A Pathway for the Practical Adoption of Federated Machine Learning Projects

Completed Research Paper

Tobias Müller
Technical University of Munich
and SAP SE
Munich, Germany
tobias1.mueller@tum.de

Milena Zahn
Technical University of Munich
and SAP SE
Munich, Germany
milena.zahn@tum.de

Florian Matthes
Technical University of Munich
Munich, Germany
matthes@tum.de

Abstract

Big data forms the fundamental basis for the success of Machine Learning. Yet, a large amount of the world's digitized data is locked up in data silos, leaving its potential untapped. Federated Machine Learning is a novel Machine Learning paradigm with the potential to overcome data silos by enabling the decentralized training of Machine Learning models through a model-to-data approach. Despite its potential advantages, most Federated Machine Learning projects fail to actualize due to their decentralized structure and incomprehensive interrelations. Current literature lacks clear guidelines on which steps need to be performed to successfully implement Federated Machine Learning projects. This study aims to close this research gap. Through a design science research approach, we provide three distinct activity models which outline required tasks in the development of Federated Machine Learning systems. Thereby, we aim to reduce complexity and ease the implementation process by guiding practitioners through the project life cycle.

Keywords: Federated Machine Learning, Activity Model, Software Engineering, AI.

Introduction

The success of Machine Learning (ML) systems roots in the emergence of big data and the ever-increasing availability and wealth of digitized information. Even though data forms the fundamental basis for powering ML systems, it also poses ML's major bottleneck. Problem domains become increasingly complex, which results in more sophisticated and therefore data-demanding ML systems. The development of such advanced, intelligent systems is often restricted through a lack of sufficient training data. Especially small and medium-sized businesses (SMEs) experience this problem and suffer from a deficiency of training data (Bauer et al., 2020). Although a considerable amount of data is freely available, vast amounts of the world's data is scattered in decentralized IoT devices and data silos. The siloed data is usually hardly accessible to external prospective parties, which leaves a significant portion of generated data largely untapped. By breaking up these data silos through collaboration and data sharing, SMEs could overcome the persisting problem of data scarcity. Thereby enabling the development of complex ML systems which were not within the realms of possibility before. However, companies are reluctant to share data due to privacy concerns and a potential loss of intellectual property (IP) (Schomakers et al., 2020). Moreover, the constrained usability of these data silos is additionally strengthened by data protection laws and regulations. Important legal regulations such as the General Data Protection Regulation (GDPR), California Consumer Privacy Act,

Cyber Security Law and the General Principles of the Civil Law justifiably aim to protect the privacy of individuals, but also lead to more data silos.

Motivated by this problem McMahan et al. (2016) introduced Federated Machine Learning (FedML), a novel ML approach which enables the training of a joint ML model on distributed datasets without the need for direct sharing training data. Through a model-to-data approach, the ML model is brought to the decentralized data, therefore data never needs to leave the individual's device which enhances privacy by design. By that, companies would be able to collaboratively train a joint ML model without the risk of losing their individual IP and the need of disclosing any data. Hence, FedML technically yields the potential of alleviating the lack of sufficient training data by enabling ML training across company borders and decentralized data silos.

However, despite its advantages, most FedML projects fail to actualize and never leave the prototype or simulation stage (Lo, Lu, Wang, et al., 2022). The reason for the lack of production ready FedML systems may be attributable to multiple aspects. Even integrating centralized ML algorithms into traditional software systems is cumbersome due to the non-deterministic behavior of ML systems, which collides with deterministic software engineering practices (Giray, 2021). Additionally, the decentralized nature of FedML introduces a further dimension of complexity. Among other things, a collaboration may need to be established and multiple parties need to be coordinated throughout the whole project life cycle (Wouters et al., 2017). Through focus group discussions and expert interviews, we recognized that the realization of FedML projects is currently hindered by missing clarity over the multi-faceted process flow. The currently incomprehensible project structure especially impedes product owners and project managers in the project planning and communication with participants. This challenge could be alleviated through a comprehensive step-by-step guide that clearly outlines the needed tasks of implementing an entire FedML project. Current academic and non-academic literature does not address this issue. We aim to close this research gap by providing activity models which describe the sequence of activities needed to successfully implement FedML applications. By this, we aim to provide clarity and guidance for practitioners throughout the project life cycle and thereby aid to facilitate the development of FedML projects. Summarized, we aim to achieve this goal by answering the following research questions (RQs):

RQ 1: What are the required activities for the implementation of Federated Machine Learning projects?

RQ 2: How can a structured model for guiding practitioners through the implementation of Federated Machine Learning projects be designed?

Federated Machine Learning

FedML is a novel, disruptive ML technique that enables the training of a joint ML model on distributed datasets without the need of sharing data. In traditional ML settings, data is usually accumulated in a central location, where the ML model is subsequently trained. Hence, data owners need to share their data with a central server, therefore potentially risk losing their data sovereignty and IP. Introduced by McMahan et al. (2016), FedML leverages a model-to-data approach and thereby alleviated the need of sharing datasets. As visualized in Figure 1, the FedML process can be divided into four steps:

1. The server chooses an initial global model, which is suitable for the use case and underlying data structure. This global model can be initially trained on an initial dataset.
2. The server distributes the global model amongst all clients.
3. Each client trains the global model on its own local dataset and stores the update gradients. After the training process, each client has its individually trained ML model based on its local dataset.
4. The clients send the individually computed update gradients back to the server. These gradients are aggregated based on a pre-defined protocol and used to update the global model.

Steps 2-4 can be repeated over several iterations until a certain accuracy level is reached or until the accuracy converges. A multitude of different FedML architecture have been proposed, however, in this work we solely refer to server-client architectures consisting of a central orchestration server and multiple clients, since this architectural pattern is the most widely used (Lo, Lu, Zhu, et al., 2022).

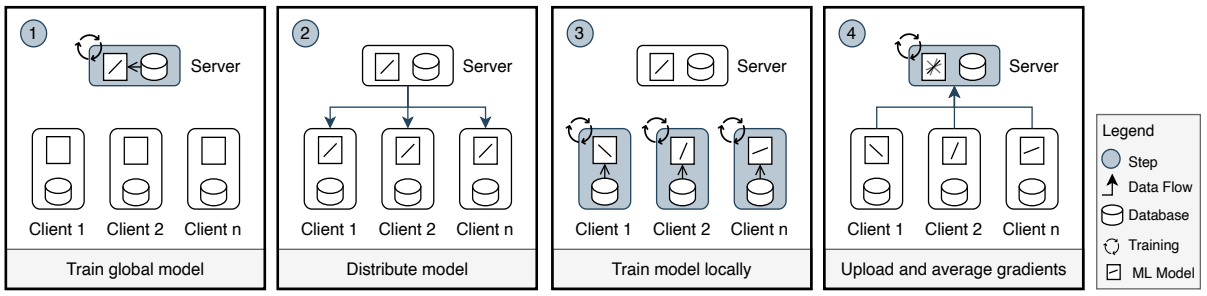

Figure 1. One Iteration of the Federated Machine Learning Process (own work)

Related Work

IEEE published a reference architecture with generalized information about the structure of a FedML model training process and high-level descriptions of activities related to defined roles (IEEE, 2021). The reference can be used as a basis for implementation, but it neither provides information about the life cycle nor detailed insights into the activity descriptions and activity sequences. Considering all life cycle stages is crucial and an existing open problem for the implementation of industrial-level FedML systems (Zhang et al., 2020). Also, it is needed to illustrate the chronological sequence as well as the interrelations of the required activities in order to successfully guide practitioners in the development. Opposed to the existing literature on FedML, detailed activity descriptions were proposed in numerous studies on life cycle models of traditional, centralized ML. For example, Studer et al. (2021) introduced CRISP-ML(Q), an ML process model which covers the full life cycle from business understanding to monitoring and maintenance. The authors followed the principles of the Cross-Industry Standard Process Model for Data Mining (CRISP-DM) (Wirth & Hipp, 2000) and constructed their ML process model with a focus on quality assurance. The development activities were organized in six phases and they described the needed activities briefly. Kreuzberger et al. (2022) developed a workflow for ML operations (MLOps) frameworks including functional components and roles. They included activity descriptions and activity sequence as part of their proposed model. Similarly, Kumara et al. (2022) proposed a reference architecture for MLOps to aid in streamlining the life cycle of ML models in production.

Work on software engineering for ML yields important insights into ML life cycles with their corresponding sequence of activities. Amershi et al. (2019) investigated how software teams at Microsoft developed AI applications. Based on their observations, they analyzed and grouped activities which are performed by the teams to construct a life cycle model with a set of best practices. The study describes the activities and illustrates a workflow of the nine stages. Besides, research on AI governance looks into the activities throughout the ML life cycle. Laato et al. (2022) conducted expert interviews to explore the incorporation of AI governance into system development life cycle models. Their study resulted in a set of governance concepts and descriptions of how AI governance stages tie into existing software development life cycles. Even though, the authors did not describe any needed activities per se, the study provides valuable information for the construction of the activity diagrams through their insights on existing best practices during the different stages of an ML project. Additionally, based on that, Ritz et al. (2022) present a process model to illustrate the dependencies and interactions within the different life cycle stages and activities. They described the activities throughout the whole life cycle including the interdependencies throughout the software development life cycle. These existing life cycle models for centralized ML systems can be used as a basis for FedML projects but need to be revisited due to the decentralized nature of FedML.

Some well-defined activity models for traditional, centralized ML models have been proposed for a multitude of application domains. Koshtura et al. (2020) proposed an UML diagram, which illustrates the set of required activities for the implementation of ML-based demand analysis systems to forecast the bicycle use in smart cities. Thanachawengsakul et al. (2019) constructed an activity diagram, which describes implementation of a knowledge repository management system architecture which uses ML, whereas Venugeetha et al. (2022) illustrated an activity diagram for ML-based breast cancer prediction. Again, these models are heavily tailored to their specific use cases and are intended for centralized ML

systems. Since the decentralized FedML process introduces additional layers of complexity, it is needed to revisit activity models and address FedML specifics for a successful industrial-level implementation.

Research Approach

We observed that many FedML projects do not evolve past prototypes. To understand why FedML projects fail to actualize, we conducted a focus group discussion with three project teams that attempted realizing FedML projects in the past. Through this focus group discussion, we aimed to understand the encountered challenges during the project. We noticed that the main barrier lies in the project initiation and planning phase. The project teams reported that it is challenging to structure the complex implementation process due to its decentralized and collaborative nature. The project flow is complicated to understand, and the division of tasks is not clear. Therefore, communicating the tasks throughout the team and potential collaborators is arduous. This barrier of intricated process structure and difficult communication seem to pose the main challenge in building FedML products. The focus group agreed that a process model, which gives a holistic overview of the process and a detailed activity model with a comprehensive sequence of tasks would alleviate the barrier. Both artifacts would help provide transparency and guidance throughout the development of FedML projects.

At this point, is important to emphasize the differences between the activity model and the process model in terms of their target group and intended purpose. The process model aims to provide a highly abstracted overview of the holistic project structure including required resources, role distributions, and resulting artifacts. This high-level overview intends to aid business stakeholder and solution architects in the project initiation phase outline the general project and facilitate communication with potential participants. The activity model, on the other hand, provides a detailed activity sequence through the needed steps of implementing a FedML project. The comprehensive activity descriptions aim to provide guidance for the implementation and therefore intends to support product owner and project manager in the planning phase of the development process. Therefore, while the process model aims to facilitate the project initiation phase for business stakeholder and solution architects, the activity model intends to ease the planning phase for product owner and project manager. In this paper, we will focus on the comprehensive activity model. We plan to communicate the process model[1] in a separate publication since both artifacts are standalone and need to be considered independently for their respective purpose.

Current literature does not offer a structure representation of a FedML project flow and does not provide guidance on the needed sequence of activities required to realize a FedML project in practice. Therefore, we aim to close this research gap by providing a detailed activity model which illustrates the tasks and interrelations of each life cycle stage in a FedML project. To develop such an activity model, we leveraged the design science research (DSR) methodology as proposed by Peffers et al. (2007) since it provides a methodical, rigorous approach for producing and evaluating innovative, purposeful artifacts for a specific problem domain (Hevner et al., 2004). Table 1 provides an overview of our research approach and short description of the conducted activities during the DSR cycle. The remainder of the paper is structured according to the steps of the DSR approach.

Step	**Short Description of Activities**
(1) Problem identification and motivation	Identified the problem and motivation through focus group discussions. See description above.
(2) Objectives of a solution	Conducted focus group discussions to derive requirements and determine relevant design principles. See chapter on Objectives of a Solution
(3) Design and development	Designed and developed the artifact to provide transparency and guide practitioners through a successful FedML project implementation. See chapter on Design and Development

[1] Process Model: https://bit.ly/3IHIrAn

(4) Demonstration	Demonstrated the artifacts in group discussion and during project ideation phase of an industrial lighthouse project. See chapter on Demonstration and Evaluation
(5) Evaluation	Evaluated the comprehensibility, completeness, usability, and value of the artifacts. See chapter on Demonstration and Evaluation.
(6) Communication	Communication is being done through this paper.

Table 1. Design Science Research Steps According to Peffers et al. (2007)

Results

Objectives of a Solution

The objectives and requirements of the solution were identified through the initial focus group discussion as described in the section on the research approach. The three project teams agreed that the activity model should aid specifically in the planning phase and provide a clear structure over the sequence of activities which are needed to successfully implement FedML projects. Therefore, the activity models should cover the entire end-to-end life cycle of a FedML project. Additionally, the model should help in the communication with non-technical stakeholders and practitioners. Hence, the model should be easily consumable and provide transparency as well as guidance throughout the whole life cycle. The contents should be closely aligned and consistent with best practices of ML life cycles, software development life cycles, and state of the art FedML practices. Lastly, the activity model should be usable for independent of the specific FedML strategy and applicable for all FedML processes with a centralized orchestrator and multiple, distributed clients. The activity model should be applicable independent of the underlying use case, application domain, data structure, and business requirements. The requirements to the artifacts are summarized and indexed in Table 2.

ID	**Short Description of the Requirement**
R1	Aligned with best practices from ML and software development
R2	Comprise an end-to-end project life cycle
R3	Generically applicable and independent of the use case
R4	Understandable and usable by non-technical stakeholders and practitioners
R5	Provide transparency and structure of the entire project
R6	Provide guidance for the implementation process

Table 2. Identified Requirements on the Activity Models

Design and Development

To build our knowledge base to meet requirements R1 and R2, we reviewed current literature on ML life cycles (Amershi et al. 2019; Kreuzberger et al. 2022; Kumara et al. 2022; Laato et al. 2022; Ritz et al. 2022), software development life cycles (Akinsola et al., 2020; Alsaqqa et al., 2020; Apoorva & Deepty, 2013; Gurung et al., 2020) and assessed which practices, procedures, and information are applicable for the activity model and its structure. Additionally, we reviewed the state-of-the-art FedML processes through current literature on FedML architectures and algorithms (Bharti et al., 2022; Bonawitz et al., 2019; IEEE, 2021; Lo et al., 2021; Lo, Lu, Zhu, et al., 2022; Zhang et al., 2020). The activity models are partially based on an expert interview study on the socio-technical challenges of FedML projects, which we describe in a separate publication (Mueller et al., 2023). The study especially provided information for strategic activities in a collaborative setting. The relevant findings from the interview study, focus group discussions and

literature review were incrementally combined. During development, we conducted regular mini focus group discussions with varying participants to iteratively assess the activity models and implement feedback.

Per definition, an *activity* represents units of work that are performed by roles. Each activity has a clear purpose and usually results in the creation or update of an artifact (Anwar, 2014). Therefore, the activity models should provide guidance on which units of work need to be followed such that the goals (or artifacts) of the stages can be successfully achieved. However, some activities and their corresponding method of execution may be dependent on technicalities, use case, or business requirements. To ensure requirement R3, we want to leave the choice of the fitting method to the practitioner. Hence, we chose the granularity, such that the goal and required activities is comprehensible, but does not preset specific context-dependent methods. We only included the happy path without exit points and did not incorporate the dependencies and interactions between the stages to enhance comprehensibility. The activity models show one iteration of each stage. It is important to highlight that in ML projects, the stages are usually performed iteratively to refine the process.

The design of the activity models is based on the UML 2.0 notation[2] since it is a commonly known and normed unified modelling language in the field of software engineering. By leveraging widely known notations, we aim to meet requirement R4. As specified in UML 2.0, every diagram has at least one initial node, which is indicated by a filled circle, and one end point, which is represented by an encircled filled circle. Activity states are illustrated through ellipses, described through verb-object activity labels, and connected with arrows to represent in which order the activities happen. Decision points are modelled through diamonds, and bars represent the start (split) or end (join) of concurrent activities. If multiple actors interact within one activity model, horizontal swimlanes group the activities performed by the same actor. Objects only show critical inputs/outputs and are modelled through rectangles. If similar objects appear multiple times, the life cycle stage is described as state name in rectangular brackets. Shared objects are placed on swimlane separators.

To meet requirement R2 and R5, we structure the activity models according to the life cycle stages of a FedML project, whereas each stage represents a set of interrelated tasks and activities which serve a clear purpose. Figure 2 provides an overview of the different life cycle stages. The first three stages comprise strategic activities, starting with the *project initiation (1)*, followed by the *project validation (2)* and *project setup (3)* phase. We clustered these three stages into a single activity diagram since these phases are interconnected and constitute the preparation for the implementation. The resulting activity diagram is illustrated in Figure 3. Thereafter, all development activities are grouped in the *System Design and Development* stage, which can be divided into the *Global Model Design (4)*, *Local Model Design (5)*, and *Global Model Aggregation (6)* stages. Stage 4 builds the technical basis for the FedML training process, and its corresponding diagram is depicted in Figure 4. The actual FedML training process is conducted in Stage 5 and 6. We clustered both stages into a single activity diagram since these two phases collectively constitute the FedML training process. The activity diagram is illustrated in Figure 5. Finally, the *Deployment and Maintenance (7)* stage serves and maintains the ML product and is shown in Figure 6. The figures display a single iteration of each stage. In practice the stages 4 to 7 are iteratively and continuously performed due to the iterative ML process. The following activity descriptions are structured according to the life cycle stages and describe the accompanying activities in relation to the activity diagrams. The descriptions should help to meet requirement R6 and simultaneously answer RQ1 and RQ2.

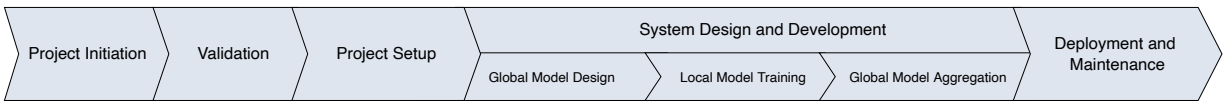

Figure 2. Stages of a Federated Machine Learning Project

[2]http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/

Stage 1: Project Initiation

The first stage represents the initial step in starting a project and lays the foundation of the project by broadly defining the goal of the project and its corresponding high-level requirements. As seen in Figure 3, this stage contains one main activity (*Initiate Project*) with two sub-activities. The first sub-activity defines the project goals and scope. Depending on the company-specific best practices, the goal definition usually requires the specification of a business model, the corresponding business requirements, and the illustration of use case diagrams to demonstrate the business case. The second sub-activity translates the resulting business problem into an ML problem to define the technical descriptions and deployment strategy.

Stage 2: Project Validation

The second stage aims to investigate the feasibility of the project and includes one main activity (*Check Project Feasibility*) with two concurrent sub-activities (see Figure 3). More specifically, the result of the previous stage is taken as input for a simultaneous business case analysis and technical assessment of the project's feasibility. These assessments determine whether the project's goals and scope are technically executable, deliver value as intended, and which steps should be taken to meet them including the overall FedML strategy. Additionally, it is determined whether collaboration partners are needed to successfully implement the project.

Stage 3: Project Setup

The third stage intends to set up the project by arranging a potential collaboration and setting up the technical foundation for the implementation of the project. Hence, the activities of this stage depend on the decision if collaboration partners are wanted for the project. In a collaborative setup, two additional main activities (*Establish Collaboration* and *Plan Collaboration)* need to be performed, before the technical foundation can be constructed. This stage is depicted in Figure 3.

(3.1) Establish Collaboration will create the collaboration and consists of three sub-activities. First, potential collaboration partners need to be identified. Then, it needs to be evaluated if the potential participants would be interested in joining the project. Lastly, the collaboration feasibility needs to be checked, to assess whether the project can be implemented jointly. This step needs to determine if all sides agree on the same objectives and assess on a high level if all participants could deliver the expected value, resources, or expertise. If the collaboration structure cannot perform as intended, all three steps need to be repeated until the collaboration can be established or until a termination decision is made.

(3.2) Plan Collaboration will plan the joint project and comprises a total of five sub-activities. To begin the collaboration planning, the overall partnership strategy needs to be described to specify the common goal and underlying partnership structure. Subsequently, the collaboration management, co-creation management, and co-creation practices are defined. The aspects which need to be covered in this step are multi-faceted and comprehensive. We describe the specifics which need to be covered in the collaboration management, co-creation management, and co-creation practices in a separate publication (Mueller et al., 2023). Thereafter, legal compliance needs to be checked. If the compliance check fails, a refinement of the collaboration planning activities is necessary, or termination decision could be made. If all activities were successfully executed, an agreement on the collaboration specifications emerges.

If a collaboration is needed and comes to fruition, all collaboration channels which are needed for the orchestration, communication, and collaboration management are created. Lastly, the required technical infrastructure needs to be set up in all cases. This activity lays the technical basis for the implementation by creating the data processing, model training, and orchestration infrastructure.

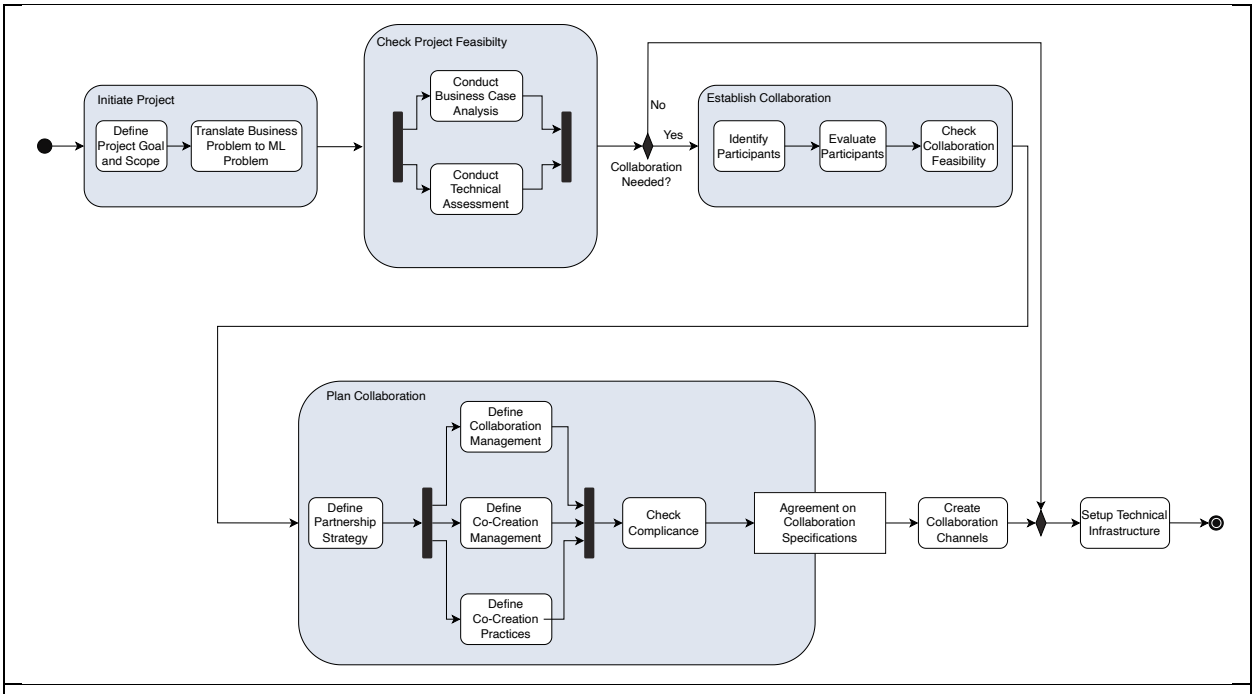

Figure 3. One Iteration of the Project Initiation, Validation, and Setup Stage

Stage 4: Global Model Design

The fourth stage yields the data specifications and source code of the initial ML model which will be the technical foundation of the FedML training process. The activities comprise an ML experimentation task as well as data preparation specifications and can be divided into three distinct main activities. As illustrated in Figure 4, this stage starts with the *Prepare Data* activity and is followed concurrently by the *Design Global Model* and *Define Data Specifications* activities.

(4.1) Prepare Data represents the data preparation step of traditional ML pipelines and consists of four sub-activities. It requires access to a data storage which contains a representative dataset for the targeted ML problem. The data is fetched from the data storage and subsequently analyzed on the underlying data distribution, data quality, and overall suitability for the business requirements. Thereafter, the data is preprocessed such that high data quality is ensured and the ML model can interpret the data's features. This task involves data cleaning, data transformation, data reduction, and feature engineering. The final data validation activity ensures the correctness, usefulness, and sufficient quality of the data.

(4.2) Design Global Model crafts the initial source code for the initial model which will be used in the FedML training process and consists of four sub-activities. An initial proposal of an ML architecture is trained on the data from step 4.1. The resulting ML model is evaluated on its performance based on the evaluation data and validated on the validation data. Based on the evaluation and validation results, it is determined whether the accuracy suffices for an initial distribution to the local clients for training. Since the model is further trained in the FedML process, this step can be seen as a further feasibility check and should validate the suitability of the ML model for the use case. If the criteria are not met, then the model is analyzed, and another design iteration is triggered. If the criteria are met, the initial ML model is stored in a model registry which is accessible to all participants.

(4.3) Define Data Specifications yields the data preparation specifications which are used by every local data contributor (or client) as a data preparation guideline to ensure homogeneous, suitable training data for the FedML process. Therefore, the data cleaning, data transformation, and data reduction rules must be defined and documented. Additionally, the feature engineering rules are defined. The resulting specification document is then communicated with each client.

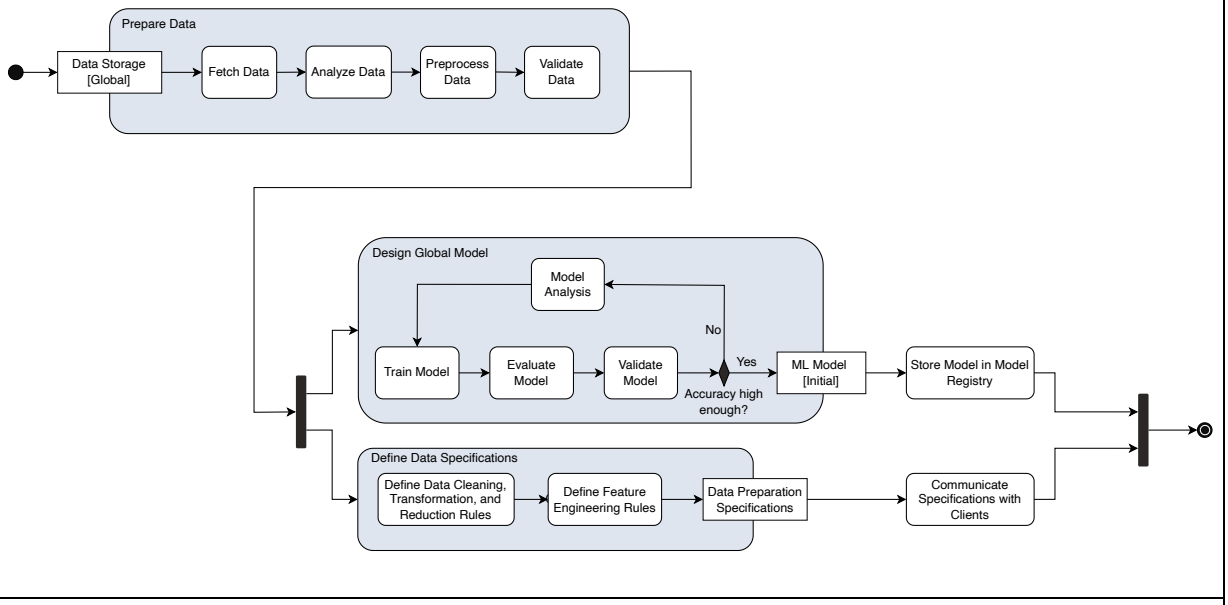

Figure 4. One Iteration of the Global Model Design Stage

Stage 5: Local Model Training

The fifth stage delineates the local training on the data contributor's (clients) side. As indicated by the swimlanes (*client 1, …, client n*) in Figure 5, every data contributor performs the same activities simultaneously, which consists of one main activity and a total of five sub-activities. Each data contributor requires access to the model registry and their individual data storage. The different clients fetch the latest model from the model registry, and concurrently prepare the local data. To prepare the local data, it is needed to first fetch the data from their data storage. Thereafter, the data is preprocessed as outlined in the data preparation specifications (see Stage 4). The data preparation specifications ensure homogeneous datasets across all clients. Naturally, in settings that allow heterogeneous data, this step can be neglected. Once preprocessed, the data is then validated on the correctness, usefulness, and sufficient quality of the data. Finally, the latest ML model is trained on the prepared data, resulting in update gradients, which are shared with the aggregator.

Stage 6: Global Model Aggregation

The sixth stage details the aggregation process of the FedML training and is structured into one main activity with six sub-activities (see Figure 5). Once enough update gradients from the local data contributors are collected, the global model can be built and evaluated. The required number of update gradients depends heavily on the use case, business requirements, or scale of the project. The aggregator selects a set of the received upgrade gradients, which are used to train the model. The gradient selection process can either follow a pre-defined sub-sampling scheme or be neglected and simply use every received gradient. This is dependent on the pre-defined FedML strategy. The selected gradients are subsequently aggregated and fused with the latest global ML model. The updated global ML model is then stored in the model registry for traceability and to make the model accessible for the involved clients. Then, the model is validated. Based on the validation results, it is determined whether the model needs to be further trained by triggering a new training iteration or if it can be packaged for deployment.

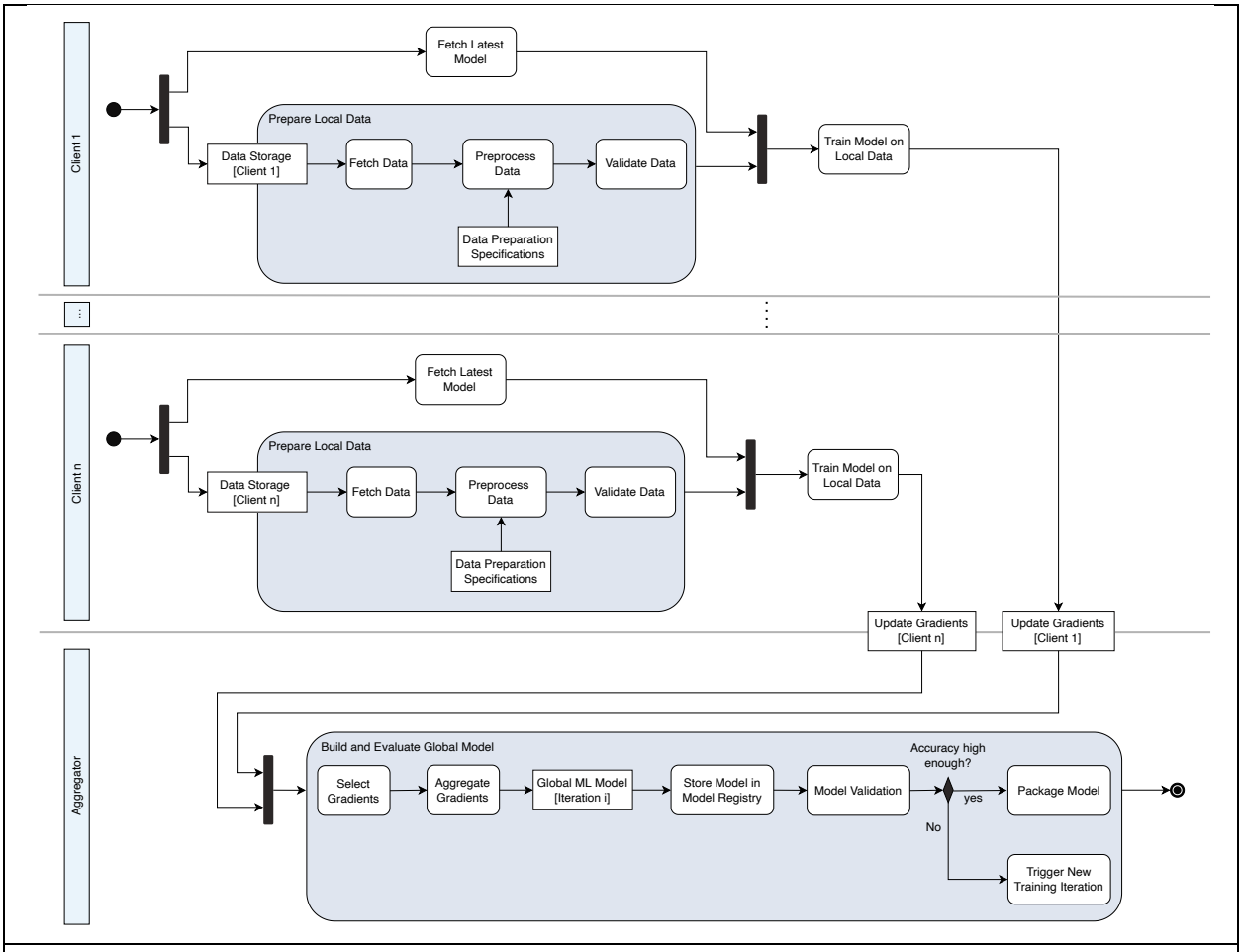

Figure 5. Activities of the Local Model Training and Global Model Aggregation Stage

Stage 7: Deployment and Maintenance

The seventh stage deploys the packaged ML model into production and yields the model service. As illustrated in Figure 6, this stage consists of seven distinct activities. First, the latest ML model is fetched from the model registry and build for deployment. Then, integration testing is performed to ensure that the built code will work as intended. If the tests are successful, the built model will be deployed and served. Again, the specific methods for building and integration testing are dependent on the use case, business requirements, or simply on the developers' preferences. Once the model is served, it is continuously maintained and monitored on its performance, potential deviations from the desired behavior, and pre-defined business key performance indicators. If the model does not perform as desired for example due to concept drift, covariate shift, or simply if an updated ML model version is available, the currently deployed model needs to be sunsetted.

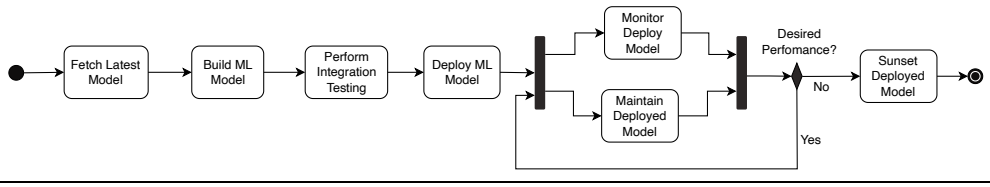

Figure 6. One Iteration of the Deployment and Maintenance Stage

Demonstration and Evaluation

To demonstrate and evaluate our activity model, we performed two iterations of demonstrations with survey-based evaluations. After each demonstration, we received feedback through an expert discussion and surveys, which were additionally filled out by each expert. The survey evaluates if the model is deemed valuable, detailed enough and additionally gathers feedback on the overall usefulness, comprehensibility, and completeness of the activity model. The survey results are shown in Figure 7. We incorporated the feedback after each round of demonstration and evaluation. In total, 14 experts from eight organizations participated in the assessments and provided valuable recommendations. An anonymized list of participants including their current position, their organization, and years of experience in their current position is shown in Table 4.

ID	Position	Organization	Experience
E1	Applied Researcher	Emerging Tech Start-up	> 2 years
E2	Applied Researcher	Industrial Software Enterprise	> 2 years
E3	Applied Researcher	Research Center for AI Security	> 1 year
E4	Research Engineer	Large Engineering Company	> 5 years
E5	ML Engineer and Senior Data Scientist	Large Software Enterprise	> 7 years
E6	Senior Researcher	Large Software Enterprise	> 7 years
E7	Senior Product Manager	Large Software Enterprise	> 6 years
E8	Product Manager	Large Software Enterprise	> 10 years
E9	Senior Consultant and Project Lead	Large Software Enterprise	> 6 years
E10	Solution Advisor	Large Software Enterprise	> 2 years
E11	Development Expert	Large Software Enterprise	> 19 years
E12	Consultant for Emerging Tech	Medium-sized Consultant Company	> 2 years
E13	Solution Architect	Large Software Enterprise	> 5 years
E14	Project Manager	Research Center for AI Security	> 1 year

Table 4. Overview of Evaluation Participants.

In the **first demonstration**, we introduced the activity model to a total of 6 experts (E1-E6). The main focus of this iteration was to evaluate the technical completeness, comprehensibility, and level of detail. Against this backdrop, we invited a diverse group of experts with technical backgrounds. The group comprised three experts on FedML and three applied researchers with expertise in related, emerging technologies but without extensive knowledge on FedML. Therefore, the iteration can be considered as an expert evaluation (Peffers et al., 2012). All experts validated the research problem and confirmed the completeness, comprehensibility, and level of detail. Additionally, the expert round experienced the activity model as a useful and valuable resource for their activities and communications with stakeholders. We received two recommendations to improve the activity model. One expert (E6) mentioned that large-scale FedML projects include an additional gradient selection step, which sub-samples the received gradients. We incorporated the feedback by adding a *Select Gradients* activity as the first task of the *Global Model Aggregation* stage. Another expert (E2) proposed that an explanation of the different steps in the *Plan Collaboration* and *Establish Collaboration* phase would be helpful. We did not adapt the activity model since these activities are dependent on the use case and business requirements.

Overall, the activity model has been received very positively. The participants stated that they will use the activity model for their future activities and discussions. More specifically, the experts remarked that the activity model will facilitate their communication with stakeholders since *"this is very much what decision-*

makers would require from practitioners in order to understand what needs to be done […]" (E2). Additionally, expert E4 stated that the model *"[…] helps [him] to structure a project for privacy-preserving FedML because [he] would know where to apply privacy-enhancing technologies"* (E4). Expert E6 validated the usefulness and wished for a similar activity model for other emerging technologies. Summarized, the models aid practitioners in communication with their stakeholders, provide a structure for their projects and thereby facilitate the development of their project. The experts expressed that the models are a *"very good resource for FedML projects"* (E3).

In the **second demonstration**, we presented the refined activity model to a group of 8 potential users and key stakeholders (E7-E14). The demonstration was part of a project ideation process to adopt FedML within a large industrial lighthouse project. The main focus of this iteration round was to evaluate the models' usefulness, comprehensibility, and value to potential users. The group of participants included a variety of potential users such as project managers, product owners, solution architects, and solution advisors. Each participant was involved in prior FedML use case discussions and therefore well-suited for this evaluation round. The group of participants reported that the activity model is intuitive to follow, helps to structure the overall process flow, and provides guidance for project implementation. Based on the feedback, we added *Maintain Deployed Model* and *Sunset Deployed* Model tasks in the *Deployment and Maintenance* stage. Besides, the activity model was considered a comprehensible, complete, and comprehensive artifact. The group further agreed that the model is useful and provides value in future activities and discussions.

Overall, the target users from the evaluation group validated the relevancy and effectiveness of the models. Expert E12 specifically emphasized that "[…] *the models are intuitive to follow"* (E12) and aid in "[…] *understanding the FedML process"* (E12). The potential users described the models as *"a very useful visual tool"* (E11) that helps in planning FedML projects due to its *"[…] detailed sequence of activities and areas where you can keep focus on"* (E11). This makes the complex FedML process for *"[…] non-technically proficient stakeholders easier to comprehend"* (E14). One target user stated that the models *"[…] help make better-informed decisions as a newbie to FedML"* (E9). According to the evaluation group, the models provide transparency, and aid non-technical stakeholders to structure and gain an understanding of FedML projects.

As illustrated in Figure 7, the evaluation results show that the model was overall well-received by all 14 experts. The model is considered useful, valuable, and detailed enough. The large majority of the participants experienced the model as comprehensible and that it covers all important aspects.

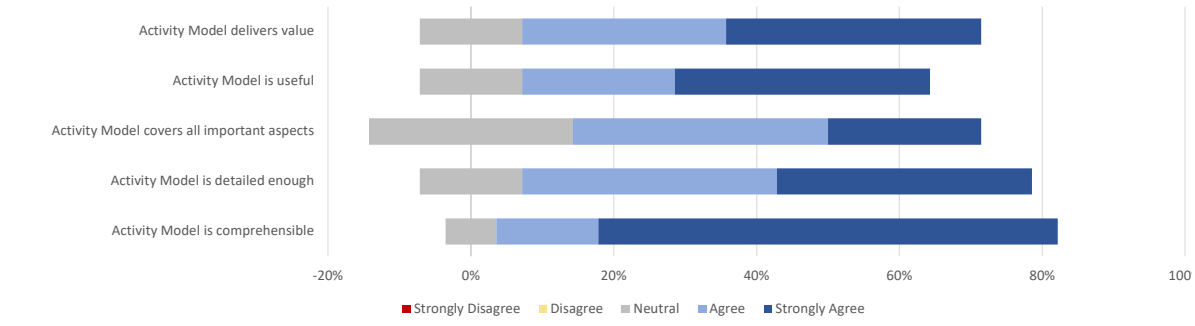

Figure 7. Consolidated Evaluation Results

Discussion

In this study, we designed three distinct activity models which represent the entire life cycle of a FedML project. The problem was identified and motivated through an initial focus group discussion with potential users from three project teams. The relevancy was additionally confirmed in the demonstrations and evaluations. This is also reflected in the survey results since each participant deems the activity models to be useful and valuable (see Figure 7).

Solution Requirements

Through the initial focus group discussion, we were able to formulate a total of six requirements to the activity models (see Table 2). Aiming to fulfill requirement R1, we reviewed and incorporated current literature on ML life cycles, software development life cycles, and state of the art FedML methods. Since technical experts and potential users validated the completeness and usability, we consider that the activity model is aligned with current best practices. Therefore, we consider requirement R1 to be met. The activity models represent the entire life cycle of a FedML project, which was validated on completeness by experts. Hence, requirement R2 should also be fulfilled. Through the two rounds of demonstrations and evaluations, we were able to gather feedback from a large variety of different profiles and backgrounds. The participants ranged from FedML experts and applied researchers to product specialists, projects managers, solution architects, and solution advisors. As reflected in the survey results (see Figure 7), every expert from this diverse set of evaluation participants considers the models valuable, useful, and with a fitting level of detail. Therefore, we consider the activity models as generically applicable and requirement R3 as met. Through the second round of demonstrations, we aimed to evaluate the comprehensibility and usability of potential users and therefore non-technical stakeholders. As reported, the feedback was consistently positive, which leads us to believe, that requirement R4 is fulfilled as well. Combined with the thoroughly positive feedback from the technical experts in the first demonstration, we can assume that the activity models provide transparency and structure to the entire project life cycle (requirement R5). After incorporating the missing aspects suggested by the experts, we consider that the activity models cover the most important aspects. Consequently, the models can be used as guidance for the implementation process and ease the planning process for product owner and project manager, which finally achieves requirement R6. Summarized, we can assume that all objectives have been met to a sufficient extent and that the activity models are useful as well as deliver value to potential users.

Generalizability

The models are currently tailored to server-client architectures with a central orchestrating server and multiple local clients since this is the most widely used architectural pattern (Lo, Lu, Wang et al., 2022),. However, more architectural patterns such as completely decentralized processes were suggested (Lo, Lu, Zhu, et al., 2022), which may not be covered by the proposed activity models of stages 5 and 6. Since the architecture choice only influences activities regarding the *local model training* (stage 5) and *global model aggregation* (stage 6), the models of stages 1-4 and stage 7 remain the same and can be used in each FedML project. Additionally, the activities of *local model training (see Figure 5)* are imminent and only the outgoing communication of stage 5 might change depending on the architectural pattern. For partially connected or fully decentralized architectures, clients might communicate amongst themselves, and the aggregation scheme might differ (Lo, Lu, Zhu, et al., 2022). Therefore, the choice of a different architectural pattern might require revisiting the outgoing communication of stage 5 and the activities of stage 6, whereas the remaining activity models can be reused as proposed in this study.

Throughout this study, we focused on designing generically usable activity models that are not specific to use cases or domains. Hence, we only included activities that are required in every FedML project. Moreover, we designed the models such that the need for optional activities is queried (see Figure 3). Thereby, we aimed for generalizable models. To evaluate the generalizability of the models to multiple areas, domains, and target users, we presented the models to a diverse group of potential users. As illustrated in Table 4, the evaluation group consisted of 14 participants, from eight organizations of different sizes and eight different job profiles. Since the large majority of the participants confirmed the comprehensibility and usefulness (see Figure 7), we consider the model to be generically applicable to a large variety of areas. Even though the models were constructed to be generically applicable, some domain-dependent and process-specific activities might be missing. Therefore, the direct application of the models in varying application domains would further validate the generalizability of the models. We encourage practitioners to test our models in their use cases to further develop the artifacts.

Limitations

It should be noted that the activity models were designed to aid in the communication with non-technical stakeholders, provide transparency, and guidance for the implementation process. Consequently, the

models are not intended to be followed down to the smallest detail since the domain, business requirements, or technology specifics might alter certain activities. In this context, we also only included the happy path without exit points throughout the life cycle. The models should be seen as a reference and basis for implementation. Lastly, experiences gathered from practical applications might help to further enhance the models. Even though we gathered insights and feedback for the artifacts from experts with a large variety of different roles and experiences, we conceivably only depict a subset of the potential user demographic. Therefore, we recommend testing the activity models in more diverse settings to receive broader feedback to further improve and develop the models.

One of the main motivational drivers to use FedML is the provided degree of privacy through its model-to-data approach. However, the send gradients could be reverse engineered which might reveal sensitive information (Jere et al., 2020). Further privacy-enhancing technologies could be implemented on top of FedML to mitigate privacy leakages. Our models do not include activities regarding the implementation of privacy-enhancing techniques since these are case-specific and reduce the generalizability of the proposed models. However, as expert E4 stated, the proposed models still aid in the implementation of further privacy-enhancing technologies. Through the detailed step-by-step description of the activity models, expert E4 recognized entry points where privacy-enhancing technologies need to be implemented.

Conclusion

In this study, we introduced three distinct activity models which together depict the sequence of activities that are needed to implement a FedML project. The activity models extend current literature on life cycle models for centralized ML projects and complement FedML reference architectures through activity sequences throughout the whole FedML project life cycle. Thereby, the models provide transparency and structure the entire life cycle of a FedML project to facilitate the comprehensibility of its complex process. The evaluation results show that potential users deem the models to be useful, deliver value and comprise the most relevant aspects with an appropriate level of detail. We showed that the activity models can help practitioners and non-technical stakeholders to gain an understanding of the structure and especially ease the project planning phase by providing guidance over the project life cycle. However, the models are currently limited to server-client architectures with a central orchestrating server and multiple local clients. Other architectural designs may not be covered by our models. Overall, the activity models aid the successful implementation of FedML projects by easing the planning phase for product owners and project managers. Additional to the sequence of activities, current literature on FedML seems to be lacking clarity over the interrelations between the stages, role distribution, and their dependencies throughout the project. We plan to discuss these aspects through a process model in a separate publication.

Acknowledgements

The authors would like to thank SAP SE for supporting this work.

References

Akinsola, J. E. T., Ogunbanwo, A. S., Okesola, O. J., Odun-Ayo, I. J., Ayegbusi, F. D., & Adebiyi, A. A. (2020). Comparative Analysis of Software Development Life Cycle Models (SDLC). In R. Silhavy (Ed.), *Intelligent Algorithms in Software Engineering* (pp. 310–322). Springer International Publishing.

Alsaqqa, S., Sawalha, S., & Abdel-Nabi, H. (2020). Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies (IJIM)*, *14*(11), 246.

Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., & Zimmermann, T. (2019). Software Engineering for Machine Learning: A Case Study. *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 291–300.

Anwar, A. (2014). A Review of RUP (Rational Unified Process). *International Journal of Software Engineering*, *5*(2), 8–24.

Apoorva, M., & Deepty, D. (2013). A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios. *International Journal of Advance Research in Computer Science and Management Studies*, *1*(5), 64–69.

Bauer, M., van Dinther, C., & Kiefer, D. (2020). Machine Learning in SME: An Empirical Study on Enablers and Success Factors. *AMCIS 2020 Proceedings*, *3*.

Bharti, S., McGibney, A., & O'gorman, T. (2022). Design Considerations and Guidelines for Implementing Federated Learning in Smart Manufacturing Applications. *IIC Journal of Innovation*, *19*, 17–35.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konecný, J., Mazzocchi, S., McMahan, H. B., Overveldt, T. V., Petrou, D., Ramage, D., & Roselander, J. (2019). Towards Federated Learning at Scale: System Design. *Proceedings of Machine Learning and Systems*, *1*, 374–388.

Chandrasekaran, V., Jia, H., Thudi, A., Travers, A., Yaghini, M., & Papernot, N. (2021). SoK: Machine Learning Governance. *ArXiv:2109.10870 [Cs]*.

Giray, G. (2021). A Software Engineering Perspective on Engineering Machine Learning Systems: State of the Art and Challenges. *Journal of Systems and Software*, *180*, 35.

Gurung, G., Shah, R., & Jaiswal, D. (2020). Software Development Life Cycle Models-A Comparative Study. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 30–37.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105.

IEEE. (2021). IEEE Guide for Architectural Framework and Application of Federated Machine Learning. *IEEE Std 3652.1-2020*, 1–69.

Jere, M. S., Farnan, T., & Koushanfar, F. (2021). A Taxonomy of Attacks on Federated Learning. *IEEE Security & Privacy*, *19*(2), 20-28.

Koshtura, D., Bublyk, M., Matseliukh, Y., Dosyn, D., Chyrun, L., & Lozyn, O. (2020). *Analysis of the Demand for Bicycle Use in a Smart City Based on Machine Learning*.

Kreuzberger, D., Kühl, N., & Hirschl, S. (2022). *Machine Learning Operations (MLOps): Overview, Definition, and Architecture* (arXiv:2205.02302). arXiv.

Kumara, I., Arts, R., Di Nucci, D., Heuvel, W. J. V. D., & Tamburri, D. A. (2022). *Requirements and Reference Architecture for MLOps:Insights from Industry*. TechRxiv.

Laato, S., Birkstedt, T., Mäantymäki, M., Minkkinen, M., & Mikkonen, T. (2022). AI governance in the system development life cycle: Insights on responsible machine learning engineering. *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*, 113–123.

Lo, S. K., Lu, Q., Paik, H.-Y., & Zhu, L. (2021). FLRA: A Reference Architecture for Federated Learning Systems. *Software Architecture*, *12857*, 83–98.

Lo, S. K., Lu, Q., Wang, C., Paik, H.-Y., & Zhu, L. (2022). A Systematic Literature Review on Federated Machine Learning: From a Software Engineering Perspective. *ACM Computing Surveys*, *54*(5), 1–39.

Lo, S. K., Lu, Q., Zhu, L., Paik, H., Xu, X., & Wang, C. (2022). Architectural Patterns for the Design of Federated Learning Systems. *Journal of Systems and Software*, *191*(111357).

McMahan, H. B., Moore, E., Ramage, D., & Arcas, B. A. y. (2016). Federated Learning of Deep Networks using Model Averaging. *CoRR*, *abs/1602.05629*.

Mueller, T., Zahn, M., & Matthes, F. (2023). Unlocking the Potential of Collaborative AI - On the Socio-Technical Challenges of Federated Machine Learning. *Proceedings of the 31st European Conference on Information Systems*.

Peffers, K., Rothenberger, M., Tuunanen, T., & Vaezi, R. (2012). Design Science Research Evaluation. In K. Peffers, M. Rothenberger, & B. Kuechler (Eds.), *Design Science Research in Information Systems. Advances in Theory and Practice* (Vol. 7286, pp. 398–410). Springer Berlin Heidelberg.

Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems,* 24*(3),* 45-77.

Ritz, F., Phan, T., Sedlmeier, A., Altmann, P., Wieghardt, J., Schmid, R., Sauer, H., Klein, C., Linnhoff-Popien, C., & Gabor, T. (2022). Capturing Dependencies within Machine Learning via a Formal Process Model. *Proceedings of the 11th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation*, *3*, 249–265.

Schomakers, E.-M., Lidynia, C., & Ziefle, M. (2020). All of me? Users' preferences for privacy-preserving data markets and the importance of anonymity. *Electronic Markets*, *30*(3), 649–665.

Studer, S., Bui, T. B., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S., & Müller, K.-R. (2021). Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. *Machine Learning and Knowledge Extraction*, *3*(2), 392–413.

Thanachawengsakul, N., Wannapiroon, P., & Nilsook, P. (2019). The Knowledge Repository Management System Architecture of Digital Knowledge Engineering using Machine Learning to Promote Software Engineering Competencies. *International Journal of Emerging Technologies in Learning (IJET)*, *14*(12), 42.

Venugeetha, Y., Harshitha, B. M., Charitha, K. P., Shwetha, K., & Keerthana, V. (2022). Breast Cancer Prediction and Trail Using Machine Learning and Image Processing. In A. Kumar, S. Senatore, & V. K. Gunjan (Eds.), *ICDSMLA 2020* (pp. 957–966). Springer. https://doi.org/10.1007/978-981-16-3690-5_89

Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a Standard Process Model for Data Mining. *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, 29–40.

Wouters, L., Creff, S., Bella, E. E., & Koudri, A. (2017). Collaborative systems engineering: Issues & challenges. *2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 486–491.

Zhang, H., Bosch, J., & Olsson, H. H. (2020). Federated Learning Systems: Architecture Alternatives. *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*, 385–394.