Future Generation Computer Systems 152 (2024) 83–98

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

FederatedTrust: A solution for trustworthy federated learning

Pedro Miguel Sánchez Sánchez [a], Alberto Huertas Celdrán [b,*], Ning Xie [b], Gérôme Bovet [c], Gregorio Martínez Pérez [a], Burkhard Stiller [b]

[a] Department of Information and Communications Engineering, University of Murcia, Murcia 30100, Spain
[b] Communication Systems Group (CSG), Department of Informatics (IfI), University of Zurich UZH, 8050 Zürich, Switzerland
[c] Cyber-Defence Campus, armasuisse Science & Technology, 3602 Thun, Switzerland

ARTICLE INFO

Keywords:
Trustworthy federated learning
Trust assessment
AI governance
Privacy
Robustness
Fairness
Explainability
Accountability

ABSTRACT

The rapid expansion of the Internet of Things (IoT) and Edge Computing has presented challenges for centralized Machine and Deep Learning (ML/DL) methods due to the presence of distributed data silos that hold sensitive information. To address concerns regarding data privacy, collaborative and privacy-preserving ML/DL techniques like Federated Learning (FL) have emerged. FL ensures data privacy by design, as the local data of participants remains undisclosed during the creation of a global and collaborative model. However, data privacy and performance are insufficient since a growing need demands trust in model predictions. Existing literature has proposed various approaches dealing with trustworthy ML/DL (excluding data privacy), identifying robustness, fairness, explainability, and accountability as important pillars. Nevertheless, further research is required to identify trustworthiness pillars and evaluation metrics specifically relevant to FL models, as well as to develop solutions that can compute the trustworthiness level of FL models. This work examines the existing requirements for evaluating trustworthiness in FL and introduces a comprehensive taxonomy consisting of six pillars (privacy, robustness, fairness, explainability, accountability, and federation), along with over 30 metrics for computing the trustworthiness of FL models. Subsequently, an algorithm named FederatedTrust is designed based on the pillars and metrics identified in the taxonomy to compute the trustworthiness score of FL models. A prototype of FederatedTrust is implemented and integrated into the learning process of FederatedScope, a well-established FL framework. Finally, five experiments are conducted using different configurations of FederatedScope (with different participants, selection rates, training rounds, and differential privacy) to demonstrate the utility of FederatedTrust in computing the trustworthiness of FL models. Three experiments employ the FEMNIST dataset, and two utilize the N-BaIoT dataset, considering a real-world IoT security use case.

1. Introduction

The last decade has been a revolutionary time for Artificial Intelligence (AI). IBM Watson, ImageNet, and AlphaGo were some of the first successful AI solutions that defined the path towards the recent ChatGPT, DALL·E 2, or Tesla Autopilot, among many others. This journey has allowed Machine and Deep Learning (ML/DL) models to learn how to play, see, speak, paint, drive, and do many other things, almost like humans. The AI hype has traditionally focused on achieving ever-higher accuracy and performance. However, performance is no longer sufficient. In recent years, we have heard more mishaps and situations in which wrong AI-based decisions negatively affect human lives. Some examples are (i) ML/DL-based systems supporting judges in pretrial recidivism scoring racially biased [1], (ii) ML/DL models of autonomous vehicles not prepared nor trained for uncommon fatalities [2], or (iii) AI-powered chatbots giving wrong answers to straightforward questions and problems [3]. These situations erode the trustworthiness of AI and raise concerns about Responsible AI (RAI) [4].

Trustworthy AI is an emerging concept towards RAI that embraces several existing terms such as explainable AI (XAI), ethical AI, robust AI, or fair AI, among others [5]. In 2021, the European Commission proposed the AI Act [6] with high-level foundations, principles, and requirements that AI systems should fulfill to be trustworthy [7]. According to these guidelines, three main foundations should be met throughout the AI system life cycle. First, AI should be lawful and comply with existing regulations. Second, it should ensure adherence to ethical principles. Last but not least, AI should be robust from technical and social perspectives. Under these three foundations, respect for

* Corresponding author.
E-mail address: huertas@ifi.uzh.ch (A. Huertas Celdrán).

https://doi.org/10.1016/j.future.2023.10.013
Received 14 February 2023; Received in revised form 13 October 2023; Accepted 20 October 2023
Available online 31 October 2023

human autonomy, prevention of harm, fairness, and explainability are ethical principles that trustworthy AI systems must respect. Finally, the European Commission translated these principles into the following seven requirements to achieve trustworthy AI: (i) human agency and oversight, (ii) technical robustness and safety, (iii) privacy and data governance, (iv) transparency, (v) diversity, non-discrimination, and fairness, (vi) societal and environmental wellbeing, and (vii) accountability. In parallel to the European Commission, researchers have also developed specific approaches and techniques that AI systems should adopt to be trustworthy. In this context, the systematic reviews on Trustworthy AI conducted in [5,8] identified robustness, privacy, fairness, explainability, and accountability as the five key pillars of trustworthy AI.

Trustworthiness is a critical aspect influencing AI, but nowadays, it is not the only one, and data privacy and protection are also highly demanded by our society. In this context, new laws and regulations have been drawn in response to this necessity. The General Data Protection Regulation (GDPR) in the European Union and the California Consumer Privacy Act (CCPA) in the state of California (USA) are well-known examples of new data protection regulation [9]. As expected, these changes affect AI systems since most ML/DL models are trained with data belonging and maintained by stakeholders in different silos. Therefore, to deal with the challenge of preserving data privacy in AI, Federated Learning (FL) [10] was proposed in 2016 by Google as a decentralized ML paradigm. FL builds collaborative models between the federation members while keeping sensitive data within the premises and control of each participant. In summary, FL is one solution to data silo and fragmentation issues caused by the new legislation that prohibits the free sharing of data and forces data to be maintained by isolated data owners [11].

In summary, trustworthiness is critical in FL to address privacy concerns, maintain model integrity, secure the aggregation process, encourage participant cooperation, enable accountability and auditing, and build user trust. By upholding these principles, FL can unlock the potential for collaborative and privacy-preserving ML in various domains while maintaining the highest trust and privacy protection standards. Trustworthiness in federated learning (FL) must be studied and compared to centralized ML/DL. Both approaches share risks such as algorithmic bias, adversarial attacks, privacy breaches, and reliability issues with centralized ML/DL. However, FL introduces additional complexities due to its diverse stakeholders, actors, information exchanges, communication infrastructures, and attack surfaces, necessitating an assessment of trustworthiness. FL also presents unique challenges related to architectural designs, privacy-preserving standards, fairness, and explainability beyond the ML/DL models [12]. Currently, there is a need for more research on trustworthiness pillars, metrics specific to FL, and tools to assess the trustworthiness of FL models. Moreover, there is a need for solutions that seamlessly integrate into existing FL frameworks to compute the trustworthiness of FL models.

To cover the previous literature gaps, the work at hand presents the following contributions:

- The creation of a novel taxonomy with the most relevant pillars, notions, and metrics to compute the trustworthiness of FL models. To create such taxonomy, crucial aspects used to evaluate the trustworthiness of classical and federated ML/DL models were studied, analyzed, and compared. More in detail, the following six pillars and more than 30 metrics were identified as the main building blocks of the taxonomy: privacy, robustness, fairness, explainability, accountability, and federation. All pillars present novel metrics compared to the literature.
- The design and implementation of FederatedTrust, an algorithm quantifying the trustworthiness of FL models based on the pillars, notions, and metrics presented in the proposed taxonomy. FederatedTrust computes global and partial trustworthiness scores by aggregating metrics and pillars dynamically and flexibly depending on the validation scenario. An algorithm prototype has been implemented in Python (available in [13]) and deployed in a well-known FL framework called FederatedScope. Then, three experiments classifying hand-written digits in a cross-device FL context using the FEMNIST dataset were performed to assess the trustworthiness of FL models. Experiments introduced differences in the number of participants in the federation, training rounds, sample rates, and countermeasures against attacks. Finally, two experiments leveraged the N-BaIoT dataset to show how different design choices can impact the trustworthiness score in a cybersecurity use case.

The remainder of this paper is structured as follows. Section 2 contains findings from the literature review on trustworthy FL. Section 3 identifies and presents a detailed analysis of the following six trustworthy FL pillars and their metrics: robustness, privacy, fairness, explainability, accountability, and federation. Section 4 presents the design detail of the proposed algorithm, while Section 5 focuses on its implementation and deployment on a real FL framework. Section 6 validates the algorithm in a use case and presents the results of the performed experiments. Finally, Section 7 provides conclusions and future work.

2. Related work

This section reviews existing solutions focused on trustworthy FL and well-defined pillars relevant to trustworthy AI, such as robustness, privacy, fairness, explainability, and accountability [8]. It is important to mention that a large body of literature on trustworthy centralized ML/DL has emerged in recent years. However, trustworthy FL is a nascent research field.

FedEval [14] is the closest solution to the one proposed in this paper because it combines several aspects relevant to trustworthy AI. More in detail, FedEval is an open-source framework for FL systems that evaluates the accuracy, communication, time efficiency, privacy, and robustness of FL models to compute their trustworthiness level. Regarding accuracy, it compares the performance of FL and the centralized training. The communication metric relies on the number of communication rounds and the total amount of data transmission during training. The time efficiency metric measures the overall time needed for getting a converged model. The privacy metric considers state-of-the-art inference attacks and their impact. Finally, robustness metrics compute the performance of different aggregation mechanisms under non-IID data. Another solution focused on quantifying the trustworthiness of AL models is presented in [15,16]. The authors propose an extensible, adaptive, and parameterized algorithm to quantify the trustworthiness level of supervised ML/DL models with tabular data according to their robustness, explainability, fairness, and accountability. The main limitation of this work is that it is not suitable for FL models.

Privacy is the central point of FL since its main objective is to protect data privacy among the federation participants. Therefore, it is crucial to preserve and quantify data privacy effectively to trust FL model predictions. In this context, several works and techniques can be categorized into three main families: (i) encryption-based, (ii) perturbation-based, and (iii) anonymization-based. In the encryption-based category, [17] designed a privacy-preserving protocol against a semi-honest adversary by combining Ternary Gradients with secret sharing and homomorphic encryption. [18] designed a secure aggregation by leveraging secure multiparty computation to perform sums of model parameter updates from individual users' devices. In the perturbation category, [19] demonstrated that global differential privacy offered a strong level of privacy when protecting sensitive health data in an FL scenario. In addition, [20] proposed a procedure using differential privacy to ensure that a learned model does not reveal whether a client participated during decentralized training. Unlike perturbation-based techniques, anonymization-based techniques can provide privacy

Table 1
Comparison of related work.

Solution (year)	FL	Privacy	Robustness	Fairness	Explainability	Accountability
[17] (2020), [18] (2017)	✓	Encryption	✗	✗	✗	✗
[19] (2019), [20] (2017)	✓	Perturbation	✗	✗	✗	✗
[21] (2021)	✓	Anonymization	✗	✗	✗	✗
[22] (2020)	✗	✗	Poisoning and Inference	✗	✗	✗
[24] (2019), [25] (2021)	✗	✗	Poisoning	✗	✗	✗
[26] (2022)	✗	✗	Assessment	✗	✗	✗
[27] (2021), [28] (2022)	✓	✗	✗	Clients Contributions	✗	✗
[29] (2021), [30] (2020), [31] (2020)	✓	✗	✗	Client Selection	✗	✗
[32] (2019), [33] (2017), [34] (2022)	✓	✗	✗	✗	Feature Importance	✗
[35] (2022)	✓	✗	✗	✗	✗	FactSheet
[36] (2021), [37] (2020), [38] (2019)	✓	✗	✗	✗	✗	Blockchain
[14] (2020)	✓	Inference Evaluation	Aggregation	✗	✗	✗
[16] (2022), [15] (2023)	✗	✗	Poisoning	Data Distribution	Features and Algorithms	Factsheet
FederatedTrust (this work)	✓	Perturbation	Poisoning and Inference	Federation	Algorithms and Statistics	Factsheet

defense without compromising data utility. In this category, to measure privacy in FL, [21] proposed a novel method to approximate the mutual information between local gradient updates and batched input data during each round of training.

Although FL provides a first level of data protection by not sharing training data, the FL paradigm is still vulnerable to adversarial attacks affecting data privacy (inference attacks) and model performance (poisoning attacks) [22]. Therefore, robustness is an important pillar to consider in trustworthy FL. To improve the FL robustness against adversarial attacks affecting data privacy and model performance, the authors of [22] proposed the usage of differential privacy, robust aggregation, and outlier detection as primary defenses. [23] showed empirical evidence that differential privacy could defend against backdoor attacks and mitigate white-box membership inference attacks in FL. [24] introduced Adaptive Federated Averaging (AFA), a Byzantine-robust FL algorithm that detects and discards malicious client updates at every iteration by comparing the similarity of the individual updates to the one for the aggregated model. From a similar perspective, [25] proposed Robust Filtering of one-dimensional Outliers (RFOut-1d), a new FL approach resilient to model-poisoning backdoor attacks. Finally, regarding mechanisms and metrics able to evaluate the robustness of FL, the literature has focused on methods to quantify robustness once the model is trained. Some examples of these metrics are loss sensitivity, empirical robustness, or CLEVER score [26].

Fairness is another essential pillar for trustworthy FL, as multiple parties contribute data to the model training and are eventually rewarded with the same aggregated global model. In this sense, [27] provided a survey and overview of fairness notions adopted in FL-based solutions. The notions include (i) accuracy parity, which measures the degree of uniformity in performance across FL client devices; (ii) selection fairness, which aims to mitigate bias and reduce underrepresentation and never representation; and (iii) contribution fairness, which aims to distribute payoff proportionately to the contributions of clients. Apart from that, the authors of [29] proposed GIFAIR-FL. This framework imposed group and individual fairness to FL settings by penalizing the spread in the loss of clients to drive the optimizer to fair solutions. FairFL [30] is another solution that facilitated fairness across all demographic groups by employing a Multi-Agent Reinforcement Learning-based scheme. This approach solved the fair classification problem in FL by enforcing an optimal client selection policy on each client. The authors of [31] proposed a long-term fairness constraint that considered an expected guaranteed chosen rate of clients that the selection scheme must fulfill. Finally, [28] proposed the Completed Federated Shapley Value (ComFedSV) to evaluate data owners' contributions in FL based on solving a low-rank matrix completion problem.

Even with active research on XAI, there are challenges specifically for FL models, as most client data are private and cannot be read or analyzed. In this context, some explainability methods, such as feature importance, reveal underlying feature information from other parties. For horizontal FL models, since clients share the same feature space, the authors of [32] suggested that predictions could be explained by calculating the Shapley value of each feature using the definition provided by [39]. For vertical FL models, their work proposed a variant version of SHAP [33] by combining the participant features into individual united feature spaces. Therefore, participants do not get information about the features of other participants. Another solution called EVFL was proposed in [34], where authors presented a credible federated counterfactual explanation method to evaluate feature importance for vertical FL models and minimize the distribution of the counterfactual and query instances in the client party.

Even though FL models are promising regarding data privacy, they require transparency and accountability, as in the case of classical centralized ML/DL models. In this sense, IBM introduced the Accountable FL FactSheet framework [35] that instruments accountability in FL models by fusing verifiable claims with tamper-evident facts. The framework requires different actors, like the project owner, data owner, or aggregator, to log claims about the various processes occurring during the FL training lifecycle. They also expanded the IBM AI FactSheet 360 [40] project to account for the complex model compositions of FL. Finally, [36,37], and [38] incorporated blockchain and smart contracts to add different auditing and accountability mechanisms to FL models by leveraging the immutability and decentralized trust properties of the blockchain.

Table 1 compares the solutions covering each one of the FL trustworthiness pillars. In conclusion, this section has reviewed the work done in each dimension or pillar relevant to trustworthy AI. As can be seen, there is a lack of solutions quantifying the trustworthiness level of FL models by combining the different pillars identified by related work. Most solutions focus on isolated pillars and improving the pillar aspect instead of assessing or quantifying its status, which is needed before deploying countermeasures. In addition, there is no solution dealing with aspects related to the architectural design of the federation.

3. Trustworthy FL: Pillars, notions, and metrics

This work identifies, introduces, and explains for the first time (to the best of our knowledge) the most relevant pillars, notions, and metrics for trustworthy FL. Additionally, it proposes a novel pillar called Federation, which has not been considered in the literature. This pillar captures complex compositions and designs of FL architectures to compute their trustworthiness.

Fig. 1 presents a taxonomy describing the pillars, notions, and metrics relevant to trustworthy FL. Under each pillar, the major aspects defining it are grouped into notions. Under each notion, the specific metrics that can be calculated to quantify the level of utility towards trustworthiness are defined. This taxonomy serves as the baseline for evaluating and assessing the trustworthiness level of FL models. The mathematical symbols employed in this section are summarized in Table 2.

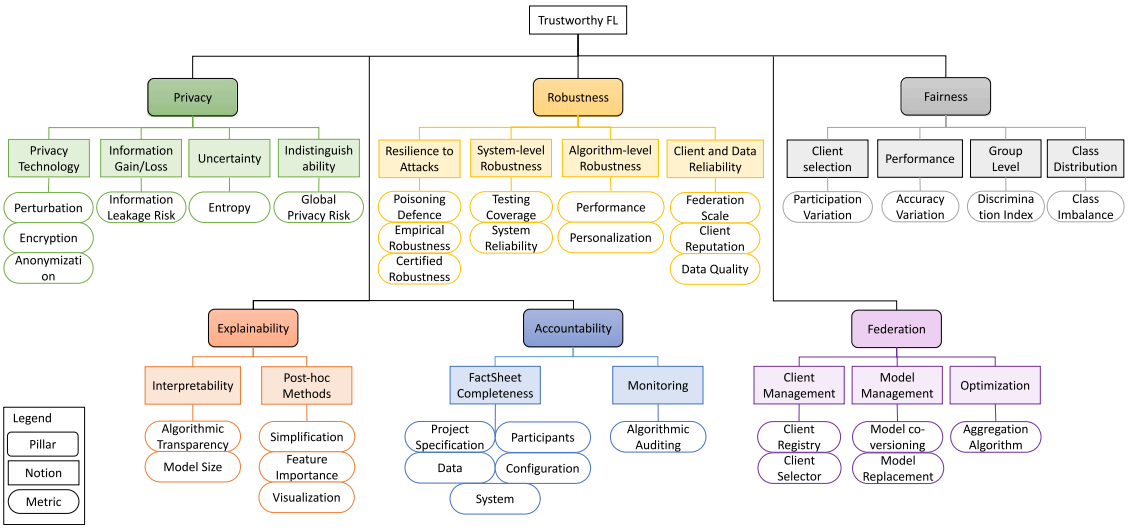

Fig. 1. Trustworthy FL taxonomy.

Table 2
Math notations employed in the document.

Symbol	Meaning
x	Model update
$p(x)$	Probability of x satisfying a condition
$H(x)$	Entropy of x
$K()$	Randomized K function
w_t	Current model
w_{t+1}	Next model
σ	Standard deviation
μ	Participation rate
CV	Coefficient of variation
$\boldsymbol{\Phi}$	Discrimination index
$F1(X)$	F1 score of samples in X

3.1. Privacy

Data privacy is the most prominent driving force for the development of FL. Therefore, FL models must preserve data privacy within their lifecycle to gain participants' trust. Even though FL already elicits a degree of data privacy by definition, assumptions have to be made about the integrity of the multiple actors and parties making up the federation. If participants are honest, but the aggregating server is honest but curious, prevention of information leakage from model parameter exchanges must be in place. If all federation members are honest but curious, then information leakage prevention should focus on secure communication. Moreover, information can still be leaked by malicious attacks from outside.

To cover these aspects, the first notion of this pillar is the usage of privacy-preserving approaches to add resilience to privacy attacks. The second focuses on metrics measuring information gain/loss based on the information leakage risk derived from the FL process. Finally, two additional notions arise from the probability of knowledge inference from the client updates.

Privacy-preserving approaches. This notion considers the following main approaches to protect data privacy in FL.

- *Perturbation*. It adds noise to raw data, so the perturbed data are statically indistinguishable from the raw data. The most widely adopted schemes are local and global Differential Privacy [41]. The global approach adds noise to the parameters shared after the local model is trained, while the local one adds noise to each client data sample used for training.
- *Encryption*. It encrypts the model parameters of each participant before sharing it. The most widely adopted scheme is Homomorphic Encryption, where the aggregation server does not decrypt the parameters to aggregate them into the global model [42]. Another popular scheme is Secure Multiparty Computation (SMC) [43], which allows participants to collaboratively calculate an objective function without revealing their data.
- *Anonymization*. The most widely adopted schemes in this approach are k-anonymity and l-diversity. K-anonymity is satisfied if each sample in the dataset cannot be re-identified from the revealed data of at least $k-1$ clients [44]. L-diversity extends on k-anonymity so that the sensitive attributes of the samples are protected.

Information Gain/Loss. This notion focuses on measuring the amount of privacy lost by participants or the amount of information gained by adversaries due to leakage or disclosure of information [45].

- *Information Leakage Risk*. In FL, the gradients can carry enough information for adversaries to reconstruct the model or infer original data. H-MINE [21] is a hierarchical mutual information estimation metric to measure the mutual information between the high-dimensional gradients and batched input data. The amount of leaked information (counting the information items disclosed by a system), relative entropy (measuring the distance between two probability distributions), or mutual information (quantifying the shared information between two random variables) are other methods to compute this metric.

Uncertainty. The uncertainty of data estimation by adversaries makes a difference in the level and effectiveness of a data privacy breach. Under normal circumstances, high uncertainty estimation correlates with high privacy. This notion considers metrics to measure uncertainty, most of which are based on entropy.

- *Entropy*. In general, entropy measures the uncertainty in predicting the value of a random variable. In FL, an adversary may be interested in identifying which data samples belong to a particular client or organization participating in the training [46]. Eq. (1) calculates the entropy of X, where X is a participating client and $p(x_i)$ is the estimated probability of this client being the target.

$$priv_{ENT} \equiv H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \tag{1}$$

Indistinguishability. Some adversaries are interested in distinguishing between two data samples of interest. In general, privacy is high if the adversary cannot distinguish between two outcomes of the model.

- *Global Privacy Risk*. It enables indistinguishability in the training data by adding random noise. It is a formal statistical guarantee that any disclosure is equally likely whether a sample is in the dataset or not [47]. Eq. (2) describes the formal differential privacy proof for privacy mechanism. A randomized function K checks if the output of random variables for two datasets D_1, D_2 that differ at most to some extent (e.g., one row of data), differ by at most $exp(\epsilon)$:

$$priv_{DP} \equiv \forall S \subseteq Range(K) : p(K(D_1) \in S) \leq \exp(\epsilon) \cdot p(K(D_2 \in S)) \tag{2}$$

3.2. Robustness

Robustness is one of the three foundations of trustworthy AI, together with lawfulness and ethics, as defined by the European Commission AI ethics guidelines [7]. AI systems must be technically robust to ensure that they are not vulnerable to malicious use or bring harm to humans. In this sense, the literature has considered three different notions, and this work proposes a new one to assess the robustness of FL models. According to the literature [5], FL models must be resilient to adversarial attacks adding perturbations or erroneous inputs. Secondly, the hardware and software of participants training and deploying FL models must be robust to avoid cyberattacks [48]. Thirdly, FL algorithms performance and customization must be reliable and robust [49]. Last but not least, this work proposes adding client and data reliability as a novel notion since reliable clients and data increase the probability of robust and reliable FL models. More details of each notion and its metrics are provided below.

Resilience to Attacks. FL models are susceptible to poisoning attacks affecting model training and robustness. Poisoning attacks can be categorized into data poisoning and model poisoning. In data poisoning, the integrity of the training data is compromised. Common methods are flipping or permuting the labels and inserting backdoor patterns or perturbations into the training data. Model poisoning attacks have a broader range, and the goal is to manipulate the training procedure. In FL, this could be gradient manipulation or model update poisoning attack, which is performed by corrupting the updates of a participant directly or during model exchanges. Therefore, this notion is usually evaluated by first checking whether the FL model is equipped with any defense mechanism and, if so, empirically verifying the model defense capabilities against representative attacks.

- *Poisoning Defense*. It focuses on providing defense mechanisms against poisoning attacks. On the one hand, *Byzantine-resilient Defense* is a popular defense mechanism where various robust aggregation methods have demonstrated their effectiveness in detecting malicious client updates and reducing their impact [12]. On the other hand, *Outlier Detection* explicitly identifies and denies negative influence as a more proactive defense against poisoning attacks. Existing approaches include rejecting updates with too large error rates, measuring the distribution of parameter updates, or looking for dormant neurons that are not frequently activated [50].
- *Empirical Robustness*. If small changes in the input data cause significant output deviations, adversarial perturbations can be used to generate undesired outcomes. It can be measured by implementing a model poisoning attack, a typical poisoning attack altering some local data (data poisoning), or the gradients (model poisoning) [51]. A mathematical explanation of how a model replacement attack works is provided by [50] and illustrated as follows. It is assumed that at least one compromised client could apply the backdoor patterns to perform a model replacement attack. Eq. (3) describes how a poisoned model update is generated. Where w_t is the current model and N is the number of clients. The global model w at time t+1 is an averaged mean of model updates from N clients at time $t+1$, and the goal is to replace the global model w at $t+1$ with the attacker's model x_{atk}:

$$x_{atk} = w_{t+1} = w_t + \frac{1}{N}\sum_{i=1}^{N}(x_{t+1}^i - w_t) \tag{3}$$

Now, x_{t+1}^m is the update from the malicious client m at time $t+1$, then rearranging Eq. (3), we have:

$$x_{t+1}^m = N \cdot x_{atk} - N \cdot w_t - \sum_{i=1}^{N-1}(x_{t+1}^i - w_t) + w_t \tag{4}$$

Assuming $\sum_{i=1}^{N}(x_{t+1}^i - w_t) \approx 0$ as explained in [52], we have the attacker's update be simplified as the following:

$$x_{t+1}^m = N \cdot (x_{atk} - w_t) + w_t \tag{5}$$

- *Certified Robustness*. It defines the least amount of perturbation required for the attacker to succeed (change the model prediction). In other words, a model is certifiably robust for an upper-bounded amount of perturbations. The CLEVER (Cross Lipschitz Extreme Value for nEtwork Robustness) metric [53] using the local Lipschitz constant for neural networks is one of the most well-known metrics. CLEVER is an attack-agnostic derivation of the universal lower bound on the minimal distortion required for a successful attack.

System-level Robustness. This notion deals with hardware and software robustness and must be considered in production environments with FL models where proper software development and deployment standards are needed. Testing coverage and system reliability are the main metrics for this notion.

- *Testing Coverage*. It guarantees that clients adhere to the federation requirements, such as broadcasting messages to distributed clients, client selection, and model aggregation. This metric can be implemented using different methodologies, such as robust system delivery, ranging from code review, unit testing, integration testing, system testing, and acceptance testing [54].
- *System Reliability*. It deals with the probability and the duration of time of failure-free operation [55]. It is measured based on error, timeout, and dropout rates. The error rate is normally calculated as the number of failures over a given amount of time. The maximum timeout measures the time the server should wait to receive client model updates. Finally, the dropout rate is the number of clients leaving the federation (to speed up convergence, optimize resources, or due to errors) divided by the total amount of clients. A high dropout rate can also indicate a less reliable FL system.

Algorithm-level Robustness. This notion deals with the performance and generalization of FL algorithms. Performance is widely used to showcase how good an ML/DL model is. However, good performance does not necessarily imply generalization. Generalization is a major challenge in FL because each client has different local data heterogeneity, and the aggregated global model might not be able to capture individual data patterns. Non-IID data [56] may cause severe learning divergence to parametric models. Therefore, metrics are desirable to measure the performance and generalization of FL [57].

- *Performance*. In FL, there are two approaches to measuring performance. One is reserving a set of test or validation data on the server side for global model evaluation. The other is by evaluating test accuracy at each client's device and aggregating the test accuracy. This metric considers these two aspects.
- *Personalization*. Several methods have been proposed for this category of metrics. Regularization is one of the personalized FL approaches aiming to minimize the disparity between global and local models. Multi-tasking Learning is a method where multiple learning tasks are solved simultaneously. This benefits FL models

since multiple organizations participating in the FL models can train their personalized models to achieve better performance. For example, MOCHA [58] generates separated but related models on local client devices using data of related tasks. Clustering is another personalization approach where clients are allocated into clusters based on their similarity [59]. Personalized Layer is a metric for neural network models that introduces custom layers for each client in the model. FedPer [60] is an example that uses the base layers as the shallow layers and personalized layers as the deep layers while keeping everything else the same as the baseline FedAvg algorithm.

Client and Data Reliability. This novel notion of robustness deals with client and data reliability. In this sense, metrics like federation scale, clients' reputation, and data quality are essential for robust and trustworthy FL models.

- *Federation Scale*. The number of clients impacts the reliability of the system. In FL, the number of clients determines the number of devices, network connections, and model parameters that must be considered to train FL models. The higher the number of clients, the network stability, computation power, and availability of clients, the higher the reliability of the FL system [61].
- *Client Reputation*. In [49], a novel reputation metric was proposed, and a subjective logic model was used to calculate the reputation score for each client interaction. After each training iteration, the server uses a poisoning attack detection scheme and the elapsed time to determine if the local update from the client is reliable. Reliable updates are treated as positive interactions and improve the reputation value, and vice versa. The client is treated as malicious and unreliable when the reputation value is below a threshold.
- *Data Quality*. It compares, in each training round, the local updates with the current global model to see if the new local updates are better or worse than the global model [62]. If a round of training at one client improves the performance of the global model, then the data from that client has a high-quality score.

3.3. Fairness

One primary source of unfairness in AI is coming from the data. In FL, different clients might contribute with heterogeneous amounts and quality of data. When data are not representative of the wider federation, participant selection bias is propagated into the model. The selection bias can be manifested by feature distribution skew or label distribution skew [12], both of which are major challenges in FL. Therefore, Client Selection Fairness is the first notion of this pillar. Apart from that, in fair AI [63], fairness is broken down into group-level and individual-level fairness. Group-level fairness means that members of a particular group should not be subject to discrimination. Individual-level fairness means that similar individuals should receive similar treatment regardless of their membership group. These requirements do not change when moving to FL. Therefore, the Group-level Fairness notion deals with the group-level aspects, and the Performance Fairness and Class Distribution notion covers individual-level ones. More in detail, Performance Fairness ensures that each client's reward should be proportional to their data contribution. Finally, Class Distribution looks at the label imbalance in the dataset of each participant.

Client Selection. Usually, in FL, only a fraction of clients are selected to participate in the training process of each round. In practice, there are several criteria for selection, such as availability, network speed, computation power, or battery level, among others. In the worst case, clients in regions with low network speed or models with weaker computation power could never be selected and represented in the training data. Therefore, under-representation is a source of selection bias that could lead to unfair model outcomes.

- *Participation Variation*. In statistics, the Coefficient of Variation (CV) measures how far the data values from a set are dispersed from the mean. This metric analyzes the distribution of participation rates among all clients. With similar clients, the more dispersed the distribution of participation rate, the less fair the client selection mechanism is, and vice versa [64]. Eq. (6) calculates the CV in the participation rate, where σ represents the standard deviation, and μ represents the average of the participation rate.

$$CV = \frac{\sigma}{\mu} \tag{6}$$

Performance. Even though performance fairness already exists in AI, in FL, another grouping needs to be considered, which is client-level fairness. [64] suggests a definition that a model provides a more fair solution to the FL learning objective on the clients if the performance is more uniform than that of another model.

- *Accuracy Variation*. This metric considers the test accuracy as a representation of performance. More in detail, the aggregated global model and test data from each client are used to measure the test accuracy. The more uniform the test accuracy among the clients, the more fair the model performance is.

Group-level. Evaluating and mitigating demographic bias in FL is more difficult than in centralized learning. First, raw training data, labels, and sensitive demographic information of each participant cannot be revealed. Second, in centralized learning, all the training data can be analyzed and pre-processed to balance the class distribution before training. In contrast, different clients pre-process their data locally in FL, so additional mechanisms are needed to adjust the global data distribution in a secure and protected manner.

- *Discrimination Index*. The discrimination index metric [30] measures the difference in the F1 score between a particular demographic group (σ) and the rest of the population. The metric value falls between $[-1, 1]$, where the ideal discrimination index should be as close to 0 as possible. Calculating this index globally would reveal sensitive attributes and statistics of the demographic group in the client data. Eq. (7) calculates the discrimination index, where $F1(X_v^+)$ represents the F1 score of all the samples from the protected group and $F1(X_v^-)$ represents the F1 score of the rest of the samples.

$$\Phi_\sigma = F1(w(X_\sigma^+)) - F1(w(X_\sigma^-)) \tag{7}$$

Class Distribution. Analyzing the class distribution of the training data used in an ML/DL model provides insight into whether the data samples are selected properly to reflect a fair representation of the wider group. In theory, this should also apply to FL models, except that in practice, this often requires access and analysis of the raw training data, which goes against data privacy. However, in [65], an estimation scheme was proposed to reveal the class distribution without accessing raw data. Furthermore, secure aggregation can also be considered for aggregating class distribution information among clients.

- *Class Imbalance*. Two approaches can be used to evaluate the class imbalance. One is the estimation scheme using a well-balanced auxiliary dataset and the gradients of a neural network model [65]. Another general way to get the class imbalance information in FL is to ask every client to submit their class distribution to the server. The secure aggregation method combines all the class distributions into one unified distribution. Then, the coefficient of variation of the class distribution can be used to calculate the level of variation of the class sample sizes to determine the class imbalance.

3.4. Explainability

Nowadays, the explainability of ML/DL/FL models is an open challenge. In this context, AI guidelines demand transparent AI processes, with the capabilities and purpose of AI systems openly communicated and decisions explainable to those directly and indirectly impacted. Transparency is often expressed as interpretability, which is often wrongly mistaken as interchangeable with explainability. Interpretability is the first notion of this pillar and can be described as a passive characteristic of a model referring to the level of understandability for humans. In contrast, explainability is the ability to describe AI systems' technical processes. Interpretable models can be explained by analyzing the model itself, but Post-hoc methods can enhance their interpretability, being the second notion of this pillar for non-interpretable models. For FL, since ML/DL models are also used in the training process, the requirement of explainability for the algorithmic model also applies. However, privacy constraints make accessing and analyzing raw data difficult.

Interpretability. This notion combines the algorithm transparency and the model size to evaluate the FL model interpretability. As already identified in the literature, some ML/DL models are interpretable by design, and some are not. In addition, model size is also widely used as a metric of interpretability. More details about these two metrics are provided below.

- *Algorithmic Transparency*. A model is considered transparent if it is understandable by itself. This definition could be very subjective to different levels of intellectual grasp. However, algorithmically transparent models must first be fully explorable through mathematical analysis and methods. Then, the assessment considers model complexity (in terms of the number of variables and interactions) and decomposability (in terms of the interpretability of each component of the model) [66]. In summary, the recognized interpretable models are linear regression, logistic regression, decision trees, decision rules, k-nearest neighbors (KNN), and Bayesian models. Even within the interpretable models, the level of interpretability slightly varies. The non-interpretable models include tree ensembles, support vector machines (SVM), multi-layer neural networks (MNN), convolutional neural networks (CNN), and recurrent neural networks (RNN).
- *Model Size*. Different algorithms have diverse definitions of model size. For instance, it could be the number of decision rules, the depth of a decision tree, the number of features in a linear/logistic regression model, or the number of trainable parameters in a neural network [67]. The larger the model size, the harder it is to understand and explain the causal relationship between input and output.

Post-hoc Methods. The three most common Post-hoc Methods are simplification, feature importance, and visualization [66]. This notion complements the previous and contributes to assessing the explainability of interpretable and non-interpretable FL models.

- *Simplification*. The idea of simplification lies in reducing the number of architectural elements or parameters in a model. One of the main techniques applied for model simplification is knowledge distillation [68].
- *Feature Importance*. Most model explanation methods can be directly used for horizontal FL because all participants share the full feature space in their local data [32]. However, exposing feature information to the server for calculating feature importance score is not ideal. For vertical FL, methods like SHAP cannot be directly used because parties do not share the full feature space. The variant version of SHAP for Vertical FL combines one party's features into a federated feature space when referenced by another party for the feature importance calculation [32].
- *Visualization*. A lifecycle dashboard that visualizes server information, from client registration to training, validation, and deployment, was proposed in [69]. The dashboard shows which clients participated in which training round and the model current status.

3.5. Accountability

Accountability is another of the seven critical requirements of Trustworthy AI defined by the EU guidelines [7]. The first main notion about accountability is FactSheet Completeness. IBM Research was the first to propose a document called FactSheet in charge of recording facts about the overall ML/DL pipeline [70]. Another important notion of accountability is Monitoring. Even with complete and detailed documentation, every participant has to make an effort to ensure that FL models are built strictly following the intended architecture, development, and deployment processes.

FactSheet Completeness. IBM extended the FactSheet approach to enable accountability in FL [35]. The accountable FL FactSheet template is a comprehensive document that contains meta-information about the project, participants, data, model configurations, and performance. Since FL is more complicated in architecture and more privacy-preserving, the FactSheet should contain information about the additional layer of configurations and avoid sensitive information about participating clients. Below, more details about the aspects considered in FactSheets are provided.

- *Project Specification*. This section of the FactSheet documents the project overview, purpose, and background. The overview explains what the project is about. The purpose details the goals, and the background elaborates on the relevant information and knowledge motivating the project.
- *Participants*. It contains the participants of the FL process. The template considers the participants' names and their organization unit names for identity verification.
- *Data*. It documents the information regarding the data used in the FL process. Two aspects are included: data provenance and pre-processing procedures. Data provenance helps trace the data origin and flow to access validity and reputation. Before training the model, pre-processing steps can tell how the raw data have been handled.
- *Configuration*. It deals with the information about the FL model configuration. First, it contains the type of optimization algorithm and the ML/DL model. Then, it indicates the global hyper-parameters the aggregator uses, for instance, the number of rounds, the maximum timeout, and the termination accuracy. Lastly, it contains the local hyper-parameters used by the trainer at each client, such as the learning rate and the number of epochs.
- *System*. This FactSheet section documents the system information for the learning process. It includes the average time spent on training, the model size, the model upload, and the download speed in bytes. This information indicates the number of resources expected to be utilized.

Monitoring. For AI systems, algorithmic auditing is a range of approaches to auditing algorithmic processing systems. The evaluation of the metrics under this notion uses a checklist-based approach, verifying if the FL system employs any external or internal algorithmic auditing.

- *Algorithmic Auditing*. This metric can be implemented in different ways. For example, it could be functional testing, performance testing, user acceptance testing, etc. It could also be system anomaly or attack monitoring. Some organizations even invite or hire external hackers to find vulnerabilities as a monitoring measure. The SMACTR framework [71] is a good option for a more systematic approach. It is an internal auditing framework with five stages: scoping, mapping, artifact collection, testing, and reflection. Each stage yields a set of documents that form a comprehensive audit report.

3.6. Federation

The major management challenges of FL deal with communication, efficiency, resource limitation, and security. In this context, it is very challenging to coordinate the learning process of thousands of clients while ensuring model integrity and security. Global models might converge slowly due to heterogeneous client data. Inconsistent clients, networks, and limited resources might cause clients to drop out, and training failures could impact model quality. In conclusion, although there is active research in FL algorithms, there is still a lack of research and guidelines on the architectural design of FL systems. In this sense, the main notions of this pillar are Client and Model Management, which considers how client and model information is administrated in the system, and Optimization Algorithm, which may impact the model performance and robustness.

Client Management. This notion proposed a Client Registry, where participants can register themselves for training, and a Client Selector to filter eligible clients for training. More details are provided below.

- *Client Registry*. It enables the system to manage client connections and track the status of all client devices. The proposed design pattern maintains the client registry in the central server for the client–server architecture. The server sends a request for information along with the initial local model to the clients when they first connect to the system. The information requested includes device ID, connection up and down time, or device computation power storage.
- *Client Selector*. It optimizes resource usage and reduces the risk of client dropout and communication latency. The proposed design pattern also maintains the client selector in the central server where the selection occurs. Before each round of training, the client selector actively selects a certain number of clients for the training according to predefined criteria to reduce convergence time and optimize the model performance.

Model Management. In a distributed learning process like FL, multiple rounds of training and aggregation of models generate numerous local model updates and aggregated global models during the process. Without recording the local and global intermediary models, there is neither traceability nor fallback when something goes wrong in the training process. A model co-versioning registry and replacement can help trace the model quality and improve system accountability. Blockchain has been proposed by recent works of the literature to populate individual models in an immutable and transparent manner, mitigating some attacks affecting FL [72].

- *Model Co-versioning*. It aligns the local model versions with their corresponding aggregated global models. It can be a registry where local model versions are stored and mapped to the associated global models. With this registry, model updates and aggregations do not always have to be synchronous because the server can refer to the mapping to perform asynchronous aggregations. Another advantage is that it allows early stopping if a model converges before the specified number of rounds.
- *Model Replacement*. It detects the global model performance dropping below a certain threshold level. For that, it compares the global modal performance in all clients to see if the performance degradation is a global event. The new global model training task is triggered if the degradation is global and persistent.

Optimization. According to the goal and the context of FL models, the choice of an optimization algorithm can impact the model performance. Therefore, various studies have conducted performance benchmarking of FL optimization algorithms [73]. This benchmarking comparison serves as a reference for this metric.

- *Aggregation Algorithm*. FedAvg is considered the baseline aggregation algorithm, but other optimization algorithms have been proposed as an extension for various purposes [73]. In addition, the aggregation can be done in a centralized or decentralized manner. Decentralization reduces network bottleneck, single point of failure, and trust dependencies while increasing network overhead in some cases [74]. This metric considers these two aspects to evaluate the trustworthiness level of the aggregation task.

4. FederatedTrust algorithm design

This section details the design of *FederatedTrust*, the algorithm proposed to quantify the trustworthiness level of FL models according to the pillars, notions, and metrics presented in Section 3. To the best of our knowledge, it is the first attempt to evaluate the trustworthiness of FL models.

Prior to delving into the specifics of the algorithm, it is crucial to mention that this work operates under the assumption that the central server, which aggregates the models of the clients, is honest, maintains data integrity, and is overseen by a dependable system administrator. Therefore, the server does not maliciously interfere with the trust calculation process. In addition, the following functional requirements (FR), non-functional requirements (NF), and privacy constraints (PC) have been considered before designing and implementing the proposed algorithm.

FR-1: Each of the six trustworthy FL pillars must be represented in the algorithm, meaning that at least one metric from each pillar must be considered in the final score.
FR-2: The final trustworthiness score must be a combination of the trustworthiness scores from all notions and pillars.
NF-1: The algorithm should add minimal computation overhead and complexity to the server, participants, and FL model.
NF-2: The algorithm should be modular and configurable.
PC-1: The algorithm must not store any sensitive data from the FL model.
PC-2: The algorithm must not leak or share any sensitive data from clients, the server, and the FL model with third parties.
PC-3: The metrics calculations can occur at the client's local devices, the central server, or collaboratively between both.
PC-4: When metrics are calculated collaboratively between clients and the server, the computation should be performed securely and privately if the individual client metrics contain sensitive information.

Once assumptions and requirements are defined, Fig. 2 shows the overview of the FederatedTrust algorithm design. First, the proposed algorithm considers the following input sources to compute the trustworthiness of FL models.

- **FL Model**. The FL model trained in a collaborative and privacy-preserving way between the federation participants. This input contains information about the model configuration and model personalization.
- **FL Framework Configuration**. The configuration parameters of the tool implementing the protocol needed to train and evaluate the FL model. This input contains information about the number of clients, the client selection mechanisms, the aggregation algorithm, and the model hyperparameters.
- **FactSheet**. As mentioned in Section 3, It provides essential details for the accountability of the training process, federation, and the individuals involved. Specifically, it encompasses information about the overview of the task to solve, data origin, techniques used for pre-processing, and the incorporation of differential privacy mechanisms.

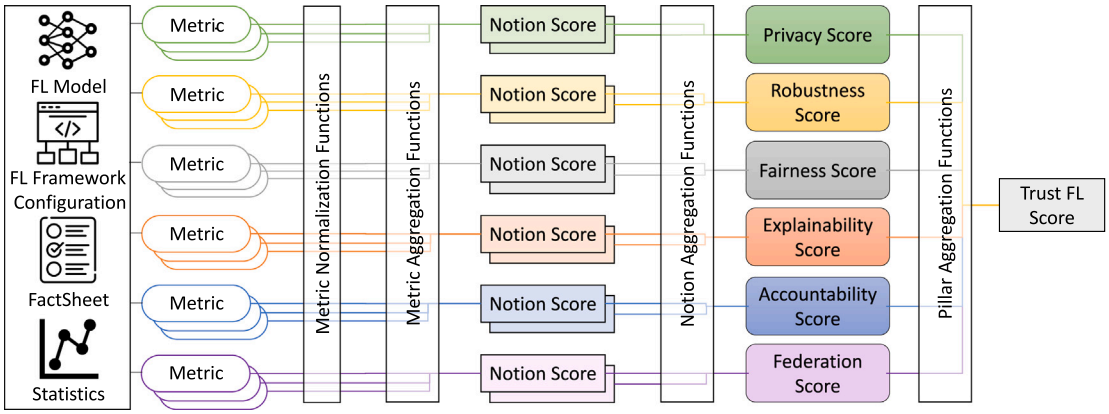

Fig. 2. Design of the FederatedTrust algorithm.

- **Statistics**. The statistical information extracted from the training dataset of each client and its model performance. It does not contain sensitive information. This input contains information about the client class balance, client test performance loss, client test accuracy, client clever score, coefficient of variation for client feature importance, coefficient of variation for client test accuracy, coefficient of variation for client participation rate, coefficient of variation for client class imbalance, client average training time, average model size, average upload bytes, and average download bytes.

These input sources are used to compute the metrics indicated in Section 3, which output values are then normalized to have a common range of values. It is important to mention that each metric can consider different input sources and can be calculated in a different phase of the FL model creation process (pre-training, during-training, or post-training) and by a different actor of the federation (client or server). These details regarding when and who computes each metric are provided later. Once the normalized outputs of metrics are calculated, they are weighted and aggregated to compute one score per notion (see Section 3 for more information about notions). Each pillar has one or more notions calculated according to predefined but configurable weights per metric. Therefore, the same process is repeated to obtain the pillar scores from weighting and aggregating notion scores. Finally, the final trust score of the FL model is a configurable combination of the pillar scores. The implementation details of the previous steps are provided in Section 5.

As indicated, some metrics of the algorithm are calculated during the training phase of the FL model. Therefore, it is necessary to integrate the proposed algorithm into an FL framework in charge of creating FL models. In this context, Fig. 3 shows the interactions between the main actors involved in the computation of the trustworthy level of FL models. As it can be seen, the central server of the federation hosts (i) the *Aggregator*, in charge of combining the parameters of the clients' models to create the FL model, (ii) the *FactSheet*, accounting the most important aspects of the FL project and participants federation, (iii) the *FL Framework Configuration*, detailing aspects of the framework, and (iv) the *FederatedTrust* algorithm, computing the FL model trust score. To compute the trust score, during the pre-training phase, the content of the FactSheet and the FL framework configuration (detailed in Section 5) is sent to the FederatedTrust algorithm (steps 1 and 2 in Fig. 3) and metrics depending on them are calculated as previously explained. Then, the model training process starts with the server sharing the type of model and its characteristics with the clients of the federation (step 3). Once clients locally train their models with their private datasets, they send the models parameters and statistics of their data to the server for aggregation (step 4). At that point, the parameters

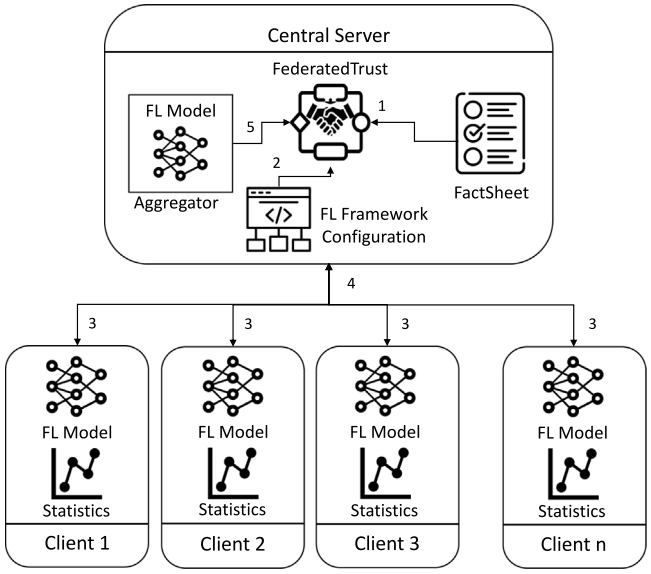

Fig. 3. Overview of the integration of FederatedTrust into an FL framework.

are aggregated, and the FL Model and statistics (about training datasets and model performance) are sent to the FederatedTrust algorithm for computing metrics during training (step 5). Steps 3, 4, and 5 are repeated until the FL model converges or the training rounds are over. At this point, the training process concludes, and FederatedTrust computes the post-training metrics. Finally, the FederatedTrust algorithm provides a score per pillar and a global one, together with a trustworthiness report. The detailed version of the previous steps is provided by Algorithm 1.

5. FederatedTrust algorithm prototype

This section contains the implementation details of the FederatedTrust algorithm when dealing with a given problem presented as a use case. The implemented prototype is available in [13].

5.1. FL framework selection and use case

Regarding the selection of the FL framework, after careful comparison and analysis of the most relevant and used frameworks in charge of training diverse FL models (TensorFlow Federated, PySyft,

Algorithm 1 Training a FL model equipped with FederatedTrust

Input: clients N, sampling size m, a central server S, number of iterations T, initial model $\tilde{w}^{(0)}$, FL Framework configuration C, FederatedTrust ft, FactSheet fs
Output: global score, pillars scores, trustworthiness report
S sends the hashed IDs of all clients $i \in [N]$, C, and fs to ft
ft creates a map of hashed client IDs to values of 0, representing the initial selection rate
S sends the model metadata to ft
S request class distribution information from all clients $i \in [N]$
for clients $i \in [N]$ **do**
    Client i uses ft function to calculate the sample size per class of local data
    ft creates or updates the class distribution map of hashed labels to sample size
end for
for $t = 0$ to T **do**
    S randomly samples $D^{(t)} \subset [N]$ clients with size of m
    S sends the hashed IDs of the selected clients to ft
    ft updates the client selection rate map
    S broadcasts the current model $\tilde{w}^{(t)}$ to all clients $i \in D^{(t)}$
    for clients $i \in D^{(t)}$ **do**
        Client i performs local training with $\tilde{w}^{(t)}$
        Client i sends new model updates $w_i^{(t+1)}$ back to S
        Client i computes evaluation metrics with local test data and local model $\tilde{w}_i{}'$
        Client i sends the evaluation results back to S
    end for
    S performs secure aggregation of all updates into a new global model $\tilde{w}^{(t+1)}$
end for
S aggregates the evaluation results and sends them to ft
ft receives the evaluation results and populates them
S asks ft to evaluate the trustworthiness of the model
ft computes the trustworthiness score and generates a report JSON and print message

Flower, FLUTE, LEAF, FederatedScope, FedEval, and FedML), FederatedScope [75] was chosen as a reference tool. More in detail, the following functionality led to the selection of FederatedScope in this work.

1. Standalone and distributed modes to set up clients experimentally or realistically.
2. Differential privacy and inference attacks.
3. Well-documented evaluation metrics.
4. Data zoo, a suite of well-known federated datasets such as FEMNIST [76], and Algo zoo, a list of optimization algorithms such as FedAvg, or FedProx.
5. Model zoo, a set of computer vision and language models.

Then, a basic use case was chosen for implementing and testing the deployment of the FederatedTrust algorithm as a proof of concept. In this context, the use case focuses on classifying handwritten digits and numbers in a federated and privacy-preserving way. For that, FederatedScope runs an FL training process in standalone mode with a variable number of clients and one central server over a defined number of iterations. The function to aggregate individual models and create the global FL Model is FedAvg, and the model is a convolutional neural network (CovNet2). The federated datasets that the clients hold locally are from the FEMNIST image dataset of handwritten digits and letters. The dataset has 62 classes (10 digits, 26 lowercase, 26 uppercase), and the images are 28 by 28 pixels. The client selector strategy is random sampling.

5.2. Metrics selection and FederatedTrust implementation

As seen in Section 3, not all metrics have a standardized way of being computed, available equations, or feasible calculation methods. Therefore, the objective is to implement a lightweight prototype with the basic pillars, notions, and metrics that can be calculated in any FL project created with the FederatedScope framework. The list of omitted notions and metrics and the reasons for not including them in the prototype implementation are the following.

- *Encryption and anonymization (Privacy)*. The usage of these two techniques is not documented in the configuration file of the FederatedScope framework. Furthermore, the implemented prototype considers differential privacy as perturbation technique to protect data privacy.
- *Information Leakage Risk (Privacy)*. This metric needs to be calculated during every round of the training process by running extra neural networks, which would incur high computation overhead.
- *Poisoning Defense (Robustness)*. It is challenging to quantify the usage of the poisoning defense mechanism unless the information is documented in the FL framework, which is not the case in FederatedScope. In addition, another way to verify this metric is to check the certified robustness against poisoning attacks, which is considered in the implementation.
- *Empirical Robustness (Robustness)*. This metric depends on each attack type, and the number of possibilities and complexity is huge. Furthermore, certified robustness could help to cover this metric.
- *Testing Coverage and System Reliability (Robustness)*. Neither testing data nor error rates are available in the FederatedScope framework.
- *Client Reputation (Robustness)*. There is no clear way to measure client reputation in a simulated environment. It requires more inspections of the client data provenance.
- *Data Quality (Robustness)*. It needs to be executed during every round of the training process, creating a high computation overhead for the FL model training process.
- *Discrimination Index (Fairness)*. It requires knowledge of protected sensitive features, and it is unclear how this calculation can be done without leaking sensitive information.
- *Visualization (Explainability)*. Graphical capabilities to show the explainability of FL models are not included in the FederatedScope framework.
- *Algorithmic Auditing (Accountability)*. It is not implemented because there are no real end users, and no attackers are present in the environment.
- *Client Registry (Federation)*. The FederatedScope framework simulates the clients participating in the federation. Therefore, no client registration is required.
- *Model co-versioning and replacement (Federation)*. Versioning and model replacement trigger functionality is not implemented as the validation scenario is simulated.

Once those metrics excluded for the prototype implementation are indicated, Table 3 describes the list of implemented metrics, with descriptions, inputs, output values, calculation moment, and federation actor in charge of computing them.

In order to understand how the trust FL score is computed, it is important to mention that the output value of each metric can have different ranges, as indicated in the metric definitions (Table 3). Therefore, to combine these values into an understandable trustworthiness score, different functions are used to translate the output values into a normalized score between [0, 1]. The logic of each normalization function is explained in Table 4. It shows for each metric the type of values it gets, the range or set of possible values, and how these values are mapped into the final output between 0 and 1. Once all metrics outputs are normalized, they are grouped (first by notions and then by pillars) to compute the Trust FL score. The algorithm prototype uses the mean average of all the metrics under a notion to calculate its final score. In the same way, the mean average of the notions of one pillar is employed as the pillar score. Finally, the Trust Score of the entire setup is the mean average of all the pillars. How metrics, notions, and pillars are combined can be changed to different aggregation methods based on the requirements of the environment, giving more importance to some metrics, notions, or pillars. The implementation of the previous life-cycle is done by using the *numpy*, *scipy*, and *sklearn* libraries. More in detail, FederatedTrust is designed as a third-party library.

Table 3
Metrics implemented by the FederatedTrust algorithm prototype.

Metric	Description	Input	Output	When	Who
Privacy					
Differential Privacy	Use of global or local differential privacy as a privacy defense	FactSheet	0/1	Pre-training	Server
Entropy	Uncertainty in predicting the value of a random variable	FL Framework Conf	[0, 1]	Pre-training	Server
Global Privacy Risk	Maximum privacy risk with differential privacy based on ϵ	Client Statistics	%	Pre-training	Server
Robustness					
Certified Robustness	Minimum perturbation required to change the neural network prediction	FL Model	Real	Post-training	Server
Performance	Test accuracy of the global model	Statistics, FL Model	%	During-training	Clients
Personalization	Use of personalized FL techniques	FactSheet	0/1	Pre-training	Server
Federation Scale	Number of clients representing the scale of the federation	FactSheet	Integer	Pre-training	Server
Fairness					
Participation Variation	Uniformity of distribution of participation rate among clients	FL Framework	[0, 1]	Post-training	Server
Accuracy Variation	Uniformity of distribution of performance among clients	Client Statistics, FL Model	[0, 1]	During-training	Clients
Class Imbalance	Average class imbalance estimation among clients	Client Statistics	[0, 1]	Pre-training	Clients
Explainability					
Algorithmic Transparency	Interpretability of the model by design	FL Model	[1, 5]	Pre-training	Server
Model Size	Model Features dimensionality, depth of decision tree, or number of parameters in neural networks	FL Model	Integer	Post-training	Server
Feature Importance	Average variance of feature importance scores	FL Model	[0, 1]	Post-training	Server
Accountability					
Project Specification	Project details and purpose	FactSheet	0/1	Pre-training	Server
Participants	Participants number, identifiers, and their organizations	FL Framework Conf	0/1	Pre-training	Server
Data	Contains Data origin and data-preprocessing steps	FactSheet	0/1	Pre-training	Server
Configuration	Information about the FL model	FL Framework Conf, FactSheet	0/1	Pre-training	Server
System	Contains training time, FL model size, and network performance	FL Framework Conf, Statistics	0/1	Post-training	Server
Federation					
Client Selector	Use of a client selector scheme rather than random selection	FactSheet	0/1	Pre-training	Server
Aggregation Algorithm	Selected aggregation function	FL Framework Conf	%	Pre-training	Server

6. Experiments

This section shows how the FederatedTrust framework can be integrated and employed with FederatedScope to evaluate trustworthiness in FL applications during model generation. The demonstration experiments are organized into two groups. The first set of three experiments using the FEMNIST dataset shows how the number of clients and their configuration impact the trustworthiness. Then, a second set of two experiments leveraging the N-BaIoT dataset about IoT network security shows how different training configurations affect a real-world use case.

6.1. Trustworthiness scores for experiments with FEMNIST

The previously defined setup combining FederatedTrust and FederatedScope was used to perform the following three experiments, which consist of training FL models that classify hand-written digits using the FEMNIST dataset.

- *Experiment 1*: The experiment considers a federation of 10 clients with a selection rate of 50% clients per training round (5 out of 10 are randomly selected in each round). The training of the FL model runs 5 rounds.
- *Experiment 2*: It considers 50 clients, and in every iteration, the server randomly selects 60% of clients. The experiment runs 25 training iterations. The main novelty of this experiment lies in the inclusion of differential privacy with ϵ of 20.
- *Experiment 3*: This experiment consists of 100 clients with a 40% client selection rate and 50 rounds of training. The ϵ value of differential privacy is set to 6.

In all experimental setups, the same configuration of the project specifications, data, and participants were employed. The specific purpose and background information of the FL project were not explicitly stated within the model. Furthermore, all models implemented a client selection method based on random sampling, and the FedAvg aggregation algorithm was utilized. As a result, the federation pillar score remained constant across all three experiments.

As seen in Fig. 4, the trustworthiness score for both first experiments is 0.56. However, there are differences in terms of pillars and metrics.

Starting from the privacy pillar, its score increased from 0.31 (Experiment 1) to 0.64 (Experiment 2) because of using differential privacy. However, the effectiveness of the privacy mechanism was not good enough because of the large value of ϵ chosen (20). It can be seen in the indistinguishability notion (covered by the global privacy risk metric), with a score of 0 in both experiments. In this context, the larger the ϵ value, the less noise was added to the data, and therefore, the higher the probability of being identified by adversaries. Dealing with the fairness pillar, its score from Experiment 1 to 2 was also increased from 0.25 to 0.47. It was mainly due to a significant increment in the selection variation metric (from 0.08 to 0.83), which resulted from the overall increase in the number of clients, the client sampling rate, and the number of rounds. The selection fairness improved with more clients participating and more training rounds. However, the performance variation metric (Fairness pillar) dropped from 0.58 to 0.50 from Experiment 1 to 2. It could also be due to the increased number of clients. More clients with different levels of heterogeneity in their data could influence the global model's generalizability, affecting the individual test accuracy at the client level. Furthermore, personalization techniques were not used, so the global model was not adapted to the clients. Regarding the explainability pillar, its global score increased from 0.59 (Experiment 1) to 0.67 (Experiment 2), increasing the feature importance metric score from

Table 4
Normalization of the metrics outputs.

Metric	Type	Output	Normalized output
Privacy			
Differential Privacy	True/False	0/1	0/1
Entropy	Uncertainty	[0, 1]	[0,1]
Global Privacy Risk	Percentage	[0, 100]	[0, 1]
Robustness			
Certified Robustness	CLEVER Score	[0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2, 2.2, 2.4, 2.6, 2.8, 3, 3.2, 3.4, 3.6, 3.8, 4.0]	{0, 0.05, 0.1, ...1}
Performance	Accuracy	[0, 1]	[0, 1]
Personalization	True/False	0/1	0/1
Federation Scale	Number of clients	[10, 10^2, 10^3, 10^4, 10^5, 10^6]	{0, 0.2, 0.4, ...1}
Fairness			
Participation Variation	Coefficient of Variation	[0, 1]	[0, 1]
Accuracy Variation	Coefficient of Variation	[0, 1]	[0, 1]
Class Imbalance	Balance Ratio	[0, 1]	[0, 1]
Explainability			
Algorithm Transparency	Random Forest, K-Nearest Neighbors, Support Vector Machine, GaussianProcessClassifier, Decision Tree, Multilayer Perceptron, AdaBoost, GaussianNB, Quadratic Discriminant Analysis, Logistic Regression, Linear Regression, Sequential, Convolutional Neural Network	{4, 3, 2, 3, 5, 1, 3, 3.5, 3,4, 3.5, 1, 1}	{0, 0.2, 0.4, 0.5, ...1}
Model Size	Number of Model Parameters	{1, 10, 50, 100, 500, 1000, 5000, 10000, 50000, 100000, 500000}	{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}
Feature Importance	SHAP Importance	[0, 1]	[0, 1]
Accountability			
Project Specification	True/False	0/1	0/1
Participation	True/False	0/1	0/1
Data	True/False	0/1	0/1
Configuration	True/False	0/1	0/1
System	True/False	0/1	0/1
Federation			
Client Selector	True/False	0/1	0/1
Aggregation Algorithm	FedAvg, FedOpt, FedProx, FedBN, pFedMe, Ditto, FedEM	{0.8493, 0.8492, 0.8477, 0.8548, 0.8765, 0.8661, 0.8479}	{0.8493, 0.8492, 0.8477, 0.8548, 0.8765, 0.8661, 0.8479}

[] denotes a continuous range and {} denotes a list of values.

0.67 to 0.92. It is impacted by the different numbers in terms of clients and the differences between their datasets. Finally, the robustness pillar score dropped from 0.36 (Experiment 1) to 0.33 (Experiment 2). It was mainly due to the decrement in the certified robustness metric (from 0.48 to 0.19) and the performance reduction from 0.96 to 0.93. The drop in the certified robustness metric could be related to the increase in the number of clients and the number of rounds. In theory, more aggregating parties provide more entries and surfaces for adversaries to insert backdoor perturbations for poisoning attacks. There are also higher chances for parties to collude when they are more in number. The higher number of rounds also means that adversaries have more chances to attack. The federation scale metric for both experiments has a similar score since both have less than 50 clients.

Before comparing Experiments 2 and 3, it is important to mention that the main difference between them is the privacy pillar (due to the ϵ change). Focusing on Experiments 2 and 3, from the robustness pillar perspective, its overall score only increased from 0.33 (Experiment 2) to 0.35 (Experiment 3), even though there was a significant increase in the federation score from 0.2 to 0.4. It was mainly due to the less relevance of that metric in the pillar than the rest. Furthermore, the certified robustness score decreased from 0.19 to 0.05. The privacy pillar score increased from 0.64 (Experiment 2) to 0.71 (Experiment 3) due to the global privacy risk metric. The increase in the number of clients caused an improvement in the indistinguishability notion (covered by the previous metric). Assuming that random guessing was used in Experiment 3, it was twice as challenging to guess the correct target among 100 clients compared to 50 clients (used in Experiment 2). Regarding explainability, the score remained constant in Experiments 2 and 3 as no changes were made in the factors impacting these metrics. Finally, the fairness score was almost identical for both experiments (0.47 and 0.5). It might be because the ratio between the increase in the number of clients and the increase in the number of rounds was the same, while the clients' sampling rate remained the same as well.

6.2. Trustworthiness scores for experiments with N-BaIoT

To perform a more exhaustive comparison, two more experiments were performed. In this case, the N-BaIoT [77] dataset was leveraged. It contains benign and attack network traces from nine different IoT devices. In the two experiments done with this dataset, the number of clients is nine, one per IoT device. Besides, the same configuration of project specifications, data, and participants was employed.

- *Experiment 4:* It employs FedAvg as the aggregation algorithm and does not use Differential Privacy. Additionally, the client selection ratio is 70%
- *Experiment 5:* It leverages Federated Median as an aggregation approach and employs local Differential Privacy with ϵ equal to 4. The client selection ratio is 90%, and local dataset balancing is applied in each client, leaving a 50/50 balance per class. In this case, the neural network contains four hidden layers of 100, 80, 70, and 50 neurons.

Fig. 5 shows the different trustworthiness metrics in Experiment 4 (up) and 5 (bottom). Comparing Experiments 4 and 5, it can be seen that even though the overall performance slightly decreases from 0.99 to 0.98, the final trust score increases from 0.57 to 0.68. The decrease

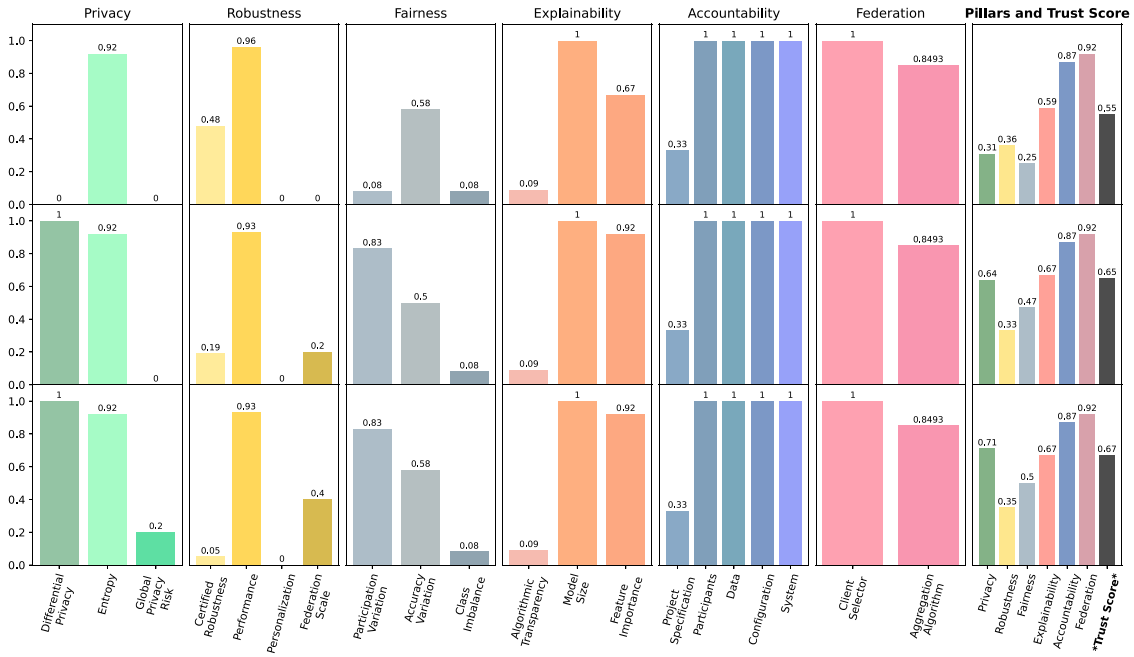

Fig. 4. Experiment comparison with FEMNIST dataset.

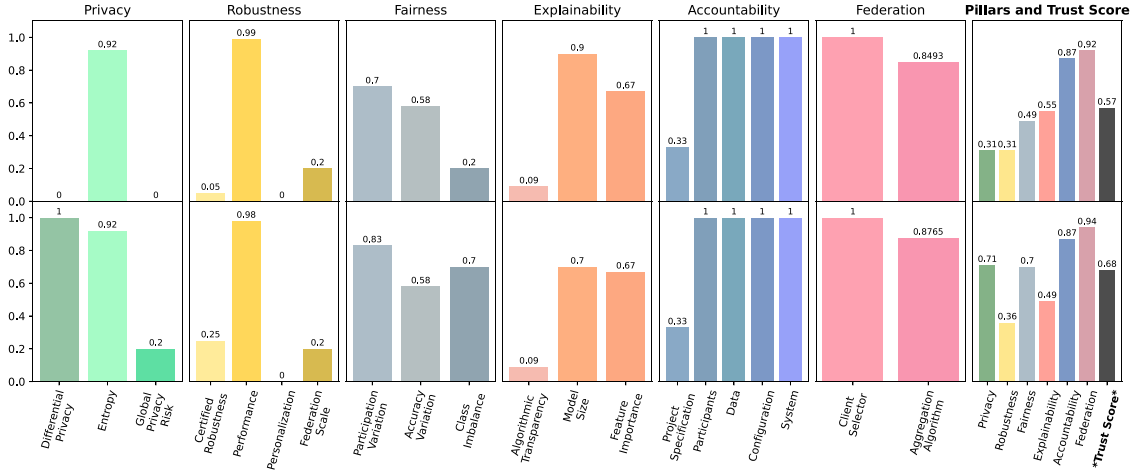

Fig. 5. Experiment comparison with N-BaIoT dataset.

in the performance is caused by the usage of a different aggregation algorithm and the usage of differential privacy, as adding noise to the training data can decrease the final model performance. The increase in the trust score is happening due to the increase in most of the evaluated pillars due to the differences in the experiment configurations. The privacy pillar goes from 0.31 to 0.71 due to the usage of differential privacy and the associated increase (from 0 to 0.2) in the global privacy risk metric. The robustness pillar is also increased even though the overall performance decreases by 0.01. It occurs because the certified robustness increases from 0.05 to 0.25, enhancing the pillar score from 0.31 to 0.36. Similarly, the fairness pillar is increased from 0.49 to 0.7 due to a better client participation ratio and class imbalance management. In contrast, the explainability pillar is decreased from 0.55 to 0.49 because the model size metric goes from 0.9 to 0.7 due to using a larger model in Experiment 5. Finally, the federation pillar is slightly increased from 0.92 to 0.94 thanks to median aggregation instead of averaging.

When comparing Experiments 4 and 5, it can be seen how a real federated application can leverage configuration optimizations during its design to increase its trustworthiness without losing notable performance. Note that more changes could be applied according to the metric to optimize depending on the exact use case.

6.3. Limitations

There are several limitations of the FederatedTrust prototype in terms of quantifying the trustworthiness level of FL models. First, some metrics, like the federation scale (robustness pillar), often have to be considered with other factors to represent the notion well. For example, the analysis performed in Section 3 shows that, in practice, the client reputation metric is also an essential factor for the client reliability notion. However, it was difficult to quantify the client reputation in the performed experiments. Another example would be the client selection fairness notion. Although the client participation variation metric is easily quantifiable by computing the dispersion of selection rate among the clients, this variation metric alone might not be the best representation of fairness for client selection in FL.

Another standalone limitation concerning the evaluated pillars is explainability. Most models employed in real scenarios are pure black-box setups, making it difficult for FederatedTrust to evaluate and show how these models make decisions. Besides, bias and AI explainability are intrinsically correlated. In this sense, AI explanations can be misleading or incomplete due to training data bias, leading to trust in biased models. Therefore, biased decisions by AI are hard to identify by FederatedTrust, even with the considered explainability metrics. These limitations also open the field for developing metrics and standard frameworks for better explainability evaluation.

Another limitation pertains to the algorithm scoring and metric aggregation systems. First, the logic of the scoring functions greatly impacts how the trust score of each metric is calculated. Based on the pillar analysis in Section 3, there were general directions of how every metric should impact the overall trustworthiness level. However, the concrete scoring maps and ranges were created based on knowledge from other studies, and their generalizability to other systems was not fully evaluated. Second, although FederatedTrust provides a flexible approach to deciding the importance of each metric and pillar, selecting optimum values is challenging because it presents a trade-off between pillars. In this context, multi-objective optimization techniques [78] could be integrated into FederatedTrust to optimize the weight selection according to the needs of each scenario where the algorithm is deployed.

Some challenges dealing with resource consumption, data leakage, governance, compliance, and scalability might appear when deploying FederatedTrust in real scenarios. For example, implementing metrics evaluated by FederatedTrust requires careful design and consumes computational resources that might not be available in resource-constrained devices. In addition, it is critical to guarantee data privacy while computing the trustworthiness score of FL models [79]. FederatedTrust must also adhere to regulatory and legal requirements, so maintaining transparency, accountability, and auditability is challenging. Finally, implementing scalable and distributed mechanisms (like Blockchain) to populate metrics outputs while maintaining privacy, integrity, and trust poses a significant challenge [80].

Another key limitation arises when comparing the proposed solution with other frameworks in the literature. The unique comparison could be made on an individual pillar basis with solutions that calculate the trustworthiness of traditional ML/DL models (privacy, robustness, explainability, and accountability). However, it is important to note that this represents a completely different scenario, lacking the privacy-preserving capabilities provided by Federated Learning. Therefore, such a comparison would not be fair and relevant in assessing the superiority of the FederatedTrust algorithm. Finally, it is important to remark that the proposed solution also covers the metrics implemented by works dealing only with particular pillars, so this comparison would show the same outputs.

7. Conclusions and future work

This work presents a comprehensive taxonomy encompassing the most relevant aspects of trustworthy FL. The proposed taxonomy expands upon the pillars previously identified by prior research, namely privacy, robustness, fairness, explainability, and accountability, by introducing a novel called federation. This new pillar quantifies the trustworthiness of FL models from both participant and FL model perspectives. Moreover, the existing pillars recognized in the literature have been augmented with various notions and novel metrics that address FL models. Based on the proposed taxonomy, a trustworthiness evaluation algorithm for FL models, named FederatedTrust, has been devised to be extensible, configurable, and flexible. To assess the effectiveness and viability of the FederatedTrust prototype, it was implemented and tested within the FederatedScope FL framework. Five experiments were conducted to validate the FederatedTrust prototype, utilizing distinct FL configurations that involved varying the number of clients, datasets, training rounds, differential privacy parameters, normalization and data balancing approaches, and model configurations. Throughout these experiments, the trustworthiness levels of each pillar and metric were compared and analyzed while classifying handwritten digits using the FEMNIST dataset and detecting IoT malware using the N-BaIoT dataset. The experiments demonstrated the intricate nature of quantifying the trustworthiness level of FL models, with the FederatedTrust algorithm representing the initial endeavor to comprehensively assess an FL model based on a holistic trustworthiness taxonomy. Finally, limitations of the current version of the algorithm prototype have been discussed.

As future work, it is planned to extend the prototype by implementing new metrics identified in the taxonomy. It is also intended to deploy the algorithm on FL frameworks, training FL models in a decentralized fashion. Furthermore, multi-objective optimization techniques will be analyzed and evaluated to help aggregate metrics while computing the global trust score.

CRediT authorship contribution statement

Pedro Miguel Sánchez Sánchez: Literature review, Design and implementation of the algorithm, Writing – original draft. **Alberto Huertas Celdrán:** Design of the algorithm, Validation of the algorithm in the proposed scenario, Writing – original draft. **Ning Xie:** Literature review, Algorithm implementation, Validation. **Gérôme Bovet:** Literature review, Paper revision, Supervision. **Gregorio Martínez Pérez:** Funding acquisition, Paper and results revision, Supervision. **Burkhard Stiller:** Funding acquisition, Results analysis, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work has been partially supported by *(a)* the Swiss Federal Office for Defense Procurement (armasuisse) with the DEFENDIS and CyberForce (CYD-C-2020003) projects and *(b)* the University of Zürich UZH.

References

[1] C. Rudin, C. Wang, B. Coker, The age of secrecy and unfairness in recidivism prediction, Harvard Data Sci. Rev. 2 (1) (2020).
[2] M. Uzair, Who is liable when a driverless car crashes? World Electr. Veh. J. 12 (2) (2021) 62.
[3] G. Wu, 5 big problems with openai's chatgpt, 2022, Make use of, URL https://www.makeuseof.com/openai-chatgpt-biggest-probelms/.
[4] V. Dignum, Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way, Springer Nature, 2019.
[5] B. Li, P. Qi, B. Liu, S. Di, J. Liu, J. Pei, J. Yi, B. Zhou, Trustworthy AI: From principles to practices, ACM Comput. Surv. 55 (9) (2023).
[6] T. Madiega, Artificial intelligence act, 2021, European Parliament: European Parliamentary Research Service.
[7] AI HLEG of the European Commission, Ethics guidelines for trustworthy AI, 2019.
[8] H. Liu, Y. Wang, et al., Trustworthy AI: A computational perspective, 2021, arXiv.
[9] A. Huertas Celdrán, M. Gil Pérez, I. Mlakar, J.M. Alcaraz Calero, F.J. García Clemente, G. Martínez Pérez, Z.A. Bhuiyan, PROTECTOR: Towards the protection of sensitive data in europe and the US, Comput. Netw. 181 (2020) 107448.
[10] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A.y. Arcas, Communication-efficient learning of deep networks from decentralized data, in: A. Singh, J. Zhu (Eds.), International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 54, PMLR, 2017, pp. 1273–1282.

[11] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, H. Yu, Federated learning, Synth. Lect. Artif. Intell. Mach. Learn. 13 (3) (2019) 1–207.

[12] P. Kairouz, H.B. McMahan, et al., Advances and open problems in federated learning, Found. Trends® Mach. Learn. 14 (1–2) (2021) 1–210.

[13] N. Xie, FederatedTrust: A trustworthiness evaluation framework, 2022, URL https://github.com/ningxie1991/FederatedTrust.

[14] D. Chai, L. Wang, K. Chen, Q. Yang, Fedeval: A benchmark system with a comprehensive evaluation model for federated learning, 2020, arXiv preprint arXiv:2011.09655.

[15] A.H. Celdran, J. Kreischer, M. Demirci, J. Leupp, P.M. Sanchez, M.F. Franco, G. Bovet, G.M. Perez, B. Stiller, A framework quantifying trustworthiness of supervised machine and deep learning models, in: SafeAI2023: The AAAI's Workshop on Artificial Intelligence Safety, 2023, pp. 2938–2948.

[16] A.H. Celdrán, J. Bauer, M. Demirci, J. Leupp, M.F. Franco, P.M. Sánchez Sánchez, G. Bovet, G.M. Pérez, B. Stiller, RITUAL: a platform quantifying the trustworthiness of supervised machine learning, in: 2022 18th International Conference on Network and Service Management, CNSM, 2022, pp. 364–366.

[17] Y. Dong, X. Chen, L. Shen, D. Wang, Eastfly: Efficient and secure ternary federated learning, Comput. Secur. 94 (2020) 101824.

[18] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H.B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth, Practical secure aggregation for privacy-preserving machine learning, in: ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 1175–1191.

[19] O. Choudhury, A. Gkoulalas-Divanis, T. Salonidis, I. Sylla, Y. Park, G. Hsu, A. Das, Differential privacy-enabled federated learning for sensitive health data, 2019, arXiv preprint arXiv:1910.02578.

[20] R.C. Geyer, T. Klein, M. Nabi, Differentially private federated learning: A client level perspective, 2017, arXiv preprint arXiv:1712.07557.

[21] Y. Liu, X. Zhu, J. Wang, J. Xiao, A quantitative metric for privacy leakage in federated learning, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2021, pp. 3065–3069.

[22] M.S. Jere, T. Farnan, F. Koushanfar, A taxonomy of attacks on federated learning, IEEE Secur. Priv. 19 (2) (2020) 20–28.

[23] M. Naseri, J. Hayes, E. De Cristofaro, Local and central differential privacy for robustness and privacy in federated learning, 2020, arXiv preprint arXiv:2009.03561.

[24] L. Muñoz-González, K.T. Co, E.C. Lupu, Byzantine-robust federated machine learning through adaptive model averaging, 2019, arXiv preprint arXiv:1909.05125.

[25] N. Rodríguez-Barroso, E. Martínez-Cámara, M.V. Luzón, F. Herrera, Backdoor attacks-resilient aggregation based on robust filtering of outliers in federated learning for image classification, Knowl.-Based Syst. 245 (2022) 108588.

[26] A. Jankovic, R. Mayer, An empirical evaluation of adversarial examples defences, combinations and robustness scores, in: ACM International Workshop on Security and Privacy Analytics, 2022, pp. 86–92.

[27] Y. Shi, H. Yu, C. Leung, A survey of fairness-aware federated learning, 2021, arXiv preprint arXiv:2111.01872.

[28] Z. Fan, H. Fang, Z. Zhou, J. Pei, M.P. Friedlander, C. Liu, Y. Zhang, Improving fairness for data valuation in horizontal federated learning, in: 2022 IEEE 38th International Conference on Data Engineering, ICDE, IEEE, 2022, pp. 2440–2453.

[29] X. Yue, M. Nouiehed, R.A. Kontar, Gifair-fl: An approach for group and individual fairness in federated learning, 2021, arXiv preprint arXiv:2108.02741.

[30] D.Y. Zhang, Z. Kou, D. Wang, Fairfl: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models, in: 2020 IEEE International Conference on Big Data, Big Data, IEEE, 2020, pp. 1051–1060.

[31] T. Huang, W. Lin, W. Wu, L. He, K. Li, A.Y. Zomaya, An efficiency-boosting client selection scheme for federated learning with fairness guarantee, IEEE Trans. Parallel Distrib. Syst. 32 (7) (2020) 1552–1564.

[32] G. Wang, Interpret federated learning with Shapley values, 2019, arXiv preprint arXiv:1905.04519.

[33] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, Vol. 30, 2017.

[34] P. Chen, X. Du, Z. Lu, J. Wu, P.C. Hung, EVFL: An explainable vertical federated learning for data-oriented artificial intelligence systems, J. Syst. Archit. 126 (2022) 102474.

[35] N. Baracaldo, A. Anwar, M. Purcell, A. Rawat, M. Sinn, B. Altakrouri, D. Balta, M. Sellami, P. Kuhn, U. Schopp, et al., Towards an accountable and reproducible federated learning: A FactSheets approach, 2022, arXiv preprint arXiv:2202.12443.

[36] H.B. Desai, M.S. Ozdayi, M. Kantarcioglu, Blockfla: Accountable federated learning via hybrid blockchain architecture, in: ACM Conference on Data and Application Security and Privacy, 2021, pp. 101–112.

[37] V. Mugunthan, R. Rahman, L. Kagal, Blockflow: An accountable and privacy-preserving solution for federated learning, 2020, arXiv preprint arXiv:2007.03856.

[38] S. Awan, F. Li, B. Luo, M. Liu, Poster: A reliable and accountable privacy-preserving federated learning framework using the blockchain, in: ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 2561–2563.

[39] C. Molnar, Interpretable Machine Learning, BOOKDOWN, 2020.

[40] IBM Research, AI FactSheets 360, IBM Research, 2022, URL https://aifs360.mybluemix.net/.

[41] K. Wei, J. Li, M. Ding, C. Ma, H.H. Yang, F. Farokhi, S. Jin, T.Q. Quek, H.V. Poor, Federated learning with differential privacy: Algorithms and performance analysis, IEEE Trans. Inf. Forensics Secur. 15 (2020) 3454–3469.

[42] H. Fang, Q. Qian, Privacy preserving machine learning with homomorphic encryption and federated learning, Future Internet 13 (4) (2021) 94.

[43] Y. Li, Y. Zhou, A. Jolfaei, D. Yu, G. Xu, X. Zheng, Privacy-preserving federated learning framework based on chained secure multiparty computing, IEEE Internet Things J. 8 (8) (2020) 6178–6186.

[44] O. Choudhury, A. Gkoulalas-Divanis, et al., A syntactic approach for privacy-preserving federated learning, in: ECAI 2020, IOS Press, 2020, pp. 1762–1769.

[45] I. Wagner, D. Eckhoff, Technical privacy metrics: A systematic survey, ACM Comput. Surv. 51 (3) (2018) 1–38.

[46] S. Zheng, X. Wang, L. Duan, Adaptive federated learning via entropy approach, 2023, arXiv preprint arXiv:2303.14966.

[47] Q. Liu, G. Wang, F. Li, S. Yang, J. Wu, Preserving privacy with probabilistic indistinguishability in weighted social networks, IEEE Trans. Parallel Distrib. Syst. 28 (5) (2016) 1417–1429.

[48] S.K. Lo, Q. Lu, C. Wang, H.-Y. Paik, L. Zhu, A systematic literature review on federated machine learning: From a software engineering perspective, ACM Comput. Surv. 54 (5) (2021) 1–39.

[49] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, M. Guizani, Reliable federated learning for mobile networks, IEEE Wirel. Commun. 27 (2) (2020) 72–80.

[50] C. Wu, X. Yang, S. Zhu, P. Mitra, Mitigating backdoor attacks in federated learning, 2020, arXiv preprint arXiv:2011.01767.

[51] L. Lyu, H. Yu, X. Ma, C. Chen, L. Sun, J. Zhao, Q. Yang, P.S. Yu, Privacy and robustness in federated learning: Attacks and defenses, IEEE Trans. Neural Netw. Learn. Syst. (2022) 1–21.

[52] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov, How to backdoor federated learning, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 2938–2948.

[53] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, L. Daniel, Evaluating the robustness of neural networks: An extreme value theory approach, 2018, arXiv preprint arXiv:1801.10578.

[54] X. Gitiaux, A. Khant, E. Beyrami, C. Reddy, J. Gupchup, R. Cutler, AURA: Privacy-preserving augmentation to improve test set diversity in noise suppression applications, in: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023, pp. 1–5.

[55] S.U. Farooq, S. Quadri, N. Ahmad, Metrics, models and measurements in software reliability, in: IEEE International Symposium on Applied Machine Intelligence and Informatics, SAMI, 2012, pp. 441–449.

[56] H. Zhu, J. Xu, S. Liu, Y. Jin, Federated learning on non-IID data: A survey, Neurocomputing 465 (2021) 371–390.

[57] S.K. Lo, Q. Lu, et al., Architectural patterns for the design of federated learning systems, J. Syst. Softw. 191 (2022) 111357.

[58] V. Smith, C.-K. Chiang, M. Sanjabi, A.S. Talwalkar, Federated multi-task learning, in: Advances in Neural Information Processing Systems, Vol. 30, 2017.

[59] F. Sattler, K.-R. Müller, W. Samek, Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints, IEEE Trans. Neural Netw. Learn. Syst. 32 (8) (2020) 3710–3722.

[60] M.G. Arivazhagan, V. Aggarwal, A.K. Singh, S. Choudhary, Federated learning with personalization layers, 2019, arXiv preprint arXiv:1912.00818.

[61] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, B. He, A survey on federated learning systems: Vision, hype and reality for data privacy and protection, IEEE Trans. Knowl. Data Eng. (2021).

[62] B. Pejó, G. Biczók, Quality inference in federated learning with secure aggregation, IEEE Trans. Big Data 9 (5) (2023) 1430–1437.

[63] S. Feuerriegel, M. Dolata, G. Schwabe, Fair AI, Bus. Inf. Syst. Eng. 62 (4) (2020) 379–384.

[64] S.A. Alvi, Y. Hong, S. Durrani, Federated learning cost disparity for IoT devices, in: 2022 IEEE International Conference on Communications Workshops, ICC Workshops, 2022, pp. 818–823.

[65] M. Yang, X. Wang, H. Zhu, H. Wang, H. Qian, Federated learning with class imbalance reduction, in: 2021 29th European Signal Processing Conference, EUSIPCO, IEEE, 2021, pp. 2174–2178.

[66] A.B. Arrieta, et al., Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, 2019, arXiv.

[67] S.R. Islam, W. Eberle, S.K. Ghafoor, Towards quantification of explainability in explainable artificial intelligence methods, in: The Thirty-Third International Flairs Conference, 2020.

[68] J. Gou, B. Yu, S.J. Maybank, D. Tao, Knowledge distillation: A survey, Int. J. Comput. Vis. 129 (6) (2021) 1789–1819.

[69] M. Ungersböck, T. Hiessl, et al., Explainable federated learning: A lifecycle dashboard for industrial settings, 2022, TechRxiv.

[70] M. Arnold, R. Bellamy, M. Hind, S. Houde, S. Mehta, A. Mojsilović, R. Nair, K.N. Ramamurthy, A. Olteanu, D. Piorkowski, D. Reimer, J. Richards, J. Tsay, K. Varshney, Factsheets: increasing trust in ai services through supplier's declarations of conformity, IBM Journal of Research and Development 63 (4/5) (2019) 6:1–6:13.

[71] I.D. Raji, A. Smart, R.N. White, M. Mitchell, T. Gebru, B. Hutchinson, J. Smith-Loud, D. Theron, P. Barnes, Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing, in: Conference on Fairness, Accountability, and Transparency, 2020, pp. 33–44.
[72] H. Baniata, R. Prodan, A. Kertesz, Machine learning for alternative mining in pow-based blockchains: Theory, implications and applications, 2022, TechRxiv.
[73] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, M. Jirstrand, A performance evaluation of federated learning algorithms, in: Workshop on Distributed Infrastructures for Deep Learning, 2018, pp. 1–8.
[74] E.T.M. Beltrán, M.Q. Pérez, P.M.S. Sánchez, S.L. Bernal, G. Bovet, M.G. Pérez, G.M. Pérez, A.H. Celdrán, Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges, IEEE Commun. Surv. Tutor. (2023) 1.
[75] Y. Xie, Z. Wang, D. Chen, D. Gao, L. Yao, W. Kuang, Y. Li, B. Ding, J. Zhou, Federatedscope: A flexible federated learning platform for heterogeneity, 2022, arXiv preprint arXiv:2204.05011.
[76] S. Caldas, S.M.K. Duddu, et al., LEAF: A benchmark for federated settings, 2018, arXiv.
[77] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, Y. Elovici, N-baiot—network-based detection of IoT botnet attacks using deep autoencoders, IEEE Pervasive Comput. 17 (3) (2018) 12–22.
[78] N. Saini, S. Saha, Multi-objective optimization techniques: A survey of the state-of-the-art and applications: Multi-objective optimization techniques, Eur. Phys. J. Spec. Top. 230 (10) (2021) 2319–2335.
[79] J.A. Alzubi, O.A. Alzubi, A. Singh, M. Ramachandran, Cloud-IIoT-based electronic health record privacy-preserving by CNN and blockchain-enabled federated learning, IEEE Trans. Ind. Inform. 19 (1) (2022) 1080–1087.
[80] M. Zarour, M.T.J. Ansari, M. Alenezi, A.K. Sarkar, M. Faizan, A. Agrawal, R. Kumar, R.A. Khan, Evaluating the impact of blockchain models for secure and trustworthy electronic healthcare records, IEEE Access 8 (2020) 157959–157973.

Pedro M. Sánchez Sánchez received the M.Sc. degree in computer science from the University of Murcia, Spain. He is currently pursuing his Ph.D. in computer science at University of Murcia. His research interests are focused on continuous authentication, networks, 5G, cybersecurity and the application of machine learning and deep learning to the previous fields.

Alberto Huertas Celdrán received the M.Sc. and Ph.D. degrees in Computer Science from the University of Murcia, Spain. He is currently a postdoctoral fellow at the Communication Systems Group CSG, Department of Informatics IfI at the University of Zurich UZH. His scientific interests include IoT, BCI, cybersecurity, data privacy, continuous authentication, semantic technology, and computer networks.

Ning Xie is an experienced software engineer who constantly seeks out innovative solutions to everyday problems. In her five years in the FinTech industry, Ning has honed her collaboration skills, web application development practices and analytical thinking in a dynamic global working environment. Currently, she is pursuing a master's degree in People-Oriented Computing at the University of Zurich with a focus on Human–Computer Interaction and Artificial Intelligence. Her thesis topic is on quantifying the trustworthiness level of federated machine learning models.

Gérôme Bovet received his Ph.D. in networks and computer systems from Telecom ParisTech, France, in 2015, and an Executive MBA from the University of Fribourg, Switzerland in 2021. He is the head of data science for the Swiss Department of Defense, where he leads a research team and portfolio of about 30 Cyber-Defence projects. His work focuses on ML and DL approaches, with an emphasis on anomaly detection, adversarial and collaborative learning applied to data gathered by IoT sensors.

Gregorio Martinez Pérez is Full Professor in the Department of Information and Communications Engineering of the University of Murcia, Spain. His scientific activity is mainly devoted to cybersecurity and networking. He is working on different national (14 in the last decade) and European IST research projects (11 in the last decade) related to these topics, being Principal Investigator in most of them. He has published 200+ papers in international conference proceedings, magazines and journals.

Burkhard Stiller received his M.Sc. degree in Computer Science and the Ph.D. degree from the University of Karlsruhe, Germany, in 1990 and 1994. Since 2004 he chairs the Communication Systems Group CSG, Department of Informatics IfI, University of Zürich UZH, Switzerland as a Full Professor. His main research interests are published in +300 papers and include decentralized systems with fully control, network and service management, IoT, and telecommunication economics.