Can the use of privacy-enhancing technologies enable federated learning for health data applications in a Swedish regulatory context?*

Rickard Brännvall†, Helena Linge†‡, Johan Östman‡

Abstract— **A recent report by the Swedish Authority for Privacy Protection (IMY) evaluates the potential of jointly training and exchanging machine learning models between two healthcare providers. In relation to the privacy problems identified therein, this article explores the trade-off between utility and privacy when using privacy-enhancing technologies (PETs) in combination with federated learning. Results are reported from numerical experiments with standard text-book machine learning models under both differential privacy (DP) and Fully Homomorphic Encryption (FHE). The results indicate that FHE is a promising approach for privacy-preserving federated learning, with the CKKS scheme being more favorable in terms of computational performance due to its support of SIMD operations and compact representation of encrypted vectors. The results for DP are more inconclusive. The article briefly discusses the current regulatory context and aspects that lawmakers may consider to enable an AI leap in Swedish healthcare while maintaining data protection.**

I. Introduction

Recent advances in artificial intelligence (AI) have shown great promise in improving diagnosis, treatment, personalized medicine [1] and disease prevention by predictions [2]. Machine learning algorithms can analyze vast amounts of medical data, such as patient records, imaging scans, and genetic information, to identify patterns and make predictions about the likelihood of diseases and the effectiveness of treatments. Additionally, computer vision algorithms can analyze medical images, e.g. X-rays and MRI scans, detect abnormalities, and provide decision support in diagnosing disease.

However, the use of AI in medical applications raises concerns about privacy, as it involves the processing of sensitive personal information protected by privacy laws and regulations, such as the General Data Protection Regulation (GDPR) in the EU and the Health Insurance Portability and Accountability Act (HIPAA) in the USA, as well as country specific patient data regulations. To facilitate the sharing of personal health information between healthcare providers and digital health services, adequate privacy protection is essential. Full anonymization (de-identification) is often not possible as it impairs full utility of the data [3]. Therefore, several alternative approaches have been proposed, including cryptographic techniques, differential privacy, synthetic healthcare data generation, federated learning, and pseudonymization [4].

One technology that shows great potential in privacy preservation is fully homomorphic encryption (FHE) [5]. It makes computation on encrypted data possible which enables privacy-by-design cloud-based services. Federated learning allows multiple parties to collaboratively train a machine learning model without exchanging actual data. However, all comprehensive solutions must have a solid foundation in conventional security technology, policies, and procedures.

There is often a utility versus privacy trade-off when using privacy-enhancing technologies (PETs). However, for medical applications, a decreased utility translates into suboptimal data use, and loss of adequacy with regard to results and outcomes. If utility loss is allowed prolonged suffering and possibly even death may result. **How can the application of advanced privacy-enhancing measures in federated learning maintain a preserved privacy without the undue compromise of utility?** We limit our exploration of this question to two privacy problems: 1) that the final model parameters can potentially disclose personal data, and 2) that the sharing of model updates in the federated learning process can potentially disclose sensitive information from the respective parties' data sets.

Motivation: Regulators are currently investigating how PETs can unlock the potential of data-driven applications. We here explore in practice how these technologies can enable an AI leap in Swedish Healthcare already within the current privacy legislation, as well as identify questions that lawmakers may consider to achieve harmony with developments and demands.

Contribution: We discuss approaches to applying PETs to improve data protection in federated learning. We compare two quantum computer resilient FHE schemes and conclude that one has an advantage in terms of computational performance. Our experiments indicate that FHE is both feasible and favorable, as it preserves utility while adhering to data use minimization and purpose limitations. We also conduct numerical experiments with differential privacy (DP), which confirm the view that it, in its strictest form, may have significant utility degradation for trained models.

Outline: The next section provides Background to this work, providing both technical details on the advanced PETs and a summary of recent regulatory developments in Sweden. The Methods section explains the proposed approach. It describes the setup for the numerical experimentation, from which Results then are presented in the next section. The Discussion section presents an analysis of the pros and cons of the proposed alternative use cases, both in relation to the results from the numerical experiments and in relation to previously published work. The article then concludes with some recommendations.

II. Background

A. Machine learning context

Machine learning is a subfield of artificial intelligence that involves training algorithms to learn patterns in data without being explicitly programmed. Deep learning refers to algorithms

*This work was supported by Vinnova grant 2022-02668 (HEIDA).

†RISE Research Institutes of Sweden.

‡AI Sweden.

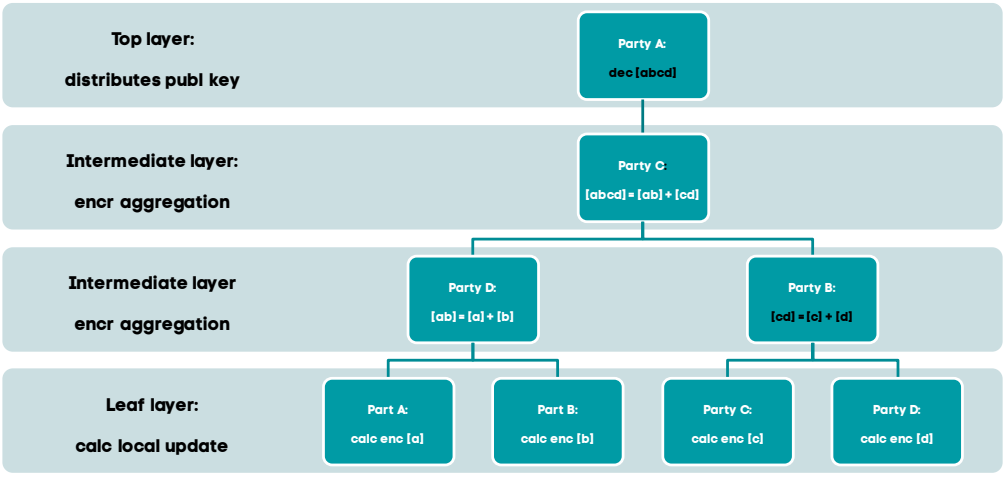

Fig. 1. Binary tree structure where all aggregation is done under Fully Homomorphic Encryption (FHE) to protect model updates of individual parties. Only the global update can be accessed in plaintext. All computations is done in Secure Processing Environments (SPEs), where intermediate results are erased immediately after used. This framework is information symmetric and supports purpose limitations, data- and storage minimization.

that are based on neural networks with many intermediate layers between the input layer and the output (prediction) layer. Deep neural networks have been demonstrated to have great capacity in identifying complex patterns. Model parameters (or weights) are variables that the machine learning algorithm adjusts to optimize its performance. For neural networks, this is often done by gradient descent (or related methods) which iteratively adjusts the parameters of a model to minimize the loss function.

Overfitting occurs when a model becomes too complex and fits the training data too closely, resulting in poor performance on unseen data. Techniques such as regularization, cross-validation, and early stopping can be used to avoid overfitting. Such settings are denoted hyperparameters, which in addition to controlling the behavior of training typically also include parameterizing the machine learning architecture, e.g., the size and depth of a neural network.

Federated learning: Federated learning is a type of machine learning that allows multiple parties to train a shared machine learning model, without directly exchanging their respective data. This makes it suitable for use in scenarios where data is distributed among different parties, and where privacy and security concerns prevent the sharing of data. The case fits well for health- and medical data [6]. For technical reviews of the current and future applications of federated learning for biomedical data, see for example [7]–[9]. Recent experimentation demonstrates that just keeping the data locally is insufficient with regard to the security of the data. Machine learning models are prone to several privacy attacks which could expose sensitive data: 1) An attacker can use the gradient information of the deep learning model to get the sensitive data. 2) Even the trained local model parameters expose information that can be used by an attacker to make an inference about the federated learning participant. The next section goes into some more detail about these types of privacy attacks.

B. Attacks on privacy

1) Membership inference attack: (MI) A membership inference attack aims to determine whether a specific data point has been used during the training of a machine learning model. This method can potentially expose sensitive information about individuals, i.e. whether a person with certain characteristics and a particular medical condition has been included in the model's training data. The attacker can either have black-box access [10], where they only have query access to the model, or white-box access [11], where they have full access to the model's parameters and architecture. Shokri et al. [10] proposed one of the first attacks, which considers an attacker who can query the target model in a black-box way to obtain confidence scores for the queried input. Among the multitude of attack procedures that were proposed later on, we mention [11] that is computationally simpler, but requires that the attacker can calculate the training loss of a candidate data point threshold and compare it with a threshold (the average training loss). A naive baseline procedure was proposed by [12], which predicts a sample as a member if it is correctly labeled by the target model and predicts it as a non-member if misclassified. In a recent experimental comparison [13], the naive model demonstrates similar performance as the more involved MI attack procedures. The two approaches both have a high false positive rate. Indeed, MI attack accuracy is reported to be highly correlated to the model's overfitting or generalization gap [19, 20, 22], and furthermore troubled by high false positive [13]. The generalization gap refers to the difference between the test set and training set performance. As low as possible is generally desired as it reflects the extent to which a model is overfitted. As overfitted models have limited practical use, it is questionable how well reported MI attack success stories can be generalized to well-trained models [14]. Despite the limitations of current MI attack strategies, it is important to study and learn from them as superior attacks might appear in the future.

2) Model inversion attack: The aim of a Model inversion attack is to learn hidden sensitive attributes of a test input given knowledge about the non-sensitive attributes. This attack is also called an attribute inference attack and is carried out as a search for the value of the sensitive attributes that maximizes the posterior probability given the non-sensitive attributes, model access, and prior knowledge about the distribution of attributes [15]. This attack exploits the correlation between the sensitive attribute

TABLE I

PRIVACY ATTACKS CONSIDERED FOR THIS WORK.

Membership inference attack Model inversion attack (attribute inference attack)	• Infer whether a specific data point has been used during the training of a machine learning model. • Infer hidden sensitive attributes of a training input given knownledge about the non-sensitive attributes. • Both attacks use similar procedure and input data (which is why they are often treated together). • A party in federated learning could use its own data and a global model to learn about other parties' data. • Reported successful attacks may rely on model overfitting. How relevant is this for more robust models? • Although perhaps not practically possible today, superior attack procedures may appear in the future.
Gradient inversion attack	• Practical to reconstruct data points from gradients averaged over several iterations or batches. • Successful attacks recovered single data points from batches of up to a hundred images or texts.

and the model output, which is encoded in the machine-learning model. Many of the proposed attack procedures are modified variations of membership inference attacks, for example, [11], why it often makes sense to discuss both these attacks together. Also related, are the memorization attack, which exploits the ability of high-capacity models to memorize certain sensitive patterns in the training data [16]; and the property inference attack, in which the attacker tries to infer whether the training data set has a specific property. Although these attacks are related to attribute inference, it is rather the overall statistical patterns of the training data that are exposed. As the topic of our discussion concerns the privacy of the individual, it is sufficient to consider attribute inference.

3) Gradient inversion attack: The gradient used to improve the model contains information about the batch of data points that were used to calculate it. Early work that recovered data from gradient information was limited to shallow networks of less relevance. Later, it was shown to be [17] possible to reconstruct up to 8 images from their batch averaged gradients also for slightly deeper neural networks. More recently, [18] explored settings encountered in practice when training deep neural networks and showed that even averaging gradients over several iterations, or several images, does not protect the privacy of an individual data point in federated learning applications. Indeed, by exploiting a magnitude-invariant loss function, it is possible to faithfully reconstruct images at high resolution from their parameter gradients for realistic deep architectures like ResNet. The reconstruction is possible even when averaging gradients over multiple epochs, using local mini-batches, or even for a local gradient averaging of up to 100 images with deep networks, appearing to be as vulnerable as shallow networks. Attacks against federated averaging of parameters (instead of gradients) have also been devised [19].

C. Privacy enhancing technologies

1) Homomorphic encryption: Fully homomorphic encryption (FHE) allows mathematical operations to be performed directly on encrypted data, without first decrypting the data, and without access to a secret key. FHE is distinguished from conventional uses of cryptography, where data is encrypted only while it is sent between parties (in motion), and during storage on a file system (at rest) but is decrypted for calculation and processing. This last step of decryption introduces a vulnerability to conventional cryptography, in that data can be read in hardware or software layers, and that a secret key must be available on the server that performs the calculations. FHE offers a solution that guarantees that even a curious computing party can not see the data. It enables privacy-preserving processing and analysis of data, for example in a cloud-based AI service (Figure 2), where the original data as well as all intermediate and final results are indistinguishable from random noise to the computing cloud.

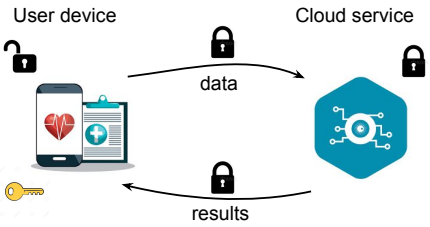

Fig. 2. A two-party solution that uses homomorphic encryption to protect data sent from a user device to be processed in the cloud. (Image from [20])

Decryption is possible only for the private key holder and considered unbreakable under very strong cryptographic guarantees for most recent FHE schemes, e.g. even against hypothetical quantum computer-based attacks [21]. We use the word *plain text* for unencrypted data and *cipher text* for encrypted data, as is conventional.

Fully Homomorphic Encryption differs from early schemes referred to as Partially Homomorphic Encryption schemes in that FHE can support both addition and multiplication. In this work, the difference in terminology is not important as we only consider schemes with full arithmetic support, and will refer to them either as (Fully) Homomorphic Encryption or simply abbreviated as FHE. Noise is added in the construction of the encrypted representation. For every operation, this noise grows such that the result of large computations can be meaningless after decryption unless noise-mitigating measures like bootstrapping are used. In a leveled

approach one carefully manages the cryptographic *noise-budget*, by which we mean the number of arithmetic operations one can carry out before bootstrapping becomes necessary. Practically this means that only a limited number of multiplications and additions are allowed. This is controlled by the parameters selected for encryption, which we will refer to loosely as "key size".

The CKKS scheme [22] named after its developers Cheon, Kim, Kim, and Song, encrypts complex numbers and can perform fixed-point arithmetics. Its security is based on the Ring Learning With Errors (RLWE) problem. While many other schemes perform exact arithmetics on encrypted integers, CKKS has become popular for applications that only require approximate calculation. The TFHE scheme developed by Chillotti and collaborators [23] supports fast bootstrapping, arithmetic operations, and univariate function evaluation thanks to its implementation of a type of look-up mechanism. An important difference between the software libraries used in this work is that the CKKS implementation [24] supports SIMD operations, while the TFHE library [25] does not. This means that the former can add (or multiply) vectors up to a certain size at constant cost, while for the latter the cost of additions of vectors is linear in the length of the vector.

Proposition for using FHE to enhance security in Federated Learning applications have recently been put forward [26]–[29]. This provides efficient defense against the above-mentioned attacks by only allowing the exchange of encrypted information between the participants of the federation such that it can be aggregated (i.e. averaged) under homomorphic encryption. However, it comes with significant overhead in terms of computation time and data transfer.

2) Differential privacy: Differential privacy is a framework for privacy-preserving data analysis, where a randomized algorithm $\mathcal{A}$ is considered (ϵ, δ)-differentially private [30] if for any neighboring datasets D_1 and D_2 that differ, in at most, one record and for any set of outputs $S \subseteq Range(\mathcal{A})$,

$$P[\mathcal{A}(D_1) \in S] \leq e^{\epsilon} P[\mathcal{A}(D_2) \in S] + \delta,$$

where δ represents the maximum allowable probability that privacy is violated. In other words, the (ϵ,δ)-differential privacy guarantee provides a limit on the overall probability of privacy violation.

The noise that is added to a differentially private algorithm's output is calibrated based on the sensitivity of the function being computed, which is defined as the maximum distance in some norm, $||\cdot||$, between the outputs of neighboring data sets, $\Delta f = \max_{D,D'} ||\mathcal{A}(D) - \mathcal{A}(D')||$.

The Rényi mechanism [31] is a variant of differential privacy that can be useful for machine learning applications as it allows for fine-grained control over the level of privacy while maintaining the accuracy of the output also over compounded applications. It can be used to perturb training data or model updates, thereby providing a privacy-preserving mechanism for training machine learning models on sensitive data also in a sequential, iterated application like gradient descent.

It is common to add noise as a perturbation to the gradients in differential privacy applications for machine learning. To ensure that the added noise is proportional to the sensitivity of the model, the gradients are often clipped before the noise is applied. This involves constraining the magnitude of the gradients to mitigate the effect of high sensitivity. By controlling the amount of perturbation and clipping, one can achieve a trade-off between privacy and model accuracy. Questions have been raised about real-world applications using very high epsilon to achieve utility over composition [14], although recent work points to more favorable trade-offs, e.g., for Stochastic Gradient Descent with noise [32].

3) Secure aggregation: Secure aggregation is a multi-party computation technique enabling non-trusting parties with sensitive data to privately compute an aggregate without depending on a trusted third party. This process typically involves the following steps: i) clients agree on pairwise private seeds, ii) each client generates a private seed, iii) clients randomly mask their model parameters using the seeds and communicate the masked model to the server, and iv) clients distribute shares of the seeds to other clients using a secret sharing scheme [33], [34]. The secret sharing is based on a (t, n) secret sharing scheme and offers resilience against dropouts and stragglers, i.e., clients not responding to the server, by allowing the random seeds of each client to be recovered from the collected shares of t out of the n clients. This property is leveraged during aggregation where the server requests shares from the available clients to reconstruct the sum of the secret masks so they can be canceled out.

Since secure aggregation occurs over a finite field, clients must convert their model parameters accordingly. The size of the finite field can impact the model utility of secure aggregation; larger field sizes preserve model utility but increase communication overhead whereas a smaller field size may result in loss of information. Secure aggregation generally incurs extra communication compared to differential privacy and homomorphic encryption.

Recently, researchers have combined secure aggregation with differential privacy to mitigate the negative impact on model utility caused by differential privacy [35]. The core concept is to protect the aggregate of local models rather than individual local models, resulting in the addition of less noise and, ultimately, a lesser effect on model utility.

D. Regulatory environment

The purpose of GDPR and other privacy legislation is to protect individuals' personal data and privacy rights by establishing clear principles for the collection, use, and processing of personal data. Important principles include the lawful, fair, and transparent processing of personal data, purpose limitation, data minimization, accuracy, storage limitation, integrity and confidentiality, accountability, and respect for individual rights. In this article, we will particularly consider:

Purpose limitation: Personal data must be collected for specified, explicit, and legitimate purposes and not processed in a manner that is incompatible with those purposes.

Data minimization: Personal data must be adequate, relevant, and limited to what is necessary for the purposes for which it is processed.

Storage limitation: Personal data must not be kept for longer than necessary for the purposes for which it is processed.

This work will now discuss how FHE and other PETs can support the above general principles, and be relevant mitigations to consider also in a specific use-case.

Changing environment: The Swedish government has launched official investigations [36], [37] in order to achieve a national data strategy aimed at increasing the access and beneficial utilization of data. Such purposes include improved health applications supported by artificial intelligence. During 2021-22, The Swedish Authority for Privacy Protection (IMY) was commissioned by the government to provide support and guidance to the innovation system on data protection matters [38] Related to this mission, IMY organized activities where experts and participants from industry and public sector could interact. These included research hearings on PETs, workshops, and seminars.

A recent report [39] commissioned by the Swedish eHealth Agency explores the benefits of a national data space for medical AI, particularly in image diagnostics and mammography. It examines the concept of a Secure Processing Environment (SPE), where data can be isolated and encapsulated to prevent unauthorized access and protect sensitive information while still allowing researchers and healthcare professionals to use the data for research and analysis purposes. The report calls for deeper investigations of how federated learning in a distributed ecosystem of SPEs can facilitate the safe sharing of resources and data, and increase opportunities for research and innovation while meeting important integrity- and legal requirements.

The Swedish innovation agency Vinnova, in a recent report commissioned by the government [40], discussed various aspects of secure data sharing, including the need for increased dissemination and utilization of conventional privacy protection techniques. It also highlights the importance of conducting research on cutting-edge technologies, especially mentioning federated learning and homomorphic encryption. Also, AI Sweden, the national center for applied artificial intelligence, views decentralized AI as one of the critical technologies for future AI development [41] across several business and industrial sectors. Although new advanced technologies for privacy protection hold much promise regarding, on the one hand, legal and security requirements, and on the other hand, exploiting the potential of data sharing and utilization in health and medicine in Sweden, legal uncertainty still remains.

Regulatory sandboxes: Regulatory sandboxes aim to bridge the gap between the rapid pace of technological development and the slower pace of regulatory and policy development. They can assist in identifying and developing new ways of working in the public sector that could enable more agile and effective regulatory and policy responses to emerging challenges. By engaging with innovators and working together to identify legal ambiguity and challenges, governing bodies that participate can promote a more effective and efficient regulation that supports innovation while still protecting the public interest.

Regulatory sandboxes have already been put in place in the UK, Norway, and France with guidance that targets the application of GDPR. The goal is to increase judicial predictability, reduce time and risk for a product or service to reach the market, and facilitate startup and small business growth by doing so. In the EU [42], regulatory sandboxes have been highlighted as a way to promote innovation and growth for companies, and the draft AI regulation being negotiated currently includes proposals for regulatory sandboxes to promote and facilitate the application of AI. Regulatory sandboxes allow for exploratory, dialogue-based guidance to be given to selected innovation projects in exchange for the work being summarized in a public report that enables learning for others. The approach helps to develop practical examples in areas where both technology and law are complex, relatively new, and untested, while also increasing regulatory authorities' understanding of new technology and how it can be applied.

Sandbox: Decentralized AI in Healthcare: IMY participated in a pilot project on regulatory sandboxing in 2022 and summarized its conclusions in a public report [43]. The project, titled "Decentralized AI in Healthcare - Federated Machine Learning between Two Healthcare Providers" focused on evaluating the potential of jointly training and exchanging machine learning models between two healthcare providers, Region Halland and Sahlgrenska University Hospital, in order to predict heart failure patient readmissions within 30 days of their last hospital stay. The project was facilitated by AI Sweden, the national center for Applied AI.

The purpose of the project was to explore the potential of regulatory sandboxing as an approach to address complex societal challenges and to help regulators and policymakers better understand and analyze new technologies that fall within their regulatory frameworks. Specifically, the project aimed to provide in-depth guidance on how data protection regulations should be interpreted and applied to a specific innovation initiative involving advanced technologies like AI and federated machine learning.

The following paragraphs summarize the parts of the report which are important for our discussion in this article, starting with its three focus questions:

Question 1: Is there a legal basis for local processing of personal data, i.e., when healthcare providers train the machine learning model locally only on their own patient data? IMY's assessment is that there is a legal basis for local processing of personal data. The key factor is that IMY believes there is support for a dynamic and technology-neutral interpretation of the purpose provisions in the Patient Data Act and the Health and Medical Services Act, which means that what falls within these provisions can change over time, with regard to technological development.

Question 2: Does personal data disclosure occur between healthcare providers in the federated machine learning in this case? IMY's assessment is that Region Halland and Sahlgrenska University Hospital are at risk of disclosing personal data to each other in the current case when the knowledge gained from local training is combined into a joint machine learning model. Either party could, if it gathers the necessary expertise and purposeful intent, launch two types of privacy-harming attacks, namely, Membership Inference Attack and Model Inversion Attack to infer information about persons in the other party's data set.

Question 3: Is there a legal basis for disclosure of personal data between healthcare providers? IMY has not made any assessment of whether any confidentiality-breaking provision could be applicable in the current case. However, if Region Halland and Sahlgrenska University Hospital, both being authorities, were to request patient data from each other with the support of the Public Access to Information and Secrecy Act, such disclosure could possibly be allowed provided that the data is not confidential. However, patient data within healthcare is generally confidential. The legal basis for personal data disclosure between healthcare providers under certain circumstances may

be in place but requires a case-by-case assessment.

IMY limited their investigation to the use case at hand: federated learning between two public health care providers. As a precaution in the project, the two healthcare providers only used data that did not contain personal information.

TABLE II
INFORMATION ASSUMPTIONS IN FRAMEWORK

Trivially known by all parties • Model architecture • Training hyperparameters such as learning rate, regularization (gradient clipping, etc) • The aggregation topology (tree)
Each party knows at the end • Content of their own data set (1) • The final model parameters (2)
Each party at each iteration • Their own (local) model update (3a) • The global model update (3b)
The central party • Knows and holds the secret key • Distributes the public keys

III. METHOD

A. Problem statement

Here we restate the objective from the Introduction: How can we design privacy-enhancing measures for a federated learning effort such that privacy is best preserved without unduly compromising the utility of the trained model? We also limited the exploration in this work to two privacy problems:

P1: the parameters of an AI model can potentially leak personal data, and

P2: in the federated learning process parties can potentially leak information about their data set through the (iterated) exchange of model updates.

The IMY report primarily discussed (P1) in how it opens up vulnerability to membership inference attacks, and model inversion attacks. Here we also want to highlight that (P2) should be considered carefully as it has been demonstrated that inference of private data from model gradients is practical for certain machine learning models and settings [18].

B. Target Framework

The overarching intention of the federated learning framework discussed below is to promote data- and storage minimization, purpose limitations, and symmetric distribution of information, such that every party is trusted only with the data it needs to perform a task according to the original intentions while avoiding trust asymmetries where some party has privileged access to the information of others.

Preparaion stage: One of the parties in the federation is selected to create a secret key which is used to produce the public keys that it distributes to the other parties.

1) Secure processing environments: First and foremost we will assume that each party carries out all computations in a Secure Processing Environment under strict access control. They are furthermore obliged by contractual agreements to follow the machine learning protocol and take appropriate storage minimization measures, like deleting intermediate model updates immediately after they are used.

2) Homomorphically encrypted: Homomorphic encryption is used throughout the federated learning process to encrypt individual parties' model updates with a public key that they have received before the model training exercise. This contributes towards the goals of data minimization and purpose limitation.

3) Aggregation over binary-tree: Encrypted model updates are aggregated between parties in a binary tree structure of Figure 1. It is strictly not necessary to use a binary tree, as long as the top party receives only the encrypted global aggregate, which it decrypts and distributes to all other parties. This removes the information advantage of the party that holds the secret key and decrypts the global model update. Every party now only has access to its own model update and the aggregated global model.

4) Differential privacy: Differential privacy has the potential to protect against both problems P1 and P2, but one carefully has to consider for each particular use case if it has adverse implications for utility. Even if one does not promise full differential privacy, the addition of noise at a higher value of ϵ (together with other regularization) can help avoid overfitting, which also combats privacy attacks.

On completion: At the end of the training effort, all hardware that has touched the data is thoroughly erased (or even destroyed). The key holding party must similarly permanently delete the decryption key.

Information symmetry: Table II summarizes the information that each party knows throughout the exercise.

C. Models and Materials

Numerical experiments were carried out to investigate the performance of differential privacy and federated learning that uses homomorphic encryption during parameter aggregation. For these experiments we used two different models chosen such that both a very simple as well as a moderately complex architecture were examined, that is a logistic regression model (LogReg) and a deep learning model for image analysis (ResNet-18) whose features are both summarized in Table IV.

1) Logistic Regression: A logistic regression model was trained to estimate the risk of future coronary heart disease (CHD) based on a patient's information such as demographic, behavioral, and medical factors. The dataset is publicly available on the Kaggle website [44], and it is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. It includes over 4,000 records and 15 attributes, from which a balanced subset of 1,000 records and 8 attributes was selected (such that positive and negative labels were equally frequent). A test set of 200 records was set aside, while the remaining 800 records were used for training the logistic regression model.

2) Deep Neural Network: ResNet-18 is a large image classification based on deep neural networks, which is described later in this section. It was trained to classify images from

TABLE III

DP ANALYSIS STATS.

	inf	0.1	0.3	1.0	3.0	10.0	30.0
LogReg	68.9±0.9	67.3±2.2	68.1±1.6	67.9±1.5	68.5±1.6	68.6±1.5	68.2±1.5
ResNet	64.7±2.1	19.2±1.1	29.7±1.1	35.6±0.4	39.6±0.0	41.5±0.2	45.1±0.1

TABLE IV

SUMMARY OF MODELS

	LogReg	ResNet
total params	10	11511784
train params	10	3591
data set name	FraminghamCHD	DermaMNIST
data size used	1000	10015

TABLE V

MEAN TEST SET ACCURACY FOR LOGREG (WITH SAMPLE STD).

	2	4	8	16
tfhe 256	68.6±1.3	68.1±1.6	64.1±4.5	58.3±6.1
tfhe 512	68.8±1.1	68.8±0.9	68.9±0.8	69.4±0.7
plain text	68.8±1.0	68.7±1.0	69.0±0.9	69.3±0.8
ckks 4096	68.7±0.9	68.8±0.8	69.1±0.9	69.2±0.7
ckks 8192	68.8±1.0	68.9±1.0	69.1±0.9	69.2±0.8

the DermaMNIST/HAM10000 [45] collection of multi-source dermatoscopic images of common pigmented skin lesions. The dataset consists of 10,015 dermatoscopic images categorized as 7 different diseases and was downloaded using the MedMNIST software library [46], [47].

Training Procedure: About 20% of the data for each model was set aside as a test set. The remaining 80% of the data was used for training. The models were trained for a total of 5 epochs.

Differential Privacy: Each model was trained under the Rényi mechanism for compounded (ϵ,δ)-differential as implemented in the Opacus [48] library for PyTorch. Standard settings were used for δ and gradient clipping, while ϵ was varied over a fixed range from $\epsilon=0.1$ (relative strong privacy) to $\epsilon=30$ (weak privacy).

Federated Learning: For each model, a federated learning set-up was simulated that at the end of each epoch, encrypted parameters were aggregated across the nodes organized in a binary tree-like topology (Figure 1), with the number of nodes taken as $n \in \{2,4,8,16\}$. The training data was split evenly between the nodes that were part of the federated exercise. Each model was trained using the two different homomorphic encryption schemes, TFHE and CKKS, each with two different parameter settings. For comparison, each model was also trained with plain text aggregation for each node configuration. The training was repeated 100 times to gather statistics about the variation in performance.

IV. RESULTS

A. Differential Privacy Experiments

Table III reports the mean accuracy with 95% confidence bands for each examined machine learning model. Each column represents a target ϵ, where the first column reports the case of no differential privacy, i.e., when no noise was added. For the LogReg model, we see that the estimated mean accuracy falls within the confidence bands of the non-private model for all but the lowest ϵ values. There doesn't seem to be a significant adverse effect on utility from adding differential privacy for this case. The results are very different for the deep learning model. The estimated mean accuracies for all the differentially private ResNet models are far below the lower confidence bound for the non-private model.

B. Federated Learning Experiments

1) Logistic regression model.: Table V displays the mean accuracy obtained for the logistic regression model, together with the observed standard deviation across the 100 repetitions of the experiment. The accuracy appears relatively unaffected by the encrypted aggregation, except for the smaller key size for the TFHE scheme for the size $n=8$ and $n=16$ node configurations where mean accuracy is more than 2 std worse than for the corresponding plain text configuration. For all other model configurations, the results using homomorphically encrypted aggregation are not significantly different from the plain text case.

Execution times were also collected throughout the numerical experiments. For the LogReg model, the measured average time in milliseconds for the central cryptographic operations is displayed in Table VI. For both schemas, encryption of the model parameters ("enc time") dominates both the time it takes to carry out the additions ("add time") and the time it takes to decrypt the aggregated results ("dec time"). Here TFHE appears to be overall faster, although we note that CKKS outperforms for the addition.

TABLE VI

TIMED OPERATIONS FOR LOGREG IN MS.

context	enc time	add time	dec time
tfhe 256	0.334	0.088	0.015
tfhe 512	0.623	0.162	0.028
ckks 4096	4.03	0.059	0.962
ckks 8192	10.071	0.149	2.916

2) ResNet-18 model.: ResNet-18 is a convolutional neural network that is 18 layers deep [49] and has more than 11 million parameters. We used a version of the network that was pre-trained on more than a million images from the ImageNet database [50] on the task to classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. The network has thus learned a rich feature representation for a wide range of images, which is can be leveraged for the medical image classification task. When data is scarce, one can sometimes take a model trained on data from a related task and then fine-tune the model by training it on the target task, which is an example of transfer learning (see

TABLE VII

MEAN TEST SET ACCURACY FOR RESNET-18 (WITH SAMPLE STD).

scheme	2	4	8	16
tfhe 256	68.1±0.7	67.3±0.8	66.3±0.9	64.7±1.5
tfhe 512	68.4±0.6	68.0±0.7	67.5±0.5	67.2±0.5
ckks 4096	68.3±0.6	67.9±0.5	67.6±0.5	67.1±0.5
ckks 8192	68.4±0.6	68.1±0.5	67.6±0.5	67.2±0.5
plain text	68.5±0.6	67.9±0.6	67.5±0.5	67.2±0.5

for example a recent review [51]) Thus we can obtain acceptable performance after only 5 epochs of training on a relatively small data set of 8007 images of skin changes. In fact, we only let the 3591 parameters of the top layer be trainable, and keep all other layers at their pre-trained values.

Similar results as for the logistic regression case are observed also for the accuracy of the ResNet18 model trained on the DermaMNIST data. Table VII shows an impairment of the accuracy of at least two standard deviations for the configurations with the smaller key size of the TFHE scheme. For the other configurations using homomorphic encryption aggregation the difference compared to the plain test case is not significant.

TABLE VIII

TIMED OPERATIONS FOR RESNET-18 IN MS.

scheme	enc time	add time	dec time
tfhe 256	46.1	128.4	1.9
tfhe 512	85.1	309.5	4.1
ckks 4096	8.0	0.6	2.0
ckks 8192	17.2	1.3	3.0

Table VIII displays the execution times for the operations that support the homomorphic encryption aggregation. Here, performance is reversed such that CKKS outperforms TFHE, since the former supports SIMD (Single Instruction Multiple Data) execution, meaning that the summation of parameters from two different nodes can be "vectorized", i.e. carried out in parallel over the entries in the same position as the encrypted representation can hold vectors. Thus we can benefit from SIMD when summing the 3591 parameters of the ResNet-18 top layer from two nodes. At the time that we carried out the experiments the TFHE software implementation [25] that we used did not support SIMD and thus had to encrypt each individual entry of a vector separately. It was supported by the CKKS implementation [24] that we used.

Data size overhead: The encryption inevitably results in storage and communication overhead since the encrypted representations are larger than the plain text. This effect is very noticeable for the two encryption schemes considered in this work, which is evident in Table IX that lists the file storage size in kilobytes (KB). Column headers identify the number of stored values, i.e. the length of a vector of real numbers. Columns with labels 10 and 3591 lists the size of the representations for the LogReg and ResNET-18 models, respectively. We note that data size requirements are lower for THFE for the LogRes model with only 10 parameters, while the requirements for ResNet are (much) smaller for CKKS than for TFHE (last column).

CKKS natively stores vectors and have more compact representations, which are in fact invariable to the size of the plain text vector for lengths up to half of the key size (poly mod); therefore CKKS 8192 can store the entire vector of trainable parameters for ResNet-18 in a single file since it has length 3591 which is smaller than $8192/2 = 4096$.

TABLE IX

FILE SIZE (KB) FOR DIFFERENT PARAMETER SIZES.

context	1	10	100	3591
tfhe 256	5.458	53.629	535.323	19223.441
tfhe 512	10.682	105.88	1057.892	37988.886
ckks 4096	80.897	80.908	80.904	161.809
ckks 8192	334.32	334.298	334.314	334.314
plain text	0.025	0.255	2.549	91.55

For the smaller LogReg model, the data size is increased by about three orders of magnitude compared to the plain text. For the larger model (ResNet-18) CKKS outperforms thanks to its compact representation and only shows a modest overhead compared to the plain text.

V. DISCUSSION

The healthcare providers that participated in IMY's regulatory test operation were data controllers for local patient data processing. The setup in the IMY study required a central party that was given plain text access to all the model updates at each iteration. This was considered a potential transfer of personal data, however, IMY did not assess whether there is a legal basis for a healthcare provider to process personal data originating from another healthcare provider.

Other examples of issues that were not considered in the pilot project, including how the right to information of the registered individuals should be met and the question of the data controller's requirement not to handle more personal data than necessary (the principle of data minimization).

Information symmetry: In the federated learning framework proposed in this article based on encrypted aggregation in a tree-like structure, all parties have symmetric information access. There is thus no party with privileged access to the plaintext model updates of all other parties. Each party knows its own local data and model update, as well as the corresponding global information (recall Table II).

A curious participant can additionally subtract their own model update from the global update, to learn the aggregate model update of all other parties (except themselves). Because of the encrypted aggregation, they do not directly access the model update from any individual party. In a set-up with more than two parties, the information from a single party is now blended with that of many other parties.

With a larger aggregate batch size (in the hundreds of data points), gradient inversion attacks like [18] should be difficult to launch successfully, especially if they target a single party's data that is now diluted in the aggregate. The addition of noise in the training process should make it harder, even if full differential privacy may not always be appropriate because of the utility-privacy trade-off.

Data minimization: The minimal information that is needed by each party for improving the model (in addition to their own

data) is the global model update. This is the only (non-trivial) information that is shared in plain text in the proposed framework. If one further requires these to be deleted after used to update the model, one additionally meets *storage minimization* criteria.

Purpose limitations: During model training, the encrypted model updates that are sent up in the binary tree have to be added, but this is also the only meaningful use (for a non-malicious participant as in our trusted-but-curious set-up). The parties that participate in the exercise, have agreed not to make unintended use of the information they are trusted with; however, data remaining on the system could be used by a future actor with other intentions. Therefore model updates should be deleted after use. The final parameters of the model are, however necessary to maintain for the deployment of the model.

Deployment: A party must know the global model parameters for deploying the resulting model on its own system. Model parameters are thus to be considered part of the minimal set in a self-deployment scenario. It is, however not necessary for every party to keep the model parameters if the model is run by a single member of the federation. This opens new privacy concerns when data is sent for inference to the centrally hosted model, which can be mitigated by the use of homomorphic encryption for the transferred data. A remaining concern in such a setup may be that the model hosting party must be trusted with plain-text access to the model parameters.

In an effort of extreme data minimization, each party could encrypt model parameters with their own secret key and outsource the running of the model to a third party (that does not have access to the decryption key). They then permanently erase all traces of the model parameters (and model updates) in their own systems and only use the hosted encrypted models, in each request sending it data encrypted with their own key. They must maintain the secret key in order to decrypt the returning results. Albeit computationally expensive, such a solution mitigate membership inference attacks and model inversion attacks by preventing an attacker from knowing the model parameters (white-box), reducing opportunities to a much harder black-box access-only scenario. However, because of the high computational cost, one should first consider plain text deployment in a secure processing environment where data is protected with conventional encryption while being transferred between client and server in the organization.

Scaling: It may be preferable to use a scheme that supports SIMD operations, as the ability to encrypt vectors and carry out parallel element-wise addition greatly improves performance for larger models. This was illustrated in this work by the comparison of training the smaller logistic regression model and the larger ResNet model. The performance versus accuracy trade-off was more favorable for CKKS as this scheme supports SIMD operations and compact representation of an encrypted vector, which reduces both the computational effort and the data file size. For the federated learning under FHE experiment, only the top layer (of 3591) parameters of ResNet-18 were trained, which could fit within two ciphertexts for CKKS with the smaller key size. If we were to train all layers in the network with 11511784 parameters, it would instead require 5621 cipher texts. Assuming linear scaling, the computation time for adding the numbers would grow to 3.4 s (although the operations could be parallelized over the ciphertexts for faster execution). Similarly, the file size would grow to almost 460 MB. For TFHE, this would require one cipher text per parameter, with clearly worse scaling of time and memory cost by factors of hundreds compared to the CKKS estimates.

Total overhead: If we exclude the loading of keys and other operations that do not cause repeated overhead and only consider the overhead for using homomorphic encryption that is part of each federated learning iteration, we have

1) τ_{enc}: encryption of parameter vectors
2) τ_{add}: addition of encrypted vectors
3) τ_{com}: transfer of encrypted vectors
4) τ_{dec}: decryption of aggregated vector

Of these, we count (1) only once as it is carried out in parallel across all parts of the federation. The overheads from (2) and (3) are counted once per level in the aggregation tree, as they happen simultaneously for all participants at that level, and hence have an impact logarithmic in the size n of the federation. Finally, (4) is done only by the aggregating party and also only happens once. The total overhead then be estimated as

$$\tau\mathrm{tot} = \tau_{\mathrm{enc}} + \tau_{\mathrm{add}} \log_2 n + \tau_{\mathrm{com}} \log_2 n + \tau_{\mathrm{dec}},$$

where we have assumed that the data processing and transport is synchronized between the parties to avoid lag.

Legal uncertainty: Homomorphic encryption, differential privacy, and federated learning are not well covered by existing privacy laws and regulations, and their use can raise questions about compliance and liability. On the other hand, it can help manage and mitigate legal and regulatory risks by providing organizations with more robust and verifiable mechanisms for complying with privacy laws. For example, an organization can provide evidence of its efforts to protect personal data. Furthermore, the combined use of the technologies may enable organizations to collaborate on new research and development projects that would otherwise not be feasible. Such efforts could improve AI-based treatment methods that lead to better health outcomes as well as commercial opportunities without compromising the privacy of the individuals who ultimately contributed the data.

VI. Conclusions

Numerical experiments where two standard text-book models were trained both under differential privacy and in a federated learning set-up that uses homomorphic encryption for model parameter aggregation. The experiments confirmed that differential privacy can have a significant adverse impact on utility for some training scenarios. For the use of FHE, it was concluded that the approach was feasible not only for the trivial regression model but also for the more advanced deep learning model. Of the two FHE schemes tested, the performance versus accuracy trade-off was more favorable for CKKS as this scheme supports SIMD operations and compact representation of an encrypted vector, which reduces both the computational effort and the data file size.

Future Work: We would like to extend the numerical experiments to also compare the framework based on FHE with alternatives that use secure aggregation in combination with differential privacy. We will continue to explore implementation together with Swedish healthcare providers to gain a roadmap to practical PET application.

ACKNOWLEDGMENT

We thank Dr. Magnus Clarin, Head of Research and Education, Region Halland, and Dr. Magnus Kjellberg, Head of AI Competence Center at Sahlgrenska University Hospital, Västra Götalandsregionen, for their contribution to the work that is described in this manuscript.

REFERENCES

[1] Schork NJ. Artificial intelligence and personalized medicine. In: Precision medicine in Cancer therapy. Springer; 2019. p. 265-83.

[2] Fitzpatrick F, Doherty A, Lacey G. Using artificial intelligence in infection prevention. Current treatment options in infectious diseases. 2020;12(2):135-44.

[3] Olatunji IE, et al. A Review of Anonymization for Healthcare Data. arXiv preprint arXiv:210406523. 2021.

[4] Torkzadehmahani, et al. Privacy-preserving AI techniques in biomedicine. Methods of Inf in Med. 2022.

[5] Gentry C. Computing arbitrary functions of encrypted data. Comm of the ACM. 2010 mar.

[6] Rieke N, et al. The future of digital health with federated learning. npj Digital Medicine. 2020 sep;3(1).

[7] Xu J, Glicksberg BS, Su C, Walker P, Bian J, Wang F. Federated Learning for Healthcare Informatics. Journal of Healthcare Informatics Research. 2020 nov;5(1):1-19.

[8] Antunes RS, da Costa CA, Küderle A, Yari IA, Eskofier B. Federated Learning for Healthcare: Systematic Review and Architecture Proposal. ACM Trans on Intelligent Systems and Tech. 2022;13(4).

[9] Kairouz P, et al. Advances and Open Problems in Federated Learning. Now Publishers; 2021.

[10] Shokri R, Stronati M, Song C, Shmatikov V. Membership Inference Attacks Against Machine Learning Models. In: 2017 IEEE Symposium on Security and Privacy (SP). IEEE; 2017. .

[11] Yeom S, Giacomelli I, Fredrikson M, Jha S. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In: 2018 IEEE 31st Computer Security Foundations Symp (CSF). IEEE; 2018. .

[12] Leino K, Fredrikson M. Stolen Memories: Leveraging Model Memorization for Calibrated White-Box Membership Inference. In: Proc of the 29th USENIX Conf on Security Symp. SEC'20. USA; 2020. .

[13] Rezaei S, Liu X. On the Difficulty of Membership Inference Attacks. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE; 2021. .

[14] Jayaraman B, Evans D. Evaluating Differentially Private Machine Learning in Practice. In: Proceedings of the 28th USENIX Conference on Security Symposium. SEC'19. USA; 2019. p. 1895–1912.

[15] Fredrikson M, Eric Lantz SJ, Lin S, Page D, Ristenpart T. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. In: Proceedings of the USENIX Security Symposium. USENIX; 2014. .

[16] Carlini N, Liu C, Erlingsson U, Kos J, Song D. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In: Proceedings of the 28th USENIX Conference on Security Symposium. SEC'19. USA: USENIX Association; 2019. p. 267–284.

[17] Zhu L, Han S. Deep Leakage from Gradients. In: Lecture Notes in Computer Science. Springer International Publishing; 2020. p. 17-31.

[18] Geiping J, Bauermeister H, Dröge H, Moeller M. Inverting Gradients - How Easy is It to Break Privacy in Federated Learning? In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS'20. Red Hook, NY, USA; 2020. .

[19] Dimitrov DI, Balunovic M, Konstantinov N, Vechev M. Data Leakage in Federated Averaging. Trans on Machine Learning Research. 2022.

[20] Brännvall R, et al. Homomorphic encryption enables private data sharing for digital health: winning entry to the Vinnova innovation competition Vinter 2021-22. In: Proceedings of the Swedish Artificial Intelligence Symposium (SAIS); 2022. .

[21] Augot D, et al. Initial recommendations of long-term secure post-quantum systems; 2015. .

[22] Cheon JH, et al. Homomorphic Encryption for Arithmetic of Approximate Numbers. In: Advances in Cryptology – ASIACRYPT 2017. Springer International Publishing; 2017. p. 409-37.

[23] Chillotti I, et al. TFHE: Fast Fully Homomorphic Encryption Over the Torus. Journal of Cryptology. 2019 apr;33(1):34-91.

[24] TenSeal library implements CKKS for the Python language;. OpenMined. Accessed: 2023-03-30. https://github.com/OpenMined/TenSEAL.

[25] Concrete library implements TFHE for the Rust langauge;. ZAMA. Accessed: 2023-03-30. https://concrete.zama.ai.

[26] Phong, et al. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. IEEE Trans on Information Forensics and Security. 2018 may.

[27] Ou W, Zeng J, Guo Z, Yan W, Liu D, Fuentes S. A homomorphic-encryption-based vertical federated learning scheme for rick management. Computer Science and Information Systems. 2020;17(3):819-34.

[28] 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 Conference on Usenix Annual Technical Conference. USENIX ATC'20. USA; 2020. .

[29] Park J, Yu NY, Lim H. Privacy-Preserving Federated Learning Using Homomorphic Encryption With Different Encryption Keys. In: 2022 13th International Conference on Information and Communication Technology Convergence (ICTC). IEEE; 2022. .

[30] Dwork C. Differential Privacy: A Survey of Results. In: Lecture Notes in Computer Science. Springer Berlin Heidelberg; 2008. p. 1-19. Available from: https://doi.org/10.1007/978-3-540-79228-4_1.

[31] Mironov I. Rényi Differential Privacy. In: 2017 IEEE 30th Computer Security Foundations Symposium (CSF). IEEE; 2017. .

[32] Altschuler* JM, Talwar* K. Privacy of Noisy Stochastic Gradient Descent: More Iterations without More Privacy Loss. In: NeurIPS; 2022. .

[33] Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, et al. Practical secure aggregation for privacy-preserving machine learning. In: Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS). Dallas, TX; 2017. p. 1175-91.

[34] Bell JH, Bonawitz KA, Gascón A, Lepoint T, Raykova M. Secure Single-Server Aggregation with (Poly)Logarithmic Overhead. In: Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS). online; 2020. p. 1253-69.

[35] Kairouz P, Liu Z, Steinke T. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In: International Conference on Machine Learning; 2021. p. 5201-12.

[36] Hälsodata som nationell resurs för framtidens hälso- och sjukvård;. Regeringskansliet, Sverige, S2022:04.

[37] Utredningen om hälsodata som nationellt intresse – en lagstiftning för interoperabilitet;. Regeringskansliet, Sverige, S2022:10.

[38] Uppdrag att genomföra kunskapshöjande insatser avseende integritets- och dataskyddsfrågor inom innovations-, utvecklings- och införandeprocesser;. Government of Sweden, N2020/01266.

[39] Kardeby V, Ardeshiri T, Eklund D. Bilaga nationellt datalagringsutrymme för bilddiagnostik. Sweden; 2022.

[40] Vinnova. [DNR I2021/02737] Slutrapport i regeringsuppdraget att kartlägga behov av utvecklingsinsatser för datadelning. Sweden; 2022.

[41] Decentralized AI;. AI Sweden. Accessed: 2023-03-30. https://www.ai.se/en/projects-9/decentralized-ai.

[42] Regulatory Sandboxes and Experimentation Clauses as tools for an innovation-friendly, future-proof and resilient regulatory framework that masters disruptive challenges in the digital age - Council conclusions, 16 November 2020.;. Accessed: 2023-03-30.

[43] IMY-2023-2602. Federerad maskininlärning mellan två vårdgivare. Slutrapport om Integritetsskyddsmyndighetens pilotprojekt med regulatorisk testverksamhet om dataskydd. Sweden; 2023.

[44] Framingham 10-year coronary heart disease risk. Kaggle;. Available from: https://www.kaggle.com/amanajmera1/framingham-heart-study-dataset/data.

[45] Tschandl P. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Harvard Dataverse; 2018. Available from: https://doi.org/10.7910/DVN/DBW86T.

[46] Yang J, Shi R, Ni B. MedMNIST Classification Decathlon: A Lightweight AutoML Benchmark for Medical Image Analysis. In: IEEE 18th International Symposium on Biomedical Imaging (ISBI); 2021. p. 191-5.

[47] Yang J, Shi R, Wei D, Liu Z, Zhao L, Ke B, et al. MedMNIST v2: A Large-Scale Lightweight Benchmark for 2D and 3D Biomedical Image Classification. arXiv preprint arXiv:211014795. 2021.

[48] Yousefpour A, et al. Opacus: User-Friendly Differential Privacy Library in PyTorch. arXiv preprint arXiv:210912298. 2021.

[49] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE; 2016. .

[50] ImageNet;. http://www.image-net.org.

[51] Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, et al. A Comprehensive Survey on Transfer Learning. Proceedings of the IEEE. 2021;109(1):43-76.