This is the accepted version of a paper presented at 2023 8th International Conference on Fog and Mobile Edge Computing, FMEC 2023, Tartu, 18/9 - 20/9 2023.

Citation for the original published paper:

In: Quwaider M., Awaysheh F.M., Jararweh Y. (ed.), 2023 8th International Conference on Fog and Mobile Edge Computing, FMEC 2023 (pp. 270-275). Institute of Electrical and Electronics Engineers (IEEE)
https://doi.org/10.1109/FMEC59375.2023.10306036

N.B. When citing this work, cite the original published paper.

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:bth-25824
ART4FL: An Agent-based Architectural Approach for Trustworthy Federated Learning in the IoT

Fahed Alkhabbas†‡§, Sadi Alawadi†‡, Majed Ayyad ‡, Romina Spalazzese§, and Paul Davidsson†§
†Internet of Things and People Research Center, Malmö University, Sweden
‡Department of Computer Science and Media Technology, Malmö University, Sweden
§Department of Computer Science, Birzeit University, Palestine
{fahed.alkhabbas, romina.spalazzese, paul.davidsson}@mau.se
†School of Information Technology, Halmstad University, Halmstad, Sweden
sadi.alawadi@hh.se
‡ Department of Computer Science, Blekinge Institute of Technology, Karlskrona, Sweden
sadi.alawadi@bth.se
§Department of Computer Science, Birzeit University, Palestine
mayyad@birzeit.edu

Abstract—The integration of the Internet of Things (IoT) and Machine Learning (ML) technologies has opened up for the development of novel types of systems and services. Federated Learning (FL) has enabled the systems to collaboratively train their ML models while preserving the privacy of the data collected by their IoT devices and objects. Several FL frameworks have been developed; however, they do not enable FL in open, distributed, and heterogeneous IoT environments. Specifically, they do not support systems that collect similar data to dynamically discover each other, communicate, and negotiate about the training terms (e.g., accuracy, communication latency, and cost). Towards bridging this gap, we propose ART4FL, an end-to-end framework that enables FL in open IoT settings. The framework enables systems’ users to configure agents that participate in FL on their behalf. Those agents negotiate and make commitments (i.e., contractual agreements) to dynamically form federations. To perform FL, the framework deploys the needed services dynamically, monitors the training rounds, and calculates agents’ trust scores based on the established commitments. ART4FL exploits a blockchain network to maintain the trust scores, and it provides those scores to negotiating agents’ during the federations’ formation phase.

Index Terms—Trustworthy Federated Learning, Internet of Things, Machine Learning, Agents

I. INTRODUCTION

The integration of Artificial Intelligence (AI) and Internet of Things (IoT) has enabled the development of novel types of systems, often referred to as AIoT, in many application domains such as healthcare, surveillance, smart buildings, and transportation. Specifically, IoT has enabled devices and objects such as sensors, actuators, and appliances to collect, exchange, and process data and accordingly achieve user goals. AI technologies (e.g., Multi-Agent Systems (MAS), and Machine Learning (ML) made IoT systems more autonomous and smarter. Due to the widespread use of IoT devices and objects and the sensitivity of the data they can collect, privacy and trust are identified as core concerns when engineering AIoT systems [3], [18].

Traditional ML approaches require data to be collected and aggregated offline on centralized servers, where ML models are trained. Therefore, such approaches do not preserve the data privacy of AIoT systems. Further, those approaches do not allow the optimal consumption of resources available across the Edge-Cloud continuum [13], [14]. Distributed ML approaches enable a more optimal resource utilization, however, they require the data to be released to the distributed servers. Thus, those approaches do not address the privacy concerns [7], [17]. To address privacy issues, Google proposed Federated Learning (FL), an emerging paradigm that enables systems to collaboratively train ML models without the need to disclose their private data [5], [16].

Several FL frameworks and approaches have been developed, however, they do not enable trustworthy FL in open IoT environments. Specifically, they have the shortcomings, including the following ones: (i) they do not support IoT systems that collect datasets with similar features to discover each other at runtime, communicate, and negotiate about training terms (e.g., accuracy, communication latency, availability, and cost). Instead, existing frameworks and approaches assume that the responsible actors communicated and reached agreements offline; (ii) although FL is a decentralized learning technique and IoT environments are also decentralized, most existing frameworks and approaches assume centralized client-server architectures [6]; (iii) they evaluate IoT systems’ trustworthiness mainly based on their local ML models’ accuracy. Thus, they overlook core aspects such as communication latency, availability, and reliability [6].

Towards addressing the aforementioned shortcomings, we propose ART4FL, a decentralized framework that enables end-to-end horizontal FL in open IoT settings. Specifically, ART4FL exploits the notion of autonomous agents and commitments to enable IoT systems that collect datasets with similar features to dynamically negotiate training terms and accordingly form federations. Further, ART4FL evaluates IoT systems’ trustworthiness based on the established commitments and considering ML models’ accuracy and other core
aspects that existing literature overlooks (e.g., communication latency, availability, and reliability). ART4FL exploits a blockchain technology to maintain the trust scores, and it provides those scores to negotiating agents’ during the federations’ formation phase.

The remainder of this paper is organized as follows. Section II discusses related studies. Section III presents the smart elderly house scenario. Section IV introduces our framework. Section V discusses the framework. Finally, Section VI concludes the paper and outlines future work directions.

II. RELATED WORK

Several (open-source) FL frameworks have been developed [14]. TensorFlow FL (TFF) is a client-server based framework that supports the simulation of distributed training of FL models but runs only on a single machine. FATE is another framework that adopts a distributed client-server architecture that can be deployed over a Kubernetes cluster. FATE assumes that servers are semi-honest, and it supports linear and logistic regression, deep learning neural networks, tree-based algorithms and transfer learning [15]. The OpenFL framework adopts also a client-server architecture where the server initiates training rounds by broadcasting a federation plan to all the clients, outlining the process flow, tasks, and instructions. OpenFL ensures data confidentiality, integrity, and remote attestation during training rounds [10].

FEDn is a hierarchical framework that adopts the client-server architecture and also follows the map-reduce paradigm. FEDn is developed to scale from pseudo-distributed development setups to real-world production networks in distributed and heterogeneous environments. Furthermore, in the enterprise edition, FEDn incorporates blockchain technology to ensure trustworthiness and security within the framework. FLOWER is a model-agnostic framework that adopts a client-server architecture and supports several aggregation algorithms. FLOWER focuses on enabling communication-efficient federated learning, particularly for heterogeneous and resource-constrained IoT edge devices [4]. As can be noted, most common frameworks adopt client-server architecture, which is not very suitable for decentralized IoT environments. Additionally, the frameworks do not support participants to dynamically discover each other, negotiate training terms, and accordingly form and enact federations.

Rehman et al. [20] proposed TrustFed, a framework for enabling trustworthy and fair federated learning. The framework supports three phases: clients selection, configuration, and reporting. Clients are selected based on their trust scores that are updated after each training round. Similarly, Yang et al. [22] presented B-FL, a blockchain based architecture that exploits a secure global aggregation algorithm to resist malicious devices. Unlike our framework, existing frameworks do not support clients to negotiate and dynamically form agreements.

Approaches have also been proposed to enable decentralized FL. Lalitha et al. [12] presented an Bayesian-like approach where users aggregate information from their one-hop neighbors and accordingly learn a model in a decentralized manner. Dai et al. [9] proposed an approach that exploits personalized sparse masks and a sparse training technique to reduce the communication and computation cost in decentralized FL. Yang et al. [21] developed an analytical model to characterize the performance of FL in wireless sensor networks. Chen et al. [8] presented a model for selecting suitable clients to execute a FL algorithm in wireless sensor networks where the communication bandwidth is limited. However, unlike our framework, most existing approaches investigate decentralized FL taking a mathematical perspective.

To summarize, most existing approaches focus on the federation formation (i.e., clients selection), training, and evaluation phases of FL shown in Figure 1. Thus, they do not enable IoT systems to dynamically discover each other and negotiate training terms. ART4FL is the first framework that exploits the notions of agents, commitments, and blockchain with the aim to cover all the phases shown in Figure 1 from a software engineering perspective.

![Fig. 1. FL phases covered by ART4FL](image)

III. SCENARIO

An elderly house is newly established in a developing country. To ensure the safety of the residents, the house managers decided to implement automated mechanisms for fall prevention and detection. For this purpose, the house facilities were equipped with sensors such as pressure sensors, and residents were offered to wear accelerometers and gyroscopes. To predict or detect the fall of residents, a ML model is executed using the sensors’ data.

The lack of data represents an obstacle to train an accurate ML model. Additionally, regulations prevent other elderly houses from sharing their data because of privacy concerns. To overcome this obstacle and preserve residents’ privacy, the technical team manager registers at ART4FL to participate in
FL and train the ML model collaboratively with trustworthy care facilities. In particular, she looks for facilities that: (i) have datasets that exhibit similar features to the dataset collected within the house; (ii) have ML models capable of accurately predicting and/or detecting resident falls with a minimum accuracy of 90%. To incentivize participation, she allocates a budget up to 70$ to compensate each facility that participates in the training.

The house receives bids from fifteen facilities that meet the specified conditions. Further, another facility sends an offer to share the training cost as it also aims to improve its ML model’s accuracy. ART4FL provides the house with information about the trustworthiness of the bidders. ART4FL enables all parties to automatically negotiate on the FL terms, including the accuracy, communication latency, training period, and the cost. The house reaches agreement with ten facilities and agrees to share the training cost equally with another facility.

At the scheduled training time, one facility did not connect to participate in the training, and another facility shared its ML model’s weights late, whereas the remaining facilities met their commitments. As a result, the house’s ML accuracy has significantly improved, and the facilities that met their commitments were rewarded. Further, the trust scores of all participants have been updated in the ART4FL framework.

IV. ART4FL

This section presents the abstract design of our ART4FL framework. Specifically, we introduce an abstract architecture and process that cover the FL phases shown in Fig. 2.

A. Architecture

Figure 2 shows ART4FL abstract architecture, which comprises the following components: agent, agent catalog, communication platform, trust manager, and the blockchain network.

Agent. An agent is a software program that autonomously participates in FL to achieve the user goal, which could be to: (1) increase its local ML model’s accuracy; and/or (2) make profit by sharing its model with other agents. The profile manager registers the agent by creating its profile in the agent catalog. The profile includes the agent goals, metadata about its datasets (e.g., input and output features, size, and data distribution), evaluation metrics of its local ML model (e.g., accuracy and Cohen Kappa), trust score, applied security measures (e.g., messages’ encryption protocol), availability, communication latency, and payment method. The rule engine supports users to configure their agents via rules (e.g., to set the maximum budget for the training). The negotiation manager negotiates with other agents to achieve the agent goal given the specified rules (e.g., agents agree on sharing the training cost).

The commitment manager creates or accepts requests for commitments when agents reach agreements. A commitment is a contract between a debtor and a creditor whereby the debtor performs a service to the creditor and in return receives a reward. If the debtor fails to deliver the service, its trust score will be lowered and a penalty can be imposed [2]. The training manager trains the agent’s local ML model using the local dataset. The evaluation manager evaluates the local ML model and the models of randomly selected agents (see Section IV-B). The adaptation manager adapts the agent behavior according to the context analyzed by the context manager. For instance, if a user reduces the budget allocated for the training, the agent will automatically (re-)negotiate or cancel its commitments for the training rounds that have not started. Finally, the knowledge base contains the agent’s local ML model and the dataset that is collected by available IoT devices and objects and processed for training.

Agent Catalog. This component registers agents in the blockchain network. As blockchain networks have performance and scalability issues when deployed in fully decentralized settings or when having a large number of nodes (i.e., agents) [1], [11], this component maintains updated clones of agents’ profiles in its knowledge base for performance purposes.

Communication Platform. This component is a distributed communication paradigm that enables a large number of loosely-coupled agents to communicate during the training and evaluation phases. Additionally, it collects metrics (e.g., latency, availability, reliability) about agents’ communications’ channels and shares it with the trust manager (see Section
Trust Manager. This is a distributed component whose instances are dynamically deployed for each group of agents that are engaged in FL rounds. It calculates the trust scores of agents participating in FL by considering: (i) the information shared by the communication platform concerning communication links between the agents; (ii) the evaluation metrics of the ML models that the agent share during FL rounds; and (iii) the commitments that the agent made during the negotiation phase.

Blockchain. The blockchain network plays a prominent role in maintaining the data integrity and performing more trustworthy FL. For this purpose, ART4FL deploys two smart contracts. The agent smart contract stores core parts of the agent profile. The commitment smart contract records the commitments established among agents participating in FL and how the commitment status evolved during the training (i.e., if the agents met the commitments or not). Finally, the blockchain oracle is responsible to interact with the smart contracts to store or retrieve data when requested by the other components.

B. Process

Figure 3 shows the ART4FL abstract process that covers the FL phases illustrated in Figure 1. The process is inspired from the Contract Net protocol, a decentralized task sharing protocol that is commonly used in the field of multi-agent systems [19].

In the registration phase, agents register their profiles in the agent catalog, which also records the profiles in the blockchain network. The blockchain returns the agents’ identifiers and the credentials (e.g., public and private keys) they should use to authenticate and communicate securely. In the discovery phase, an agent queries the catalog about registered agents that have datasets that exhibit similar features to its dataset and meet specific conditions (e.g., related to their trust scores and accuracy of ML models). The agent catalog, which receives continuous updates from the blockchain network regarding registered agents, responds to the request by providing a list
of suitable agents that also includes agents that have the same goal of the requester (if available).

Then, in the federation formation phase, the agent sends a Call for Proposals (CFP) to the list of agents provided by the catalog. A CFP includes information about the agent’s goal, dataset (input and output) features, initial training terms, and how it can reward other agents that participate in the training. The agents receive the call and can either accept the terms, send alternative proposals, decline the call, or not respond. For instance, an agent can ask for a higher reward as it has a trust score higher than the score specified in the initial terms. Accordingly, the agent initiating the call creates commitments to reward the agents that will participate in the training, and these agents willingly accept the commitment. Next, the established commitments are registered in the blockchain network. After that, based on the number of agents willing to participate in the training, agents are grouped into sub-federations based on aspects such as their dataset distribution, and communication latency.

In the training phase, the agents train their ML models locally and share their models with a distributed, highly available, reliable, and cost-efficient communication node (e.g., Amazon Kinesis or Kafka) that broadcasts the model to all the agents that belong to the same sub-federation. The communication node also collects metrics about the communication latency, availability, and reliability of the agents. In the evaluation phase, we envision the framework to work in three modes:

1) extensive evaluation: in this mode, each agent in a sub-federation aggregates each model shared with it, evaluates the aggregated model using its local dataset, and reports the model’s evaluation metrics to the communication node. Afterwards, sub-federations exchange the models that scored the highest on the evaluation metrics. Then, the agent aggregates only the models that improve its performance and rejects the rest. Consequently, the agents might have different models when the training ends. This mode enables agents to produce their customized models and evaluate thoroughly the trustworthiness of other agents. However, this mode is resource demanding and time consuming.

2) medium evaluation: in this mode, in each sub-federation, an agent is selected based on its trust score, available resources, and communication latency to aggregate all the produced models and share them with other agents that belong to its sub-federation. Additionally, each produced model is evaluated once by an agent that is selected randomly. Sub-federations then exchange the models that scored highest on the evaluation metrics. As a result, agents in all sub-federations will have the same model. This mode demands less resources and produces a global model faster than the extensive evaluation mode. However, the global model might not be useful to all the agents.

3) lightweight evaluation: in this mode, a global model is generated following the same approach applied in the medium evaluation mode. However, the agents only evaluate the global model instead of assessing all individual models. Among the three modes, this mode demands the fewest resources. However, the resulting model may be the least trustworthy.

Communication nodes report metrics about the communication links and ML models to the trust manager, which calculates and updates agents’ trust scores in the blockchain network. Finally, in the rewarding phase the agents that participated in the training and met their commitments are rewarded (or penalized) according to the established commitments.

V. DISCUSSION AND CHALLENGES

ART4FL is a decentralized framework that is engineered with the aim of enabling FL in open settings, which represents a step towards an online marketplace of FL. Realizing the framework requires addressing several challenges, including the following ones:

C1- Semantic alignment and ML models’ structure. In horizontal FL settings, it is essential that participating agents’ datasets are aligned. Specifically, the datasets’ features need to match in terms of number, types, and meanings. Similarly, the concepts included in the datasets need to be semantically aligned (i.e., a concept has the same meaning across the datasets). A possible direction to address this issue is that agents need to use a reference ontology when defining their datasets’ structure and the concepts included. However, mechanisms are needed to automate this process and reduce human intervention.

Further, aggregating heterogeneous models that are trained with different algorithms can introduce complexities during the aggregation process and result in an unreliable global model. Therefore, it is recommended that agents agree on the ML models’ structure and algorithm that they will use to train the models locally.

C2- Concept and data drifts. Concept drifts occur when the relationship between input and output features change over time. They can happen because of several factors, including changes in user behavior or varying environmental conditions. For instance, elderly individuals might experience gradual or slow falls due to health conditions. As the agents’ models were not trained on datasets that comprise this new behavior, their accuracy of predicting it will degrade.

Data drifts occur due to unforeseen changes in input data. For instance, the layout of elderly houses might vary with respect to the placement of sensors and furniture. Thus, a ML model that performs well in an environment might underperform in another.

To address concepts and data drifts, the quality of ML models need to be monitored, the collected data need to be inspected to discover new insights or patterns, and ML models
need to be continuously updated and retrained to account for the variations. Additionally, new ML algorithms need to be evaluated to assess if they can improve the quality of the local ML models.

C3- Trade-offs and quality attributes. Evaluating the trustworthiness of individual agents by assessing their ML models and computational and communication capabilities is a resource-demanding process, especially when a large number of agents participate in the training. Enabling secure communications between the agents would increase the time required for agents to communicate. Therefore, the trade-offs between qualities (e.g., performance, security, and trustworthiness) should be considered when refining the ART4FL architecture. Further, in large-scale FL, maintaining knowledge about agents and their commitments in the blockchain can lead to scalability issues [1]. To address this problem, the knowledge can be automatically cloned from the blockchain and stored in an external knowledge base (e.g., a NoSQL database), which is used as the main knowledge provider to the other components. Furthermore, storing commitments on the blockchain can lead to privacy issues especially in sensitive application domains. Towards addressing this issue, we plan to use a hybrid blockchain that can shield sensitive data.

VI. CONCLUSIONS AND FUTURE WORK

Enabling FL in open IoT settings where IoT systems can communicate, negotiate, and reach consensus on training terms is essential to unlock greater benefits of FL. Towards achieving this goal, we introduced a work in progress on a decentralized agent-based framework for trustworthy FL in the IoT.

As future work, we plan to refine the proposed framework by developing a more concrete architecture and process and considering non-functional requirements (e.g., security, performance, and availability) and a wider range of IoT application types. Furthermore, we plan to assess different consensus algorithms, develop a prototype of the framework, and perform an extensive evaluation. Furthermore, we plan to investigate how to extend the framework to enable trustworthy vertical FL where agents’ datasets have different features.

ACKNOWLEDGMENT

This work was partially funded by the Knowledge Foundation (KK-Stiftelsen) via the project Intelligent and Trustworthy IoT Systems (Grant 20220087).

REFERENCES


