A Process Model for the Practical Adoption of Federated Machine Learning

Completed Research Full Paper

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

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

Florian Matthes
Technical University of Munich
matthes@tum.de

Abstract

The wealth of digitized data forms the fundamental basis for the disruptive impact of Machine Learning. Yet a significant amount of data is scattered and locked in data silos, leaving its full potential untouched. Federated Machine Learning is a novel Machine Learning paradigm with the ability to overcome data silos by enabling the training of Machine Learning models on decentralized, potentially siloed data. Despite its advantages, most Federated Machine Learning projects fail in the project initiation phase due to their decentralized structure and incomprehensive interrelations. The current literature lacks a comprehensible overview of the complex project structure. Through a Design Science Research approach, we provide a process model of a Federated Machine Learning life cycle including required activities, roles, resources, artifacts, and interrelations. Thereby, we aim to aid practitioners in the project initiation phase by providing transparency and facilitating comprehensibility over the entire project life cycle.

Keywords

Federated Machine Learning, Process Model, Software Engineering, Applied AI, Design Science Research.

Introduction

The disruptive potential of Machine Learning (ML) roots in the emergence of big data and the ever-increasing wealth of digitized data. Yet, the lack of sufficient training data is still a persevering bottleneck in creating sophisticated, data-demanding ML systems. Even though vast amounts of data is freely available, a considerable amount of the world's data is still scattered, stored and locked up in data silos, therefore hardly accessible. This lack of available, suitable training data creates a competitive disadvantage especially for small and medium-sized enterprises (SMEs) (Bauer et al. 2020) leaving their full economic potential unreached. SMEs could overcome data scarcity by breaking up data silos, sharing data and collaborating. However, the companies' willingness to share data is low due to privacy concerns and a potential loss of intellectual property (IP) (Schomakers et al. 2020).

Federated Machine Learning (FedML) is a novel ML technique which allows the creation of a joint ML model on decentralized and therefore siloed datasets (McMahan et al. 2016). Through this model-to-data approach, companies could collaboratively train an ML model without the need for direct data sharing. Technically, FedML has the potential to enable SMEs overcome data silos, use currently untapped data, and thereby leverage the full potential of ML without privacy leakage. Despite its advantages, there are currently only a few production-level applications and most work on FedML comprises prototypes or simulations (Lo, Lu, Wang, et al. 2022). The missing operationalization of FedML may be attributable to

multiple aspects. For example, the persisting discrepancy between engineering traditional, deterministic software and engineering non-deterministic ML systems makes the integration of ML cumbersome (Giray 2021). Additionally, FedML requires the coordination of multiple parties across different life cycle stages due to its decentralized nature. We recognized through focus group discussions and expert interviews, that FedML is currently missing clarity over its complex and multi-faceted process flow. This missing clarity poses a key challenge for practitioners in the project initiation and communication with potential participants. Current literature usually takes a technical perspective when designing FedML systems and lacks a comprehensible overview of the complex project structure. Against this backdrop, we aim to provide transparency by developing a comprehensible process model of an end-to-end FedML project life cycle. The process model intends to break down, structure and illustrate the different project stages including required tasks, resources, roles, artifacts, and their interrelations. In summary, we aim to answer the following research questions (RQs):

RQ 1: What are the most relevant components and aspects needed for a comprehensible overview of a FedML process flow?

RQ 2: How can a generic, structured process model of an end-to-end Federated Machine Learning project life cycle be designed?

Theoretical Background and Related Work

FedML is a novel, disruptive ML paradigm 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 and risk losing their IP. Introduced by McMahan et al. (2016), FedML counteracts the need of sharing datasets through a model-to-data approach. As visualized in Figure 1, the FedML process can be divided into four steps. First, the server chooses an initial global model which is suitable for the use case and underlying data structure. The global model can be initially trained by the server. Secondly, the server distributes the global model amongst all clients. Thirdly, each client trains the global model on its own local dataset and stores the update gradients. Consequently, each client owns its individually trained ML model based on its local dataset. Finally, the clients send the individually computed update gradients back to the server, which are aggregated based on a pre-defined protocol and used to update the global model. Steps 2-4 can be repeated until a certain accuracy level is reached or until the accuracy converges. Even though FedML usually uses a client-server architecture consisting of a central orchestration server and multiple clients, other architectures have been proposed as well. We refer to client-server architectures with a central orchestrating server since it is the most widely used architectural pattern (Lo, Lu, Zhu, et al 2022).

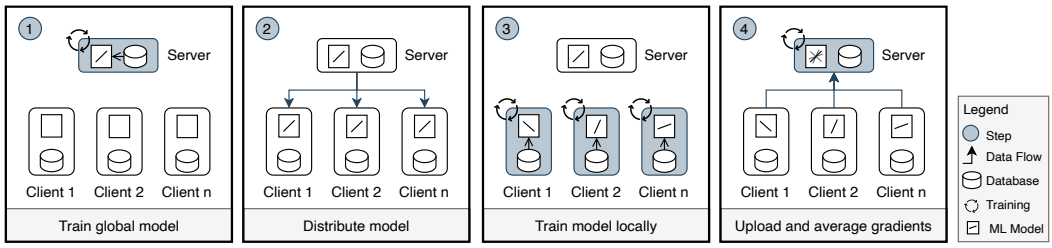

Figure 1. One Iteration of the Federated Machine Learning Training Process (own work)

IEEE published a reference architecture with generalized information about the structure of FedML model training processes and involved roles (IEEE 2021). The standard can be used as a basis for implementation, but it does not offer information about the entire life cycle. However, considering all life cycle stages is crucial and an existing open problem for an industrial-level implementation of FedML (Zhang et al. 2020). In contrast to the current literature on FedML, such life cycle models can be found in numerous studies on traditional, centralized ML systems. Studer et al. (2021), for example, introduced CRISP-ML(Q), an standard ML process model overarching the entire life cycle from business understanding to monitoring and maintenance with a focus on quality assurance. Kreuzberger et al. (2022) developed an end-to-end workflow including functional components and roles tailored to ML operations (MLOps) principles. Similarly, Kumara et al. (2022) proposed a reference architecture for

MLOps to aid in streamlining the life cycle of ML models in production. Besides, work on software engineering for ML as well as studies on AI governance yielded insights into ML life cycles. Amershi et al. (2019) investigated how software teams at Microsoft developed AI applications and presented a life cycle model including a set of best practices and challenges based on their observations. Laato et al. (2022) explored the incorporation of AI governance into system development life cycle models by conducting expert interviews. These interviews resulted in a set of governance concepts and a visualization of how AI governance stages tie into existing software development life cycles. The studies on ML life cycles yielded valuable insights into existing practices and the different stages of ML project. Based on that, Ritz et al. (2022) presented a process model to additionally illustrate the dependencies and interactions within the different stages and activities. Their model describes the activities and resulting artifacts including the interdependencies throughout the ML software development cycle. The existing life cycle and process models on centralized ML systems can be used as a basis for FedML projects but need to be revisited due to the decentralized nature of FedML. The decentralization introduces additional layers of complexity and coordination, which need to be addressed for a successful industrial-level implementation. In contrast to studies on ML systems, the current literature on FedML lacks such a comprehensible overview of the complex project structure. We aim to close this research gap and enhance the current literature corpus on FedML through a process model which breaks down, structures, and illustrates the different project stages including required tasks, resources, roles, artifacts, and their interrelations.

Research Approach

To investigate why FedML projects fail to actualize, we initially conducted a focus group discussion. We invited three project teams that attempted to realize FedML projects prior and discussed the encountered challenges during the projects. The main challenges have already arisen in the project initiation phase and was two-fold. The first challenge is the structuring of the process flow including the activities, resources, data, and information exchange. Secondly, communicating the division of tasks and coherent interrelations with the collaboration partners posed another main challenge. The focus group participants agreed that a structured end-to-end process model could alleviate both problems. The process model would facilitate the planning phase and communication with external participants by providing an abstract overview of the process flow including its required resources, roles, activities, and their interrelations throughout the project. We recognized that current research on FedML lacked such a holistic process model. To develop a process model for operationalizing FedML projects we used the design science research (DSR) methodology as proposed by (Peffers et al. 2007). We chose the DSR approach since it provides a methodical, rigorous approach for producing and evaluating innovative, purposeful artifacts for a specified problem domain (Hevner et al. 2004). Our research activities according to the DSR steps by Peffers et al. (2007) can be summarized as follows:

(1) Problem Identification and Motivation: Through a focus group, we recognized that structuring the FedML process is complex. Understanding and communicating the division of tasks as well as the coherent interactions with potential partners is challenging. Current literature does not offer a structured, comprehensible overview of the FedML project flow and lacks guidance in the project initiation phase.

(2) Objectives of a Solution: Our objective was the development of a generically applicable end-to-end process model for FedML projects to facilitate the planning phase and communication with external participants. The model should comprise an entire life cycle and provide a clear understanding of interacting entities, their corresponding activities, interactions, and dependencies along with the required resources and resulting artifacts. The content and structure should be closely aligned and consistent with best practices offered by literature on ML life cycles, and software development life cycles. Furthermore, the model should be useful and easily understandable by non-technical stakeholders and practitioners.

(3) Design and Development: We reviewed current literature on ML life cycles, software development life cycles and assessed which practices, procedures and information are applicable for our process model. We gained our knowledge base on the structure of FedML processes from current literature. We conducted an interview study to collect information on the FedML project structure, separation of roles, activities, and interactions between the entities throughout the project life cycle. We describe the interview study in another publication (Müller et al. 2023). According to the described objectives, we designed and structured the elements of our process model. We incorporated the relevant findings from the literature review and interview study. During development, we conducted regular mini focus group discussions with

varying participants to assess the model regularly and to iteratively implement feedback. Additionally, we presented the initial resulting process model to a diverse group of 6 experts with technical backgrounds to gather feedback and change requests on the completeness, comprehensibility, and level of detail.

(4) Demonstration. We demonstrated the process model during an expert discussion and within a large industrial lighthouse project as part of a project ideation process to adopt FedML for varying use cases.

(5) Evaluation: We conducted two survey-based iterations to evaluate the artifact on the objectives.

(6) Communication: Communication is being done through this paper.

Results

A process model should describe an abstract representation of reality and reduce complexity by eliminating details which do not influence relevant behavior (Curtis et al. 1992). We adapted the seven process modeling guidelines (7PMG) by Mendling et al. (2010) wherever possible to design such a process model, which is comprehensible and comprises solely the important details of the underlying process. Following the 7PMG, we use as few elements as possible by clustering activities where possible and only incorporate the most essential elements for the FedML project life cycle. This guideline concurs with our objective to produce an easily understandable process model. To follow the 7PMGs, we also aimed to minimize the routing paths per element by only including the most relevant inputs, outputs, and feedback loops. The full process model is designed so that it can be easily decomposed according to the life cycle stages. Finally, we used verb-object activity labels. Figure 2 displays the resulting process model.

Components of the Process Model

In this section, we answer RQ1 by presenting the components of the process model. These components comprise life cycle *stages*, *activities*, *roles*, *resources*, as well as *artifacts* and are described as follows:

Stages of the life cycle represent interrelated tasks and activities and are structured according to software development life cycle stages in combination with MLOps and FedML-specific stages. Our model comprises five stages: *Project Initiation*, *Project Validation*, *Project Setup*, *System Design and Development*, and *Deployment and Maintenance*. The *System Design and Development* stage contains the FedML training process and can be divided into three sub-stages: *Global Model Design*, *Local Model Training* and *Global Model Aggregation*. The stages are described in the section on the process flow.

Roles describe the behavior and responsibility of an individual or a group. One individual may take multiple roles, but each role describes a distinct set of behaviors and responsibilities attributed to a role. The following gives a short description of the nine different roles and their intended area of responsibility.

(1) Business Stakeholder has the business need or problem to be solved and represents the initiator. The Business Stakeholder defines the business requirements, establishes a project strategy and agrees on the project definition with the success criteria (Kreuzberger et al. 2022; Too and Weaver 2014; Zwikael and Meredith 2018). In the process model, the business stakeholder is solely involved in strategic activities.

(2) Project Manager is responsible and held accountable by the business stakeholder for the success of the project. The project manager oversees and leads the team to achieve the project's objectives and is responsible for planning and managing the project efficiently (Too and Weaver 2014; Zwikael and Meredith 2018). Therefore, the project manager is involved in strategic activities.

(3) Subject Matter Expert (also called domain expert) is an expert in a specific domain and deeply understands the business problem. The Subject Matter Expert aids the Business Stakeholder in the definition of goals during the project initiation stage (Amershi et al. 2019; Studer et al. 2021).

(4) Solution Architect analyzes the functional, technical and business requirements and defines the overall solution architecture and technologies to be used (Kreuzberger et al. 2022). In the process model, the Solution Architect is mainly involved in the technical assessment of the ML problem.

(5) Data Scientist is an ML expert responsible for analyzing the business problem and building a suitable ML model which solves the business problem. This includes conducting the algorithm selection, feature engineering and performing hyperparameter tuning (Kim et al. 2018; Kreuzberger et al. 2022; Kumara et

al. 2022). Hence the Data Scientist is involved in operational activities such as translating the business problem into an ML problem and further ML-related operational activities such as designing the model.

(6) Data Engineer is responsible for pulling and engineering raw data such that the curated data is usable and accessible to the Data Scientist. These activities comprise building and managing data pipelines together with performing data cleaning and feature engineering (Kreuzberger et al. 2022; Kumara et al. 2022). The Data Engineer performs operational activities like preparing data. The Data Engineer is the only role which performs local activities.

(7) Software Engineer applies mature software engineering techniques to ensure that the ML system is built in a robust manner. The Software Engineer is responsible for packaging, testing as well as assuring the quality and robustness of the ML model along with the infrastructure (Kreuzberger et al. 2022; Serban et al. 2020). Hence, the Software Engineer is involved in software-related operational activities.

(8) Development-Operations (DevOps) Engineer aims to automate and combine the processes of model development and model operations to deploy and serve the ML model. This includes incident management, monitoring, support and delivery of models in production (Kreuzberger et al. 2022; Kumara et al. 2022; Laato et al. 2022). Hence, the DevOps Engineer performs operational activities in the deployment and maintenance stage.

(9) Legal Representative checks compliance with legal frameworks (e.g., GDPR or HIPAA) and clarifies the conformity of the potentially established collaboration.

Activities summarize units of work performed by roles. An activity has a clear purpose and usually results in the creation or update of an artifact (Anwar 2014). Specific roles are assigned to each activity. We chose the granularity such that the goal of the activity is comprehensible and such that the flow of activities provides a good understanding of the underlying process. Activities are described with verb-object labels. They either use information or data flows as input or can be triggered through prior activities. Some activities are optional depending on the use case-specific set-up and are indicated through dashed lines. The different activities are arranged horizontally by their chronological order and vertically by the type of activity. Activity types are indicated by swimlanes and are categorized into *strategic activities*, *operational activities,* and *local activities*. Strategic activities comprise management tasks such as planning, analysis, strategy formulation and organization. Operational and local activities cluster technical tasks which are concerned with the development and serving of the software product. We introduced an additional swimlane for local activities specific to FedML. These activities are executed solely in the environment of the local data contributors and accentuate which activities are performed by the decentralized entities. The separation into operational activities and local activities demonstrates the division, interrelations and dependencies between the central entity and local data contributors.

Artifacts are tangible by-products which are produced, modified, or used by a process. Artifacts are used as input to perform an activity or are the result of activities (Anwar 2014). We differentiate between *document artifacts* and *code artifacts* Documents comprise specifications, descriptions, diagrams, reports, and other documented outputs. Code artifacts include output resulting from implementations such as ML models, infrastructures, or logs. Optional activities are indicated through a dashed outline.

Resources are hardware or software components which are required to fulfil the activities but do not result from the activities of the process. We differ between global and local resources. Global resources can be used by every participant, whereas local resources are used exclusively by the corresponding local data contributor. Resources comprise for example data storage, log storage, model registry and metric storage. We include resources in our process model to fulfill the objective of providing transparency and showcasing the global and local interactions and dependencies within the FedML project life cycle.

Process Flow of a Federated Machine Learning Life Cycle

In this section, we answer RQ 2 by providing a walkthrough of the process model as displayed in Figure 2. We based the stages of our process life cycle on the different phases of software development life cycle models (Akinsola et al. 2020; Alsaqqa et al. 2020; Apoorva and Deepty 2013; Gurung et al. 2020) in combination with ML life cycle models (Amershi et al. 2019; Kreuzberger et al. 2022; Kumara et al. 2022; Laato et al. 2022; Ritz et al. 2022). Finally, we adjusted the stages to the specifics of FedML processes and architectures (Bharti et al. 2022; Bonawitz et al. 2019; IEEE 2021; Lo et al. 2021; Lo, Lu, Zhu, et al. 2022;

Zhang et al. 2020). Subsequently, we will describe the process model according to the life cycle stages. The listed activities represent a high-level abstraction of the needed tasks. The level of abstraction should give non-technical stakeholders and other practitioners a good understanding of the needed activities but should not be too detailed to keep the process model comprehensible and manageable.

Project Initiation stage is the first step in starting a project. The business stakeholder and subject matter expert establish the goal and scope of the project. They define how business value will be delivered and which use cases will be tackled. Hence, a business model, business requirements and use case diagrams will emerge. The goals and scope are communicated with the data scientist, which will translate the business problem into an ML problem. The data scientist produces a technical description of the problem, data descriptions, and a deployment strategy according to the business requirements. Any challenges will be communicated with the business stakeholder to refine the goals and scope if necessary.

Project Validation stage ascertains if the project is feasible and can meet the expected requirements. The ML problem is the input for the validation task which needs to be performed in the strategic and operational dimension. The business stakeholder, project manager, and legal representative conduct the business case analysis resulting in a business case report. The data scientist and solution architect carry out technical assessments. The technical assessment yields data understanding and the infrastructure specifications describing the needed infrastructure for data processing, model training and orchestration.

Project Setup stage lays the basis for the implementation. The strategic activities are concerned with the collaboration creation if the project is implemented in a collaborative setting. Hence, this strategic activity the project setup stage is optional. The business stakeholder and project manager aim to establish and plan the collaboration, whereas the legal representative checks the compliance and legal regulations of the collaboration. The collaboration creation activities result in a collaboration agreement, which also specifies the used communications channels of the collaboration. We provide a comprehensive description of the socio-technical aspects which need to be addressed in the collaboration agreement in a separate publication (Müller et al. 2023). The software engineers set up the infrastructure as specified in the infrastructure specifications and technical description. Therefore, the infrastructure for data processing, model training and orchestration emerges from this stage. Optionally, communication channels are established if the project is implemented within a collaboration.

System Design and Development stage aims to create the software product as well as the ML model and is divided into three substages:

(1) Global Model Design substage intends to define the source code for the initial global model which is subsequently trained during the following substages. This requires access to existing data storage and log storage for traceability. The data engineer prepares the data as needed by the specified ML problem. The data scientist builds the corresponding ML model, which is prepared by the software engineer for distribution to the local datasets. Additionally, the data engineer defines data specifications rules, which can be followed by the local data engineers to achieve data homogeneity. The data preparations and features can be iteratively adapted dependent on the ML model performance. Finally, the initial model is stored in a central model registry. Summarized, this stage yields logs to ensure traceability, data preparation specifications to achieve data homogeneity and a global model implementation.

(2) Local Model Training substage is conducted by the local participating data contributors. The local data engineers extract their local data from their corresponding data storage and prepare the data according to the defined data preparation specifications. After preparation, they pull the latest model from the model registry and train the local model. This results in a local version of the ML model. The gradients from the training process need to be extracted and sent to the central entity. All training logs need to be stored in a local log storage to ensure traceability. This local data and log storage is not public to guarantee privacy and should only be exchanged in exceptional cases e.g., for accountability matters.

(3) Global Model Aggregation substage collects and aggregates the resulting update gradients from the local data contributors. A data scientist builds the global model through the collected update gradients. The model is evaluated and stored in the model registry. Logs are stored in a log storage for traceability. If the model performance is insufficient, another training iteration can be triggered. The evaluation results can also be used to adjust the scope and goals or adapt the data specifications and model design. If the model performs as desired, it is packaged by the software engineer for deployment.

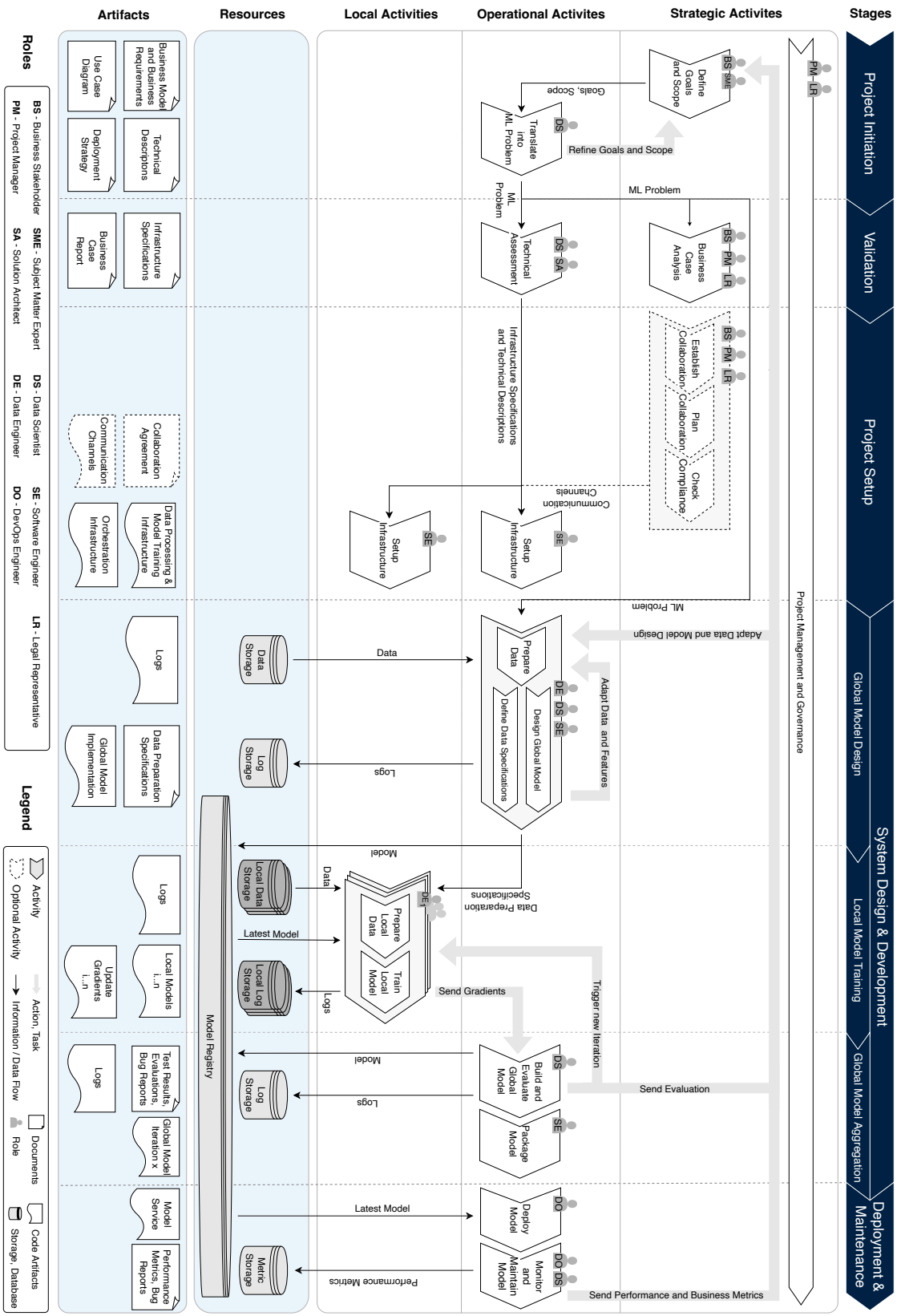

Figure 2. Process Model of an End-to-End FedML Project Life Cycle

Deployment and Maintenance stage intends to serve and maintain the model service. The DevOps engineer deploys the final packaged model. The DevOps engineer and data scientist maintain and monitor the model while storing the recorded metrics in a metric storage. This includes the monitoring report of business metrics. Hence, this stage produces a model service, metrics, and bug reports. If ML service deviates from the desired behavior, the performance metrics can be used to trigger changes in the data preparation scheme and model design as well as trigger an adjustment of the goals and scope.

Demonstration and Evaluation

We conducted two iterations of demonstrations with subsequent survey-based evaluations. The first iteration was part of the *Design and Development* stage as described in the section on the *Research Approach*. We demonstrated the process model and discussed the model with the experts. We received feedback through the discussion and asked all participants to fill out a survey. The survey evaluates the relevance and usefulness of each element with its overall comprehensibility, level of detail, completeness, usefulness, and value. The demonstration was part of a project ideation process to adopt FedML within a large industrial lighthouse project. In total, 14 experts participated in the assessments. Overall, the group of experts comprised a variety of affiliations from startups, big tech companies and research institutions with experience ranging from one to 12 years in their respective field. For each demonstration, all experts confirmed the identified problem and validated the idea, structure, components, and importance of the process model. We incorporated the feedback after each round of discussions.

To evaluate the models' usefulness, we finally presented the refined model to 8 potential users and key stakeholders such as project managers, product specialists, solution architects, and solution advisors. Each participant was involved in prior FedML use case discussions. We aimed to assess if the model can be used in practice, or if further modification is required. One expert stated that the model seems complex at a first glance due to its size. Since the other experts liked that the model displays an entire project life cycle, that the level of detail is fitting, and all components should be included. Based on these statements, we have not made any changes. The involved experts aim to use the model in future discussions.

The overall evaluation results in Figure 3 show that the model was well-received by all participants. A large majority of the participants described the model as detailed enough, complete, comprehensive, and valuable. The model is unanimously considered useful, which implies that the set objective of a useful and understandable model is met. Regarding the components, the stages seem to be relevant for each expert, whereas very few experts deem the artifacts and roles somewhat irrelevant. Three experts stated that the last stage of a project life cycle is missing *Model Maintenance* activities and therefore strongly disagreed that the model covers all important aspects. We agreed and updated our model accordingly. Therefore, we consider the model as presented in Figure 2 as complete. All experts confirmed the structure, comprehensiveness, and level of detail. Thereby, we assume that the objective of a clearly understandable and generically applicable model is met as well. All experts agreed that the process model includes all relevant details and stated that they will use the model in future discussions on FedML with their stakeholders and potential collaborators, which confirms the usefulness for the target user again.

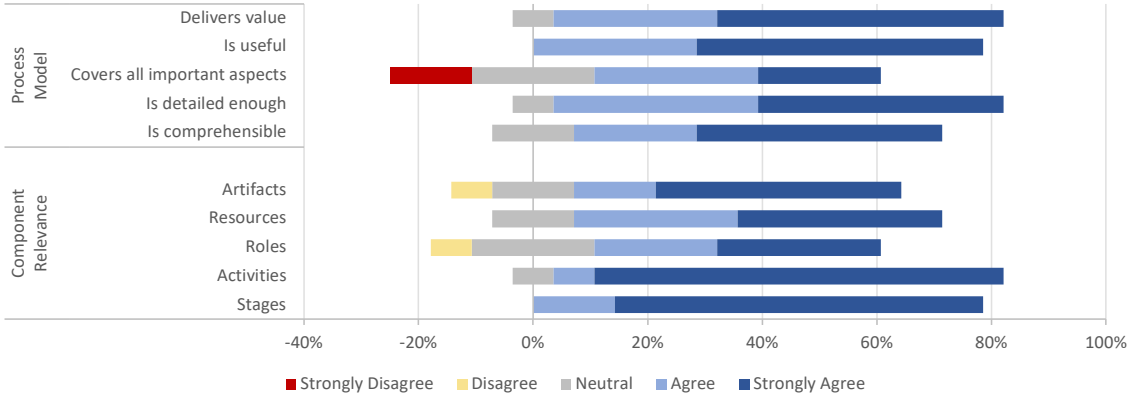

Figure 3. Results of the Evaluation

Conclusion

In this study, we introduced a process model of an end-to-end FedML project to provide transparency and facilitate the comprehensibility of the complex project life cycle. The process model extends current literature on traditional, centralized ML life cycles and complements existing reference architectures through a holistic overview of the FedML project life cycle. By investigating related work, conducting focus group discussions and an interview study, we identified *stages*, *activities*, *roles*, *resources*, and *artifacts* as the most relevant components for a comprehensible overview of the FedML process (RQ1). Finally, based on our research, we designed and constructed the process model with the corresponding interactions and dependencies of the different components (RQ2). The evaluation results show that potential users deem the model to be useful, deliver value and comprise the most relevant aspects with an appropriate level of detail. The process model can help practitioners to gain an understanding of the structure and interrelations within a FedML project. As stated in the expert evaluations, the model can be used for simplified communication with potential collaboration partners and decision-makers. Additionally, the model can ease the project initiation phase and successful implementation by providing guidance and an overview of the whole project life cycle.

The process model is currently limited to server-client architectures with a central orchestrating server and multiple local data contributors since it is the most widely used architectural pattern. However, more architecture designs are possible and cannot be abstracted by our process model. In the future, adapted process models for novel or different architectures could be established. Also, the model has only been evaluated by 14 experts and within one project ideation phase. We recommend testing and evaluating the model in more scenarios to gather more feedback for improvement. Practical validation might also lead to a more thorough alignment with current MLOps rather than DevOps paradigms to further streamline the model to current best practices. Overall, the process model aids the successful implementation of FedML projects. By that, our model contributes towards more privacy-enhancing ML projects and might help improve the overall performance of ML models through leveraging decentralized and currently untapped dataset. Furthermore, we aim to publish an additional activity model with more detailed activities for each stage of the life cycle to provide further guidance on more in-depth technical aspects of FedML projects.

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., and Adebiyi, A. A. 2020. "Comparative Analysis of Software Development Life Cycle Models (SDLC)," *Intelligent Algorithms in Software Engineering,* In R. Silhavy (Ed.), Springer International Publishing, pp. 310–322.

Alsaqqa, S., Sawalha, S., and Abdel-Nabi, H. 2020. "Agile Software Development: Methodologies and Trends," *International Journal of Interactive Mobile Technologies (IJIM)* (*14*:11), pp. 246-270.

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

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

Apoorva, M., and 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), pp. 64–69.

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

Bharti, S., McGibney, A., and O'gorman, T. 2022. "Design Considerations and Guidelines for Implementing Federated Learning in Smart Manufacturing Applications," *IIC Journal of Innovation* (*19:1)*, pp. 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., and Roselander, J. 2019. "Towards Federated Learning at Scale: System Design," in *Proceedings of Machine Learning and Systems 1*, Standford, California, pp. 374–388.

Curtis, B., Kellner, M. I., and Over, J. 1992. "Process modeling," *Communications of the ACM* (*35:1)*, pp. 75–90.

Giray, G. (2021). "A Software Engineering Perspective on Engineering Machine Learning Systems: State of the Art and Challenges," *Journal of Systems and Software* (*180:1)*, pp. 35-97.

Gurung, G., Shah, R., and Jaiswal, D. 2020. "Software Development Life Cycle Models-A Comparative Study," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (10:4)*, pp. 30–37.

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

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

Kim, M., Zimmermann, T., DeLine, R., and Begel, A. 2018. "Data Scientists in Software Teams: State of the Art and Challenges," *IEEE Transactions on Software Engineering* (*44*:11), pp. 1024–1038.

Kreuzberger, D., Kühl, N., and Hirschl, S. 2022. "Machine Learning Operations (MLOps): Overview, Definition, and Architecture," *IEEE Access (11:1),* pp. 31866-31879.

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

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

Lo, S. K., Lu, Q., Paik, H.-Y., and Zhu, L. 2021. "FLRA: A Reference Architecture for Federated Learning Systems," in *Proceedings of the 15th European Conference on Architecture*, pp. 83–98.

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

Lo, S. K., Lu, Q., Zhu, L., Paik, H., Xu, X., and Wang, C. 2022. "Architectural Patterns for the Design of Federated Learning Systems," *Journal of Systems and Software*, (191:3).

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

Mendling, J., Reijers, H. A., and van der Aalst, W. M. P. 2010. "Seven process modeling guidelines (7PMG)," *Information and Software Technology* (*52*:2), pp. 127–136.

Müller, T., Zahn, M., and Matthes, F. 2023. "Unlocking the Potential of Collaborative AI - On the Socio-Technical Challenges of Federated Machine Learning," in *ECIS 2023 Proceedings,* Kristiansand.

Peffers, K., Tuunanen, T., Rothenberger, M., and Chatterjee, S. 2007. "A design science research methodology for information systems research," *Journal of Management Information Systems* (*24*:1), pp. 45–77.

Ritz, F., Phan, T., Sedlmeier, A., Altmann, P., Wieghardt, J., Schmid, R., Sauer, H., Klein, C., Linnhoff-Popien, C., and Gabor, T. 2022. "Capturing Dependencies within Machine Learning via a Formal Process Model," in *Proceedings of the 11th ISoLA, 3*, Rhodes, pp. 249–265.

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

Serban, A., van der Blom, K., Hoos, H., and Visser, J. 2020. "Adoption and Effects of Software Engineering Best Practices in Machine Learning" in *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, Torino, pp. 1–12.

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

Too, E. G., and Weaver, P. 2014. "The management of project management: A conceptual framework for project governance," *International Journal of Project Management* (*32*:8), pp. 1382–1394.

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

Zwikael, O., and Meredith, J. R. 2018. "Who's who in the project zoo? The ten core project roles," *International Journal of Operations & Production Management* (*38*:2), pp. 474–492.