to 1. This model is referenced as "full-scale model". Moreover, we included an evaluation of these full-scale models in the dichotomized and in a trichotomized representation. This enables us to compare the binary prediction performance of the dichotomized and the full-scale model and also to compare our results to the literature (where full-scale mRS is not classically predicted). For the trichotomy we used:

- mRS 0 – 2 -> favourable outcome,
- mRS 3 – 4 -> intermediate outcome,
- mRS 5 – 6 -> miserable outcome

Model

The model we employed was a neural network, with batch normalization layer before hidden layers and dropout after them. The number of hidden layers, number of neurons in each layer and rate of dropout was tuned through hyperparameter search on the validation sets.

Experimental setup

For training and evaluating models in a robust way we employed a 4-fold cross-validation mechanism, with non-overlapping test sets and random validation sets (25%) separated from training folds for hyperparameter tuning. A visualization is shown in Figure 5. Zero mean, unit variance standardization was applied to continuous variables with training statistics. We report average test performance across the 4 test sets. For hyperparameter search we used Random Search in each fold. This results in 4 models and parameter sets, from which we pick the best for further evaluation on the MRCLEAN and HDB datasets. Models were evaluated using the Area Under the Receiver Operator Characteristics curve (ROC AUC), Accuracy, Balanced accuracy (average of class recalls) and F1 score in case of binary predictions and using macro ROC AUC and macro F1 score for multi-class predictions.

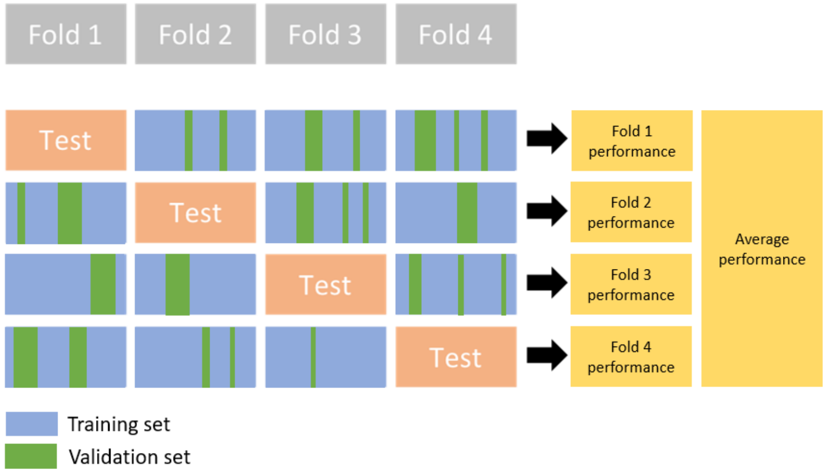

Figure 5 Visualisation of the cross-fold validation process

Results

In the following sections we present our results for all executed experiments. As a summary we evaluate dichotomized and full-scale model performance on all evaluation metrics and confusion matrices on:

- GSR test sets
- All patients from MRCLEAN, HDB
- EVT only and without EVT patients in MRCLEAN
- Patients from the top 6 most represented centers of MRCLEAN

Comparison of test performance

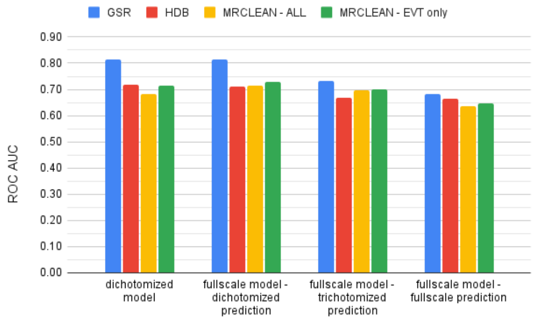

Figure 6 ROC AUC. Comparison of test performances on GSR, HDB and MRCLEAN dataset.

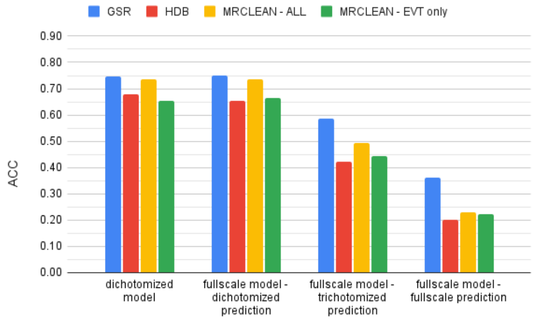

Figure 7 Accuracy. Comparison of test performances on GSR, HDB and MRCLEAN dataset.

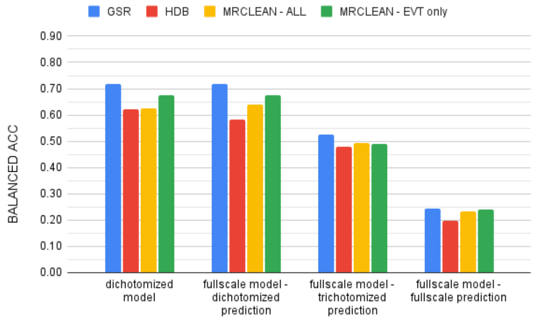

Figure 8 Balanced Accuracy. Comparison of test performances on GSR, HDB and MRCLEAN dataset.

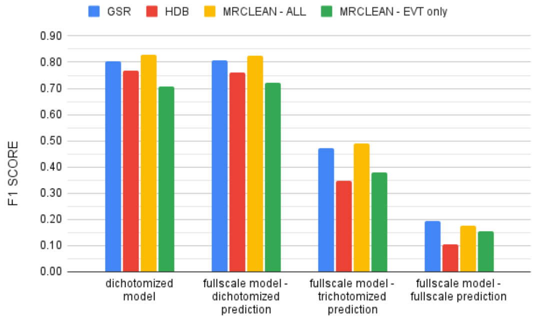

Figure 9 F1 score. Comparison of test performances on GSR, HDB and MRCLEAN dataset.

Confusion matrices for test predictions

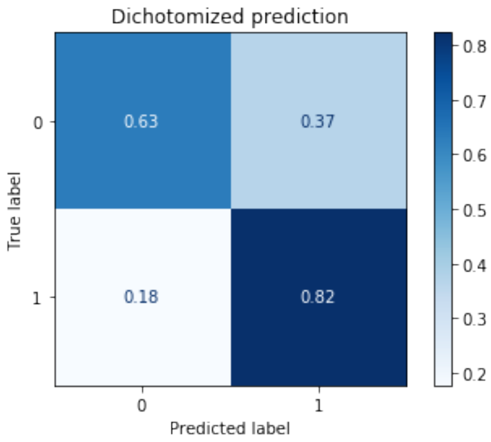

Figure 10 Confusion matrix normalized over the true labels. Test predictions on GSR dataset. Dichotomized model.

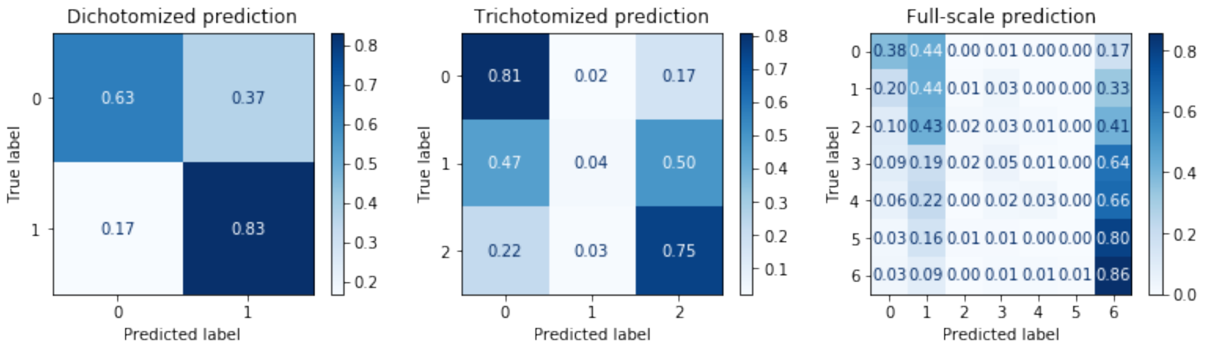

Figure 11 Confusion matrix normalized over the true labels. Test predictions on GSR dataset. Full-scale model.

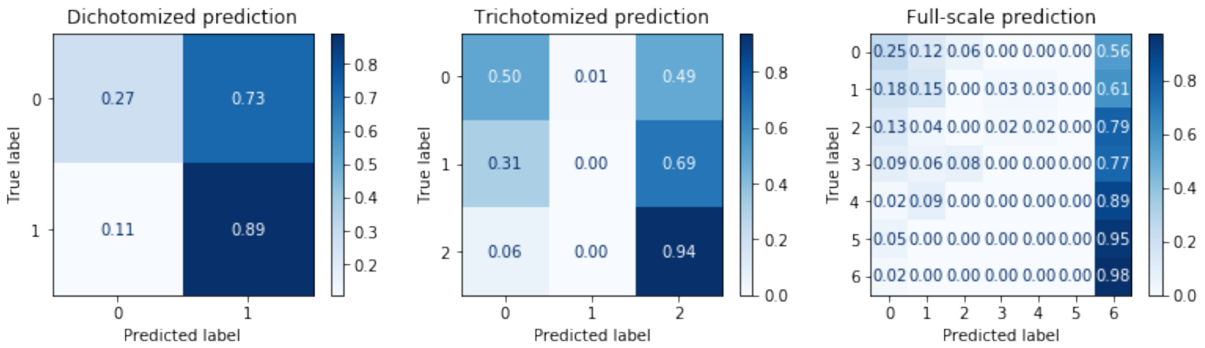

Figure 12 Confusion matrix normalized over the true labels. Test predictions on HDB dataset. Full-scale model.

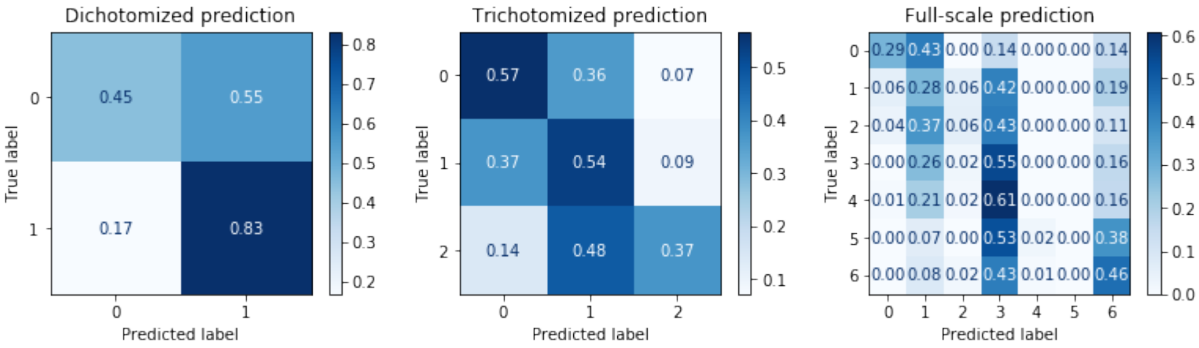

Figure 13 Confusion matrix normalized over the true labels. Test predictions on MRCLEAN dataset. Full-scale model.

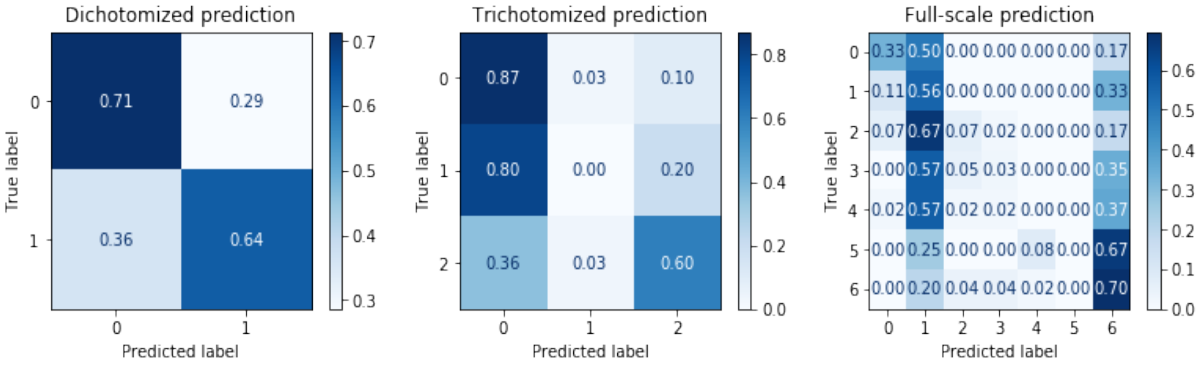

Figure 14 Confusion matrix normalized over the true labels. Test predictions on MRCLEAN dataset – patients with EVT. Full-scale model.

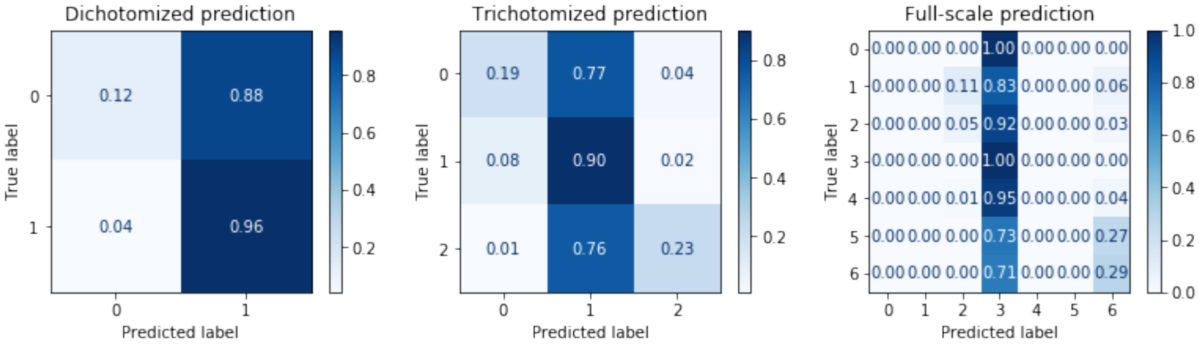

Figure 15 Confusion matrix normalized over the true labels. Test predictions on MRCLEAN dataset – patients without EVT. Full-scale model.

Test performance for diverse datasets

Table 9 Performance of models trained on GSR data, evaluated on GSR data. Average of 4 non-overlapping test folds. 1353 patients per fold.

	Dichotomized model	Full-scale model		
Metrics on test sets – mean (std)	Dichotomized performance	Dichotomized performance	Trichotomized performance	Full-scale performance
ROC AUC	0.813 (0.008)	0.813 (0.007)	0.733 (0.008)	0.681 (0.006)
ACC	0.747 (0.003)	0.749 (0.006)	0.588 (0.010)	0.362 (0.012)
BALANCED ACC	0.718 (0.008)	0.718 (0.011)	0.526 (0.007)	0.243 (0.012)
F1	0.805 (0.005)	0.807 (0.005)	0.473 (0.004)	0.193 (0.010)

Table 10 Performance of models trained on GSR data, evaluated on HDB data. All 267 patients.

	Dichotomized model	Full-scale model		
	Dichotomized performance	Dichotomized performance	Trichotomized performance	Full-scale performance
ROC AUC	0.717	0.711	0.666	0.664
ACC	0.678	0.655	0.423	0.202
BALANCED ACC	0.621	0.583	0.480	0.197
F1	0.768	0.762	0.349	0.106

Table 11 Performance of models trained on GSR data, evaluated on MRCLEAN data. All 500 patients.

	Dichotomized model	Full-scale model		
	Dichotomized performance	Dichotomized performance	Trichotomized performance	Full-scale performance
ROC AUC	0.683	0.716	0.696	0.635
ACC	0.736	0.734	0.494	0.230
BALANCED ACC	0.626	0.640	0.492	0.234
F1	0.828	0.823	0.489	0.176

Table 12 Performance of models trained on GSR data, evaluated on MRCLEAN data. Patients with EVT - 217 patients.

	Dichotomized model	Full-scale model		
	Dichotomized performance	Dichotomized performance	Trichotomized performance	Full-scale performance
ROC AUC	0.715	0.728	0.700	0.647
ACC	0.654	0.664	0.442	0.221
BALANCED ACC	0.674	0.677	0.492	0.239
F1	0.708	0.720	0.380	0.155

Table 13 Performance of models trained on GSR data, evaluated on MRCLEAN data. Patients without EVT - 283 patients.

	Dichotomized model	Full-scale model		
	Dichotomized performance	Dichotomized performance	Trichotomized performance	Full-scale performance
ROC AUC	0.682	0.683	0.700	0.677
ACC	0.799	0.788	0.534	0.237
BALANCED ACC	0.500	0.539	0.442	0.192
F1	0.888	0.878	0.428	0.114

Experiment 2: Tree-based Methods

Rationale

As mentioned early in this deliverable, the development and application of tree-based FL are still very limited compared to FL with neural network models. Furthermore, studies have reported that tree-based models may outperform deep learning approaches on tabular data [30]. To better estimate the potential of using tree-based models to predict post-stroke outcome of patients at 90 days after stroke using clinical data, we performed a single-site in-lab experiment using a number of tree-based machine learning approaches on the MRCLEAN data. The work reported here was completed at the TU Dublin site and we are currently working on transferring the trained models to CUB and MBH for a broader validation on the datasets available at these sites, and after that to work with CUB and MBH on the implementation of a tree-based FL for medical applications together with guidelines on the use of the framework and validation and verification methods.

Pre-processing of MRCLEAN data

To train the tree-based models, we use MRCLEAN data. Prior to training the models the data were pre-processed as follows:

i. Selecting the common input features for modelling (see Table 5)
ii. Adding new features that detail the absence of the considered features such as age, NIHSS at baseline, MRS before 90 days after stroke. In this way, we can keep the majority of patients for the data after filtering.
iii. Applying standard normalization to the continuous features, such as age, NIHSS at bas- line, MRS before 90 days after stroke and the onset-to-groin time.
iv. Extracting MRS scores at 90 days after stroke as the output label for prediction, such as the original full-scale of MRS, the binarized and the trichotomized mRS scale generated from the full-scale MRS (as described in Experiment 1: Neural Networks)

After being pre-processed, data is then used for training and validation of the tree-based models. In this work, we developed three commonly used tree-based machine learning models namely Decision Tree, Random Forest and XGBoost. The next section will briefly describe those approaches.

Description of tree-based methods

1. Decision tree

A Decision Tree (DT) is an information-based machine learning model that makes predictions based on “if-then-else” rules in a hierarchical tree structure [35]. Each tree is composed of multiple nodes that link together in a hierarchical way, where each node defines one attribute to test. The structure of the tree starts from a root node that links to internal nodes and terminates at the leaf nodes; the linkages between nodes are called branches.

Generally, decision trees are built in a recursive and depth-first manner, i.e., the algorithm starts building the tree at the root node and then iteratively grows the tree by extending the leaf nodes. Briefly, the algorithm for creating a decision tree from data follows three main steps:

i. Selecting the best descriptor using a feature selection measure (such as information gain) as the split criterion at the root node
ii. Adding the root node to the tree and labelling it with the split criterion of the selected descriptor found in i)
iii. Splitting the training dataset using the criterion of the node from ii) into partitions. For each partition, a branch is then grown from the considered node.

Steps i), ii) and iii) are then repeated for each newly created branch using the relevant partitions of the training dataset in iii) while excluding the descriptors that were used for the splitting. These steps are repeated until all the data points in the relevant partition have the same class label (or some other convergence criterion is reached, e.g., maximum depth of the tree, minimum number of examples, etc.). At that time, a leaf node is then created for predicting the data point with the majority class label for the training examples that have reached the node.

In this work, we use one of the most popular decision tree algorithms named CART (classification and regression trees) [36] which is available in Scikit-Learn library for predicting the outcome of patients with stroke.

2. Random Forest

A Random Forests (RF) is an ensemble learning method that trains in parallel a set of decision trees [37]. Each tree is trained on a different random bootstrap subset that is sampled from the training dataset, i.e., the subset sample has the same number of data points as the training dataset and sampling is done with replacement. When predicting a new data point, each tree will vote for the class label that the unknown data point belongs to. The class label returned from the ensemble is the label that receives the majority of votes from across the trees of the RF.

The application of RF in this work is done by using RandomForestClassifier class available in Scikit-Learn library.

3. XGBoost

Boosting is an ensemble machine learning strategy that aims to combine multiple models together and generate a better model. Gradient Boosting [38] is one of the best-known boosting methods. This method creates a tree ensemble by sequentially adding decision trees to the ensemble. Each tree is created in the way that it corrects for the errors made by the trees previously added to the ensemble; i.e., this method tries to fit a new decision tree to the residual errors that are made by the previous tree.

We used the python library XGBoost (Extreme Gradient Boosting) to train and implement our XGBoost models.

Parameters and Evaluation metrics

In these experiments, 20% of the dataset was randomly selected to create a hold-out test set. The remaining of 80% of the dataset is used to train the models. To find the optimal parameters for each tree-based model, we applied randomized cross-validated search on the training set using k=4 as the number of k-fold cross-validation. All the experiments were run with 10 random seeds, going from seed value 0 to 9. The details of the values of parameters used for training each model are detailed in Table 14 below.

Table 14 Parameters of tree-based models used for randomized cross-validated search

Model	Parameter	Values
Decision Tree		
	Feature selection measure	Gini, Entropy
	Maximum depth of a tree	[3, 5, 7, 9, 11, 13, 15, 20, 25, no limit]
Random Forest		
	Number of estimators	[10, 20, …, 180]
	Feature selection measure	Gini, Entropy
	Maximum depth of a tree	[3, 5, 7, 9, 11, 13, 15, 20, 25, no limit]
	Using boostrap	True, False
XGBoost		
	Learning rate	[0.01, 0.05, 0.1, 0.15, 0.20,]
	Maximum depth of a tree	[3, 5, 7, 9, 11, 13, 15, 20, 25, no limit]
	Maximum number of nodes to be added	[5, 10, … 45]
	Number of trees	[10, 20, …, 180]

The evaluation metric used in this work is mainly F1-score which is a harmonic mean precision and recall of the model [39]. Similar to the previous experiment (as in Section Experiment 1: Neural Networks) macro-averaged F1-score was calculated and we used this metric to evaluate the prediction between models.

Results

The tree models were trained to predict the dichotomized mRS score and the full-scale mRS. Then, from the prediction of the full-scale mRS model, we generated the trichotomized and dichotomized mRS values for evaluating the prediction results between methods. Table 15 below shows the performance of the three studied tree-based models after training and validation on MRCLEAN data.

Table 15 TTT Prediction results of the tree-based modelling for predicting dichotomized and full-scale mRS of MRCLEAN data. STD - Standard deviation, CI - Confidence interval at 95%.

Tree-based model	Metrics	Dichotomized mRS			Full-scale mRS			Trichotomized from predicted full-scale mRS			Dichotomized from predicted full-scale mRS		
		Mean	STD	CI	Mean	STD	CI	Mean	STD	CI	Mean	STD	CI
Decision Tree	F1-score	0.550	0.037	0.023	0.170	0.025	0.015	0.443	0.040	0.025	0.564	0.059	0.036
	Precision	0.656	0.066	0.041	0.177	0.047	0.029	0.444	0.041	0.025	0.567	0.059	0.037
	Recall	0.557	0.026	0.016	0.190	0.031	0.019	0.459	0.046	0.029	0.567	0.059	0.037
	Accuracy	0.754	0.018	0.011	0.263	0.050	0.031	0.461	0.029	0.018	0.680	0.041	0.025
Random Forest	F1-score	0.490	0.054	0.033	0.192	0.027	0.017	0.486	0.031	0.019	0.574	0.032	0.020
	Precision	0.594	0.181	0.112	0.211	0.045	0.028	0.505	0.039	0.024	0.608	0.061	0.038
	Recall	0.527	0.027	0.017	0.213	0.025	0.016	0.502	0.032	0.020	0.569	0.028	0.017
	Accuracy	0.750	0.016	0.010	0.310	0.047	0.029	0.518	0.034	0.021	0.724	0.033	0.020
XGBoost	F1-score	0.540	0.023	0.014	0.187	0.027	0.017	0.455	0.062	0.038	0.541	0.044	0.027
	Precision	0.606	0.073	0.045	0.204	0.058	0.036	0.456	0.059	0.036	0.548	0.050	0.031
	Recall	0.545	0.020	0.012	0.200	0.031	0.019	0.469	0.065	0.040	0.541	0.040	0.025
	Accuracy	0.728	0.033	0.020	0.272	0.038	0.023	0.480	0.056	0.034	0.681	0.032	0.020

For all three models the F1-score obtained for predicting Dichotomized mRS is much higher than that for the Full-Scale mRS target. For full-scale mRS, Random Forest obtains the best performance (F1-score of 0.192, standard deviation of 0.027, confidence interval at 95% is 0.017) and for dichotomized mRS, Decision Tree is the optimal one (F1-score of 0.550, standard deviation of 0.037, confidence interval at 95% is 0.023).

Interestingly, for all three model types, if we dichotomize the predicted mRS from a full-scale mRS model, the performance results are better than the corresponding model that was trained to predict a dichotomized score. This suggests that in many instances the errors of the full-scale mRS model are caused by predictions that are for neighbouring labels to the true label (i.e., the model has many near misses). Extending this approach to trichotomized results shows a decrease in performance compared to either directly predicting or posthoc mapping full-scale mRS predictions to dichotomised prediction results, but an improvement compared to the original full-scale prediction. Overall the best F1-score is obtained by the RF model using a dichotomized from the predicted full mRS (F1-score of 0.574, a standard deviation of 0.032, confidence interval at 95% is 0.020). Similar trends of improvement can also be seen for DT and XGBoost when we dichotomize the predicted mRS from the full-scale mRS models rather than predict dichotomized scores directly. This suggests that an interesting direction for future work may be to explore the trade-off between predicting full-scale mRS and the level of detail that a clinician requires for their decisions. However, overall none of these models have obtained sufficiently high performance to be useful in a clinical setting. We believe that using a larger dataset would improve performance, and we will explore the benefits of this in later experiments when we apply FL for tree-based methods across multiple sites. These results will provide a useful baseline for these future experiments. Also, if we compare these results with the results from the neural network experiments on MRCLEAN reported in Figure 9 we see that both the neural networks and tree-based models obtain similar F1 scores on this data in the full-scale mRS setting, however the neural models tend to outperform the tree-based models in the other settings. Better understanding what is causing this difference in another direction for future research.

To understand which feature contributes most to the predictions of the trained models (dichotomized mRS and full-scale mRS), feature importance was extracted from these tree-based models. Figure 16 and Figure 17 respectively illustrate the feature importance returned by the models of predicting dichotomized and full-scale mRS. We annotated the name of the input features as follows:

- AGE: age of the patient at randomization
- AGE_MISSING: the feature that states if age of the patient is missing.
- SEX_F: gender of the patient
- NIHSS_BL: NIHSS score at randomization.
- NIHSS_BL_MISSING: the feature that states if NIHSS is missing.
- MRS_PRE: pre-stroke MRS.
- MRS_PRE_MISSING: the feature that states if PRE_MRS is missing.
- AF: Atrial fibrillation
- DM: Diabetes mellitus
- IVT: Intravenous thrombolysis
- EVT: Endovascular treatment
- ONSET_TO_GROIN: Time from stroke onset to groin
- ONSET_TO_GROIN_MISSING: the feature that states if the time from stroke onset to groin is missing.

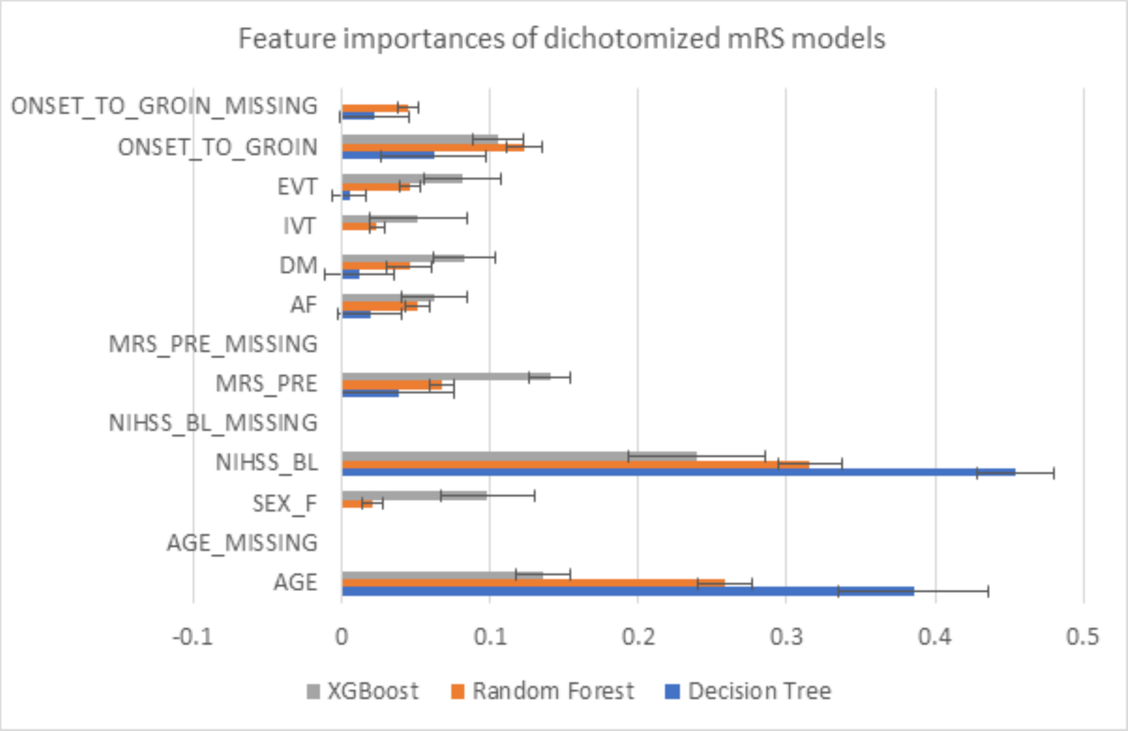

Figure 16 Feature importance of tree-based models trained for predicting the dichotomized mRS of the patient at 90 days after stroke.

As shown in Figure 16, the four features that contribute most to the predictions of the dichotomized mRS models are NIHSS_BL, AGE, ONSET_TO_GROIN and MRS_PRE. Depending on the tree-based approach, the order of importance of these features is different. For the DT model (which was the best performing model on the dichotomized mRS prediction task), the most important feature is NIHSS_BL, then followed by AGE, ONSET_TO_GROIN and MRS_PRE. The features EVT, IVT, DM and AF are the feature that present a low importance to the model. Moreover, the additional feature like AGE_MISSING, MRS_PRE_MISSING, NIHSS_BL_MISSING and ONSET_TO_GROIN_MISSING were shown to be the least or no importance for the prediction.

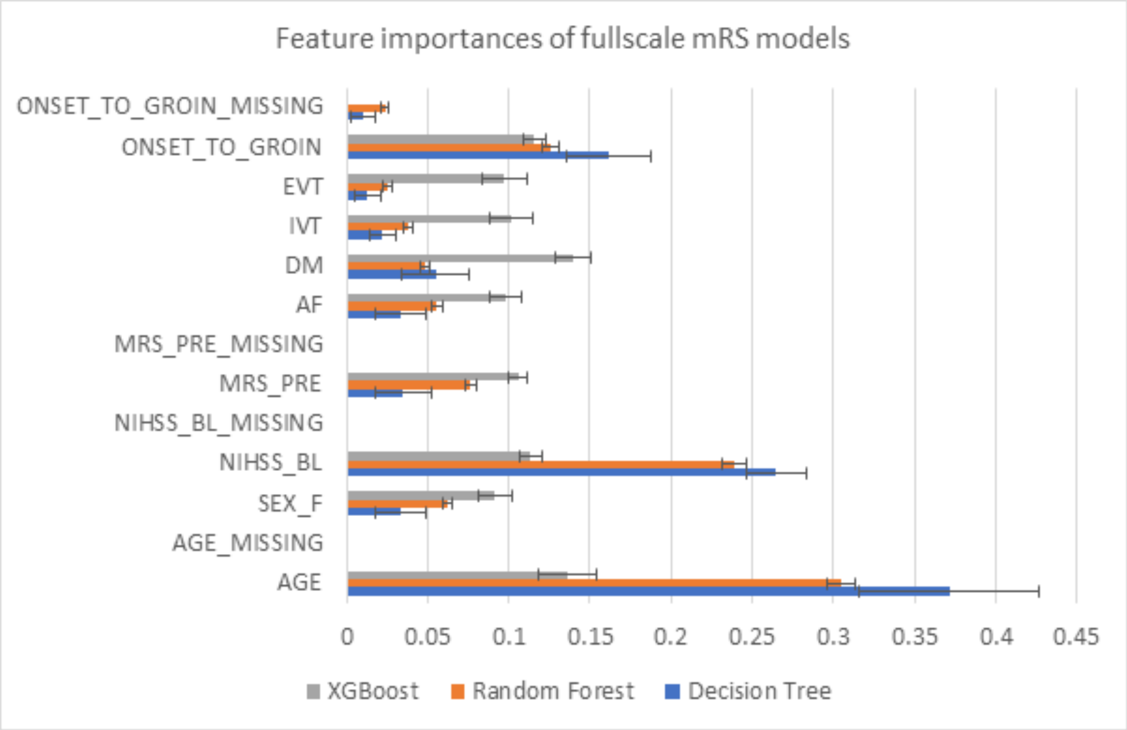

Figure 17 Feature importance of tree-based models trained for predicting the full-scale mRS of the patient at 90 days after stroke.

A similar feature importance analysis is shown in Figure 17 for the full-scale mRS prediction models. The top four features that contribute to the full-scale mRS models are AGE, NIHSS_BL, ONSET_TO_GROIN and MRS_PRE (or DM for XGBoost). For the RF model (the best performing model for predicting full-scale mRS score), the most important feature is AGE, then followed by NIHSS_BL, ONSET_TO_GROIN and MRS_PRE. The features EVT, IVT, DM and AF are the feature that present a low importance to the model. The additional feature like AGE_MISSING, MRS_PRE_MISSING, NIHSS_BL_MISSING and ONSET_TO_GROIN_MISSING were shown to be the least or no importance for the prediction.

Comparing the feature importance of these tree-based models we conclude that the age of the patient, the NIHSS score at randomization, the onset to groin time and the mRS score before 90 days after stroke are the most important information that contribute to the prediction of the outcome of the patient at 90 days after stroke. Furthermore, the procedure of dichotomizing the predicted mRS from the full-scale mRS improves the prediction of dichotomized mRS and Random Forrest is shown to be the most adaptable machine-learning method for this procedure. The feature importance obtained by the models seems sensible. At this point, the models are not obtaining sufficiently high performance to be considered for clinical use, and so we did not ask clinicians to review the feature ranking by the models as we did not wish to waste clinicians' time on assessing a weak model. However, as the models become more mature and their performance improves this type of feature analysis will be used as one of the methods for discussing and explaining model decisions with clinicians.

Although the obtained performance is low, this preliminary result is used as fundamental work that will feed into later iterations of model development, validation and verification. In particular, we are keen to explore how the performance of tree-based models developed using FL across multiple sites compares with single-site models.

Appendix C Literature Review on Federated Learning Applications

In this appendix, we provide an overview of the federated learning literature, and where the paper reports an experiment we include a brief summary of the experimental methods and training scheme reported in each paper. This review is ongoing and the information reported here is best understood as working notes that are tracking our preparations towards the review publication.

Title	DOI	Methods, Experiments	Training scheme
Distributed learning: Developing a predictive model based on data from multiple hospitals without data leaving the hospital – A real life proof of concept	https://doi.org/10.1016/j.radonc.2016.10.002	distributed learning with aggregation server	- CPTs were obtained by learning locally from each hospital. - 5-fold cross-validation using 80% of all patients selected at random for training - The CPTs were sent to the central location, where they were combined by weighted averaging: Individual table entries were weighted in proportion to the number of patients available at the hospital. - The weighted CPTs, which comprise the global model, were sent back to each site to be validated on the remaining 20% of patients on each site. - This was repeated 5 times.
Communication-Efficient Learning of Deep Networks from Decentralized Data	https://doi.org/10.48550/arXiv.1602.05629	IID data partition non-IID data partition: for MNIST and CIFAR-10 2 classes per client - both partitions are balanced: all clients have the same number of examples (600) Experiments: - comparison of FedSGD and FedAvg (with varying B, E and C) on IID data and non-IID data	-100 clients - most experiments done on MNIST - on CIFAR-10 only one IID experiment
Developing and Validating a Survival Prediction Model for NSCLC Patients Through Distributed Learning Across 3 Countries	https://doi.org/10.1016/j.ijrobp.2017.04.021	distributed learning with aggregation server	used a formerly adapted method (Distributed learning)

A Performance Evaluation of Federated Learning Algorithms	10.1145/3286490.3286559	FL with aggregation server 1. Federated Averaging (FedAvg) 2. Federated StochasticVariance Reduced Gradient (FSVRG) 3.. CO-OP 4. collaborative data sharing (centralized dataset)	FedAvg: - a total of 56 parameter configurations tuned in random search - First, we searched for learning rate and learning rate decay with fixed C, E and B, - after which we try to improve by further exploring C and E.
Federated Learning with Non-IID Data	https://doi.org/10.48550/arXiv.1806.00582	FL with aggregation server non-IID data setting: data divided to 2 extreme cases: 1. 1-class non-IID: each client receives data partition from only a single class 2. 2-class non-IID: each client receives data partition from 2 classes Experiments: 1. SGD - model trained on centralized data 2. FedAvg IID - trained on IID data 3. FedAvg non-IID - trained on non-IID data (1-class, 2-class) Proposed method: - small subset of data from each client is globally shared, which contains a uniform distribution over classes - warm-up model can be trained ofn the globally shared data and then distributed to the clients for FL instead of random weight initialization	FedAvg: - 10 clients - batch size: 10 and 100 - local epochs: 1 and 5 - 500 federated rounds SGD: - batch size is 10 times larger. This is because the global model from FedAvg is averaged across 10 clients at each synchronization. FedAvg with IID data should be compared to SGD with shuffling data and a batch size K times larger, where K is the number of clients included at each synchronization of FedAvg. - all models initialized with the same weights - cross-validation over 5 distributions

Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data	-	FL with aggregation server 1. federated learning (FL) 2. institutional incremental learning (IIL) 3. cyclic institutional incremental learning (CIIL) 4. collaborative data sharing (centralized dataset. CDS)	FedAvg: - 10 clients - unbalanced number of patients in clients - up to 200 federated rounds - 1 local epoch per round - all clients selected in each round Collaborative cross validation: - each institution's dataset is partitioned into 5 folds - for every experiment five runs are performed, using 1 fold for validation and the other 4 folds for training Final model selection: - Each institution locally validates the received model at the start of each federated round. - Local validation results are sent to the server with the model updates. - Global validation is averaged local validation results. - loss function: negative log of DICE - Adam optimizer hyperparameter tuning: - tuned on the central trained model, final hyperparameters used for FL model
Federated learning in a medical context: A systematic literature review	https://doi.org/10.1145/3412357	-	-
Federated learning: a collaborative effort to achieve better medical imaging models for individual sites that have small labelled datasets	10.21037/qims-20-595	-	-
A collaborative online AI engine for CT-based COVID-19 diagnosis	10.1101/2020.05.10.20096073, preprint	FL with aggregation server 1. FL - publicly available Unified CT-COVID AI Diagnostic Initiative (UCADI) framework - data from 3 hospitals: 3 Tongji hospitals in Wuhan 2. FL - UCADI framework - data from 4 hospitals: 3 Tongji hospitals in Wuhan + Wuhan Union hospital (WU)	1. FL - FedAvg - 200 federated rounds, 1 local epoch per each round 2. FL - Federated transfer learning - FedAvg - 30 federated rounds, 1 local epoch per each round, start training with the global model coming from experiment 1 5-fold cross-validation on train/validation set No hyperparameter tuning: same set of

		3. model trained on centralized data from 3 Tongji hospitals in Wuhan (CDS)	local training hyperparameters for all clients.
Federated Learning on Clinical Benchmark Data: Performance Assessment	10.2196/20891	FL with aggregation server 1. centralized ML method (CML) 2. basic FL: IID data distribution, each client same number of samples 3. imbalanced FL: each client different number of samples ranging from 1 to 600 for MNIST and 50%, 30%, 20% for MIMIC-III and ECG 4. extremely skewed FL: each client had just one label from MNIST and 600 samples 5. imbalanced and skewed FL	- no cross-validation - don't specify hyperparameter tuning - early stopping - SGD FedAvg: MNIST: - 10 clients - up to 3000 federated rounds, 5 local epochs MIMIC-III: - 3 clients - up to 30 federated rounds, 2 local epochs ECG: - 3 clients - up to 30 federated rounds, 3 local epochs Test set bootstrapping: K=100 for MNIST and ECG, K=10000 for MIMIC-III
FEDERATED OPTIMIZATION IN HETEROGENEOUS NETWORKS	https://doi.org/10.48550/arXiv.1812.06127	FedProx is a generalization and re-parametrization of FedAvg: 1. adds a **proximal term scaled by μ** to FedAvg objective, which limits the impact of local updates - minimizes the distance between local and global models 2. tolerates partial work from all selected devices – allows devices to train for less than specified number of local epochs FedAvg is a special case of FedProx with μ = 0, SGD as the local optimizer, and no systems heterogeneity (equal number of epochs on each device).	For each dataset, the learning rate is tuned on FedAvg, the same learning rate is then used for all experiments on that dataset. Number of randomly selected devices per round is 10. Various number of stranglers (devices that would be dropped in FedAvg due to incomplete local training in a federated round) 0%, 50%, 90%.

Inverse Distance Aggregation for Federated Learning with Non-IID Data	https://doi.org/10.48550/arXiv.2008.07665	IDA = Inverse Distance Aggregation - changes the weighting coefficients in FedAvg - the proposed coefficients are based on the inverse distance of each client parameters to the average model of all clients. - this allows to reject or weigh less the models who are poisoning, i.e. out-of-distribution. INTRAC = INverse TRaining ACcuracy - weighting scheme that uses clients training accuracy to penalize over-fitted models and encourage under-trained models in the aggregated model.	90% of data for training, 10% for evaluation - 5000 federated rounds, 1 local epoch on ech client - report classification accuracy: 1. on local clients' test sets 2. on union of the tests sets of clients
Siloed Federated Learning for Multi-centric Histopathology Datasets		SiloBN: Instead of treating all BN parameters equally as in FedAvg, we propose to take into account the separate roles of BN statistics (μ,σ2)(μ,σ2) and learned parameters (γ,β)(γ,β). SiloBN consists in only sharing the learned BN parameters across different centers, while BN statistics remain local. Parameters of non-BN layers are shared in the standard fashion.	Data : ○ H&E stained whole slide images (WSI) of lymph node sections drawn from breast cancer patients from 2 and 5 hospitals, respectively ○ each dataset is partitioned into a training set (60%), a validation set (20%), and a test set (20%) using per-hospital stratification. Tiles from same slide (resp. tiles from same patient) are put into the same partition. ○ compare the proposed SiloBN method to two standard Federated algorithms, FedAvg and FedProx, which treat BN statistics as standard parameters Training info: ○ two deep convolutional neural network (DCNN) architectures that only differ by the presence or absence of BN layers ○ Each FL algorithm is run with E =1 or E = 10 local batch updates. Each training session is repeated 5 times (different random seeds for weight initialization and data ordering). Local optimization with

			Adam Evaluation: intra-center generalization performance: AUCROC of the trained model's predictions on each center's held-out data for single-model methods (Pooled, FedAvg,or FedProx) and personalized models are tested on held-out data for their specific training domain (Local and SiloBN)
Optimized Federated Learning on Class-biased Distributed Data Sources	preprint	FL with aggregation server FedBGVS = Federated Balanced Global Validation Score - employing a balanced global validation dataset available on the server side - aggregation algorithm is refined by using the Balanced Global Validation Score	training set on clients - 20% used for local validation, validation set - for global validation on the server, hold-out test set - on the server - 50 federrated rounds, 1 local epoch on each client per each round - 4 clients
FEDERATED LEARNING BASED ON DYNAMIC REGULARIZATION	https://doi.org/10.48550/arXiv.2111.04263	FedDyn - a dynamic regularizer for each device and each federated round - the regularizer dynamically modifies the device objective with a **penalty term scaled by α** so that, in the limit, when model parameters converge, they do so to stationary points of the global empirical loss. - the penalty terms are linear or quadratic.	- 100, 500, 1000 clients

Federated learning for predicting clinical outcomes in patients with COVID-19	https://doi.org/10.1038/s41591-021-01506-3	FL with aggregation server x-rays preprocessed to 2D images	70%, 10%, and 20% of the cases were used for training, validation, and testing - no cross-validation - no hyperparameter tuning: same set of local training hyperparameters for all clients. - cross-entropy with learning rate decay - Adam optimizer - normalization to zero-mean and unit variance FedAvg: - 20 clients - unbalanced number of patients in clients - 200 federated rounds, 1 local epoch per each round evaluation: - each **client** site selects its **best local model** by tracking the model's performance on its **local validation set** - **server** determines the **best global model** based on the **average validation scores** sent from each client site to the server after each FL round
Federated Learning of Electronic Health Records to Improve Mortality Prediction in Hospitalized Patients With COVID-19: Machine Learning Approach	10.2196/24207	FL with aggregation server 1. models trained by using only local data 2. pooled model - centralized data from all sites 3. federated model	FedAvg: - 5 clients - all client models initialized with the same weights - multiple federated rounds, 1 local epoch in each round cross-validation: - 490-fold bootstrapping - 70:30 train and test data split - AVG AUCROC across 490 folds reported hyperparameter tuning: - tuned on the central trained model, final hyperparameters used for FL model
Federated Learning for Healthcare: Systematic Review and Architecture Proposal	https://doi.org/10.1145/3501813		

Federated learning for violence incident prediction in a simulated cross-institutional psychiatric setting	https://doi.org/10.1016/j.eswa.2022.116720	FL with aggregation server 1. FL model 2. data-centralized model (DC) 3. local A model 4. local B model	5-fold cross-validation: - local dataset split into train/val and test set - grid search with 5-fold CV to find the best hyperparameters on trian/val data (for each HP combination 5 models trained, their predictions and labels are concatenated and used for performance assessment) - F1 score used for fine-tuning - final model trained on all 5 folds and tested on test set - sigmoid, binary crossentropy - class balancing using class weights - early stopping on validation loss, max 120 epochs FedAvg: - 2 clients - all client models initialized with the same weights - 1 local epoch - in each round val loss and perf. measures are tracked by: 1. sending the global model after averaging back to the clients, 2. making predictions on local val set, 3. sending the predictions and labels to server, 4. the server concats the predictions and labels and computes the loss and perf. measures
Closing the Generalization Gap of Cross-silo Federated Medical Image Segmentation	https://doi.org/10.48550/arXiv.2203.10144	FedSM = Federated Super model - consists of global model, personalised models and a model selector global model - generalizes better on joint data personalized models - generalize better on local data model selector - combines the predictions from models above SoftPull = personalized FL optimization formulation, to produce the personalized models	50%, 25%, 25% - train, validation, test split - no cross-validation - 6 clients - 150 federated rounds, 1 local epoch in each round - Dice loss function, Adam optimizer - tune the learning rate, how? - performance metric Dice

Federated Learning in Medical Imaging: Part I: Toward Multicentral Health Care Ecosystems	https://doi.org/10.1016/j.jacr.2022.03.015	-	-

Federated algorithms

Federated stochastic gradient descent (FedSGD)

FedSGD (2015) is the direct transposition of SGD to the federated setting by using a random fraction of clients and using all the data on those clients. In each round, just one batch update is performed. The gradients are averaged by the server proportionally to the number of training samples on each node and used to make a gradient descent step.

This requires a v**ery large number of communication rounds** of training to produce a good model.

Variation of FedAvg, where B = all local data, and E = 1.

Federated Averaging Algorithm (FedAvg)

FedAvg (2016) is a generalization of FedSGD, where rather than gradients the updated weights are exchanged. The rationale behind this generalization is that in FedSGD, if all local nodes start from the same initialization, averaging the gradients is strictly equivalent to averaging the weights themselves.

Advantages of FedAvg to FedSGD:

- FedAvg converges to a higher level of test-set accuracy than the baseline FedSGD models.
- FedAvg lowers communication costs.
- FedAvg produced a regularization benefit similar to that achieved by dropout.

Non-IID data problem

Zhao et al. observed that if the data is IID, then the learned model parameters in FedAvg are similar to those learned using centralized SGD, but they differ for non-IID data. And highly skewed non-IID data significantly reduces the accuracy. Possible **causes for the weight divergence** could be:

1. **varying initial weights** on each client.
2. the **earth mover's distance** (EMD) between the data distributions of each client and the global distribution. The proposed solution **is sharing a fraction of data globally**, reducing the earth mover's distance (EMD) and in turn improving the achieved accuracy. Additionally, the server can pre-train the model on the globally shared data that jumpstarts the learning process on the client side. With those measures in place, the paper reports an improvement of ≈ 30% for CIFAR-10 in the 1-class non-IID case. **However, this rarely is possible for medical use-cases.**

Moreover, they experimented with **highly skew data**:

1. 1-class non-IID: each client receives data partition from only a single class
2. 2-class non-IID: each client receives data partition from 2 classes

McMahan et al. have demonstrated in the original FL paper that FedAvg can work with certain non-IID data.

Optimizations and modifications of FedAvg

FedProx, 2020 [40]

- Addresses:
  - **System heterogeneity** (differences in system environment on devices, devices are dropped in FedAvg when they fail to complete a certain amount of training, e.g. due to a weak computational power or communication capabilities)
  - **Statistical heterogeneity**
- Both heterogeneities have negative effects on convergence. Larger heterogeneity results in worse convergence.
- Adds a **proximal term scaled by μ** to FedAvg objective, which limits the impact of local updates.
- **Tolerates partial work** from all selected devices – allows devices to train for less than specified number of local epochs.
- With IID-data FedAvg outperforms FedProx.
- With non-IID data FedAvg starts to diverge, increasing heterogeneity leads to worse convergence. Using FedProx with $\mu > 0$ improves the convergence.
- μ needs to be tuned.
- implementation: https://github.com/litian96/FedProx

IDA, 2020 [41]

- Addresses:
  - **Statistical heterogeneity**, unbalanced data and skewed data in terms of labels
- IDA changes the weighting in FedAvg based on the **inverse distance of each client's parameters to the average parameters**. This allows to reject or weigh less models that are poisoning, i.e. out-of-distribution.
- Also propose **INTRAC** – weighting scheme that uses clients training accuracy to penalize over-fitted models and encourage under-trained models in the aggregated model.
- IDA has on-par or slightly better performance than FedAvg both with IID and non-IID data.
- IDA is more robust than FedAvg in high non-IID data.
- On clinical dataset HAM10k global accuracy of IDA is on par with FedAvg, but local accuracy of IDA (clients on their own test set) is superior to FedAvg.

FedBGVS, 2021 [42]

- Addresses:
  - **Statistical heterogeneity**
- **Requires a balanced global validation** dataset available on server side.
- FedAvg is refined using the global dataset and Balanced Global Validation Score (BGVS).
- Report that FedBVGS outperforms FedAvg, IDA and FedProx in general.

FedDyn, 2020 [43]

- Addresses:
  - **Communication costs**
- Adds a dynamic regularizer for each device and each federated round.
- The regularizer dynamically modifies the device objective with a **penalty term scaled by α** so that, in the limit, when model parameters converge, they do so to stationary points of the global empirical loss.
- Compare proposed FedDyn to FedAvg, FedProx, SCAFFOLD.

- FedDyn always achieves the target accuracy with fewer rounds than all competing algorithms.
- significantly faster convergence than all compared algorithms in both IID and non-IID data distribution.
- α needs to be tuned.

Bibliography

[1] S. Studer *et al.*, 'Towards CRISP-ML (Q): a machine learning process model with quality assurance methodology', *Mach. Learn. Knowl. Extr.*, vol. 3, no. 2, pp. 392–413, 2021.

[2] C. and T. (European C. Directorate-General for Communications Networks and Grupa ekspertów wysokiego szczebla ds. sztucznej inteligencji, *Ethics guidelines for trustworthy AI*. LU: Publications Office of the European Union, 2019. Accessed: Apr. 29, 2023. [Online]. Available: https://data.europa.eu/doi/10.2759/346720

[3] Kincso Izsak and Apolline Terrier, 'Artificial Intelligence-based software as a medical device', Executive Agency for Small and Medium-sized Enterprises (EASME), European Commission, Jul. 2020. Accessed: Apr. 29, 2023. [Online]. Available: https://ati.ec.europa.eu/reports/product-watch/artificial-intelligence-based-software-medical-device

[4] R. Zhang, Y. Wang, Z. Zhou, Z. Ren, Y. Tong, and K. Xu, 'Data Source Selection in Federated Learning: A Submodular Optimization Approach', in *Database Systems for Advanced Applications: 27th International Conference, DASFAA 2022, Virtual Event, April 11–14, 2022, Proceedings, Part II*, Springer, 2022, pp. 606–614.

[5] A. A. de Hond *et al.*, 'Guidelines and quality criteria for artificial intelligence-based prediction models in healthcare: a scoping review', *NPJ Digit. Med.*, vol. 5, no. 1, p. 2, 2022.

[6] D. Leslie, 'Understanding artificial intelligence ethics and safety', *ArXiv Prepr. ArXiv190605684*, 2019.

[7] T. Weikert *et al.*, 'Machine learning in cardiovascular radiology: ESCR position statement on design requirements, quality assessment, current applications, opportunities, and challenges', *Eur. Radiol.*, vol. 31, pp. 3909–3922, 2021.

[8] US DoD, 'Technology Readiness Levels in the Department of Defense (DoD)', 2010. https://api.army.mil/e2/c/downloads/404585.pdf (accessed Apr. 29, 2023).

[9] European Commission, 'Horizon 2020 - Work Programme 2014-2015 (General Annexes)', *Technology Readiness Levels*, 2014. https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf (accessed Apr. 29, 2023).

[10] L. Wheatcraft, 'Technology Readiness Levels applied to Medical Device Development', *ArgonDigital | Making Technology a Strategic Advantage*, Nov. 30, 2015. https://argondigital.com/blog/product-management/technology-readiness-levels-applied-to-medical-device-development/ (accessed Apr. 29, 2023).

[11] MIT4LS SUB2022, 'Life Sciences Technology Readiness Level (pharma, medical devices, digital health)', 2022. https://meetinitalylifesciences.eu/wp-content/uploads/2022/03/MIT4LS_SUB2022_TRL_life_sciences.pdf (accessed Apr. 29, 2023).

[12] R. R. Seva, A. L. S. Tan, L. M. S. Tejero, and M. L. D. S. Salvacion, 'Multi-dimensional readiness assessment of medical devices', *Theor. Issues Ergon. Sci.*, vol. 24, no. 2, pp. 189–205, Mar. 2023, doi: 10.1080/1463922X.2022.2064934.

[13] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, 'Revisiting deep learning models for tabular data', *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 18932–18943, 2021.

[14] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, 'Deep neural networks and tabular data: A survey', *IEEE Trans. Neural Netw. Learn. Syst.*, 2022.

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

[16] N. Rieke *et al.*, 'The future of digital health with federated learning', *Npj Digit. Med.*, vol. 3, no. 1, Art. no. 1, Sep. 2020, doi: 10.1038/s41746-020-00323-1.

[17] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, 'Communication-efficient learning of deep networks from decentralized data', in *Artificial intelligence and statistics*, PMLR, 2017, pp. 1273–1282.

[18] Y. Wang, L. Lin, and J. Chen, 'Communication-efficient adaptive federated learning', in *International Conference on Machine Learning*, PMLR, 2022, pp. 22802–22838.

[19] J. M. Montgomery and S. Olivella, ‘Tree-Based Models for Political Science Data’, *Am. J. Polit. Sci.*, vol. 62, no. 3, pp. 729–744, 2018.

[20] W. B. Chaabene, M. Flah, and M. L. Nehdi, ‘Machine learning prediction of mechanical properties of concrete: Critical review’, *Constr. Build. Mater.*, vol. 260, p. 119889, 2020.

[21] N. Singh, P. Singh, and M. Gupta, ‘An inclusive survey on machine learning for CRM: a paradigm shift’, *Decision*, vol. 47, no. 4, pp. 447–457, 2020.

[22] P. Doupe, J. Faghmous, and S. Basu, ‘Machine learning for health services researchers’, *Value Health*, vol. 22, no. 7, pp. 808–815, 2019.

[23] M. LeBlanc and J. Crowley, ‘A review of tree-based prognostic models’, *Recent Adv. Clin. Trial Des. Anal.*, pp. 113–124, 1995.

[24] Y. J. Ong, N. Baracaldo, and Y. Zhou, ‘Tree-Based Models for Federated Learning Systems’, in *Federated Learning: A Comprehensive Overview of Methods and Applications*, Springer, 2022, pp. 27–52.

[25] A.-C. Hauschild *et al.*, ‘Federated Random Forests can improve local performance of predictive models for various healthcare applications’, *Bioinformatics*, vol. 38, no. 8, pp. 2278–2286, 2022.

[26] Y. J. Ong, Y. Zhou, N. B. Angel, and H. Ludwig, ‘Adaptive Histogram-based Gradient Boosted Trees For Federated Learning’, in *INFORMS Annual Meeting*, 2020.

[27] K. Cheng *et al.*, ‘Secureboost: A lossless federated learning framework’, *IEEE Intell. Syst.*, vol. 36, no. 6, pp. 87–98, 2021.

[28] L. Zhao *et al.*, ‘Inprivate digging: Enabling tree-based distributed data mining with differential privacy’, in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, IEEE, 2018, pp. 2087–2095.

[29] S. Wassan *et al.*, ‘Gradient Boosting for Health IoT Federated Learning’, *Sustainability*, vol. 14, no. 24, p. 16842, 2022.

[30] L. Grinsztajn, E. Oyallon, and G. Varoquaux, ‘Why do tree-based models still outperform deep learning on tabular data?’, *ArXiv Prepr. ArXiv220708815*, 2022.

[31] F. A. Wollenweber *et al.*, ‘Functional outcome following stroke thrombectomy in clinical practice’, *Stroke*, vol. 50, no. 9, pp. 2500–2506, 2019.

[32] F. Quandt *et al.*, ‘Machine Learning–Based Identification of Target Groups for Thrombectomy in Acute Stroke’, *Transl. Stroke Res.*, Jun. 2022, doi: 10.1007/s12975-022-01040-5.

[33] O. A. Berkhemer *et al.*, ‘A Randomized Trial of Intraarterial Treatment for Acute Ischemic Stroke’, *N. Engl. J. Med.*, vol. 372, no. 1, pp. 11–20, Jan. 2015, doi: 10.1056/NEJMoa1411587.

[34] M. A. Mutke *et al.*, ‘Comparing Poor and Favorable Outcome Prediction With Machine Learning After Mechanical Thrombectomy in Acute Ischemic Stroke’, *Front. Neurol.*, vol. 13, 2022, Accessed: Apr. 29, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fneur.2022.737667

[35] J. R. Quinlan, ‘Induction of Decision Trees’, *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986, doi: 10.1023/A:1022643204877.

[36] L. Breiman, J. Friedman, R. Olshen, and C. Stone, ‘Cart’, *Classif. Regres. Trees*, 1984.

[37] L. Breiman, ‘Random forests’, *Mach. Learn.*, vol. 45, pp. 5–32, 2001.

[38] J. H. Friedman, ‘Greedy function approximation: a gradient boosting machine’, *Ann. Stat.*, pp. 1189–1232, 2001.

[39] J. D. Kelleher, B. Mac Namee, and A. D’arcy, *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT press, 2020.

[40] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, ‘Federated optimization in heterogeneous networks’, *Proc. Mach. Learn. Syst.*, vol. 2, pp. 429–450, 2020.

[41] Y. Yeganeh, A. Farshad, N. Navab, and S. Albarqouni, ‘Inverse distance aggregation for federated learning with non-iid data’, 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 2*, Springer, 2020, pp. 150–159.

[42] Y. Mou, J. Geng, S. Welten, C. Rong, S. Decker, and O. Beyan, ‘Optimized Federated Learning on Class-Biased Distributed Data Sources’, in *Machine Learning and Principles and Practice of Knowledge Discovery in Databases: International Workshops of ECML PKDD 2021, Virtual Event, September 13-17, 2021, Proceedings, Part I*, Springer, 2022, pp. 146–158.

[43] A. E. Durmus, Z. Yue, M. Ramon, M. Matthew, W. Paul, and S. Venkatesh, ‘Federated Learning Based on Dynamic Regularization’, in *International Conference on Learning Representations*, 2021.