ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Secure Federated Learning with Kernel Affine Hull Machines

Mohit Kumar, Bernhard A. Moser, and Lukas Fischer *

Software Competence Center Hagenberg GmbH
A-4232 Hagenberg, Austria

Abstract. The concept of Kernel Affine Hull Machine (KAHM) was recently introduced for representing data via learning in Reproducing Kernel Hilbert Spaces. KAHM defines a bounded geometric body in data space such that a distance measure from the geometric body can be used to aggregate local KAHM-based models to build a global model. This study leverages KAHMs for secure federated learning where data is protected from an aggressive aggregator by fully homomorphic encryption. An accurate and computationally efficient federated learning architecture, that combines local KAHMs-based classifiers in a robust and flexible manner such that the global model can be homomorphically evaluated in an efficient manner, is provided.

1 Introduction

Privacy-preserving distributed machine learning is increasingly gaining attention. As a result, there has been a recent surge in the interest on advanced privacy-preserving methods such as Fully Homomorphic Encryption (FHE) and differential privacy. A practical secure privacy-preserving distributed machine (deep) learning requires addressing some of the fundamental issues [1]. In particularly, the following three issues are identified: 1) Despite recent advances in FHE schemes, machine learning with fully homomorphic encrypted data remains impractical due to the large computational overhead. 2) The major limitation of the differentially private machine learning is that a sufficiently low value of privacy-loss bound results in a considerable loss of accuracy and it is not clear how to practically choose the value of privacy-loss bound. 3) The scalable and fast machine (deep) learning demands development of computationally efficient algorithms for a) training models outside the realm of slow gradient-based learning algorithms, and b) automatically determining the model size matching the complexity of the problem.

To address the issue arising from the large computational cost associated with fully homomorphic encrypted data, a membership-mappings based approach to secure distributed deep learning was suggested [2]. This approach relies on defining fuzzy attributes such that fuzzy attributes allow combining local models by

*The research reported in this paper has been supported by the Austrian Research Promotion Agency (FFG) COMET-Modul S3AI; FFG Grant SMiLe; BMK, BMAW, and the State of Upper Austria in the frame of the SCCH competence center INTEGRATE [(FFG grant no. 892418)] part of the FFG COMET Competence Centers for Excellent Technologies Programme.

means of a rule-based fuzzy system and the global model can be homomorphically evaluated efficiently. Differential privacy is another approach that preserves the privacy of the data via adding random noise to ensure that an adversary can not infer any single data instance by observing model parameters or model outputs. The amount of noise depends upon the value of privacy-loss bound and an obvious effect of adding noise is the loss in accuracy. To address the accuracy-loss issue of differential privacy, efforts have been made to optimize the privacy-accuracy tradeoff. The studies [3, 4] derive the probability density function of noise that minimizes the expected noise magnitude together with achieving differential privacy. The optimal differentially private noise adding mechanism has been applied to distributed machine learning in [4, 5], where fuzzy sets and rules were used to aggregate the local privacy-preserving deep models for building the global model.

There are three main issues pertaining to deep neural networks: determination of the optimal model structure, gradient-based iterative nature of learning algorithms, and requirement of large training data. These issues motivate an alternative kernel-based nonparametric approach. For example, fuzzy theoretic kernel based autoencoders have been introduced [6, 7, 8, 9, 10, 11, 2, 12, 4, 5], such that solutions for the learning of models are derived analytically using variational optimization technique. More recently, [13] introduced the concept of Kernel Affine Hull Machine (KAHM) such that KAHM defines a bounded region in the affine hull of data samples having learned the representation of data samples in reproducing kernel Hilbert spaces via solving a kernel regularized least squares problem. KAHM based federated learning has been previously considered in [13], however, under differential privacy. Although [13] suggests the smoothing of data for mitigating the accuracy-loss issue of the differential privacy, a low value of privacy-loss bound still leads to a drop in the accuracy. To address this limitation, this study extends the KAHM based federated learning approach to FHE setting so that privacy is protected without an accuracy loss. Moreover, the federated learning method remains computationally efficient since the method requires only the locally computed distance measures which are encrypted and can be efficiently processed for the homomorphic evaluation of the global model.

2 KAHM Based Secure Federated Learning

We consider a scenario of federated learning with Q different parties such that $q-$th party owns a dataset, $\{Y_1^q, \cdots, Y_C^q\}$, that can be partitioned into C different classes (here Y_c^q refers to the $c-$th class labelled points in $\mathbb{R}^p$ owned by $q-$th party). Following [13], our approach is of combining together the local KAHM based classifiers using the distance functions induced by local KAHMs to build a global classifier. The global classifier assigns a label to an input $y \in \mathbb{R}^p$ as

$$\mathcal{GC}(y) \quad = \quad \arg \min_{c \in \{1,2,\cdots,C\}} \left(\min_{q \in \{1,2,\cdots,Q\}} \Gamma_{\mathcal{W}_{Y_c^q}}(y) \right) \tag{1}$$

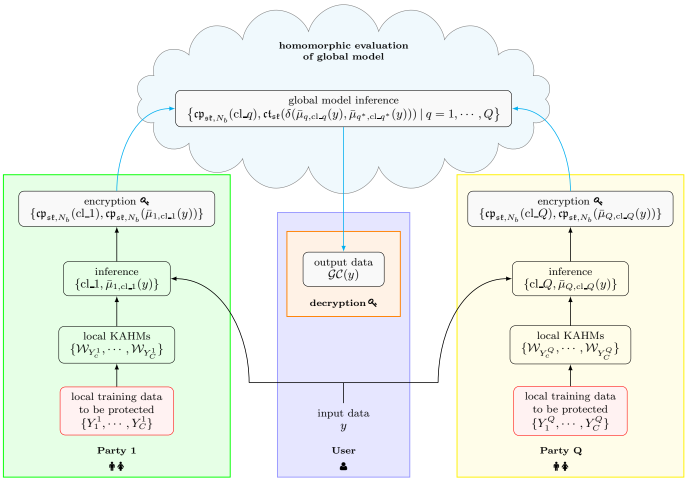
homomorphic evaluation of global model
global model inference
$\{\mathfrak{cp}_{\mathfrak{sk},N_b}(\text{cl\_}q), \mathfrak{ct}_{\mathfrak{sk}}(\delta(\bar{\mu}_{q,\text{cl\_}q}(y), \bar{\mu}_{q^*,\text{cl\_}q^*}(y))) \mid q = 1, \cdots, Q\}$
encryption
$\{\mathfrak{cp}_{\mathfrak{sk},N_b}(\text{cl\_}1), \mathfrak{cp}_{\mathfrak{sk},N_b}(\bar{\mu}_{1,\text{cl\_}1}(y))\}$
inference
$\{\text{cl\_}1, \bar{\mu}_{1,\text{cl\_}1}(y)\}$
local KAHMs
$\{\mathcal{W}_{Y_c^1}, \cdots, \mathcal{W}_{Y_C^1}\}$
local training data to be protected
$\{Y_1^1, \cdots, Y_C^1\}$
Party 1
output data
$\mathcal{GC}(y)$
decryption
input data
y
User
encryption
$\{\mathfrak{cp}_{\mathfrak{sk},N_b}(\text{cl\_}Q), \mathfrak{cp}_{\mathfrak{sk},N_b}(\bar{\mu}_{Q,\text{cl\_}Q}(y))\}$
inference
$\{\text{cl\_}Q, \bar{\mu}_{Q,\text{cl\_}Q}(y)\}$
local KAHMs
$\{\mathcal{W}_{Y_c^Q}, \cdots, \mathcal{W}_{Y_C^Q}\}$
local training data to be protected
$\{Y_1^Q, \cdots, Y_C^Q\}$
Party Q

Fig. 1: Secure federated learning based on KAHM and FHE.

where $\Gamma_{\mathcal{W}_{Y_c^q}}$ is the distance function induced by the KAHM (refer to [13] for details). Under the assumption that the minimum is reached by a unique party, the global classifier assigns to an arbitrary point y the label of the class which has the minimum distance between y and y's image onto the affine hull of samples of that class. An important feature of the global classifier evaluation using (1) is that the evaluation doesn't require individual KAHMs (that are owned by different parties) but only the distance measures. This allows to design a KAHM based secure federated learning method using TFHE scheme [14, 15]. For this, the distance measure is rescaled to lie in $[0, 1]$ via defining

$$\bar{\mu}_{q,c}(y) \quad := \quad 1 - \exp\left(-\frac{1}{2p}\left|\Gamma_{\mathcal{W}_{Y_c^q}}(y)\right|^2\right). \tag{2}$$

Define cl_q as the class-label predicted by the $q-$th local classifier, i.e.,

$$\text{cl\_}q \quad = \quad \arg \min_{c \in \{1,2,\cdots,C\}} \bar{\mu}_{q,c}(y). \tag{3}$$

Let $\delta(\mathfrak{m}_1, \mathfrak{m}_2)$ be the Kronecker delta function of $\mathfrak{m}_1, \mathfrak{m}_2 \in [0, 1]$, i.e.,

$$\delta(\mathfrak{m}_1, \mathfrak{m}_2) \quad = \quad \begin{cases} 1 & \text{if } \mathfrak{m}_1 = \mathfrak{m}_2, \\ 0 & \text{if } \mathfrak{m}_1 \neq \mathfrak{m}_2. \end{cases} \tag{4}$$

Now, (1) can be alternatively expressed as

$$\mathcal{GC}(y) \quad = \quad \sum_{q=1}^{Q} \text{cl\_}q \; \delta\left(\bar{\mu}_{q,\text{cl\_}q}(y), \bar{\mu}_{q^*,\text{cl\_}q^*}(y)\right), \text{ where} \tag{5}$$

$$q^* \quad = \quad \arg \min_{q \in \{1,2,\cdots,Q\}} \bar{\mu}_{q,\text{cl\_}q}(y). \tag{6}$$

For a given positive integer $N_b \in \mathbb{Z}_{>0}$, let $\mathfrak{pt}_{N_b} : [0,1] \to \{0, 1, \cdots, 2^{N_b} - 1\}$ be a function defined as

$$\mathfrak{pt}_{N_b}(\mathfrak{m}) \quad := \quad \lceil (2^{N_b} - 1)\mathfrak{m} \rceil, \; \mathfrak{m} \in [0,1]. \tag{7}$$

In our setting, $\mathfrak{pt}_{N_b}(\mathfrak{m})$ is the plaintext that encodes a message $\mathfrak{m}$ as unsigned N_b-bit integer. Let $\mathrm{BitDec}_{N_b} : \{0, 1, \cdots, 2^{N_b} - 1\} \to \{0,1\}^{N_b}$ be the binary representation of a N_b-bit unsigned integer. That is,

$$(\mathfrak{bt}_1(\mathfrak{m}), \cdots, \mathfrak{bt}_{N_b}(\mathfrak{m})) \quad = \quad \mathrm{BitDec}_{N_b}(\mathfrak{pt}_{N_b}(\mathfrak{m})), \tag{8}$$

where $\mathfrak{bt}_k(\mathfrak{m}) \in \{0,1\}$ for all $k \in \{1, 2, \cdots, N_b\}$. Let N_c be the ciphertext dimension set for a given value of security bits, say 128 bits security. Let $\mathfrak{sk} \in \{0,1\}^{N_c}$ be a secret key generated for TFHE encryption. Let $\mathfrak{ct}_{\mathfrak{sk}}(\mathfrak{bt}) \in \mathbb{T}^{N_c+1}$, where $\mathbb{T} = \mathbb{R}/\mathbb{Z}$, be the TFHE encryption of a bit $\mathfrak{bt} \in \{0,1\}$, i.e.,

$$\mathfrak{ct}_{\mathfrak{sk}}(\mathfrak{bt}) \quad = \quad \text{TFHE.Encryption}(\mathfrak{bt}; \mathfrak{sk}). \tag{9}$$

Let $\mathfrak{cp}_{\mathfrak{sk},N_b} : [0,1] \to \mathbb{T}^{N_b(N_c+1)}$ be a function defined as

$$\mathfrak{cp}_{\mathfrak{sk},N_b}(\mathfrak{m}) \quad := \quad \left(\mathfrak{ct}_{\mathfrak{sk}}\left(\mathfrak{bt}_1(\mathfrak{m})\right), \cdots, \mathfrak{ct}_{\mathfrak{sk}}\left(\mathfrak{bt}_{N_b}(\mathfrak{m})\right)\right) \tag{10}$$

where $\mathfrak{ct}_{\mathfrak{sk}}\left(\mathfrak{bt}_k(\mathfrak{m})\right)$ is the TFHE encryption of bit $\mathfrak{bt}_k(\mathfrak{m})$. Thus, $\mathfrak{cp}_{\mathfrak{sk},N_b}(\mathfrak{m})$ homomorphically encrypts the message $\mathfrak{m} \in [0,1]$ with N_b-bit precision.

Algorithm 1 Implementing secure federated learning based on KAHM and FHE

```
Require: Input data vector $y$; $Q$ different parties participating in collaborative learning each of which owns local KAHMs (such that $q-$th party's local KAHMs, $\{\mathcal{W}_{Y_c^q}\}_{c=1}^{C}$, are built with private dataset $\{Y_1^q, \cdots, Y_C^q\}$); and a secret key $\mathfrak{sk}$.
Choose bits precision $N_b \in \mathbb{Z}_{>0}$, e.g., $N_b = 16$.
The output of each local classifier to the input $y$ is encrypted and then exported to the cloud. That is, the $q-$th party exports $\{\mathfrak{cp}_{\mathfrak{sk},N_b}(\mathrm{cl\_}q), \mathfrak{cp}_{\mathfrak{sk},N_b}(\bar{\mu}_{q,\mathrm{cl\_}q}(y))\}$ to the cloud.
The global classifier is homomorphically evaluated in the cloud from the encrypted data sent by all parties using TFHE [16], and the resulting output (which remains encrypted) is returned to the owner of input data. That is, data $\{\mathfrak{cp}_{\mathfrak{sk},N_b}(\mathrm{cl\_}q), \mathfrak{ct}_{\mathfrak{sk}}(\delta(\bar{\mu}_{q,\mathrm{cl\_}q}(y), \bar{\mu}_{q^*,\mathrm{cl\_}q^*}(y))) \mid q = 1, \cdots, Q\}$ are returned by the cloud to the owner of input data vector $y$.
The user (i.e. owner of the input data) decrypt the encrypted data provided by the cloud. That is, user obtains after decryption $\{\mathrm{cl\_}q, \delta(\bar{\mu}_{q,\mathrm{cl\_}q}(y), \bar{\mu}_{q^*,\mathrm{cl\_}q^*}(y)) \mid q = 1, \cdots, Q\}$.
The user determines the class-label $\mathcal{GC}(y)$ associated to the input $y$ using (5).
return $\mathcal{GC}(y)$.
```

The proposed approach to homomorphically evaluate the global classifier (1) is illustrated in Fig. 1 and Algorithm 1 provides implementation procedure.

3 Experiments

Algorithm 1 was implemented using MATLAB R2017b and TFHE library [16] on a MacBook Pro laptop with a 2.2 GHz Intel Core i7 processor and 16 GB of memory. The secret key is generated using TFHE library for 128-bits of security and

method	accuracy	time (sec.)
Algorithm 1 (16-bits precision)	**0.9859**	**5**
differentially private federated learning ($\epsilon = 0.1$) [13]	0.7552	n/a
differentially private federated learning ($\epsilon = 1$) [13]	0.9470	n/a
NN-20 [17]	<u>0.971</u>	<u>115.52</u>
NN-50 [17]	0.947	233.55
NN-100 [17]	0.830	481.61

Table 1: Experimental results on MNIST dataset.

experiments are performed with the precision of 16-bits. The experiments are on the widely used MNIST digits dataset containing 28×28 sized images divided into training set of 60000 images and testing set of 10000 images. The 28×28 normalized values of each image were flattened to an equivalent $784-$dimensional data vector. A two-party scenario is considered such that Party-A owns all the training images of odd digits while Party-B owns the rest training images of even digits. The aim of the experiments is to 1) compare Algorithm 1 in-terms of test data accuracy with the alternative KAHM based federated learning method [13] where differentially private fabricated data are used for privacy-preservation; and 2) compare Algorithm 1 in-terms of test data accuracy and computational time required for secure homomorphic computations in the cloud (i.e. the time required for computing the encrypted global output for a given input) with the state-of-art study [17] on homomorphic inference of deep neural networks. The study in [17] evaluates neural networks with different depths (referred to as NN-20, NN-50, and NN-100) over TFHE fully homomorphic encrypted data. The experimental results, reported in Table 1, lead to the following inferences:

1. FHE based Algorithm 1 is more accurate than the reference differential privacy based alternative [13] in high-privacy regime (i.e. in lower range of privacy-loss bound ϵ).

2. Algorithm 1 is more accurate and several times computationally more efficient than the homomorphically evaluated deep neural networks by the reference method of [17].

4 Concluding Remarks

The recently introduced geometrically inspired kernel machines facilitate secure federated learning using FHE while offering simultaneously accuracy and computational efficiency.

References

[1] M. Kumar, B. Moser, L. Fischer, and B. Freudenthaler. Towards practical secure privacy-preserving machine (deep) learning with distributed data. In Kotsis et al., editor, *Database and Expert Systems Applications - DEXA 2022 Workshops*, pages 55–66, Cham, 2022. Springer International Publishing.

[2] M. Kumar, W. Zhang, L. Fischer, and B. Freudenthaler. Membership-mappings for practical secure distributed deep learning. *IEEE Trans. Fuzzy Systems*, pages 1–14, 2023.

[3] M. Kumar, M. Rossbory, B. A. Moser, and B. Freudenthaler. Deriving an optimal noise adding mechanism for privacy-preserving machine learning. In Anderst-Kotsis et al., editor, *Proceedings of the 3rd International Workshop on Cyber-Security and Functional Safety in Cyber-Physical (IWCFS 2019), August 26-29, 2019, Linz, Austria*, pages 108–118, Cham, 2019. Springer International Publishing.

[4] M. Kumar, M. Rossbory, B. A. Moser, and B. Freudenthaler. An optimal $(\epsilon, \delta)-$differentially private learning of distributed deep fuzzy models. *Information Sciences*, 546:87–120, 2021.

[5] M. Kumar, Michael Rossbory, B. A. Moser, and B. Freudenthaler. Differentially private learning of distributed deep models. In *Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '20 Adjunct, pages 193–200, New York, NY, USA, 2020. Association for Computing Machinery.

[6] M. Kumar and B. Freudenthaler. Fuzzy membership functional analysis for nonparametric deep models of image features. *IEEE Trans. Fuzzy Systems*, 28(12):3345–3359, 2020.

[7] M. Kumar, W. Zhang, M. Weippert, and B. Freudenthaler. An explainable fuzzy theoretic nonparametric deep model for stress assessment using heartbeat intervals analysis. *IEEE Trans. Fuzzy Systems*, 29(12):3873–3886, 2021.

[8] M. Kumar, S. Singh, and B. Freudenthaler. Gaussian fuzzy theoretic analysis for variational learning of nested compositions. *International Journal of Approximate Reasoning*, 131:1–29, 2021.

[9] W. Zhang, M. Kumar, W. Ding, X. Li, and J. Yu. Variational learning of deep fuzzy theoretic nonparametric model. *Neurocomputing*, 506:128–145, 2022.

[10] M. Kumar, B. Moser, L. Fischer, and B. Freudenthaler. Membership-mappings for data representation learning: Measure theoretic conceptualization. In Kotsis et al., editor, *Database and Expert Systems Applications - DEXA 2021 Workshops*, pages 127–137, Cham, 2021. Springer International Publishing.

[11] M. Kumar, B. Moser, L. Fischer, and B. Freudenthaler. Membership-mappings for data representation learning: A bregman divergence based conditionally deep autoencoder. In Kotsis et al., editor, *Database and Expert Systems Applications - DEXA 2021 Workshops*, pages 138–147, Cham, 2021. Springer International Publishing.

[12] M. Kumar. Differentially private transferrable deep learning with membership-mappings. *Advances in Computational Intelligence*, 3(1):1–27, 2023.

[13] M. Kumar, B. A. Moser, and L. Fischer. On mitigating the utility-loss in differentially private learning: A new perspective by a geometrically inspired kernel approach, 2023. `https://arxiv.org/abs/2304.01300` [Accessed: (02.07.2023)].

[14] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In Jung Hee Cheon and T. Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016*, pages 3–33, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[15] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. Faster packed homomorphic operations and efficient circuit bootstrapping for tfhe. In T. Takagi and T. Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 377–408, Cham, 2017. Springer International Publishing.

[16] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. TFHE: Fast fully homomorphic encryption library, August 2016. https://tfhe.github.io/tfhe/.

[17] I. Chillotti, M. Joye, and P. Paillier. Programmable bootstrapping enables efficient homomorphic inference of deep neural networks. In Dolev et al., editor, *Cyber Security Cryptography and Machine Learning - 5th International Symposium, CSCML 2021, Be'er Sheva, Israel, July 8-9, 2021, Proceedings*, volume 12716 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2021.