

# Secure Neuroimaging Analysis using Federated Learning with Homomorphic Encryption

Dimitris Stripelis<sup>\*a</sup>, Hamza Saleem<sup>\*c</sup>, Tanmay Ghai<sup>\*c</sup>, Nikhil Dhinagar<sup>\*b</sup>, Umang Gupta<sup>a</sup>, Chrysovalantis Anastasiou<sup>c</sup>, Greg Ver Steeg<sup>a</sup>, Srivatsan Ravi<sup>a</sup>, Muhammad Naveed<sup>c</sup>, Paul M. Thompson<sup>b</sup>, and José Luis Ambite<sup>a</sup>

<sup>a</sup>Information Sciences Institute, University of Southern California

<sup>b</sup>Imaging Genetics Center, Mark and Mary Stevens Institute for Neuroimaging and Informatics, Keck School of Medicine, University of Southern California

<sup>c</sup>Viterbi School of Engineering, University of Southern California

## ABSTRACT

Federated learning (FL) enables distributed computation of machine learning models over various disparate and remote data sources, without requiring to transfer any individual sample to a centralized location. This results in improved model generalization and efficient scaling of computation as more sources and larger datasets are added to the federation. Nevertheless, recent membership inference attacks show that private or sensitive personal data can sometimes be leaked or inferred when model parameters or summary statistics are shared with a central site, requiring improved security solutions. In this work, we propose a framework for *secure* FL using fully-homomorphic encryption (FHE). Specifically, we use the CKKS construction, an approximate, floating point compatible scheme that benefits from ciphertext packing and rescaling. In our evaluation on a large-scale brain MRI dataset, we use our proposed secure FL framework to train a deep learning model to predict a person’s age from distributed MRI scans, a common benchmarking task, and demonstrate that there is no degradation in the learning performance between the encrypted and non-encrypted federated models.

**Keywords:** federated learning, homomorphic encryption, secure computation, neuroimaging, MRI

## 1. INTRODUCTION

Deep learning and traditional machine learning methods are now widely applied across all fields of medical imaging to facilitate image reconstruction and enhancement, automated segmentation and labeling of key structures, as well as computer aided diagnosis, pathology detection, disease subtyping, and predictive analytics (e.g., modeling future recovery or decline). In brain imaging in particular, there is great progress in automated diagnostic classification and subtyping of diseases such as Alzheimer’s disease and Parkinson’s disease, to assist in patient management and monitoring, and to screen patients for eligibility for clinical trials. Some recent MRI-based classifiers have merged data from over 80,000 individuals for diagnostic classification.<sup>1</sup>

Even so, for many tasks, publicly available training data is limited, and vast repositories of raw neuroimaging data may not be shared due to privacy concerns, as well as ethical and legal limitations on data sharing, and patient consent. In addition to local regulations that may govern or restrict data transfer, national and international laws, such as HIPAA or the General Data Protection Rule (GDPR) in Europe, severely limit the transfer of personal data across organizations and countries, even when data is anonymized as much as possible.

Federated learning offers a distributed machine learning training paradigm to perform computations on diverse datasets that may be geographically distributed across the world, or on multiple servers, by relaxing the need to centralize the data in any single location for training or testing. A federation consists of a set of learning sites (i.e., learners/clients), each performing computations on its local data, and a centralized controller that integrates information from the participating sites, and coordinates the distributed computation. The ability to

---

\* equal contribution. Send correspondence to D.S (stripeli@isi.edu), H.S. (hsaleem@usc.edu), P.M.T (pthomp@usc.edu) and J.L.A (ambite@isi.edu)

learn from multiple remote data sources results in models that can generalize better than the models learned from a specific data source, while maintaining computational efficiency as the number of sites scales and datasets at each site increase in size and content.

## 2. RELATED WORK

Several recent papers tackle the problem of learning over distributed neuroimaging datasets. The ENIGMA Consortium<sup>2</sup> has coordinated meta-analyses of statistical models from over 150 neuroimaging centers in 45 countries. Typically, parameters from models are estimated locally, and parameter estimates are shared with a central site for meta-analysis. Plis et al. (2016) introduced the COINSTAC system<sup>3</sup> for decentralized processing of medical imaging data, allowing iterative computations, such as N-site PCA and ICA, fitting models to local datasets and merging parameters without transferring raw data. Recent workshops on distributed and collaborative learning in biomedical imaging have focused on federated gradient averaging and multivariate linear modeling,<sup>4,5</sup> learning from non-IID or unbalanced data at the participating sites,<sup>6</sup> as well as adaptive weighting, asynchronous and semi-synchronous schemes that allow distributed learning in heterogeneous computing environments.<sup>7,8</sup>

Other works focus on *personalized* FL,<sup>9,10</sup> where models are trained on distributed data but customized to perform well on a specific dataset; in this context, the federation offers an auxiliary source of relevant data for a given task, similar in spirit to transfer learning or pre-training on auxiliary tasks. A similar approach based on Batch-Normalization (BN) has also been proposed,<sup>11</sup> where only the BN-transformation parameters are shared during federated training, leading to learn a per-site model instead of a single global model, and thereby accounting for heterogeneity via model personalization. Frameworks addressing fairness in federated biomedical applications<sup>12</sup> have also been proposed through the decomposition of locally learned representations into target and sensitive codes. FL has been applied in real-world collaborative medical settings, such as the Breast imaging-reporting and data system (BI-RADS) tool, where a federated model for the breast density classification problem was jointly trained across seven medical institutions.<sup>13</sup>

### 2.1 Data Privacy and Security

Several authors have tackled data privacy and security aspects of distributed computations.<sup>3,11,14</sup> Andreux et al<sup>11</sup> noted that even in FL, sharing aggregated statistics or model parameters can leak sensitive personal information. In a study of membership inference attacks,<sup>14</sup> we recently showed that it is possible to infer if a person’s data was used to train a model given only access to the model prediction (black-box) or access to the model itself (white-box), and some leaked samples from the training data distribution. We correctly identified whether specific MRI scans were used in model training with a 60% to over 80% success rate depending on model complexity and security assumptions. Differential privacy<sup>15</sup> constitutes another proposed privacy-preserving data mining solution for biomedical settings.<sup>3</sup> However, as already shown in,<sup>14</sup> even though it is an effective mechanism, it can lead to significantly increased model training costs and large drops in model performance.

### 2.2 Neuroimaging Analysis

The focus of our work is on the problem of predicting subjects’ brain age from 3D MRI scans in an encrypted federated learning environment. Brain Age Gap Estimation (BrainAGE) from brain structural MRIs is a challenging biomedical task, which is an important biomarker to assess and diagnose an individual’s risk of neurological diseases. The age value difference between the predicted and chronological brain age has been shown to be abnormally increased in a variety of neurological and psychiatric disorders, and even associated with all-cause mortality.<sup>16</sup> Recently, this task has been used for *benchmarking*, as ground truth (the person’s real age) is known. Deep learning methods have been used to predict an individual’s brain age both in centralized<sup>17–21</sup> and federated learning settings.<sup>8</sup> In our study, we perform the BrainAge prediction task using a 2D Convolutional Neural Network (CNN), which was shown<sup>21</sup> to yield better predictive performance compared to its 2D-Slice-RNN<sup>19</sup> and 3D-CNN<sup>17,20</sup> counterparts.

## 2.3 Federated Learning

Federated Learning was introduced in the seminal work of McMahan et al. (2017).<sup>22</sup> The task was to jointly train machine learning models over mobile phones using the privately held user data without any data ever leaving the training device. The originally proposed algorithm, *Federated Average* (FedAvg), assigns a fixed number of training steps to each participating device and periodically aggregates the locally trained models to compute a new community (global) model. This aggregation (synchronization) point is expressed in terms of federation rounds. Once the new community model is computed, it is distributed to all the devices and a new federation round begins. The FedAvg algorithm has been widely adopted.<sup>23–27</sup> In our work, we employ the FedAvg algorithm to distributively train our encrypted federated model (see section 5). However, our architecture is general and supports other training policies.

Depending on the federated learning setting, the number of participating learners, and the amount of data, two distinct federated learning environments are commonly considered:<sup>25,26,28</sup> the *cross-device* and the *cross-silo* environments. The cross-device environments characterize federations with a large number of participating learners, such as internet of things (IoT) and mobile devices, with each learner holding a relatively small amount of data, and low client availability. The cross-silo federated environments consist of a relatively small number of learners, such as organizations and data centers, with larger amounts of data and high client availability. Moreover, depending on the execution workflow and computing plan, different federated learning topologies may exist<sup>28</sup> with *centralized* and *decentralized* being the two most prominent. In the centralized case, a central entity exists (federation controller), which is responsible to coordinate the participating learners and aggregate their local models, while in the decentralized case the learners can communicate directly to each other (peer-to-peer) and aggregate models without the presence of a centralized controller. In the biomedical domain that we investigate in this work, our primary focus is on the cross-silo, centralized federated learning environments, such as a network of hospitals or a research consortium, that want to collaboratively train a machine learning model over their privately owned biomedical data.

## 2.4 Homomorphic Encryption in Federated Settings

Homomorphic Encryption (HE) has also been used in various federated settings in recent years. Zhang et al. present BatchCrypt,<sup>29</sup> a system for secure FL in the cross-silo setting. They present a quantization scheme to encode weight updates as a batch of gradients, which can then be processed by leveraging single instruction multiple data (SIMD) methods. Using the Paillier scheme, they demonstrate that their technique improves the training efficiency, with a small drop in accuracy. Truex et al.<sup>30</sup> use both differential privacy and additive HE to train ML models in the FL setting. However, using the Paillier scheme naively for HE operations (i.e. without any optimizations, such as batching), introduces a large computational overhead. POSEIDON<sup>31</sup> uses FHE in FL for model training. However, they consider a model in which the complete FL process, including the local training done by individual learners, is encrypted; such a formulation adds significant computational overhead. Their work suggests to combine the gradient updates using a tree-like network to reduce the learners' communication, instead of using a centralized federated learning topology where all learners interact directly with a central controller. Ma et al.<sup>32</sup> applied the CKKS scheme to FL, but through a multi-key HE approach, called xMK-CKKS, which employs an aggregated public key with model decryption occurring after clients share information of their individual secret keys.

## 3. FEDERATED LEARNING

In Federated Learning settings, the learning task is to find a set of optimal parameters  $w^*$  that can minimize the global objective function:

$$w^* = \underset{w}{\operatorname{argmin}} f(w) \quad \text{where} \quad f(w) = \sum_{k=1}^N \frac{p_k}{\mathcal{P}} F_k(w) \quad (1)$$

with  $N$  representing the total number of participating learners,  $p_k$  the contribution value of every learner  $k$  in the federation,  $\mathcal{P} = \sum p_k$  the normalization factor ( $\sum_k \frac{p_k}{\mathcal{P}} = 1$ ), and  $F_k(w)$  the local objective function of learner  $k$ . We refer to the model computed using Equation 1 as the community (global) model  $w_c$ . During

federated training, every learner computes its local objective function by minimizing the empirical risk over its local training dataset  $D_k^T$  as  $F_k(w) = \mathbb{E}_{x_k \sim D_k^T}[\ell_k(w; x_k)]$ , with  $\ell_k$  denoting the loss function. In the original FedAvg algorithm,<sup>22</sup> the weighting (contribution) value for any learner  $k$  in the federation is equal to the size of its local training set,  $p_k = |D_k^T|$  and every learner minimizes its local objective function for a predefined number of steps, expressed as epochs, using Stochastic Gradient Descent (SGD) as its local solver ( $\eta =$  learning rate):

$$w_{t+1} = w_t - \eta \nabla F_k(w_t) \quad (2)$$

Once every learner completes its local training task, it shares its local model with the federation controller, and a new community model is computed.

## 4. HOMOMORPHIC ENCRYPTION

### 4.1 Introduction to HE

A homomorphic encryption (HE) scheme, unlike regular cryptographic schemes, allows for certain operations (e.g. addition, multiplication) to be performed directly over encrypted data without a need for decryption. Formally, such a scheme is *homomorphic* if it satisfies the following equation:  $E(m_1) * E(m_2) = E(m_1 * m_2)$   $\forall m_1, m_2 \in M$ , where  $*$  represents a homomorphic operation, and  $M$  represents the set of all possible messages.<sup>33</sup>

A HE scheme can be primarily described by 4 main algorithms: *KeyGen*, *Enc*, *Dec*, and *Eval*; they are individually discussed below:

- $KeyGen(1^\lambda) \rightarrow (p_k, s_k)$ : Takes as input the security parameter  $\lambda$ , and outputs a pair of keys: a public key  $p_k$  and a private key  $s_k$ .
- $Enc(p_k, m) \rightarrow c$ : Takes as input the public key  $p_k$ , and message  $m$ , and outputs the ciphertext  $c$ .
- $Dec(s_k, c) \rightarrow m$ : Takes as input the private key  $s_k$ , and ciphertext  $c$  and outputs the message  $m$ .
- $Eval(p_k, F, c_1, c_2, \dots, c_n) \rightarrow c^*$ : Takes as input the public key  $p_k$ , a permitted evaluation function  $F$ , and the ciphertexts  $c_1, \dots, c_n$  and computes  $F(c_1, \dots, c_n)$ . The evaluation is correct if the following holds:  $Dec(s_k, Eval(p_k, F, c_1, \dots, c_n)) = F(m_1, \dots, m_n)$ , where  $\{c_1, \dots, c_n\}$  is the encryption of  $\{m_1, \dots, m_n\}$

The security parameter  $\lambda$ , informally, refers to how *hard* (computationally) it is for an adversary to successfully break the encryption scheme. In general, a message  $m \in M$  can be a string, integer, or other type of encoding, but for our purposes, we work with vectors of real (floating point) numbers. Additionally, in many instances, a HE scheme can be classified as one of the following: partially-homomorphic (PHE), somewhat-homomorphic (SHE), or fully-homomorphic (FHE). PHE schemes allow for either additive or multiplicative operations over ciphertexts, SHE allow for both but up to a pre-defined limit, and FHE allow for an arbitrary number of additions and multiplications (un-bounded). In our study, we employ a fully-homomorphic encryption scheme due its inherent support for an unbounded number of additive and multiplicative operations, which are essential during the aggregation of encrypted models in federated settings.

### 4.2 CKKS

In this work, we apply the Cheon-Kim-Kim-Song (CKKS)<sup>34</sup> fully-homomorphic construction, which is based on the *hardness* of the Learning-With-Error (LWE),<sup>35</sup> or its ring variant (RLWE)<sup>36</sup> problem. Unlike other FHE schemes, such as the BGV<sup>37</sup> and BFV<sup>38</sup> (integer) constructions, CKKS allows for approximate arithmetic over real and floating point numbers. It is an approximate scheme in the sense that it provides limited (configurable) precision by treating the encryption noise as natural error incurred through approximate computations, and through dropping the least significant bits of computations via the *rescaling* of encrypted data.

Rescaling refers to the underlying process of limiting ciphertext noise, and keeping the scale (controlling the precision) constant throughout a pre-defined number of multiplications allowed in the computation. Due to this, compounded computations (specifically multiplications), scale much more efficiently compared to the integer schemes as they no longer need to be exact. Furthermore, CKKS benefits from *packing*, the process of slotting multiple data values into one ciphertext, which allows for encrypted computations to be done in a Single Instruction Multiple Data (SIMD) fashion. These properties make CKKS useful for our particular use case. We refer the reader to<sup>34</sup> for more specific details.

## 5. FEDERATED LEARNING WITH HOMOMORPHIC ENCRYPTION

In a centralized federated learning environment, each learner trains on its local dataset for an assigned number of local epochs, and upon completion of its local training task, it sends the local model to the federation controller to compute the new community model. In an encrypted centralized federation environment, the procedure is similar with the addition of three pivotal key steps: *encryption*, *encrypted-aggregation*, *decryption*. During the encryption step, every learner encrypts its locally trained model with an HE scheme using the public key, and sends the encrypted model (ciphertext) to the controller. For each learner, its encrypted model is treated as a vector of ciphertext objects, each object corresponding to a model array. With this approach, the encrypted data is represented as a (concatenated) collection of flattened data-vectors, each of them representing the local data for a particular learner. The controller receives all the encrypted local models, and then performs the encrypted weighted-aggregation to compute the new encrypted community model without ever decrypting any of the individual models. Subsequently, the controller sends the new community model to all the learners, and the learners decrypt it using the private key. Once the decryption is complete, the learners train the (new) decrypted model on their local data set and the entire procedure repeats. This pipeline is represented schematically in Fig. 1, and algorithmically in Alg. 1. In our setup, the (encrypted) weighted aggregation rule applied by the controller on learners' local models is based on the local training dataset size of each learner (i.e., FedAvg).

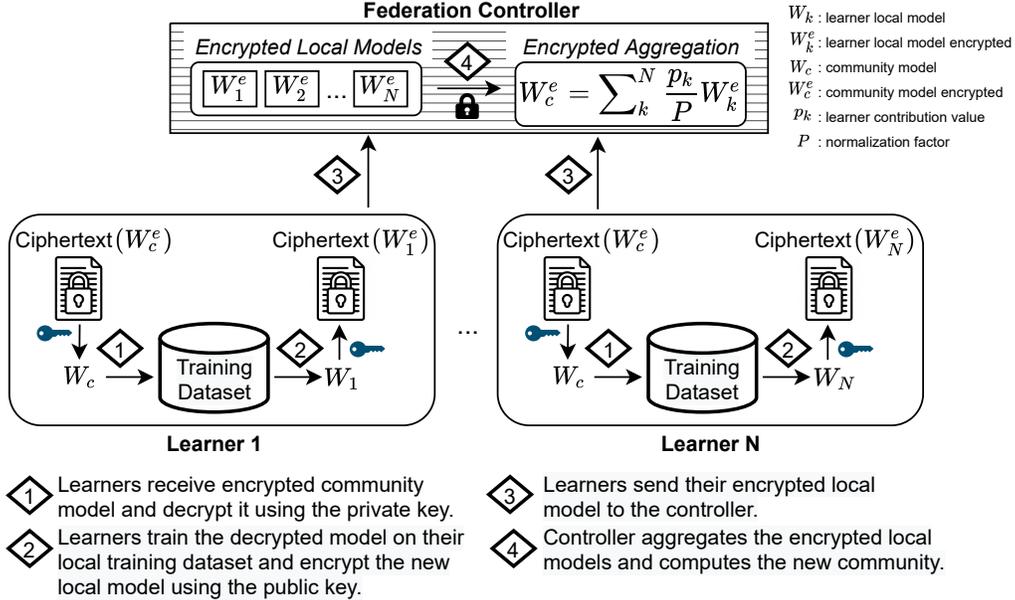


Figure 1. Federated System Architecture with Encryption

---

**Algorithm 1** Federated Learning with HE. Encrypted community model  $w_c^e$  is computed with  $N$  clients, each indexed by  $k$ ;  $\beta$  is the batch size;  $\eta$  is the learning rate;  $E$  are the local epochs.

---

**Initialize:**  $w_c, \eta$

**Controller:**

```

 $w_c^e$  = encrypt initial community model
for  $t = 0, \dots, T - 1$  do
  for each learner  $k \in N$  in parallel do
    send encrypted community model  $w_c^e$ 
     $w_k = \text{LEARNEROPT}(w_c^e)$ 
   $w_c^e = \text{encrypted aggregation of all } w_k$ 
  
```

**LearnerOpt( $w_t^e$ ):**

```

 $w_t = \text{decrypt community model } w_t^e$ 
 $\mathcal{B} \leftarrow \text{Split training data } D_k^T \text{ into batches of size } \beta$ 
for  $i \in E$  do
  for  $b \in \mathcal{B}$  do
     $w_{t+1} = w_t - \eta \nabla F_k(w_t; b)$ 
   $w_{t+1}^e = \text{encrypt } w_{t+1}$ 
  send  $w_{t+1}^e$  to controller
  
```

---

## 6. EVALUATION

We evaluate the learning performance of FedAvg with and without encryption on the BrainAge prediction task over the UK Biobank (UKBB) neuroimaging dataset<sup>39</sup> for several data distributions environments.

**UKBB Dataset.** For our evaluation, we use as a total of 10,446 MRI scans<sup>19</sup> from subjects with no psychiatric diagnosis as defined by ICD-10 criteria out of the 16,356 available subjects in the UK Biobank dataset.<sup>39</sup> The scans were processed using a standard preprocessing pipeline with non-parametric intensity normalization for bias field correction<sup>1</sup> and brain extraction using FreeSurfer and linear registration to a  $(2mm)^3$  UKBB minimum deformation template using FSL FLIRT, with the final dimension of the images being equal to  $91 \times 109 \times 91$ . The dataset is split into a training and test sets of size 8,356, and 2,090, respectively.

**Federation Environments.** Our evaluation testbed consists of three federated learning environments\* with diverse data distributions across 8 learners.<sup>8</sup> Each learner runs on a dedicated GPU card on a single server. In terms of data distribution, we investigate two distinct cases: *IID*, where each learner holds training examples that follow the global data distribution, and *Non-IID*, where local distributions may differ substantially from the global. Specifically, for the BrainAge prediction task, the IID scenario refers to the case where learners hold examples from all possible age ranges, while in Non-IID each learner holds subjects over a subset of ages. In terms of data amount across learners, we explore two partitioning schemes, with *Uniform* representing the cases where learners hold equal number of examples, and *Skewed*, the case where learners hold different amounts of examples (i.e., the assignment follows a rightly skewed distribution). For every federated learning environment, the data distribution is provided as an inset inside the federated learning curves of Fig. 3.

**CKKS Parameters.** For the implementation of the CKKS scheme, we utilize the lattice-based cryptographic & HE library PALISADE.<sup>40</sup> In order to meet standards set by the Homomorphic Encryption Standard,<sup>41</sup> PALISADE allows the configurability of scheme parameters that achieve certain precision, performance and security goals that a user may have. For our work, we configured the following parameters: *multiplicative depth*=2, *scale factor bits*=52, *batch size*=8192, and *security level*=128 bits. Multiplicative depth refers to the maximum multiplicative *path*-length that may occur in a computation (e.g. multiplying  $1 \cdot 2$  and then multiplying that quantity with 3 has a path-length of 2). Scale factor bits represent the bit-length of the scaling factor  $D$  present in the CKKS scheme; choosing it controls the bit-level accuracy of the desired computation. Batch size is precisely the number of plaintext slots used in a single ciphertext. As discussed in Section 4.2, CKKS can pack multiple plaintext values in each ciphertext. Lastly, security level controls the bit-security that such an implementation would achieve according to FHE standards.

**Federated BrainAge Model.** Our 2D-CNN BrainAge model (1 million parameters in total) takes a 3D scan as input and encodes each slice using a 2D-CNN encoder. Subsequently, it combines the slice encodings using an attention-based aggregation module, resulting in a single embedding for the scan, which is then passed through the feed-forward layers to predict the age of the scan. An overview of the model architecture can be seen in Fig. 2. During federated training, all learners receive the same model architecture and hyperparameter values, ( $E = 4$ ,  $\eta = 5 \times 10^{-5}$ ,  $\beta = 1$ ), and use SGD as their local optimization function solver.

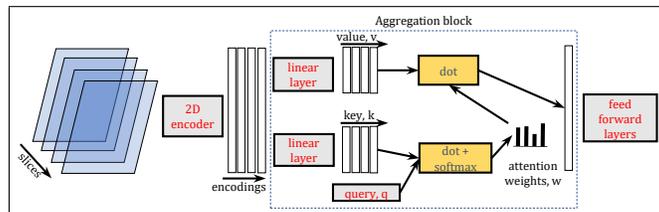


Figure 2. 2D-CNN BrainAge Model Architecture

**Results.** We compare the convergence rate of FedAvg with and without encryption in terms of federation rounds on the three previously described federated learning environments. As shown in Fig. 3, encryption does not penalize the learning performance of the federated model, and in some learning scenarios (e.g., Skewed & Non-IID) the encrypted model can lead to faster convergence compared to its non-encrypted counterpart. This

\*<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/2RKAQP>

can be attributed to the stochastic effect that the encryption scheme introduces to the weights of the federated learning model during encoding, decoding and private-aggregation and thereby acting as a regularizer during federated training. Eventually as training progresses, both the encrypted and non-encrypted models reach the same final score. As the environments become more heterogeneous (from Uniform & IID, to Uniform & Non-IID, to Skewed & Non-IID), they are harder to learn and we observe a small degradation on the final prediction error.

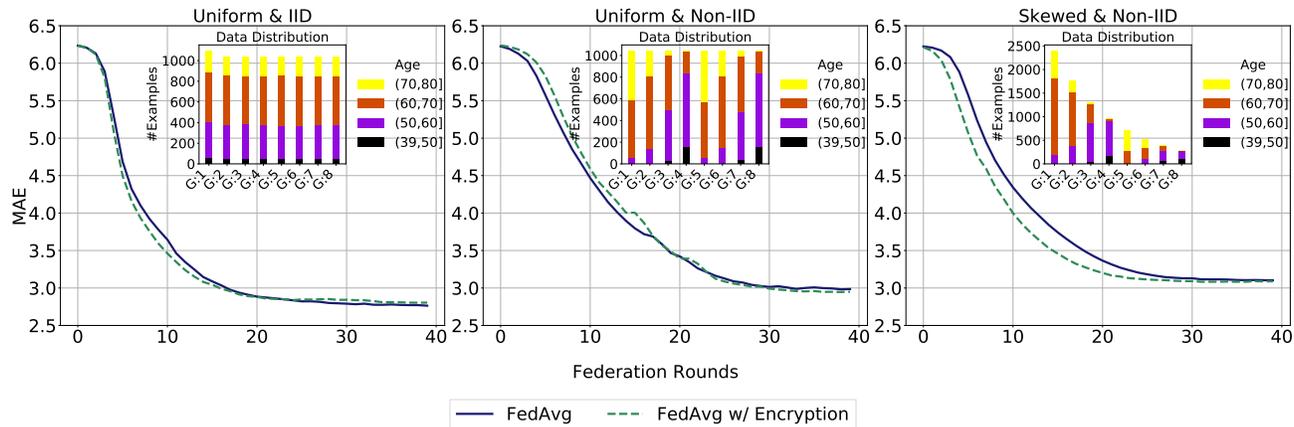


Figure 3. BrainAge Evaluation with and without Encryption

## 7. DISCUSSION

We have presented a federated learning architecture supporting fully-homomorphic encryption for privacy-preserving biomedical data analysis. We evaluated the performance of a non-encrypted and an encrypted federated model using the CKKS FHE scheme on the BrainAge prediction task over several data distributions of the UK BioBank dataset, showing no performance degradation under encryption.

In our immediate future work, we plan to investigate other homomorphic encryption schemes, such as Pailler,<sup>42</sup> BGV,<sup>37</sup> BGN,<sup>43</sup> and their respective encryption and federated training performance, as well as further adaptations that might be necessary for their successful integration into our federated training procedure (e.g., numerical scaling, protocol and data type tuning). We also plan to extend our encrypted federated neuroimaging analysis to harder learning domains, such as neurodegenerative disease prediction (e.g., Alzheimer’s and Parkinson’s diseases). Finally, we plan to explore how more efficient federated training protocols can be coupled with homomorphic encryption schemes to accelerate the convergence rate of the federation.

## 8. ACKNOWLEDGMENTS

This research was supported by DARPA contract HR0011-2090104. This research has been conducted using the UK Biobank Resource under Application Number 11559.

## REFERENCES

- [1] Lu, B., Li, H.-X., Chang, Z.-K., Li, L., Chen, N.-X., Zhu, Z.-C., Zhou, H.-X., Li, X.-Y., Wang, Y.-W., Cui, S.-X., Deng, Z.-Y., Fan, Z., Yang, H., Chen, X., Thompson, P. M., Castellanos, F. X., Yan, C.-G., and for the Alzheimer’s Disease Neuroimaging Initiative, “A practical alzheimer disease classifier via brain imaging-based deep learning on 85,721 samples,” *bioRxiv* (2021).
- [2] Thompson, P. M., Jahanshad, N., Ching, C. R., Salminen, L. E., Thomopoulos, S. I., Bright, J., Baune, B. T., Bertolin, S., Bralten, J., Bruin, W. B., et al., “Enigma and global neuroscience: A decade of large-scale studies of the brain in health and disease across more than 40 countries,” *Translational psychiatry* **10**(1), 1–28 (2020).
- [3] Plis, S. M., Sarwate, A. D., Wood, D., Dieringer, C., Landis, D., Reed, C., Panta, S. R., Turner, J. A., Shoemaker, J. M., Carter, K. W., et al., “Coinstac: a privacy enabled model and prototype for leveraging and processing decentralized brain imaging data,” *Frontiers in neuroscience* **10**, 365 (2016).

- [4] Remedios, S. W., Butman, J. A., Landman, B. A., and Pham, D. L., “Federated gradient averaging for multi-site training with momentum-based optimizers,” in [*Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*], 170–180, Springer (2020).
- [5] Silva, S., Altmann, A., Gutman, B., and Lorenzi, M., “Fed-biomed: A general open-source frontend framework for federated learning in healthcare,” in [*Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*], 201–210, Springer (2020).
- [6] Yeganeh, Y., Farshad, A., Navab, N., and Albarqouni, S., “Inverse distance aggregation for federated learning with non-iid data,” in [*Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*], 150–159, Springer (2020).
- [7] Stripelis, D. and Ambite, J. L., “Semi-synchronous federated learning,” *arXiv:2102.02849* (2021).
- [8] Stripelis, D., Ambite, J. L., Lam, P., and Thompson, P., “Scaling neuroscience research using federated learning,” in [*2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*], 1191–1195, IEEE (2021).
- [9] Grimberg, F., Hartley, M.-A., Jaggi, M., and Karimireddy, S. P., “Weight erosion: An update aggregation scheme for personalized collaborative machine learning,” in [*Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*], 160–169, Springer (2020).
- [10] Chen, Y., Qin, X., Wang, J., Yu, C., and Gao, W., “Fedhealth: A federated transfer learning framework for wearable healthcare,” *IEEE Intelligent Systems* **35**(4), 83–93 (2020).
- [11] Andreux, M., du Terrail, J. O., Beguier, C., and Tramel, E. W., “Siloed federated learning for multi-centric histopathology datasets,” (2020).
- [12] Sarhan, M. H., Navab, N., Eslami, A., and Albarqouni, S., “On the fairness of privacy-preserving representations in medical applications,” in [*Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*], 140–149, Springer (2020).
- [13] Roth, H. R., Chang, K., Singh, P., Neumark, N., Li, W., Gupta, V., Gupta, S., Qu, L., Ihsani, A., Bizzo, B. C., and et al., “Federated learning for breast density classification: A real-world implementation,” *Lecture Notes in Computer Science*, 181–191 (2020).
- [14] Gupta, U., Stripelis, D., Lam, P. K., Thompson, P., Ambite, J. L., and Ver Steeg, G., “Membership inference attacks on deep regression models for neuroimaging,” in [*Medical Imaging with Deep Learning*], (2021).
- [15] Dwork, C., “Differential privacy,” in [*International Colloquium on Automata, Languages, and Programming*], 1–12, Springer (2006).
- [16] Franke, K. and Gaser, C., “Ten years of brainage as a neuroimaging biomarker of brain aging: what insights have we gained?,” *Frontiers in neurology* **10**, 789 (2019).
- [17] Cole, J. H., Poudel, R. P., Tsagkrasoulis, D., Caan, M. W., Steves, C., Spector, T. D., and Montana, G., “Predicting brain age with deep learning from raw imaging data results in a reliable and heritable biomarker,” *NeuroImage* **163**, 115–124 (2017).
- [18] Jónsson, B. A., Bjornsdottir, G., Thorgeirsson, T., Ellingsen, L. M., Walters, G. B., Gudbjartsson, D., Stefansson, H., Stefansson, K., and Ulfarsson, M., “Brain age prediction using deep learning uncovers associated sequence variants,” *Nature communications* **10**(1), 1–10 (2019).
- [19] Lam, P. K., Santhalingam, V., Suresh, P., Baboota, R., Zhu, A. H., Thomopoulos, S. I., Jahanshad, N., and Thompson, P. M., “Accurate brain age prediction using recurrent slice-based networks,” in [*16th International Symposium on Medical Information Processing and Analysis*], **11583**, 1158303, International Society for Optics and Photonics (2020).
- [20] Peng, H., Gong, W., Beckmann, C. F., Vedaldi, A., and Smith, S. M., “Accurate brain age prediction with lightweight deep neural networks,” *Medical image analysis* **68**, 101871 (2021).
- [21] Gupta, U., Lam, P. K., Ver Steeg, G., and Thompson, P. M., “Improved brain age estimation with slice-based set networks,” in [*2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*], 840–844, IEEE (2021).
- [22] McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A., “Communication-efficient learning of deep networks from decentralized data,” in [*Artificial Intelligence and Statistics*], 1273–1282, PMLR (2017).

- [23] Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S., “Federated multi-task learning,” in [*Advances in Neural Information Processing Systems*], 4424–4434 (2017).
- [24] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, B., Van Overveldt, T., Petrou, D., Ramage, D., and Roselander, J., “Towards federated learning at scale: System design,” in [*Proceedings of Machine Learning and Systems*], Talwalkar, A., Smith, V., and Zaharia, M., eds., **1**, 374–388 (2019).
- [25] Li, T., Sahu, A. K., Talwalkar, A., and Smith, V., “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Processing Magazine* **37**(3), 50–60 (2020).
- [26] Kairouz, P. and McMahan, H. B., “Advances and Open Problems in Federated Learning,” *Foundations and Trends® in Machine Learning* **14**(1), – (2021).
- [27] Yang, Q., Liu, Y., Chen, T., and Tong, Y., “Federated machine learning: Concept and applications,” *ACM Transactions on Intelligent Systems and Technology (TIST)* **10**(2), 1–19 (2019).
- [28] Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H., Albarqouni, S., Bakas, S., Galtier, et al., “The future of digital health with federated learning,” *npj Digital Medicine* **3**(119) (2020).
- [29] Zhang, C., Li, S., Xia, J., Wang, W., Yan, F., and Liu, Y., “Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning,” in [*2020 USENIX Annual Technical Conference (USENIX ATC 20)*], 493–506, USENIX Association (July 2020).
- [30] Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., and Zhou, Y., “A hybrid approach to privacy-preserving federated learning,” in [*Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*], *AISec’19*, 1–11 (2019).
- [31] Sav, S., Pyrgelis, A., Troncoso-Pastoriza, J., Froelicher, D., Bossuat, J.-P., Sousa, J. S., and Hubaux, J., “Poseidon: Privacy-preserving federated neural network learning,” *ArXiv abs/2009.00349* (2021).
- [32] Ma, J., Naas, S.-A., Sigg, S., and Lyu, X., “Privacy-preserving federated learning based on multi-key homomorphic encryption,” *arXiv preprint arXiv:2104.06824* (2021).
- [33] Acar, A., Aksu, H., Uluagac, A. S., and Conti, M., “A survey on homomorphic encryption schemes: Theory and implementation,” (2017).
- [34] Cheon, J. H., Kim, A., Kim, M., and Song, Y., “Homomorphic encryption for arithmetic of approximate numbers,” in [*Advances in Cryptology – ASIACRYPT 2017*], Takagi, T. and Peyrin, T., eds., 409–437 (2017).
- [35] Regev, O., “On lattices, learning with errors, random linear codes, and cryptography,” *J. ACM* **56** (Sept. 2009).
- [36] Lyubashevsky, V., Peikert, C., and Regev, O., “On ideal lattices and learning with errors over rings,” **60** (Nov. 2013).
- [37] Brakerski, Z., Gentry, C., and Vaikuntanathan, V., “(leveled) fully homomorphic encryption without bootstrapping,” in [*Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*], *ITCS ’12*, 309–325, Association for Computing Machinery (2012).
- [38] Fan, J. and Vercauteren, F., “Somewhat practical fully homomorphic encryption,” *IACR Cryptol. ePrint Arch.* , 144 (2012).
- [39] Miller, K. L., Alfaro-Almagro, F., Bangerter, N. K., Thomas, D. L., Yacoub, E., Xu, J., Bartsch, A. J., Jbabdi, S., Sotiropoulos, S. N., Andersson, J. L., et al., “Multimodal population brain imaging in the uk biobank prospective epidemiological study,” *Nature neuroscience* **19**(11), 1523–1536 (2016).
- [40] team, P., “PALISADE Lattice Cryptography Library (release 1.10.6),” (2020).
- [41] Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., Lokam, S., Micciancio, D., Moody, D., Morrison, T., Sahai, A., and Vaikuntanathan, V., “Homomorphic encryption standard.” *Cryptology ePrint Archive*, Report 2019/939 (2019). <https://eprint.iacr.org/2019/939>.
- [42] Paillier, P., “Public-key cryptosystems based on composite degree residuosity classes,” in [*Advances in Cryptology — EUROCRYPT ’99*], 223–238 (1999).
- [43] Boneh, D., Goh, E.-J., and Nissim, K., “Evaluating 2-dnf formulas on ciphertexts,” in [*Theory of Cryptography*], 325–341 (2005).