Journal of
Sensor and Actuator Networks

MDPI

Article

Federated Learning for Privacy-Friendly Health Apps: A Case Study on Ovulation Tracking

Nikolaos Pavlidis [1,2,*], Andreas Sendros [1,2], Theodoros Tsiolakis [1,2], Periklis Kostamis [1,2], Christos Karasoulas [1,2], Eleni Briola [1,2], Christos Chrysanthos Nikolaidis [1,2], Vasilis Perifanis [1,2], George Drosatos [1], Eleftheria Katsiri [1,2], Despoina Elisavet Filippidou [3], Anastasios Manos [3] and Pavlos S. Efraimidis [1,2]

[1] Institute for Language and Speech Processing, Athena Research Center, 67100 Xanthi, Greece; asendros@athenarc.gr (A.S.); ttsiolak@ee.duth.gr (T.T.); periklis.kostamis@athenarc.gr (P.K.); c.karasoulas@athenarc.gr (C.K.); eleni.briola@athenarc.gr (E.B.); christos.nikolaidis@athenarc.gr (C.C.N.); vperifan@athenarc.gr (V.P.); gdrosato@athenarc.gr (G.D.); eli@athenarc.gr (E.K.); pefraimi@athenarc.gr (P.S.E.)
[2] Department of Electrical and Computer Engineering, Democritus University of Thrace, Kimmeria, 67100 Xanthi, Greece
[3] OPSIS Research, Strada Corbita 30, Parter, Sector 5, 51083 Bucharest, Romania; eli@opsis-research.ro (D.E.F.); tasos.manos@gmail.com (A.M.)
* Correspondence: nikolaos.pavlidis@athenarc.gr

Abstract: In an era of increasing reliance on digital health solutions, safeguarding user privacy has emerged as a paramount concern. Health applications often need to balance advanced AI functionalities with sufficient privacy measures to ensure user engagement. This paper presents the architecture of FLORA, a privacy-first ovulation-tracking application that leverages federated learning (FL), privacy-enhancing technologies (PETs), and blockchain to protect user data while delivering accurate and personalized health insights. Unlike conventional centralized systems, FLORA ensures that sensitive information remains on users' devices, with predictive algorithms powered by local computations. Blockchain technology provides immutable consent tracking and model update transparency, further improving user trust. In addition, FLORA's design incentivizes participation through a token-based reward system, fostering collaborative data contributions. This work illustrates how the integration of cutting-edge technologies creates a secure, scalable, and user-centric health application, setting a new standard for privacy-preserving digital health platforms.

Keywords: federated learning; blockchain; privacy; machine learning; encryption

Academic Editors: Giovanni Paragliola, Laura Verde, Fiammetta Marulli and Rosario Catelli

Received: 17 December 2024
Revised: 21 January 2025
Accepted: 27 January 2025
Published: 29 January 2025

Citation: Pavlidis, N.; Sendros, A.; Tsiolakis, T.; Kostamis, P.; Karasoulas, C.; Briola, E.; Nikolaidis, C.C.; Perifanis, V.; Drosatos, G.; Katsiri, E.; et al. Federated Learning for Privacy-Friendly Health Apps: A Case Study on Ovulation Tracking. *J. Sens. Actuator Netw.* **2025**, *14*, 11. https://doi.org/10.3390/jsan14010011

1. Introduction

In recent years, the use of digital health applications has become increasingly widespread, offering individuals innovative tools to manage and monitor their well-being [1]. Among these, ovulation-tracking apps have gained notable popularity, providing women with valuable insights into their reproductive health, including menstrual cycles, fertility windows, and overall wellness [2]. These applications play a critical role in empowering users to make informed decisions about family planning and health management. However, as the adoption of these technologies has grown, so have concerns regarding data privacy and security [3]. Studies [4,5] have revealed significant issues with many existing apps, particularly those that collect sensitive health data, which are often shared without explicit user consent. This raises ethical and legal concerns, especially in contexts where such information could be misused, potentially leading to discrimination, profiling, or legal repercussions.

The challenges that can arise from poor quality of privacy in such applications are profound. Sensitive reproductive health data, including menstrual cycle details and ovulation patterns, can reveal many aspects of users' lives. For instance, research conducted by the Mozilla Foundation revealed that 72% of reproductive health apps exhibit serious privacy flaws, often sharing sensitive data with third parties for advertising or analytics purposes [6]. Such practices undermine user trust and deter individuals from adopting potentially beneficial digital health solutions. This underscores the need for health applications that prioritize data privacy and user control.

Existing ovulation-tracking apps offer a range of features, including period tracking, fertility predictions, and wellness recommendations [7]. Although most popular apps currently available in the market have gained recognition for their usability and predictive accuracy, they often rely on centralized data storage and processing, posing inherent risks to user privacy. Other apps [8] that have attempted to focus on privacy in response to users' growing concerns often lack advanced features, like AI-powered predictions, limiting their appeal to a broader audience.

To address these challenges, the FLORA project introduces an innovative solution that aims to set some general guidelines and good practices on how health applications handle sensitive user data. FLORA is designed as a privacy-first and user-centric ovulation-tracking app that leverages advanced technologies in the field of privacy such as federated learning (FL) [9] and blockchain [10], as well as some key privacy-enhancing technologies (PETs) like fully homomorphic encryption (FHE) [11], differential privacy (DP) [12], and proxy re-encryption (PRE) [13]. Unlike conventional app architectures that centralize user data for analysis, FLORA ensures that sensitive information remains on users' devices, empowering individuals with trust and control over their personal information. The app employs FL to collaboratively train machine learning models locally on user devices, with only encrypted model updates being shared. This decentralized approach minimizes privacy risks while maintaining the accuracy and reliability of predictions.

In addition to federated learning, FLORA incorporates state-of-the-art cryptographic techniques, including FHE and DP, to enhance data privacy during parameters exchange between clients and safeguard the application against model-related attacks, e.g., model inversion [14]. These methods set a high level of privacy level by ensuring that user data remain encrypted even when processed, effectively eliminating the possibility of unauthorized access or data breaches. The app also uses PRE to enable secure and controlled access to encrypted data with explicit user consent, further bolstering privacy protection, even when access to raw data is imposed by other factors, i.e., medical history.

FLORA also investigates the integration of blockchain technology into the FL framework, in order to enhance transparency and accountability. A blockchain-based consent system allows users to manage and audit their data-sharing preferences, ensuring compliance with privacy regulations. The app's Model Vault provides an immutable record of all machine learning model updates on the server side, fostering trust in the system's predictive capabilities. Additionally, FLORA incentivizes user participation through a token-based reward system, which compensates individuals for contributing to the federated learning process.

The proposed architecture of FLORA aims to serve as a reference architecture for digital health application design by addressing critical privacy concerns and aligning with the growing demand for ethical and transparent handling of sensitive health data. The general architecture of FLORA is visually presented in Figure 1. The main contributions of this paper are as follows:

- We systematically evaluate the integration of advanced PETs, like FHE and DP, into an FL framework. Specifically, we focus on PETs' impact on data privacy, predictive

accuracy, and computational overhead, contributing to the optimization of privacy-preserving machine learning models.

- We examine the use of blockchain technology to enhance FL systems, focusing on the role of immutable consent management and the integrity of model updates. This work demonstrates how blockchain can address trust and transparency challenges in distributed systems. Additionally, we introduce a token-based reward system designed to incentivize user participation.
- We integrate FL, PETs, and blockchain technologies into a unified system and evaluate their combined performance in a real-world ovulation-tracking application. The deployment demonstrates the feasibility and effectiveness of privacy-preserving frameworks in addressing sensitive health problems. To the best of our knowledge, this is the first time that the integration and deployment of all these technologies into a single real-world testbed are being discussed in a study.
- Based on direct feedback from end users during the co-design process, this paper describes best practices for designing and deploying federated learning systems in real-world applications. Insights include technologies and architectural choices aimed at balancing privacy, predictive accuracy, and usability.

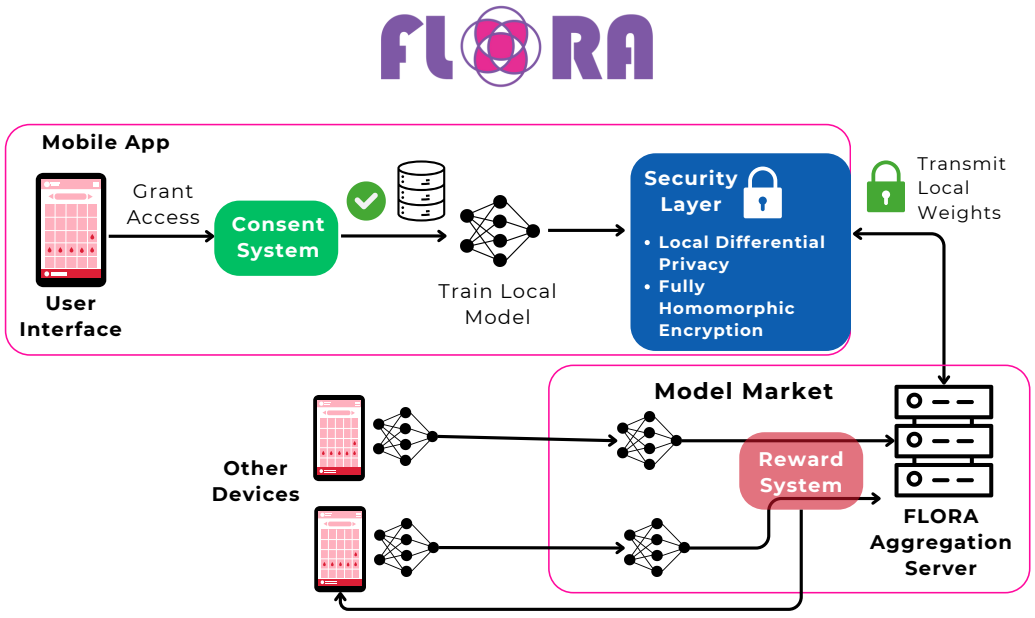

Figure 1. FLORA general architecture.

The rest of this paper is structured as follows. Section 2 describes the detailed architecture and technologies used in every aspect of the application development, including machine learning processes and user engagement strategies to foster end-user co-design. Section 3 presents measurable outcomes of the federated learning framework, comparing its accuracy and computational overhead to centralized counterparts, in both simulation and real-world settings. Section 4 evaluates the advantages of the proposed solution, addressing challenges and limitations. Finally, Section 5 summarizes the work and discusses possible future contributions.

2. System Model

The FLORA project adopts a privacy-centric design approach, integrating cutting-edge technologies to address the challenges associated with safeguarding user data in health applications. This section describes the core methodologies and technical components used in

FLORA, with a focus on the user co-design process, the mobile application design, federated learning framework architecture, cryptographic enhancements, and blockchain integration.

2.1. User Needs Evaluation

Understanding the needs and preferences of the target audience of the application was an essential step for the development of an ovulation-tracking app. The target users of such an application include women of reproductive age who actively engage in managing their reproductive health through digital tools, and healthcare providers who guide these individuals. To gain comprehensive insights, we conducted a multi-step research process that included desk research and a quantitative survey using a structured questionnaire. This holistic approach ensured that we could capture a diverse range of user insights and requirements.

The questionnaire included a series of Likert scale questions, ranging from 1 (strongly disagree) to 10 (strongly agree), to capture user opinions. All the participants were first informed about the scope of the survey and were reassured that their answers would remain anonymous. A total of 40 individuals participated in the questionnaire, representing a diverse demographic profile. The summarized results of the survey are presented in Table 1.

Table 1. The average scores from a user needs evaluation questionnaire, indicating the extent to which respondents agree with the statement.

Statement	Average Score (0 to 10)
1. I use apps regularly to track my menstrual cycle.	7.35
2. I find period-tracking apps to be accurate in predicting my menstrual cycle.	7.12
3. I would like more personalized insights based on my period-tracking data.	7.7
4. How satisfied are you with the current features of ovulation-tracking apps?	6.5
5. Having reminders and alerts related to my period is useful to me.	8.3
6. I would like the app to support other aspects of health and well-being.	7.95
7. I am concerned about my privacy when using ovulation-tracking apps.	6.02
8. I am comfortable with sharing anonymized data to improve app services.	7.3
9. I regularly review and adjust privacy settings in health-related apps.	5.05
10. I feel informed about how my data are used by health apps.	4.27
11. I am likely to adopt a new ovulation-tracking app that prioritizes privacy.	7.07
12. Receiving rewards or incentives for sharing data or models is appealing to me.	6.4

The key insights gathered from the responses can be summarized as follows:

- Participants were asked to rate how important they consider the use of an app for tracking their ovulation and menstrual cycles. Approximately 71% of respondents responded that they found ovulation-tracking apps to be important or very important in managing their reproductive health. This underscores the essential role these apps play in users' lives, providing valuable insights for family planning and menstrual health management. However, around 77% of the respondents stated that the apps they currently use could be improved with new features.
- Respondents were asked about their level of concern regarding the privacy of their health data when using ovulation-tracking apps. Around 60% of participants expressed moderate to high concern over data privacy, highlighting a significant apprehension about how their sensitive information might be handled or shared without consent. However, many users are unaware of how important privacy is in such applications. Also, an alarming 50% of the participants stated that they do not often review the privacy settings in health-related applications.
- Users were questioned about their concerns regarding data sharing with third parties and whether they find it significant to receive rewards or incentives for sharing knowledge (data or models) with third parties. Approximately 65% expressed a significant desire to receive rewards when their personal data are shared with third parties.

Overall, 70% of the users answered that it is highly possible to adopt a new ovulation-tracking app that prioritizes privacy, showing the pressing need for a privacy-first ovulation-tracking app.

2.2. Ovulation Tracking with ML Methods

Machine learning (ML) offers innovative solutions for ovulation tracking by analyzing patterns in menstrual cycle data. One approach involves using time series forecasting algorithms such as AutoRegressive Integrated Moving Average (ARIMA) and Seasonal and Trend decomposition using Loess (STL), which predict future values based on historical data. These algorithms can be trained on past menstrual cycle data to predict the timing of future cycles accurately. Neural networks, which model complex data patterns, can also be used to predict the next period date by analyzing cycle length and symptoms. In addition, ensemble methods like random forest and gradient boosting can classify and predict ovulation dates, and support vector machines (SVMs) can handle classification tasks effectively [15].

Another ML application involves using menstrual cycle tracking apps (MCTAs) that collect real-time data on bleeding days and symptoms. These apps help users track their cycles accurately and provide personalized notifications. A study evaluating the characteristics of MCTA users found that app users, non-trackers, and those using other tracking methods are largely comparable in demographic and menstrual cycle characteristics. This suggests that MCTAs can be reliable tools for a diverse population, enhancing the generalizability of data collected for epidemiological studies [16].

ML also aids in personalized ovulation tracking by incorporating additional relevant data, such as stress levels, weight changes, and symptoms. By analyzing data from multiple users, ML algorithms can identify trends and patterns, improving the overall accuracy of predictions. This personalized approach helps in understanding and managing menstrual cycles more effectively, providing women with better tools for reproductive health management. For instance, a deep learning approach to menstrual cycle tracking has demonstrated enhanced accuracy in predicting ovulation and menstrual dates, thereby empowering women with more precise reproductive health information [17]. Furthermore, a study on unsupervised deep learning applied to longitudinal follicular growth tracking in IVF cycles demonstrates the potential of ML in automating and improving the accuracy of ovulation monitoring. This approach reduces manual errors and provides reliable tracking of follicular growth, which is essential for effective assisted reproduction techniques [18].

In our application, we employed a multi-layer perceptron (MLP), designed to handle a multi-class classification problem. The learning task involves predicting the menstrual cycle regularity of users, represented by three target classes: 0 (regular, 25–35 days), 1 (long, >35 days), and 2 (short, <25 days). The selection of a classification task, rather than a regression one, can be justified due to the limited amount of data and the complexity of ovulation date predictions, as also presented in [19,20].

The MLP model consists of an input layer with 14 features corresponding to the collected dataset's variables. It has a single hidden layer with 64 neurons, where each neuron is followed by a ReLU activation function. The model's parameters are optimized using a cross-entropy loss function, suitable for multi-class classification tasks, and a stochastic gradient descent optimizer with a learning rate of 0.01. The dataset was divided into training and testing sets, with a batch size of 32.

2.3. Dataset

The dataset used in this study was constructed through a combination of real and synthetic data to address challenges related to limited sample size and class imbalance. As

our main focus was to deploy and test a distributed ML framework in a real-world scenario, rather than a machine learning task itself, we used a hybrid solution. The initial dataset consisted of responses from 20 individuals, who completed a comprehensive questionnaire designed to capture key variables related to reproductive health. While the sample size was sufficient to capture the structure of the data, it was necessary to augment the dataset to ensure robustness and avoid bias. The collected data included variables that are described in detail in Table 2.

Table 2. Dataset feature description.

Feature Name	Description
Age	Participant's age in years
Height (SI units - m)	Height in meters
Mass (SI units - kg)	Weight in kilograms
Pregnancy	Whether the participant is currently pregnant
Number of pregnancies	Total number of previous pregnancies
Number of Miscarriages	Number of miscarriages
Number of Abortions	Number of abortions
Race Group	Categorical variable for participant's race
Gyno surgeries	Number of gynecological surgeries undergone
Breastfeeding	Whether the participant is currently breastfeeding
Most recent family planning method	The most recent method used for family planning
Stress level until the last period	Reported stress level before the last period
Sleep quality until the last period	Reported sleep quality before the last period
BMI	Body mass index
Target	Three target classes: 0: regular (25–35 days period), 1: long (>35 days), 2: short (<25 days)

Given the limited sample size and the need for a balanced dataset, two synthetic data generation techniques were applied: SMOTE (synthetic minority oversampling technique) [21] and ADASYN (adaptive synthetic sampling) [22]. These methods were used in conjunction with Gaussian noise augmentation to create diverse and realistic data samples.

SMOTE was employed to generate synthetic samples by interpolating between minority class instances. Gaussian noise was then added to these synthetic samples to introduce variability and improve robustness. This approach is particularly effective for datasets with continuous numerical features, as it prevents overfitting and enhances the model's ability to generalize to unseen data. ADASYN generates synthetic samples by adaptively focusing on instances that are harder to classify. Gaussian noise was similarly added to the ADASYN-generated samples, creating an alternative synthetic dataset with different properties compared to the SMOTE-augmented data. This dual approach allowed us to analyze the impact of synthetic data generation methods on model performance. Gaussian noise augmentation was chosen for its ability to perform the following:

- Improve robustness: Adding noise helps the model generalize to noisy or imperfect data.
- Increase variability: When working with a small dataset, Gaussian noise introduces variability without creating artificial patterns.
- Act as regularization: By exposing the model to slight variations of the same data points, overfitting is minimized.

However, Gaussian noise is less suitable for structured or highly sensitive features (e.g., categorical data) and requires careful application to ensure that the augmented data remain valid.

The characteristics of the augmented datasets are visualized using t-SNE plots (Figure 2), showing the distribution of synthetic samples generated by SMOTE and ADASYN. Both methods effectively expanded the dataset while preserving the original data structure.

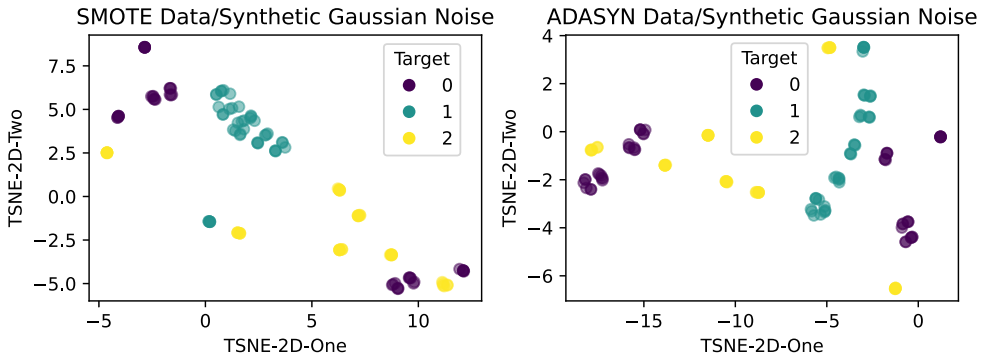

Figure 2. TSNE representation of the synthetic dataset.

After a series of experiments using ML on both synthetic datasets, we found that the SMOTE-augmented dataset exhibited superior performance, with more stable loss and accuracy metrics during training and testing. For this reason, we selected this dataset, consisting of 130 unique records, for the rest of the experiments in this work.

2.4. Mobile Application Design

The mobile application serves as the gateway through which users interact with FLORA. It is developed using *Kotlin*, a language suitable for Android platforms, ensuring a seamless and responsive user experience. The app was designed with a user-first approach, by effectively integrating end-user feedback, emphasizing simplicity and accessibility for a broad demographic of users. Some key screens of the app interface are presented in Figure 3.

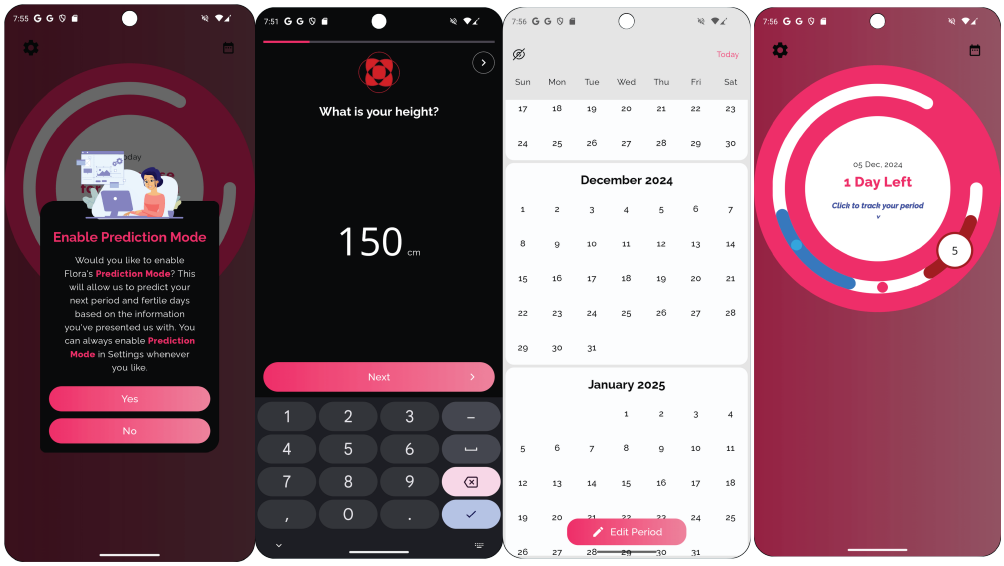

Figure 3. FLORA app interface.

Key features of the mobile application include data input interfaces for logging menstrual cycle information, tracking symptoms, and viewing personalized health insights. The app provides predictive analytics on ovulation and fertility windows, enabling users to make informed decisions about family planning or menstrual health management.

The app also provides users with the option to subscribe to the 'Sharing4Good' mode. This mode is designed to facilitate collective benefits by enabling users to share their raw private data with external actors, such as clinicians and researchers. In this mode, the user's data are securely processed, and the proxy re-encryption (PRE) protocol is employed to ensure the secure transfer of data to external actors. Users can revoke access at any time,

according to their preferences. An overview of how the Sharing4Good mode operates is illustrated in Figure 4.

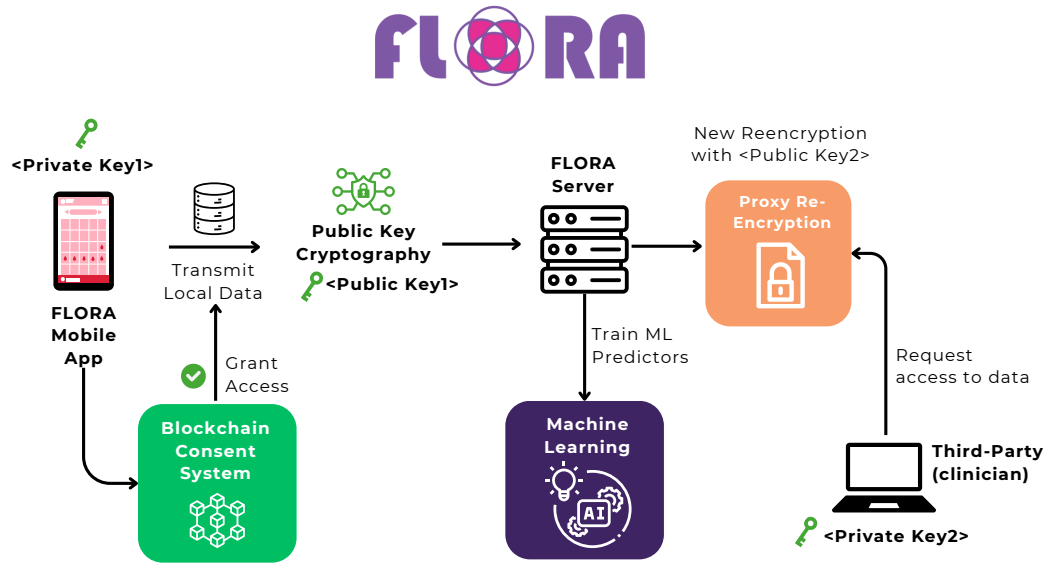

Figure 4. Sharing4Good mode architecture.

To support its functionalities, the app employs a modular architecture based on microservices. This ensures that each feature operates independently, allowing for seamless updates and scalability. The local storage of data is managed using the *Android Room* database, which provides an efficient mechanism for secure, offline data handling. For network communication, the app leverages *Ktor*, a robust framework that ensures secure API interactions, further enhancing the security of data exchanges between the app and the FL orchestrator.

2.5. Federated Learning Architecture

Federated learning [9] is particularly beneficial for health-related applications as sensitive information is not exposed during the training process. Hence, FL holds the potential for generating high-quality and personalized predictions and insights without compromising privacy.

Recent studies have demonstrated the effectiveness of FL in several healthcare applications. For instance, Patel et al. [23] reviewed state-of-the-art techniques for adopting FL in smart healthcare, analyzing market needs, industry trends, and challenges in adopting FL frameworks. Liu et al. [24] proposed a secure and efficient smart healthcare system based on FL, addressing critical issues such as reducing system overhead and authenticating user devices. Nikolaidis et al. [25,26] showed that FL can achieve similar predictive accuracy to centralized settings in the context of predicting early dropouts from healthy aging applications, focusing on patients with Parkinson's disease, while also demonstrating the potential for improving training efficiency and reducing environmental impact.

Despite these recent advances, most studies simulate FL processes, which may not fully reflect real-world scenarios where data are distributed across actual individuals. In addition, to the best of our knowledge, no study integrates FL to enhance user privacy into period-/ovulation-tracking applications, leaving significant room for exploring privacy-preserving techniques in this domain. To address this gap, FLORA introduces a generic, practical, privacy-preserving FL framework that can be integrated into any application (whether web or mobile), using ovulation-tracking applications as a case study, an area that has received limited attention concerning user privacy needs. Moreover, FLORA's

architecture can be integrated with any ML algorithm that can be executed under FL settings such as neural networks, random forests, logistic/linear regression, and gradient boosting algorithms. On top of this, FLORA can be integrated with state-of-the-art PETs like HE and DP, technologies that are already integrated into the platform.

In our federated learning implementation, we began by experimenting with popular Python libraries to establish a robust foundation for our models in a simulation environment. Initially, we utilized *Pandas* and *NumPy* for efficient data processing, manipulation, and analysis. For traditional machine learning algorithms, we will employ *scikit-learn*, which provides a comprehensive suite of tools for learning tasks, and pre-processing methods such as normalization and evaluation metrics for our models. *PyTorch* was used to handle more advanced models, such as artificial neural networks, due to its flexibility and extensive support for deep learning.

During the simulation phase, we incorporated visualizations to analyze data characteristics, as well as to observe the convergence of our models and compare their performance across different metrics.

As we transitioned from simulation to production, our focus shifted to creating a hardware-agnostic training pipeline, with models that can be deployed across platforms with different characteristics, including web and mobile applications. For this, we leveraged *ONNX (Open Neural Network Exchange)* and its runtime, *onnxruntime*, to enable model interoperability and optimize execution on different devices.

In production mode, we also developed a job scheduler—a service orchestrating the FL process. The service distributes the global model to clients, receives locally trained models, and aggregates them to update the global model.

Some screens of the developed web FL orchestrator are presented in Figure 5.

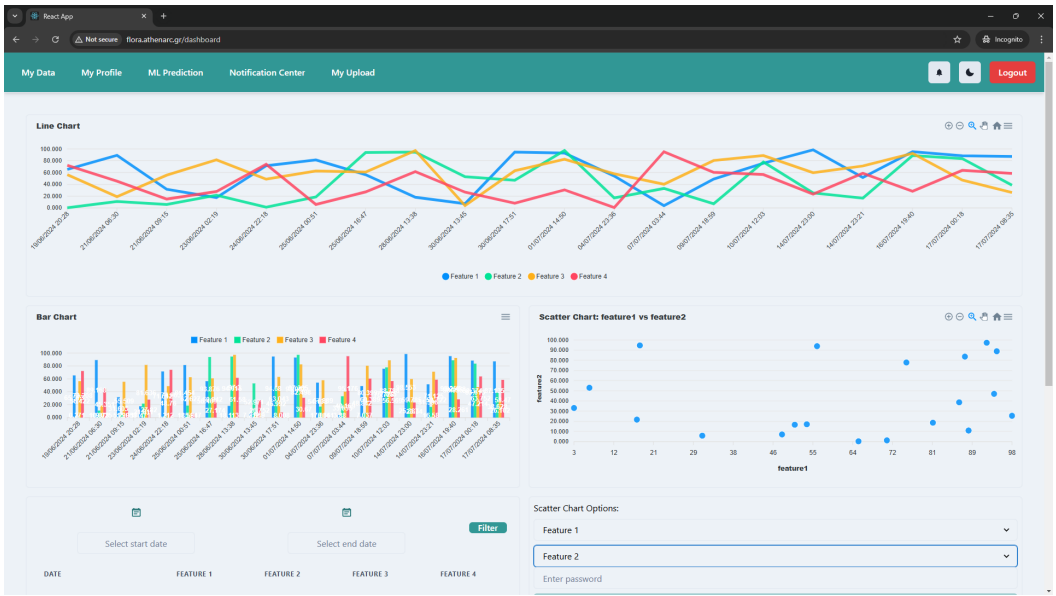

Figure 5. Web FL orchestrator.

2.6. Privacy-Enhancing Techniques

2.6.1. Fully Homomorphic Encryption

Fully homomorphic encryption is a cryptographic technique that allows computations to be performed on encrypted data without requiring decryption. This capability is critical for preserving privacy in scenarios where sensitive data must be processed without exposing the underlying information. Even though FL ensures that raw data remains with the data owner, model updates shared during the training process can still leak sensitive information [27]. We introduce FHE to our FL framework to counter potential inference

attacks to ensure that neither our FL orchestrator nor any other adversary can access local model parameters. Our approach ensures that the aggregation service, operating on encrypted data, performs the aggregation blindly and only clients can decrypt the aggregated outcome.

FHE supports both additive and multiplicative operations, making it ideal for complex computations necessary in machine learning models. Unlike partially homomorphic encryption (PHE), which only allows a limited set of operations on encrypted data (e.g., only addition or only multiplication), FHE permits arbitrary computation sequences, providing a higher level of flexibility [28]. For instance, PHE cannot perform operations such as weighted averaging, thus making FHE the preferred option [29]. Since we are working on floating point numbers, i.e., neural network weights, we will employ the CKKS cryptosystem [30].

Secure aggregation using FHE protocols in FL has been extensively studied in the literature. Hijazi et al. [31] proposed a method for secure aggregation on IoT devices using FHE, where model weights are encrypted before being sent to the server, and only users (with access to the private key) can decrypt the aggregated result. Similarly, SafeLearn [32] employs a multi-party FHE scheme, where the secret key is divided among clients, allowing each client to decrypt the global model after the encrypted aggregation process. Finally, Ref. [28] introduced a protocol that utilizes a multi-party key variant of the CKKS cryptosystem for secure aggregation.

Despite these advancements in ensuring privacy and security through homomorphic encryption, most research efforts have been conducted using simulation devices. Hence, overseeing the computational overhead introduced by FHE protocols.

In our FL implementation, participants encrypt their model updates (i.e., weights) using a shared public key before transmission to the central server. The server aggregates these encrypted updates without decrypting them, thus preventing any potential data leakage while ensuring privacy. The encrypted global model, aggregated by the server, is then sent back to participants who decrypt it using their private keys to continue training. Overall, the secure aggregation process ensures the privacy, accuracy, and integrity of the machine learning models [33].

To facilitate this, we use Microsoft's Simple Encrypted Arithmetic Library (*SEAL*) to incorporate FHE, specifically the *CKKS* cryptosystem, into our FL setup. SEAL is an open-source library that supports several FHE schemes and is optimized for several platforms, including Android applications [34]. By incorporating SEAL into our FL system, we maintain a platform-agnostic solution that can be deployed across a wide range of devices and applications without compromising security or performance.

FLORA employs a trusted secure server for key generation for consistency and integrity in key management. Specifically, the following workflow is implemented:

1. A trusted secure server generates a public–private key pair for the CKKS cryptosystem using the SEAL library.
2. The private key is securely transmitted to clients that participate in the FL process via a secure channel, with end-to-end encryption during transmission.
3. The public key is stored both on the server and the clients, allowing for encrypted aggregation.
4. After local model training, each client encrypts the model parameters using the provided public key.
5. The encrypted parameters are sent to the FL orchestrator for aggregation. Note that no plaintext data leave the client device.
6. The FL server aggregates the encrypted updates using homomorphic operations, i.e., it performs blind aggregation using the weighting averaging technique of the FedAvg algorithm [9]. The CKKS natively supports operations on floating-point

numbers, enabling computations of encrypted neural network weights, which is ideal, in our context.

7. The aggregated encrypted global model is transmitted to the clients selected for the subsequent federated round. The clients then decrypt the aggregated model using their shared private key. Steps 4–7 are repeated until model convergence.

A single point of failure could be the trusted secure server. To prevent possible unauthorized access, this server is isolated and protected by access control mechanisms to ensure that private keys remain confidential. In addition, periodic key rotation is implemented to improve security and minimize risks associated with compromised keys.

2.6.2. Differential Privacy

While secure aggregation techniques using FHE or similar approaches enable blind aggregation, which enhances client privacy, the risk of inferring additional information from the aggregated global model still persists [35]. To further strengthen data privacy and mitigate these risks, differential privacy (DP) techniques are employed. DP techniques may add noise at different stages of the machine learning pipeline; either to the original data, during model training, after training, after aggregation, or using a combination of these methods [36]. However, the introduction of noise can hinder convergence, and finding the optimal balance between privacy and model utility (accuracy) remains a challenge.

DP has gained significant attention due to its strong mathematical foundation and theoretical privacy guarantees. In the context of FL, Yang et al. [37] introduced a novel approach that balances model accuracy with privacy through an adaptive DP mechanism. Similarly, Wei et al. [38] proposed a DP-based federated framework focused on optimizing the trade-off between convergence and privacy levels. Hidayat et al. [39] introduced an adaptive Gaussian clipping method in FL to enhance privacy while maintaining model accuracy, which complements the goals of dynamic adaptive differential privacy. However, many recent studies have attempted to optimize the trade-off between privacy and utility using simulation approaches and re-executing federated training from scratch, which is impractical in real-world applications. In our context, we will first simulate DP scenarios and then deploy the most viable one based on the privacy-utility trade-off, prioritizing approaches that require minimal effort (i.e., avoiding extensive hyper-parameter search).

In our approach, we started by simulating different combinations of DP techniques to identify the most suitable for our practical application. We explored both local DP (LDP) methods, where noise is added under the client's control on the model parameters during training, and central DP (CDP) methods, where the server introduces noise to the aggregated output, as well as combinations of both approaches. Based on the insights gained from these simulations, particularly the measurements of the privacy-utility trade-off, we selected LDP as the mechanism for ensuring differential privacy, which provides stronger privacy guarantees than CDP. From the experiments, it is evident that LDP improves privacy without heavy costs on the model's predictive performance. Specifically, we employed several frameworks, each with unique characteristics, to integrate the following DP mechanisms: OpenMined's PyDP for noise addition into the original data, IBM's Diffprivlib for integrating DP into simple models like linear regression, and PyTorch's Opacus for noise addition during neural network model training.

2.6.3. Proxy Re-Encryption

Data transmission and secure storage are critical aspects of data management, especially in scenarios requiring stringent security measures. Ensuring that data are encrypted during transmission and that only the data owner can access the data later (e.g., in the case of lost access to local data) is essential for maintaining continuous application usability.

Traditionally, when only secure storage is required, symmetric encryption algorithms like AES are commonly used. In this approach, the client generates a secret key derived from a personal secret, such as the user's password. The data are then encrypted and the ciphertext is transmitted to the server. The server cannot infer any additional information from the ciphertext and only the user can decrypt it using the underlying secret key.

However, in more complex scenarios, such as those where users can choose whether another entity, e.g., a service provider or another user can access their data with their consent, traditional symmetric encryption approaches are insufficient, since clients have to share their private key, breaking the concept of key confidentiality. In our context, users have the option to participate in the Sharing4Good mode, where they can send their data, and with their consent, clinicians might access the data for further analysis. The data must remain encrypted to prevent potential data leakage that could reveal sensitive user information. Clinicians can only access the data with the user's explicit consent.

To address this challenge, we utilize proxy re-encryption [13], a public-key cryptosystem that allows data encrypted on the user's side to be transformed into another ciphertext that can be decrypted under a different secret key (in this case, the FLORA backend's key).

To this end, users generate a public–secret key pair, with the secret key kept local and never transmitted. They encrypt their data and send them to the server, where the data remain secure as they cannot be decrypted without the secret key. When the FLORA backend or another entity needs access to specific user data, they request the user's consent. If the user agrees, they generate a re-encryption key using their secret key and the entity's public key. This re-encryption key does not reveal any information about the original secret key but allows the third party to access the data. The re-encryption key, along with the encrypted data, can be transmitted to a proxy for re-encryption. This process does not expose any details about the underlying plaintext; the re-encrypted ciphertext can only be decrypted by the entity that requested access using its secret key.

In recent years, PRE has been used for both secure storage and to secure transactions in machine and FL pipelines. Keshta et al. [40] proposed a blockchain-based data-sharing scheme using PRE to enhance privacy protection and access control in distributed environments. The proposed system divides blockchain nodes into different roles that independently manage re-encryption key parameters, thus facilitating the dynamic and secure updating of access rights. In the context of FL, Shen et al. [41], introduced a privacy-preserving multiparty deep learning scheme leveraging a novel homomorphic PRE approach to ensure user privacy in deep learning processes. The scheme is primarily designed to thwart collusion between semi-honest servers by employing a fog node as a proxy, which utilizes a one-way homomorphic proxy re-encryption scheme to securely transform user-end ciphertext into server-end ciphertext. Finally, Zhang et al., in the context of homomorphic encryption-based aggregation in FL, presented a method that allows for fine-grained control over users who can access re-encrypted homomorphic encryption keys, ensuring that only authorized participants can access underlying keys [42].

In the FLORA project, we implemented PRE using NuCypher's Umbral PRE library, which supports modern elliptic-curve cryptography. Specifically, the workflow for the PRE mechanism in FLORA is as follows:

1. Each user generates a public–private key pair, automatically, during app registration.
2. The private key is securely stored on the user's device and never transmitted, while the public key is directly sent to the FLORA server. Note that the public key does not expose any information about the associated private key.
3. When a third party (e.g., a clinician) requests access to a user's data, the user is presented with a notification to provide their explicit consent. If the user declines the request, the process stops and the third party cannot observe any user's data.

On the other hand, if the user accepts the data access request, a re-encryption key is generated locally using their private key and the third party's public key.

4. The re-encryption key is transmitted to the FLORA proxy server. Note that this key does not reveal anything about the user's private key.
5. The proxy server re-encrypts the ciphertext without accessing the plaintext and transforms it into a format that can be decrypted using the third party's private key.
6. The proxy server transmits the re-encrypted ciphertext and then, the third party decrypts the re-encrypted data using their private key.

2.7. Blockchain Integration

FLORA leverages blockchain technology as a robust solution to enhance transparency and foster user trust. The blockchain framework is centered around three components: a secure model vault to store the FL server's global models, an advanced consent management system, and a blockchain-based reward system for incentivizing users to participate in the FL process.

The Model Vault is designed to be a secure, blockchain-based repository for storing, managing, and sharing ML models within the context of FL. The primary objective of the Model Vault is to provide a tamper-proof environment where ML models can be stored securely and shared among authorized participants. By using blockchain technology, we ensure that every model version is recorded immutably, offering an auditable history of model development. This solution will facilitate secure collaboration among researchers and developers, ensuring that models can be trusted and used across different stages of federated learning processes. This feature is particularly critical in medical applications, where it is essential to ensure that models used in healthcare decisions have not been tampered with. The application of blockchain technology in managing and securing ML models, particularly within federated learning frameworks, is an emerging area of research. Blockchain has been increasingly adopted to ensure the integrity, traceability, and immutability of stored models. For instance, Kleinaki et al. [43] introduced a blockchain-based notarization service specifically aimed at providing secure and verifiable storage of biomedical data. This approach is especially relevant in healthcare, where maintaining the integrity of machine learning models is critical to ensuring reliable patient outcomes. However, when considering the storage of models on public blockchains, several factors must be taken into account. Refs. [44,45] highlights the trade-offs between cost and performance in Ethereum-based decentralized applications, particularly when dealing with high-volume data like machine learning models. While blockchain offers unparalleled security and transparency, the cost of transactions and storage can be a challenge. For this reason, the use of a hybrid approach, where off-chain storage solutions like distributed file systems such as IPFS or Swarm are combined with on-chain hash verification, is recommended to balance security with cost-effectiveness. This approach emphasizes the need to carefully select the blockchain infrastructure based on specific application requirements. In our system, we have implemented both a solution where everything is stored on the blockchain, which can be used in private EVM-based systems where the cost does not affect us, as well as a hybrid architecture utilizing IPFS and Ethereum, to adapt to the needs of each different case.

Moreover, FLORA introduces a blockchain-based reward system designed to incentivize user participation and data contributions within collaborative ML and FL frameworks. Blockchain-based reward systems have emerged as a promising solution for incentivizing participation in FL and ML [46]. These systems use blockchain tokens to reward participants based on their contributions, such as data provision and computational resources. Smart contracts automate the issuance and management of these tokens, ensuring

transparency and fairness in reward distribution. In healthcare, blockchain rewards are particularly valuable, promoting user participation in a transparent and incentivized manner. For instance, Chen et al. [47] explored blockchain's role in FL to incentivize patient data contributions, which are crucial for developing accurate ML and FL models while maintaining data privacy. Similarly, Refs. [48,49] highlight the potential of blockchain tokens in rewarding participants fairly in FL setups, ensuring that contributors are compensated based on the value of their input. Additionally, the integration of blockchain in FL systems helps prevent dishonest behavior, such as data tampering or unfair token distribution, by ensuring accurate records and fair rewards, thereby enhancing trust in FL applications. To ensure that, FLORA evaluates user contributions using a reputation-based mechanism inspired by Shapley Value approximations [49,50]. This system dynamically measures the impact of individual contributions on the overall model performance, preventing misuse and ensuring that rewards are distributed fairly based on data quality. In FLORA, each participating user is rewarded with a blockchain token as evidence of their contribution, whether through engaging in FL or sharing data with a doctor or server for ML purposes. These tokens are non-exchangeable and include mechanisms for revocation in cases of dishonest behavior. Users can later redeem these tokens for third-party services, such as free appointments with doctors. To preserve user anonymity in reward transactions, FLORA employs a "use-and-burn" token mechanism, where tokens are immediately burned upon usage, ensuring no persistent on-chain record exists that could reveal user activity patterns. These transactions are conducted within a zk-rollup environment on Polygon zkEVM [51], an off-chain aggregation solution that uses cryptographic proofs to validate transactions on-chain. Specifically, each token is minted within this environment and burned instantly upon redemption, while a corresponding zero-knowledge proof is generated and submitted to Ethereum. This approach confirms the validity of the transaction without exposing sensitive details, leveraging the low-cost batch processing capabilities of zk-rollups. FLORA's blockchain implementation utilizes a customized ERC-20 token called FLORA Token (FLT). FLT has no subdivisions, ensuring each token represents an indivisible unit of contribution. Additionally, FLT is designed with non-transferable functionality, inspired by the concept of soulbound tokens (SBTs), ensuring tokens remain bound to their original recipients.

FLORA will also integrate ConInSe [52], a blockchain-based consent system, to manage and record user consent securely and transparently. ConInSe, developed as part of the TRUSTCHAIN project, is designed to empower users with full control over their data-sharing preferences, ensuring that their consent decisions are immutably recorded on the EVM-based blockchain. To further enhance user anonymity and prevent potential data leaks, we deployed it on Polygon's zkEVM. This architecture processes consent transactions off-chain, generating zero-knowledge proofs that validate these actions without exposing sensitive details. This consent system empowers users to make informed choices about their data, fostering trust in the app's privacy policies. In medical applications, the use of blockchain-based consent management systems has been proposed [53], allowing doctors or researchers to request user data while ensuring that users maintain full control over whom they grant access to and for what duration, ensuring both transparency and alignment with GDPR requirements. In our application, doctors or researchers can request access to user data or participate in FL models. Users decide whom to grant access to and for how long, thus maintaining full control over their data-sharing decisions. By providing a transparent and auditable record of consent transactions, FLORA enhances user accountability and prevents unauthorized data usage. Our integration of ConInSe also includes features such as allowing users to select specific data they wish to share, as well as incorporating the data in our reward system. All of these options are seamlessly integrated into the mobile

application via an easy-to-use interface, enabling users to manage their consents effortlessly and reinforcing FLORA's user-centric approach.

2.8. Integration and System Architecture

The FLORA architecture adopts a modular design based on microservices, ensuring flexibility, scalability, and ease of integration. The system integrates key components such as federated learning, privacy-enhancing technologies, blockchain, and mobile applications through well-defined APIs and containerized environments. Figure 6 provides an overview of the system's flow and interactions among its components.

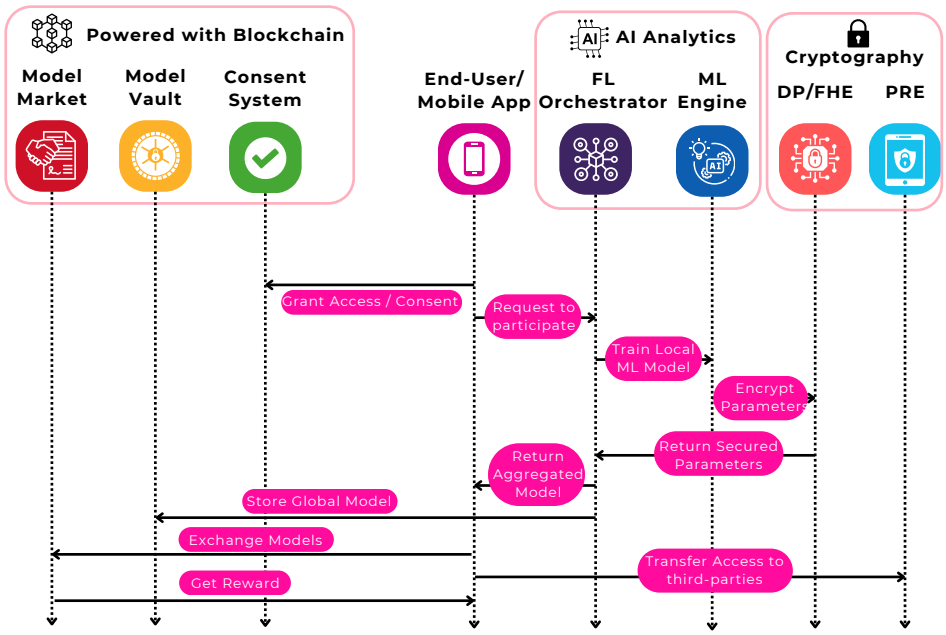

Figure 6. FLORA interaction flow diagram.

The system is designed as a collection of microservices, each responsible for a specific functionality, including:

- **Federated learning orchestrator:** Manages the distribution of models to user devices, aggregation of local updates, and synchronization of the global model.
- **Cryptographic engine:** Implements privacy-enhancing technologies (PETs) such as differential privacy (DP), fully homomorphic encryption (FHE), and proxy re-encryption (PRE) to secure data and model parameters.
- **Blockchain layer:** Handles user consent management, model storage (via the Model Vault), and token-based reward distribution through smart contracts.

This microservices architecture allows independent development, deployment, and scaling of each component, ensuring the system can evolve and adapt to future requirements.

Each microservice is containerized using Docker, enabling consistent environments across development, testing, and production. Docker containers encapsulate the required dependencies, libraries, and runtime, ensuring seamless deployment across different infrastructure setups. This approach enhances the system's portability and reliability, particularly when scaling across multiple servers or cloud environments.

The integration between components is facilitated by RESTful APIs, providing clear interfaces for communication in the following ways:

- The federated learning orchestrator interacts with user devices via APIs to distribute models, receive encrypted updates, and provide global model feedback. Specifically,

the users are connected via WebSocket, so the server can be aware of the users' availability. The orchestrator samples at least three clients or the 10% of available (if more than 3) and sends them a trigger message in the WebSocket. Then clients receive the model via an API endpoint, train it locally, and distribute it into an API endpoint.
- The blockchain layer exposes APIs for recording consent transactions, retrieving model metadata, and processing token rewards.
- The cryptographic engine communicates with the orchestrator through APIs to apply encryption, decryption, and re-encryption processes seamlessly.

These APIs ensure interoperability and simplify the integration of new functionalities or third-party services.

2.9. Privacy and Security Analysis

As mentioned in the previous sections, FLORA integrates advanced PETs, i.e., FHE, DP, and PRE to minimize user data exposure.

FHE is integrated to ensure that local model updates are never exposed to the aggregator server, enabling blind aggregation and minimizing potential inference attacks.

Formally, let w_i^t represent the model weights in plaintext format for client i at federated round t. The encryption function is denoted as $E_k(\cdot)$, where k is the public key from the CKKS cryptosystem, and the decryption function as $D_c(\cdot)$, where c is the shared private key. FHE ensures the following:

$$D_c(f(E_k(w_i^t), E_k(w_j^t), \ldots)) \approx f(w_i^t, w_j^t, \ldots),$$

where f represents a computation, in our context, the aggregation function. In FLORA, the server receives encrypted updates $E_k(w)$ from each client, aggregates them blindly, and returns $E_k(w_{\text{global}})$, where w_{global} is the updated global model.

Hence, the orchestrator or any other intermediate adversary cannot infer W_i^t or any intermediate computation without the decryption key c, ensuring confidentiality in local model updates. Since FHE enables the aggregation of encrypted data, it effectively mitigates inference attacks that exploit intermediate model updates.

DP provides an additional layer of protection against information leakage by introducing statistical noise into the local models, just before encryption, on the clients' side. Note that we selected this approach after experiments with the CDP and LDP approaches.

In general, DP ensures that the inclusion or exclusion of any single data point $x \in \mathcal{D}$ for a $\mathcal{D}$ does not significantly modify the output of a function $f(\mathcal{D})$. For any two datasets, $\mathcal{D}$ and $\mathcal{D}'$, differing by at most one element ($\|\mathcal{D} - \mathcal{D}'\|_1 \leq 1$), and for all possible outputs $S \subseteq \text{Range}(f)$:

$$\Pr[f(\mathcal{D}) \in S] \leq e^{\epsilon} \Pr[f(\mathcal{D}') \in S] + \delta,$$

where $\epsilon > 0$ is the privacy budget, and $\delta > 0$ accounts for the probability of failure.

The FLORA system employs LDP, where noise is added to model updates on the client side before encryption and transmission to the central server.

- Noise $\eta \sim \mathcal{N}(0, \sigma^2)$ to the local model parameters w_i^t, resulting in $(w_i^t)' = w_i^t + \eta$.
- The variance σ^2 is calibrated based on ϵ and the sensitivity Δw of the model updates the following:

$$\sigma^2 = \frac{2\ln(1.25/\delta)\Delta w^2}{\epsilon^2}.$$

In our context, DP ensures that individual user contributions are obfuscated within the aggregated model. Thus, DP effectively guarantees that adversaries attempting to extract

individual-level information from the global model cannot infer any additional knowledge about individual contributions, bounded by (ϵ, δ).

Beyond FHE and DP, which minimize privacy risks during the FL process, FLORA integrates **PRE**, which allows users to participate in the 'Sharing4Good' mode. In this mode, users willing to contribute to the community by providing additional data for algorithmic development or users who wish to provide data to their clinicians (e.g., for health-related reasons), are allowed to share their submitted data from the app with their selected external party under strong security guarantees.

PRE allows encrypted user data to be transformed into ciphertexts, which are decryptable by authorized third parties, such as clinicians, only with the user's explicit consent. Specifically, let the user generate a key pair (sk_u, pk_u), and an external party, e.g., a clinician, owns a different key pair (sk_c, pk_c). The user encrypts their data $\mathcal{D}$ as follows:

$$C_u = E_{pk_u}(\mathcal{D}).$$

The third party (clinician) sends a data access request to the user. To allow the access, the user generates a re-encryption key $r_{u \to c}$ using:

$$r_{u \to c} = f(sk_u, pk_c),$$

where f is a cryptographic function that binds sk_u and pk_c. The proxy uses $r_{u \to c}$ to transform C_u into $C_c = E_{pk_c}(M)$, which is decryptable under the third party's key sk_c.

Without sk_u or sk_c, neither the proxy nor any adversary can decrypt C_u or C_c, ensuring data confidentiality and controlled sharing. The above mechanism ensures that sensitive user data remain inaccessible to unauthorized parties and enables secure data sharing with trusted entities.

The combination of FHE, DP, and PRE measures ensures that the FLORA application effectively protects user privacy and mitigates risks associated with data sharing and collaborative learning in a federated environment.

3. Results

This section presents the outcomes of machine learning experiments comparing centralized and federated learning (FL) approaches for a multi-class classification task. Key metrics include prediction accuracy and convergence, as well as an analysis of FL's practical deployment characteristics such as latency and resource efficiency.

3.1. ML Predictive Accuracy-Simulation Mode

The centralized learning setup achieved an accuracy of 81%, compared to 78% for standard federated learning, without any use of extra encryption techniques. While the centralized approach benefits from direct access to the entire dataset, FL demonstrated comparable performance despite its decentralized nature, highlighting its potential as a privacy-preserving alternative.

The experiments utilized 20 epochs for centralized learning and 10 federated rounds of 2 local epochs to ensure an equal number of data accesses across both setups. This ensured a fair comparison of the two approaches under similar training conditions.

Figure 7 presents the convergence curves for the different learning setups. The key points that can be identified are as follows:

- **Centralized learning:** Exhibited faster and more stable convergence, with both training and testing loss curves stabilizing by epoch 15.

- **Federated learning:** Showed a slightly slower convergence rate, with minor fluctuations in test loss across the federated rounds. This can be attributed to the non-IID (non-independent and identically distributed) nature of local data on participating devices.

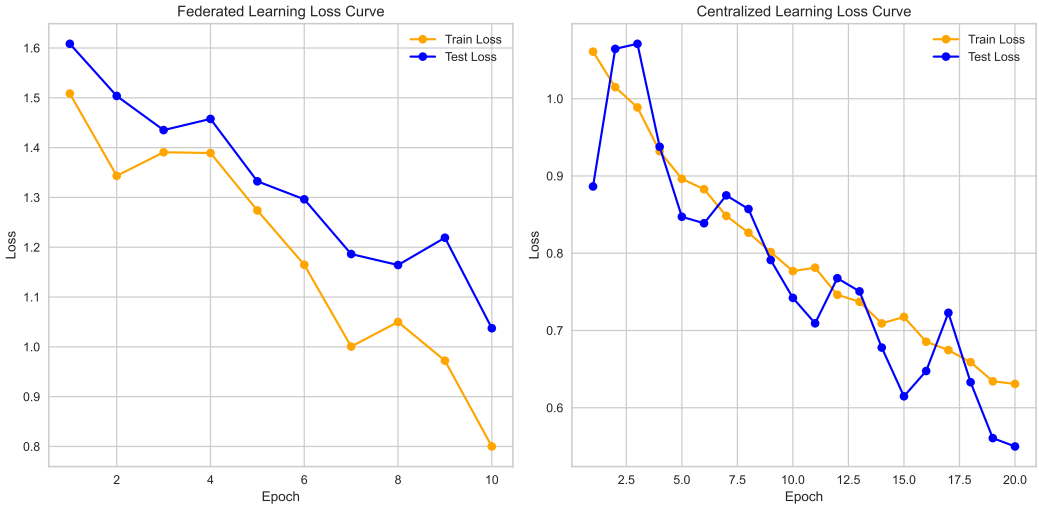

Figure 7. Loss curves for centralized and federated setups.

Despite these differences, the small accuracy gap (3%) between centralized and federated learning underscores FL's viability for real-world applications where privacy is paramount.

3.2. FL Framework Performance-Actual Deployment

The FL framework was deployed in a real-world environment involving four end-user devices. The deployment setup was designed to mimic realistic conditions, ensuring that the results provide valuable insights into the framework's practical feasibility. The evaluation focused on key performance metrics, including training time and predictive accuracy. This section compares the performance of centralized learning, standard FL, and FL with FHE and DP.

Each client device was assigned an equal share of the dataset, ensuring a balanced distribution of data instances across the four devices. Specifically, each client processed approximately 25% of the total dataset, which included artificially generated data to address the limited availability of real-world instances.

The deployment assumed a theoretical uniformity in hardware specifications across all devices. While the devices were not physically distinct, the framework emulated a consistent computational environment, including processing speed, memory availability, and network latency, to ensure fairness in the evaluation. This theoretical uniformity allowed the results to focus on the performance of the FL framework itself, independent of hardware variability.

Table 3 summarizes the performance metrics, including execution time per epoch and predictive accuracy, for the four configurations. Centralized learning demonstrated the fastest execution time, achieving 0.10 s per epoch with a predictive accuracy of 81%. Plain federated learning incurred a marginal increase in execution time, reaching 0.12 s per epoch with accuracy ranging from 78%. Federated learning with FHE required 0.14 s per epoch due to the additional computational overhead of encryption and decryption processes, achieving an error rate of 77%, very close to the standard FL. As anticipated, FL with DP performed worse (74%), due to the associated inserted noise.

Table 3. Execution time per epoch and error rate comparison in real deployment.

Configuration	Execution Time per Epoch	Accuracy
Centralized Learning	0.10 s $\pm$ 0.01 s	81%
Standard Federated Learning	0.12 s $\pm$ 0.02 s	78%
Federated Learning with FHE	0.14 s $\pm$ 0.02 s	77%
Federated Learning with DP	0.12 s $\pm$ 0.03 s	74%

Execution time per epoch reflects the computational demands of each configuration. Centralized learning achieved the shortest execution time because it directly accessed the entire dataset without the need for distributed updates or encryption. Federated learning added latency due to its reliance on local training and server aggregation, while federated learning with FHE and DP introduced further delays as a result of encryption and decryption operations. Despite these differences, the training times for all configurations remained efficient and suitable for practical deployment.

The accuracy across the configurations indicates the robustness of the models. Federated learning configurations achieved comparable predictive accuracy to centralized learning, while FHE and DP introduced a slight performance degradation due to the computation noise.

The capacity of the FL framework to handle varying workloads was tested by increasing the number of participating devices from two to four. The results demonstrated a linear increase in latency per round as the number of devices grew, indicating the framework's ability to manage additional devices efficiently. Importantly, this behavior was achieved without significant degradation in accuracy or computational performance, validating the system's design for practical real-world applications.

In conclusion, the FL framework demonstrated strong performance across all key metrics, balancing accuracy, efficiency, and privacy preservation. Centralized learning achieved the fastest execution times, while FL configurations offered comparable accuracy with added privacy guarantees. The integration of FHE and DP enhanced security but with a slight increase in computational overhead. These findings confirm the feasibility of federated learning as a practical and scalable solution for privacy-preserving applications in real-world health settings. The system's performance metrics highlight its potential to address privacy concerns while maintaining robust predictive capabilities.

4. Discussion

The results of the FLORA project illustrate its innovative approach to addressing privacy concerns in digital health applications through the integration of federated learning (FL), privacy-enhancing technologies (PETs), and blockchain. These results reveal significant insights into the balance between privacy preservation and functionality and the practical challenges and opportunities presented by deploying FL in real-world settings.

4.1. Implications of the Proposed Architecture

FLORA successfully demonstrates the feasibility of utilizing federated learning to preserve user privacy without sacrificing predictive performance. The decentralized nature of FL ensures that sensitive personal data remain on user devices, mitigating the risks associated with centralized data storage. Furthermore, including PETs, such as fully homomorphic encryption (FHE) and differential privacy (DP), enhances data security by protecting model updates against potential privacy attacks. Blockchain integration further bolsters transparency by providing immutable consent tracking and verifiable model update histories.

These features collectively highlight the potential of FLORA's architecture to address longstanding challenges in digital health. The framework's dual operational modes allow users to tailor their privacy preferences, fostering trust and engagement. Additionally, the real-world deployment of FLORA illustrates that advanced privacy-preserving technologies can achieve practical scalability while maintaining acceptable latency and overhead.

4.2. Challenges and Areas for Improvement

The computational overhead introduced by FHE and other cryptographic measures remains a significant challenge. While strategies such as compression and hardware optimization mitigated these issues, further work is required to streamline these technologies for broader, large-scale applications. Additionally, the variability in device capabilities and network conditions occasionally affected the consistency of local model training during real-world tests. This highlights the need for adaptive algorithms capable of dynamically adjusting training workloads.

Another challenge encountered was user education. Privacy-preserving technologies are often complex, and many users lack familiarity with terms like federated learning or homomorphic encryption. Simplified explanations and interactive interfaces could bridge this knowledge gap, ensuring informed participation and greater trust in the system.

4.3. Broader Implications

FLORA's design demonstrates that it is possible to combine privacy preservation and practical application in digital health, setting a benchmark for the development of privacy-friendly systems across other domains. The framework is particularly relevant for applications involving sensitive data, such as mental health monitoring or chronic disease management, where privacy concerns are paramount.

Moreover, the token-based reward system highlights an ethical model for incentivizing user participation. By aligning user benefits with collective goals, FLORA offers a sustainable approach to data sharing that avoids commodification or exploitation.

5. Conclusions

FLORA represents a significant step forward in the development of privacy-preserving health applications. By combining federated learning, advanced cryptographic techniques, and blockchain, the project demonstrates that it is possible to deliver predictive functionality while safeguarding user privacy. The system's flexibility and scalability, validated through real-world deployment, underscore its potential as a viable solution for sensitive digital health domains.

While FLORA achieved promising results, its challenges underscore the need for further refinement. Reducing the computational demands of privacy-preserving technologies, improving user interfaces for privacy education, and enhancing system adaptability are critical next steps. Additionally, integrating FLORA with broader healthcare infrastructures, such as electronic health records and clinical systems, could expand its utility and impact.

Beyond ovulation tracking, FLORA's architecture provides a framework for other privacy-critical applications. The project's results illustrate how ethical and secure data practices can be aligned with advanced functionality, offering a roadmap for the future of digital health technologies. In a landscape increasingly defined by the trade-offs between data utility and user privacy, FLORA sets a standard for how both goals can be achieved simultaneously.

Author Contributions: N.P.: Conceptualization, methodology, validation, formal analysis, writing—original draft, review and editing, visualization, investigation; A.S.: conceptualization, methodology, validation, formal analysis, software; T.T.: methodology, validation, software, data curation, visualization; P.K.: methodology, validation, software; C.K.: methodology, validation, software, visualization; E.B.: methodology, validation, investigation; C.C.N.: methodology, validation, investigation; V.P.: conceptualization, methodology, validation, writing—review and editing, G.D.: conceptualization, methodology, validation, supervision; E.K.: conceptualization, methodology, validation, supervision, P.S.E.: conceptualization, methodology, validation, supervision, project administration; D.E.F.: conceptualization, methodology, validation; A.M.: conceptualization, methodology, validation, project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by "Cascade Funding" received by the H2020 project entitled "TRUSTCHAIN—Fostering a Human-Centred, Trustworthy and Sustainable Internet" (Grant Agreement N° 101093274 with the European Commission).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The implementation code and all related data used in this study are publicly available at the following GitHub repository: https://github.com/FLORA-TRUSTCHAIN/FLORA_Ovulation_Tracker.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Triantafyllidis, A.K.; Tsanas, A. Applications of machine learning in real-life digital health interventions: Review of the literature. *J. Med. Internet Res.* **2019**, *21*, e12286.
2. Johnson, S.; Marriott, L.; Zinaman, M. Can apps and calendar methods predict ovulation with accuracy? *Curr. Med. Res. Opin.* **2018**, *34*, 1587–1594.
3. Shipp, L.; Blasco, J. How private is your period?: A systematic analysis of menstrual app privacy policies. *Proc. Priv. Enhancing Technol.* **2020**, , 491–510.
4. Parker, L.; Halter, V.; Karliychuk, T.; Grundy, Q. How private is your mental health app data? An empirical study of mental health app privacy policies and practices. *Int. J. Law Psychiatry* **2019**, *64*, 198–204.
5. Cao, J.; Laabadli, H.; Mathis, C.H.; Stern, R.D.; Emami-Naeini, P. "I Deleted It After the Overturn of Roe v. Wade": Understanding Women's Privacy Concerns Toward Period-Tracking Apps in the Post Roe v. Wade Era. In Proceedings of the CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 11–16 May 2024; pp. 1–22.
6. Malki, L.M.; Kaleva, I.; Patel, D.; Warner, M.; Abu-Salma, R. Exploring Privacy Practices of Female mHealth Apps in a Post-Roe World. In Proceedings of the CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 11–16 May 2024; pp. 1–24.
7. Earle, S.; Marston, H.R.; Hadley, R.; Banks, D. Use of menstruation and fertility app trackers: A scoping review of the evidence. *BMJ Sex. Reprod. Health* **2021**, *47*, 90–101.
8. Song, Q.; Hernandez, R.H.; Kou, Y.; Gui, X. "Our Users' Privacy is Paramount to Us": A Discourse Analysis of How Period and Fertility Tracking App Companies Address the Roe v Wade Overturn. In Proceedings of the CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 11–16 May 2024; pp. 1–21.
9. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, PMLR, Ft. Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
10. Nguyen, D.C.; Ding, M.; Pham, Q.V.; Pathirana, P.N.; Le, L.B.; Seneviratne, A.; Li, J.; Niyato, D.; Poor, H.V. Federated learning meets blockchain in edge computing: Opportunities and challenges. *IEEE Internet Things J.* **2021**, *8*, 12806–12825.
11. Gentry, C. *A Fully Homomorphic Encryption Scheme*; Stanford University: Stanford, CA, USA, 2009.
12. Dwork, C. Differential privacy. In Proceedings of the International Colloquium on Automata, Languages, and Programming, Venice, Italy, 10–14 July 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1–12.
13. Green, M.; Ateniese, G. Identity-based proxy re-encryption. In Proceedings of the Applied Cryptography and Network Security: 5th International Conference, ACNS 2007, Zhuhai, China, 5–8 June 2007; Proceedings 5; Springer: Berlin/Heidelberg, Germany, 2007; pp. 288–306.
14. Huang, Y.; Gupta, S.; Song, Z.; Li, K.; Arora, S. Evaluating gradient inversion attacks and defenses in federated learning. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 7232–7241.

15. Thakur, T.; Kadam, S.; Patil, N.; Achrekar, C. Machine Learning in Period, Fertility and Ovulation Tracking Application. *TechRxiv* **2023**. https://doi.org/10.36227/techrxiv.22041683.v1.
16. Adnan, T.; Li, H.; Peer, K.; Peebles, E.; James, K.; Mahalingaiah, S. Evaluation of Menstrual Cycle Tracking Behaviors in the Ovulation and Menstruation Health Pilot Study: Cross-Sectional Study. *J. Med. Internet Res.* **2023**, *25*, e42164.
17. Suman, S.; Mukherjee, S.; Selvan, M.P.; Mary, V.A.; Jancy, S.; Shyry, S.P. Menstrual Cycle Tracking Using Deep Learning. In Proceedings of the 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, 19–20 June 2023; IEEE: Piscataway, NJ, USA, 2023, pp. 146–152.
18. Srivastava, D.; Gupta, S.; Kudavelly, S.; Ga, R.; et al. Unsupervised deep learning based longitudinal follicular growth tracking during IVF cycle using 3D transvaginal ultrasound in assisted reproduction. In Proceedings of the 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Virtual, 1–5 November 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 3209–3212.
19. Fehring, R.J.; Schneider, M. Randomized Comparison of Two Internet-Supported Methods of Natural Family Planning. 2013. Available online: https://epublications.marquette.edu/data_nfp/2/ (accessed on 1 December 2024).
20. Odirichukwu, J.; Njoku, O.; Sp.C, O.; C., N.; Nwachukwu, C.D.; U., N.J.; C., O.I.; C., D.; Odii, J.; N., J.C.; et al. *Improving Menstrual Cycle Prediction Accuracy using Advanced Machine Learning Model Methods*; QTanalytics India: Delhi, India, 2023. https://doi.org/10.48001/JoITML.
21. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357.
22. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1322–1328.
23. Patel, V.A.; Bhattacharya, P.; Tanwar, S.; Gupta, R.; Sharma, G.; Bokoro, P.N.; Sharma, R. Adoption of federated learning for healthcare informatics: Emerging applications and future directions. *IEEE Access* **2022**, *10*, 90792–90826.
24. Liu, W.; Zhang, Y.; Han, G.; Cao, J.; Cui, H.; Zheng, D. Secure and efficient smart healthcare system based on federated learning. *Int. J. Intell. Syst.* **2023**, *2023*, 8017489.
25. Nikolaidis, C.C.; Perifanis, V.; Pavlidis, N.; Efraimidis, P.S. Federated Learning for Early Dropout Prediction on Healthy Ageing Applications. In Proceedings of the 2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC), Tartu, Estonia, 18–20 September 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 135–142.
26. Nikolaidis, C.C.; Efraimidis, P.S. Advancing elderly social care dropout prediction with federated learning: Client selection and imbalanced data management. *Clust. Comput.* **2024**, *28*, 114. https://doi.org/10.1007/s10586-024-04850-4.
27. Perifanis, V.; Efraimidis, P.S. Federated neural collaborative filtering. *Knowl.-Based Syst.* **2022**, *242*, 108441.
28. Ma, J.; Naas, S.A.; Sigg, S.; Lyu, X. Privacy-preserving federated learning based on multi-key homomorphic encryption. *Int. J. Intell. Syst.* **2022**, *37*, 5880–5901.
29. Perifanis, V.; Drosatos, G.; Stamatelatos, G.; Efraimidis, P.S. FedPOIRec: Privacy-preserving federated poi recommendation with social influence. *Inf. Sci.* **2023**, *623*, 767–790.
30. Cheon, J.H.; Kim, A.; Kim, M.; Song, Y. Homomorphic encryption for arithmetic of approximate numbers. In Proceedings of the Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, 3–7 December 2017; Proceedings, Part I 23; Springer: Berlin/Heidelberg, Germany, 2017; pp. 409–437.
31. Hijazi, N.M.; Aloqaily, M.; Guizani, M.; Ouni, B.; Karray, F. Secure federated learning with fully homomorphic encryption for iot communications. *IEEE Internet Things J.* **2023**, *11*, 4289–4300.
32. Fereidooni, H.; Marchal, S.; Miettinen, M.; Mirhoseini, A.; Möllering, H.; Nguyen, T.D.; Rieger, P.; Sadeghi, A.R.; Schneider, T.; Yalame, H.; et al. SAFELearn: Secure aggregation for private federated learning. In Proceedings of the 2021 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 27 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 56–62.
33. Zhang, C.; Li, S.; Xia, J.; Wang, W.; Yan, F.; Liu, Y. {BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning. In Proceedings of the 2020 USENIX annual technical conference (USENIX ATC 20), Online, 15–17 July 2020; pp. 493–506.
34. Chen, H.; Laine, K.; Player, R. Simple encrypted arithmetic library-SEAL v2. 1. In Proceedings of the Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, 7 April 2017; Revised Selected Papers 21; Springer: Berlin/Heidelberg, Germany, 2017; pp. 3–18.
35. Gu, Y.; Bai, Y.; Xu, S. CS-MIA: Membership inference attack based on prediction confidence series in federated learning. *J. Inf. Secur. Appl.* **2022**, *67*, 103201.
36. El Ouadrhiri, A.; Abdelhadi, A. Differential privacy for deep and federated learning: A survey. *IEEE Access* **2022**, *10*, 22359–22380.
37. Yang, X.; Huang, W.; Ye, M. Dynamic personalized federated learning with adaptive differential privacy. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 72181–72192.

38. Wei, K.; Li, J.; Ma, C.; Ding, M.; Chen, W.; Wu, J.; Tao, M.; Poor, H.V. Personalized federated learning with differential privacy and convergence guarantee. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 4488–4503.
39. Hidayat, M.A.; Nakamura, Y.; Dawton, B.; Arakawa, Y. Agc-dp: Differential privacy with adaptive gaussian clipping for federated learning. In Proceedings of the 2023 24th IEEE International Conference on Mobile Data Management (MDM), Singapore, 3–6 July 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 199–208.
40. Keshta, I.; Aoudni, Y.; Sandhu, M.; Singh, A.; Xalikovich, P.A.; Rizwan, A.; Soni, M.; Lalar, S. Blockchain aware proxy re-encryption algorithm-based data sharing scheme. *Phys. Commun.* **2023**, *58*, 102048.
41. Shen, X.; Luo, X.; Wang, B.; Chen, Y.; Tang, D.; Gao, L.; et al. Privacy-preserving multi-party deep learning based on homomorphic proxy re-encryption. *J. Syst. Archit.* **2023**, *144*, 102983.
42. Zhang, Y.; Zhang, Z.; Ji, S.; Wang, S.; Huang, S. Conditional Proxy Re-Encryption-Based Key Sharing Mechanism for Clustered Federated Learning. *Electronics* **2024**, *13*, 848.
43. Kleinaki, A.S.; Mytis-Gkometh, P.; Drosatos, G.; Efraimidis, P.S.; Kaldoudi, E. A blockchain-based notarization service for biomedical knowledge retrieval. *Comput. Struct. Biotechnol. J.* **2018**, *16*, 288–297.
44. Kostamis, P.; Sendros, A.; Efraimidis, P. Exploring ethereum's data stores: A cost and performance comparison. In Proceedings of the 2021 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), Paris, France, 27–30 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 53–60.
45. Kostamis, P.; Sendros, A.; Efraimidis, P.S. Data management in Ethereum DApps: A cost and performance analysis. *Future Gener. Comput. Syst.* **2024**, *153*, 193–205.
46. Zhu, J.; Cao, J.; Saxena, D.; Jiang, S.; Ferradi, H. Blockchain-empowered federated learning: Challenges, solutions, and future directions. *ACM Comput. Surv.* **2023**, *55*, 1–31.
47. Chen, Y.; Lin, F.; Chen, Z.; Tang, C.; Jia, R.; Li, M. Blockchain-based Federated Learning with Contribution-Weighted Aggregation for Medical Data Modeling. In Proceedings of the 2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS), Denver, CO, USA, 19–23 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 606–612.
48. Majeed, U.; Khan, L.U.; Hassan, S.S.; Han, Z.; Hong, C.S. FL-incentivizer: FL-NFT and FL-tokens for federated learning model trading and training. *IEEE Access* **2023**, *11*, 4381–4399.
49. Pandey, S.R.; Nguyen, L.D.; Popovski, P. Fedtoken: Tokenized incentives for data contribution in federated learning. *arXiv* **2022**, arXiv:2209.09775.
50. Pandl, K.D.; Leiser, F.; Thiebes, S.; Sunyaev, A. Reward systems for trustworthy medical federated learning. *arXiv* **2022**, arXiv:2205.00470.
51. Chaliasos, S.; Reif, I.; Torralba-Agell, A.; Ernstberger, J.; Kattis, A.; Livshits, B. Analyzing and Benchmarking ZK-Rollups. In Proceedings of the 6th Conference on Advances in Financial Technologies (AFT 2024), Vienna, Austria, 23–25 September 2024.
52. Global, U. ConInSe. 2024. Available online: https://github.com/NGI-TRUSTCHAIN/IS-CIS (accessed on 1 December 2024).
53. Rantos, K.; Drosatos, G.; Demertzis, K.; Ilioudis, C.; Papanikolaou, A.; Kritsas, A. ADvoCATE: A Consent Management Platform for Personal Data Processing in the IoT Using Blockchain Technology. In *Innovative Security Solutions for Information Technology and Communications: 11th International Conference, SecITC 2018, Bucharest, Romania, 8–9 November 2018*; Lanet, J.L., Toma, C., Eds.; Springer: Cham, Switzerland, 2019; pp. 300–313.