Membership Mappings for Practical Secure Distributed Deep Learning

Mohit Kumar, Weiping Zhang, Lukas Fischer, and Bernhard Freudenthaler

Abstract—In this article, we consider the problem of privacy-preserving distributed deep learning where data privacy is protected by fully homomorphic encryption. The aim is to develop a method for practical and scalable distributed deep learning with fully homomorphic encrypted data. The method must address the issue arising from the large computational cost associated with fully homomorphic encrypted data to offer a practical and scalable solution. An approach that leverages fuzzy-based membership mappings for data representation learning is considered for distributed deep learning with fully homomorphic encrypted data. The method introduces globally convergent and robust variational membership mappings to build local deep models. The local models are combined in a robust and flexible manner by means of fuzzy attributes to build a global model such that the global model can be homomorphically evaluated in an efficient manner. The membership-mappings-based privacy-preserving distributed deep learning method is accurate, practical, and scalable. This is verified through numerous experiments that include demonstrations using MNIST and Freiburg Groceries datasets, and a biomedical application related to the detection of mental stress on individuals. This study develops globally convergent and robust variational membership mappings for their application to accurate, practical, and scalable privacy-preserving distributed deep learning.

Index Terms—Distributed deep learning, fully homomorphic encryption (FHE), fuzzy attributes, membership mappings, privacy.

I. INTRODUCTION

FULLY homomorphic encryption (FHE) is a method to preserve the privacy of data in the cloud computing scenario. The first FHE scheme [1] is based on ideal lattices, and the bootstrapping procedure is introduced to reduce the noise contained in a ciphertext for allowing arbitrary computations. The bootstrapping operation is performed on a ciphertext via evaluating the decryption function homomorphically using the bootstrapping key (which is the encryption of the private decryption key under the public encryption key). Bootstrapping is the computationally most expensive part of a homomorphic encryption scheme. The theoretical breakthrough of [1] was followed by several attempts to develop more practical FHE schemes. The scheme introduced in [2] was improved in [3] with reduced public key size, extended in [4] to support encrypting and homomorphically processing a vector of plaintexts as a single ciphertext, and generalized to nonbinary messages in [5]. Schemes based on the learning with errors (LWE) problem were constructed in [6]. An FHE scheme constructed in [7] is based solely on the standard LWE assumption. Ring learning with errors (RLWE) is a variant of LWE problem where the algebraic structure of the underlying hard problem reduces the key sizes and speeds up the homomorphic operations. A leveled fully homomorphic encryption scheme based on LWE or RLWE, without bootstrapping procedure, was proposed in [8]. The ciphertexts contain a certain amount of noise for security purposes that grows with homomorphic operations. A tensoring technique for LWE-based FHE that reduced ciphertext noise growth after multiplication from quadratic to linear was introduced in [9]. As the scheme of [9] no longer requires the rescaling of the ciphertext, this scheme was called a scale-invariant fully homomorphic encryption scheme. Fan and Vercauteren [10] provided the RLWE version of the scale-invariant scheme of [9]. A technique for building LWE-based FHE scheme was called as approximate eigenvector method in which homomorphic addition and multiplication are just matrix addition and multiplication [11]. The essence of this scheme is that the secret key is an approximate eigenvector of the ciphertext matrix and the message is the corresponding eigenvalue. Several works that followed the theoretical breakthrough of [1] were aimed at improving the bootstrapping as the bootstrapping remained the bottleneck for an efficient FHE in practice. A much faster bootstrapping, based on a scheme similar to the type of [11] that allows to homomorphically compute simple bit operations and refresh (bootstrap) the resulting output in less than a second, was devised in [12]. Finally, the TFHE scheme was proposed in [13] and [14] that features an improved bootstrapping procedure that is considerably more efficient than the previous state of the art. The TFHE scheme generalizes previous structures and schemes over the torus and improves the bootstrapping dramatically.
For practical applications, TFHE is an open-source C/C++ library [15] implementing the ring variant of [11] together with the optimizations of [12], [13], and [14].

Remark 1 (Research Gap and Central Research Problem): Despite recent advances in FHE schemes, machine (deep) learning with fully homomorphic encrypted data remains impractical due to the large computational overhead. Therefore, development of a method for practical and scalable distributed deep learning with fully homomorphic encrypted data is the central research problem to be addressed in this study.

Recent years have witnessed a surge of interest in application of deep neural networks for representation learning. However, there are three inherent issues of deep neural networks: requirement of large training data, determining the optimal model structure, and iterative time-consuming nature of numerical learning algorithms. To address these issues, an alternative approach based on nonparametric fuzzy theoretic deep models [16], [17], [18], [19] has been suggested. The fuzzy systems’ capability of handling uncertainties in a rigorous mathematical manner has motivated the combining fuzzy theory with deep models [20], [21], [22], [23], [24], [25], [26], [27]. The concept of fuzzy mapping [16], [17], [18], [19] has been introduced for representing a mapping through a fuzzy set such that a deep model formed via a composition of fuzzy mappings can be learned analytically via a variational optimization technique. The idea of fuzzy mapping was conceptualized using the measure theory and was referred to as membership mapping in (28). The membership-mappings-based deep models have been successfully applied for data representation learning [29], [30].

Fuzzy set and fuzzy rules can potentially provide a robust and flexible mechanism to combine local models in a distributed learning scenario. The idea of using fuzzy sets and fuzzy rules to aggregate the local privacy-preserving deep models for building the global model was previously considered in [31] and [32], however, under differential privacy framework. Differential privacy preserves the privacy of the training dataset via adding random noise to ensure that an adversary cannot infer any single data instance by observing model parameters or model outputs. The amount of noise depends upon the value of privacy-loss bound. A major limitation of the differential privacy is that a sufficiently low value of privacy-loss bound results in a loss of accuracy. Moreover, it is not clear how to practically choose the value of privacy-loss bound. The FHE approach on the other hand does not lead to the loss of accuracy, however, requires a large computational time. A recent study [33] is the state-of-art on privacy-preserving inference of deep neural networks using the TFHE fully homomorphic encryption scheme. However, the framework of [33] does not address the practicality and scalability issues in a distributed learning scenario.

Remark 2 (Research Problems): Aiming at practical secure distributed deep learning using a fully homomorphic encryption, this study addresses the following research problems.

1) Problem 1: Develop a variational learning algorithm with the mathematical proof of global convergence and robustness to leverage the data representation capability of membership-mappings-based deep models.

2) Problem 2: Apply fuzzy sets and rules to build a global model via combining distributed local models such that homomorphic evaluation of the global model remains practical and scalable.

This study introduces a methodology that:
1) leverages data representation learning capability of globally convergent and robust membership-mappings to build local deep models;
2) uses local deep models to induce fuzzy attributes such that defined fuzzy attributes learn data representation;
3) combines local deep models in a robust and flexible manner by means of fuzzy attributes and fuzzy rules to define a global model;
4) defines the global model in such a way that the global model can be evaluated homomorphically in an efficient manner using a boolean circuit composed of bootstrapped binary gates;
5) implements very fast gate-by-gate bootstrapping to homomorphically evaluate the global model (that combines the distributed local models) to predict the output.

Remark 3 (Contributions and Novelities): The overall contribution of this study is to address the central research problem (as stated in Remark 1) after addressing Problems 1 and 2 (as stated in Remark 2). This study is novel in the following aspects.
1) Transformation to fuzzy attributes for efficient homomorphic inference: A novelty of this work lies in the idea that instead of performing machine (deep) learning on fully homomorphic encrypted high-dimensional data points (that obviously is computationally challenging), the high-dimensional data points are transformed by means of data representation models to fuzzy attribute values, which are then encrypted for an efficient fully homomorphic evaluation of the model. This is illustrated as in Fig. 1.
2) Global convergence and robustness of fuzzy deep learning algorithm: The study suggests an algorithm for the learning of variational membership-mappings-based deep models with the mathematical proofs of global convergence and robustness. The mathematical analysis regarding convergence and robustness of variational membership mappings is novel.

Remark 4 (Contributions w.r.t. Previous Works): Table I lists the differences of this work from previous related works. This work extends the membership-mappings models [16], [17], [18], [19], [28], [29], [30] via providing a variational learning algorithm with mathematical proofs for global convergence and robustness that the previous learning algorithms were lacking. This study further extends the fuzzy-based approach of aggregating distributed models (as suggested in [31] and [32]) to FHE
setting for protecting data privacy without an accuracy loss. Therefore, this study offers, unlike all previous related works, simultaneously the privacy preservation, accuracy, computational efficiency, global convergence, robustness, mathematical theory of deep learning, and sufficient experimentation.

This article is organized into sections. Section II reviews the membership mappings from previous works. An estimation algorithm for the variational learning of membership mappings together with convergence and robustness analysis is provided in Section III. Section IV considers the application of membership mappings to secure distributed deep learning. The experiments are provided in Section V, and finally, Section VI concludes this article.

II. REVIEW OF MEMBERSHIP MAPPINGS

This section presents the mathematical background of membership mappings from our previous works [28], [29], [30].

A. Notations

Let \( n, N, p, M \in \mathbb{N} \). Let \( \mathcal{B}(\mathbb{R}^N) \) denote the Borel \( \sigma \)-algebra on \( \mathbb{R}^N \), and let \( \mathcal{L}^N \) denote the Lebesgue measure on \( \mathcal{B}(\mathbb{R}^N) \). Let \((\mathcal{X}, \mathcal{A}, \rho)\) be a probability space with unknown probability measure \( \rho \). Let us denote by \( \mathcal{S} \) the set of finite samples of data points drawn i.i.d. from \( \rho \), i.e.,

\[
\mathcal{S} := \{(x^i \sim \rho)_{i=1}^N \mid N \in \mathbb{N}\}.
\]

For a sequence \( x = (x^1, \ldots, x^N) \in \mathcal{S} \), let \(|x|\) denote the cardinality, i.e., \(|x| = N\). If \( x = (x^1, \ldots, x^N) \), \( a = (a^1, \ldots, a^M) \in \mathcal{S} \), then \( x \wedge a \) denotes the concatenation of the sequences \( x \) and \( a \), i.e., \( x \wedge a = (x^1, \ldots, x^N, a^1, \ldots, a^M) \). Let us denote by \( \mathcal{F}(\mathcal{X}) \) the set of \( \mathcal{A} \)-\( \mathcal{B}(\mathbb{R}) \) measurable functions \( f : \mathcal{X} \to \mathbb{R} \), i.e.,

\[
\mathcal{F}(\mathcal{X}) := \{ f : \mathcal{X} \to \mathbb{R} \mid f \text{ is } \mathcal{A} \text{-}\mathcal{B}(\mathbb{R}) \text{ measurable} \}.
\]

For a sequence \( x \in \mathcal{S} \), assume that a membership function \( \zeta_x : \mathbb{R}^{|x|} \to [0, 1] \) satisfies the following properties.

1) \( \zeta_x(y) > 0 \) for all \( y \in \mathbb{R}^{|x|} \), i.e.,

\[
\text{supp}[\zeta_x] = \mathbb{R}^{|x|}.
\]

2) The functions \( \zeta_x \) are absolutely continuous and Lebesgue integrable over the whole domain such that for all \( x \in \mathcal{S} \) we have

\[
0 < \int_{\mathbb{R}^{|x|}} \zeta_x \, d\lambda^{|x|} < \infty.
\]

3) The membership function induced probability measures \( \mathbb{P}_{\zeta_x} \), defined on any \( A \in \mathcal{B}(\mathbb{R}^{|x|}) \), as

\[
\mathbb{P}_{\zeta_x}(A) := \frac{1}{\int_{\mathbb{R}^{|x|}} \zeta_x \, d\lambda^{|x|}} \int_A \zeta_x \, d\lambda^{|x|}
\]

are consistent in the sense that for all \( x, a \in \mathcal{S} \)

\[
\mathbb{P}_{\zeta_x}(A \times \mathbb{R}^{|x|}) = \mathbb{P}_{\zeta_a}(A).
\]

Let \( \Theta \) be a set defined as

\[
\Theta := \{ \zeta_x : \mathbb{R}^{|x|} \to [0, 1] \mid (3), (4), (6), x \in \mathcal{S} \}.
\]

B. Membership-Mappings-Based Models

Definition 1 (Student-t Membership Mapping [28]): A Student-t membership-mapping, \( \mathcal{F} \in \mathcal{F}(\mathcal{X}) \), is a mapping with input space \( \mathcal{X} = \mathbb{R}^n \) and a membership function \( \zeta_x \in \Theta \) that is Student-t like

\[
\zeta_x(y) = \left(1 + \frac{1}{\nu - 2} (y - m_y)^T K_{xx}^{-1} (y - m_y) \right)^{-\frac{\nu + |x|}{2}}.
\]

where \( x \in \mathcal{S}, y \in \mathbb{R}^{|x|}, \nu \in \mathbb{R}_+ \setminus [0, 2] \) is the degrees of freedom, \( m_y \in \mathbb{R}^{|x|} \) is the mean vector, and \( K_{xx} \in \mathbb{R}^{|x| \times |x|} \) is the covariance matrix with its \((i,j)\)th element given as

\[
(K_{xx})_{i,j} = kr(x^i, x^j).
\]

where \( kr : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \) is a positive definite kernel function defined as

\[
kr(x^i, x^j) = \sigma^2 \exp \left(-0.5 \sum_{k=1}^n w_k |x^i_k - x^j_k|^2 \right)
\]

where \( x^i_k \) is the \( k \)th element of \( x^i \), \( \sigma^2 \) is the variance parameter, and \( w_k \geq 0 \) (for \( k \in \{1, \ldots, n\} \)).

Definition 2 (Membership-Mapping Autoencoder [29]): A membership-mapping autoencoder, \( \mathcal{G} : \mathbb{R}^p \to \mathbb{R}^p \), maps an input vector \( y \in \mathbb{R}^p \) to \( \mathcal{G}(y) \in \mathbb{R}^p \) such that

\[
\mathcal{G}(y) := \left[ \mathcal{F}_1(Py) \cdots \mathcal{F}_p(Py) \right]^T
\]

where \( \mathcal{F}_j \) (\( j \in \{1, 2, \ldots, p\} \)) is a Student-t membership mapping, \( P \in \mathbb{R}^{n \times p} (n \leq p) \) is a matrix such that the product \( Py \) is a lower dimensional encoding for \( y \).

Definition 3 (Conditionally Deep Membership-Mapping Autoencoder (CDMMA) [29]): A conditionally deep membership-mapping autoencoder, \( \mathcal{D} : \mathbb{R}^p \to \mathbb{R}^p \), maps a vector \( y \in \mathbb{R}^p \) to \( \mathcal{D}(y) \in \mathbb{R}^p \) through a nested composition of finite number of

| Studies | Privacy-preserving | FHE accuracy | Convergence analysis | Robustness analysis | Computational efficiency | Mathematical theory | Sufficient experiments |
|---------|-------------------|--------------|---------------------|--------------------|------------------------|--------------------|--------------------|
| [16], [19] | No | No | Yes | No | Yes | Yes | Yes |
| [28], [29] | No | No | Yes | No | Yes | Yes | Yes |
| [20]–[24], [24]–[27], [33] | No | No | Yes | No | No | Yes | No |
| [30], [32] | Yes | No | No | No | Yes | Yes | Yes |
| [34] | Yes | Yes | Yes | No | No | Yes | No |
| This work | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
membership-mapping autoencoders such that
\[
y^l = (G_{l} \circ \cdots \circ G_2 \circ G_1)(y) \quad \forall l \in \{1, 2, \ldots, L\} \quad (12)
\]
\[
l^* = \arg \min_{l \in \{1, 2, \ldots, L\}} \|y - y^l\|^2 \quad (13)
\]
\[
D(y) = y^n \quad (14)
\]
where \(G_l\) is membership-mapping autoencoder (Definition 2).

**Definition 4 (A Wide CDMMA [29]):** A wide CDMMA, \(WD : \mathbb{R}^p \to \mathbb{R}^p\), maps a vector \(y \in \mathbb{R}^p\) to \(WD(y) \in \mathbb{R}^p\) through a parallel composition of \(S (S \in \mathbb{Z}_+)\) number of CDMMAs such that
\[
WD(y) = D_s(y) \quad (15)
\]
\[
s^* = \arg \min_{s \in \{1, 2, \ldots, S\}} \|y - D_s(y)\|^2 \quad (16)
\]
where \(D_s(y)\) is the output of the \(s\)th CDMMA.

**C. Variational Membership Mappings**

Given a dataset \(\{(x^i, y^i) \mid x^i \in \mathbb{R}^n, y^i \in \mathbb{R}^p, i \in \{1, \ldots, N\}\}\), a modeling scenario is created assuming that there exist zero-mean Student-t membership mappings \(F_1, \ldots, F_p \in \mathbb{R}^p \) such that
\[
y^i = \begin{bmatrix} F_1(x^i) & \cdots & F_p(x^i) \end{bmatrix}^T \quad (17)
\]
For \(j \in \{1, 2, \ldots, p\}\), define
\[
y_j = \begin{bmatrix} y_j^1 & \cdots & y_j^N \end{bmatrix}^T \in \mathbb{R}^N \quad (18)
\]
where \(y_j^i\) denotes the \(i\)th element of \(y_j^i\).

**Definition 5 (Variational Membership-Mapping Model [30]):** Given a dataset \(\{(x^i, y^i) \mid x^i \in \mathbb{R}^n, y^i \in \mathbb{R}^p, i \in \{1, \ldots, N\}\}\) and a set of auxiliary inducing points \(a = \{a^m \in \mathbb{R}^n \mid m \in \{1, \ldots, M\}\}\), a variational membership-mapping model maps an input \(x \in \mathbb{R}^n\) to an output \(\begin{bmatrix} \tilde{F}_1(x) & \cdots & \tilde{F}_p(x) \end{bmatrix}^T \in \mathbb{R}^p\) such that for all \(j \in \{1, 2, \ldots, p\}\)
\[
\tilde{F}_j(x) = G(x) K_{xa} K_{x \tau} + \frac{\text{tr}(K_{xa} - K_{xa} K_{x \tau} K_{x \tau} K_{xa})}{\nu + M - 2} K_{xa}
\]
\[
+ \frac{1}{\beta} K_{xa} K_{xa}^{-1} K_{xa}^T \nu_j \quad (19)
\]
where
1. \(G(x) \in \mathbb{R}^{1 \times M}\) is a vector-valued function defined as
\[
G(x) := [kr(x, a^1) \cdots kr(x, a^M)] \quad (20)
\]
where \(kr : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}\) is defined as in (10);
2. \(K_{xa} \in \mathbb{R}^{N \times N}, K_{xa} \in \mathbb{R}^{M \times M}\), and \(K_{xa} \in \mathbb{R}^{N \times M}\) are matrices with their \((i, j)\) elements given as
\[
(K_{xa})_{i,j} = kr(x^i, a^j) \quad (21)
\]
\[
(K_{xa})_{i,j} = kr(a^i, a^j) \quad (22)
\]
\[
(K_{xa})_{i,j} = kr(x^i, a^j) \quad (23)
\]
3. \(\nu \in \mathbb{R}_+ \setminus [0, 2]\) is the degrees of freedom and \(\beta > 0\) is the precision value.

**D. Estimation of Membership-Mapping Model Parameters**

Define a vector \(\alpha_j \in \mathbb{R}^M\) as
\[
\alpha_j(\beta^{-1}) := \left( K_{xa}^T K_{xa} + \tau K_{aa} + \beta^{-1} K_{aa} \right)^{-1} (K_{xa})^T y_j \quad (24)
\]
where
\[
\tau := \frac{\text{tr}(K_{xa} - K_{xa} K_{xa}^T K_{xa})}{\nu + M - 2} \quad (25)
\]
Now, the output of the variational membership-mapping model could be expressed as
\[
\tilde{F}_j(x) = (G(x)) \alpha_j(\beta^{-1}), \quad j \in \{1, 2, \ldots, p\}. \quad (26)
\]
It follows from (26) that estimation of the variational membership-mapping model output requires computing \(\alpha_j\) via (24), which in-turn requires estimating the inverse precision value \(\beta^{-1}\). The inverse precision value \(\beta^{-1}\) is iteratively estimated as the inverse of the mean squared error between data and membership-mappings outputs. That is,
\[
\beta^{-1} = \frac{1}{pN} \sum_{j=1}^{p} \sum_{i=1}^{N} \left| y_j^i - \tilde{F}_j(x^i) \right|^2 \quad (27)
\]
where \(\tilde{F}_j(x^i)\) is given as in (26). Expression (27) using (26) can be expressed as
\[
\beta^{-1} = \frac{1}{pN} \sum_{j=1}^{p} \sum_{i=1}^{N} \left| y_j^i - (G(x^i) \alpha_j(\beta^{-1})) \right|^2. \quad (28)
\]
As \(G(x^i)\) is equal to the \(i\)th row of the matrix \(K_{xa}\), (28) can be expressed as
\[
\beta^{-1} = \frac{1}{pN} \sum_{j=1}^{p} \left| y_j^i - K_{xa} \alpha_j(\beta^{-1}) \right|^2. \quad (29)
\]

**III. GLOBALLY CONVERGENT LEARNING ALGORITHM AND ROBUSTNESS ANALYSIS FOR VARIATIONAL MEMBERSHIP MAPPINGS**

**A. Convergence Analysis**

In this subsection, we study the convergence of the estimation algorithm (24), (29). In particular, we derive a sufficient condition for the estimation algorithm (24), (29) to converge. For this, consider the singular value decomposition of \(K_{xa}\) as
\[
K_{xa} = U \begin{bmatrix} S & \emptyset \end{bmatrix}_V^T \quad (30)
\]
where \(U \in \mathbb{R}^{N \times N}\) and \(V \in \mathbb{R}^{M \times M}\) are orthogonal, and \(S = \text{diag}(s_1, \ldots, s_M)\) is a diagonal matrix with \(s_1 \geq s_2 \geq \cdots \geq s_M\) \(\geq 0\) being the singular values of \(K_{xa}\). The vectors \(b_1^j \in \mathbb{R}^M\) and \(b_2^j \in \mathbb{R}^{N-M}\) are defined as
\[
\begin{bmatrix} b_1^j \\ b_2^j \end{bmatrix} = U^T y_j. \quad (31)
\]
The expression (24) for \( \alpha_j \) can be rewritten as
\[
\alpha_j(\beta^{-1}) = (VS^2V^T + \tau K_{aa} + \beta^{-1} K_{aa})^{-1} V S b_j. \tag{32}
\]
Consider
\[
y_j - K_{xx} \alpha_j(\beta^{-1}) = U U^T y_j - U \left[ S 0 \right] V^T (V S^2 V^T + (\tau + \beta^{-1}) K_{aa})^{-1} V S b_j. \tag{33}
\]
Using the matrix inversion lemma
\[
y_j - K_{xx} \alpha_j(\beta^{-1}) = U \left[ I + \frac{1}{\tau + \beta^{-1}} S V^T K_{aa}^{-1} V S \right]^{-1} b_j \tag{34}
\]
and hence,
\[
\|y_j - K_{xx} \alpha_j(\beta^{-1})\|^2 = \|b_j\|^2 + (\tau + \beta^{-1})^2 (b_j^T) ((\tau + \beta^{-1}) I + SV^T K_{aa}^{-1} VS)^{-2} b_j. \tag{35}
\]
Using (35) in (29)
\[
\beta^{-1} = \frac{1}{p N} \sum_{j=1}^{p} \|b_j\|^2 + \frac{(\tau + \beta^{-1})^2}{p N} \sum_{j=1}^{p} (b_j^T) \left((\tau + \beta^{-1}) I + SV^T K_{aa}^{-1} VS\right)^{-2} b_j. \tag{36}
\]
**Result 1 (A Function in \( \beta^{-1} \)):** Let \( \mathcal{R} \) be a function in \( \beta^{-1} \) defined as
\[
\mathcal{R}(\beta^{-1}) := \frac{1}{p N} \sum_{j=1}^{p} \|b_j\|^2 + \frac{(\tau + \beta^{-1})^2}{p N} \sum_{j=1}^{p} (b_j^T) \left((\tau + \beta^{-1}) I + SV^T K_{aa}^{-1} VS\right)^{-2} b_j. \tag{37}
\]
We have followings.

1) \( \mathcal{R}(\beta^{-1}) \in (\beta^{-1}_{\text{low}}, \beta^{-1}_{\text{up}}) \) \( \forall \beta^{-1} \in \mathbb{R}_{>0} \), where
\[
\beta^{-1}_{\text{low}} = \frac{1}{p N} \sum_{j=1}^{p} \|b_j\|^2 \tag{38}
\]
\[
\beta^{-1}_{\text{up}} = \frac{1}{p N} \sum_{j=1}^{p} \|y_j\|^2. \tag{39}
\]

2) The lower and upper bounds on the derivative of \( \mathcal{R} \) w.r.t. \( \beta^{-1} \) are given as
\[
0 < \frac{d \mathcal{R}(\beta^{-1})}{d \beta^{-1}} < \frac{2}{p N} \tau + \beta^{-1} \sum_{j=1}^{p} \|b_j\|^2. \tag{40}
\]

3) \( \mathcal{R}(\beta^{-1}) \) has at least one fixed point in \( (\beta^{-1}_{\text{low}}, \beta^{-1}_{\text{up}}) \).

**Proof:** The proof is provided in Appendix A.

**Result 2 (Convergence):** If \( \tau \) is chosen such that
\[
\tau > \frac{2}{p N} \sum_{j=1}^{p} \|y_j\|^2 - \frac{1}{p N} \sum_{j=1}^{p} \|b_j\|^2 \tag{41}
\]
then the iterations
\[
\beta^{-1}_{it+1} = \mathcal{R}(\beta^{-1}_{it}), \quad \text{it} \in \{0, 1, 2, \cdots \} \tag{42}
\]
with \( \beta^{-1}_{0} \in (\beta^{-1}_{\text{low}}, \beta^{-1}_{\text{up}}) \), converge to the unique fixed point of \( \mathcal{R}(\beta^{-1}) \).

**Proof:** The proof is provided in Appendix B.

### B. Learning Algorithm

Result 2 allows to design a globally convergent algorithm for the variational learning of membership mappings via ensuring the sufficient condition (41). For this, it is observed that \( \tau \), defined as in (25), is a function of parameters set \( \{M, \sigma^2, \{w_i\}_{i=1}^{n}, x, a, \nu\} \). It follows from the kernel function definition (10) that
\[
\tau(M, \sigma^2, \{w_i\}_{i=1}^{n}, x, a, \nu) = \sigma^2 \tau(M, 1, \{w_i\}_{i=1}^{n}, x, a, \nu). \tag{43}
\]
Using (43), the condition (41) can be rewritten as
\[
\sigma^2 > \frac{1}{\tau(M, 1, \{w_i\}_{i=1}^{n}, x, a, \nu)} \sum_{j=1}^{p} \|y_j\|^2 - \frac{1}{p N} \sum_{j=1}^{p} \|b_j\|^2. \tag{44}
\]
To hold the inequality (44), the value of \( \sigma^2 \) can be adjusted as in steps 10–14 of Algorithm 1. Furthermore, some of the parameters must be chosen priorly, which are suggested to be chosen as in the following.

1) **Auxiliary inducing points:** The auxiliary inducing points are suggested to be chosen as the cluster centroids:
\[
a = \left\{ a^m \right\}_{m=1}^{M} = \text{cluster_centroid} \left( \{x^i\}_{i=1}^{N}, M \right) \tag{45}
\]
where \( \text{cluster_centroid} \left( \{x^i\}_{i=1}^{N}, M \right) \) represents the k-means clustering on \( \{x^i\}_{i=1}^{N} \).

2) **Degrees of freedom:** \( \nu \in \mathbb{R}_+ \setminus [0, 2] \) is chosen as
\[
\nu = 2.1. \tag{46}
\]

3) **Parameters \( \{w_1, \ldots, w_n\} \):** The parameters \( \{w_1, \ldots, w_n\} \) for the kernel function (10) are chosen such that \( w_k \) (for \( k \in \{1, 2, \ldots, n\} \)) is given as
\[
w_k = \left( \max_{1 \leq i \leq N} x^i_k - \min_{1 \leq i \leq N} x^i_k \right)^{-2} \tag{47}
\]
where \( x^i_k \) is the \( k \)th element of vector \( x^i \in \mathbb{R}^n \).

Finally, Algorithm 1 presents a systematic procedure for the variational learning of membership mappings while ensuring the sufficient condition (41) for the convergence.

**Definition 6 (Membership-Mappings Prediction):** Given the parameters set \( \mathbb{M} = \{\alpha, \alpha, M, \sigma, w, \beta\} \) returned by Algorithm 1, the learned membership mappings could be used to predict output corresponding to any arbitrary input data point.
Algorithm 1: A Globally Convergent Algorithm for the Variational Learning of Membership-Mappings.

Require: Dataset \( \{(x_i, y_i) \mid i \in \{1, \ldots, N\}\} \) and maximum possible number of auxiliary points \( M_{\text{max}} \in \mathbb{Z}_+ \) with \( M_{\text{max}} \leq N \).

1: Choose \( \nu \) and \( w = (w_1, \ldots, w_n) \) as in (46) and (47), respectively.
2: Set iteration count \( it = 0, M|_0 = M_{\text{max}} \), and determine \( a_0 = \{a_{0m}|_{m=1}^{M_{\text{max}}} \} \) using (45).
3: while \( \tau(M|_it, 1, \{w_i\}_{i=1}^n, x, a, \nu) \leq 0 \) do
4: \( M|_{it+1} = M|_it - 1 \)
5: Determine \( a|_{it+1} = \{a_{m}|_{it+1}^{M_{\text{max}}} \} \) using (45).
6: \( it \leftarrow it + 1 \)
7: end while
8: Set \( M = M|_{it} \) and compute \( a = \{a_{m}\}_{m=1}^{M_{\text{max}}} \) using (45).

Compute \( K_{\text{xx}} \) using (23) taking \( \sigma^2 = 1 \) and perform singular value decomposition of \( K_{\text{xx}} \) to compute orthonormal matrix \( U \) such that (30) holds. Further compute \( b_{1} \) and \( b_{2} \) using (31) for all \( j \in \{1, \ldots, p\} \).

10: if \( \tau(M, 1, \{w_i\}_{i=1}^n, x, a, \nu) > \frac{\sum_{i=1}^{p}(2\|y_i\|^2 - 2\|y_i\|^2)^2}{pn} \) then
11: \( \sigma^2 = 1 \)
12: else
13: \( \sigma^2 = \frac{1.1}{\tau(M, 1, \{w_i\}_{i=1}^n, x, a, \nu)} \cdot \frac{\sum_{i=1}^{p}(2\|y_i\|^2 - 2\|y_i\|^2)^2}{pn} \).
14: end if
15: Compute \( a = \{a_{m}\}_{m=1}^{M_{\text{max}}} \) using (45), \( K_{\text{xx}} \) using (9), \( K_{\text{aa}} \) using (22), and \( K_{\text{xa}} \) using (23).
16: Compute \( \tau \) using (25).

17: Set iteration count \( it = 0 \) and \( \beta^{-1}|_0 = 0.5(\beta^{-1}|_{\text{low}} + \beta^{-1}|_{\text{up}}) \), where \( \beta^{-1}|_{\text{low}} \) and \( \beta^{-1}|_{\text{up}} \) are given by (38) and (39), respectively.
18: Determine the unique fixed point of \( R(\beta^{-1}) \), say \( \beta^{-1} \), using iterations (42).
19: Compute matrix \( \alpha = [\alpha_1(\beta^{-1}) \cdots \alpha_p(\beta^{-1})] \in \mathbb{R}^{M \times p} \), where \( \alpha_j(\beta^{-1}), j \in \{1, \ldots, p\} \), is computed using (24).
20: return the parameters set \( \mathcal{M} = \{\alpha, a, M, \sigma, w, \beta\} \).

\[ x \in \mathbb{R}^n \text{ as} \]
\[ \hat{y}(x; \mathcal{M}) = \begin{bmatrix} \mathcal{F}_1(x) & \cdots & \mathcal{F}_p(x) \end{bmatrix}^T \] (48)

where \( \mathcal{F}_j(x) \) is defined as in (26). It follows from (26) that
\[ \hat{y}(x; \mathcal{M}) = (\alpha(\beta))^T G(x)^T \] (49)

where \( G(\cdot) \in \mathbb{R}^{1 \times M} \) is a vector-valued function (20).

As a CDMMMA consists of membership-mapping compositions, Algorithm 1 can be directly applied for their learning as in Algorithms 2 and 3. For practical applications, Algorithm 3 is suggested for the learning of wide CDMMMA where a computational optimization of a free parameter is performed via minimizing the estimated variance of the mean squared error between data and membership-mappings outputs.

Algorithm 2: A Globally Convergent Algorithm for the Variational Learning of CDMMMA.

Require: Dataset \( \mathbf{Y} = \{y_i \in \mathbb{R}^p \mid i \in \{1, \ldots, N\}\} \); the subspace dimension \( n \in \{1, 2, \ldots, p\} \); maximum number of auxiliary points \( M_{\text{max}} \in \mathbb{Z}_+ \) with \( M_{\text{max}} \leq N \); the number of layers \( L \in \mathbb{Z}_+ \).

1: for \( l = 1 \) to \( L \) do
2: Set subspace dimension associated to \( l \)th layer as \( n_l = \max(n - l + 1, 1) \).
3: Define \( P_l \in \mathbb{R}^{n_l \times \pi} \) such that \( \pi \)th row of \( P_l \) is equal to transpose of eigenvector corresponding to \( \pi \)th largest eigenvalue of sample covariance matrix of dataset \( \mathbf{Y} \).
4: Define a latent variable \( x^{l,i} \in \mathbb{R}^{n_l} \), for \( i \in \{1, \ldots, N\} \), as
\[ x^{l,i} := \begin{cases} P_l y^{i} & \text{if } l = 1 \\ P_l g^{l-1}(x^{l-1,i}, M^{l-1}) & \text{if } l > 1 \end{cases} \] (50)

where \( g^{l-1} \) is the estimated output of the \((l - 1)\)th layer computed using (49) for the parameters set \( M^{l-1} = \{\alpha_1, \ldots, \alpha_l, M^{l-1}, \sigma_l, w_l\} \).
5: Define \( M^{l}_{\text{max}} \) as
\[ M^{l}_{\text{max}} := \begin{cases} M_{\text{max}} & \text{if } l = 1 \\ M^{l-1} & \text{if } l > 1 \end{cases} \] (51)

6: Compute parameters set \( \mathcal{M} = \{\alpha, a, M^{l}, \sigma, w, \beta^{l}\} \), characterizing the membership mappings associated to \( l \)th layer, using Algorithm 1 on dataset \( \{\{x^{l,i}, y_i\} \mid i \in \{1, \ldots, N\}\} \) with maximum possible number of auxiliary points \( M^{l}_{\text{max}} \).
7: end for
8: return the parameters set \( \mathcal{M} = \{\{M^1, \ldots, M^L\}, \{P^1, \ldots, P^L\}\} \).

Definition 7 (CDMMMA Filtering): Given a CDMMMA with its parameters being represented by a set \( \mathcal{M} = \{\{M^1, \ldots, M^L\}, \{P^1, \ldots, P^L\}\} \), the autoencoder can be applied for filtering a given input vector \( y \in \mathbb{R}^P \) as follows:
\[ x^{l}(y; \mathcal{M}) = \begin{cases} P_l y & \text{if } l = 1 \\ P_l g^{l-1}(x^{l-1}, M^{l-1}) & \text{if } l \geq 2 \end{cases} \] (52)

Here, \( g^{l-1} \) is the output of the \((l - 1)\)th layer estimated using (49). Finally, CDMMMA’s output, \( \hat{y}(y; \mathcal{M}) \), is given as
\[ \hat{y}(y; \mathcal{M}) = g^{\ell^*}(x^{\ell^*}; M^{\ell^*}) \] (53)

\[ \ell^* = \arg \min_{l \in \{1, \ldots, L\}} \|y - g^{l}(x^{l}; M^{l})\|^2. \] (54)

Definition 8 (Wide CDMMMA Filtering): Given a wide CDMMMA with its parameters being represented by a set \( \mathcal{P} = \{\mathcal{M}^s\}_{s=1}^S \), the autoencoder can be applied for filtering a given input vector \( y \in \mathbb{R}^P \) as follows:
\[ \hat{y}(y; \mathcal{P}) = \hat{D}(y; \mathcal{M}^s) \] (55)
Algorithm 3: A Globally Convergent Algorithm for the Variational Learning of Wide CDMMMA.

**Require**: Dataset \( Y = \{y_i \in \mathbb{R}^p \mid i \in \{1, \ldots, N\} \} \); the subspace dimension \( n \in \{1, 2, \ldots, p\} \); an array of possible \( r \) values \( \{r_1, \ldots, r_N \} \) with \( 0 < r_1 < r_2 < \cdots < r_N \leq 1 \); and the number of layers \( L \in \mathbb{Z}_+ \).

1. Partition \( Y \) (using, e.g., k-means clustering) into \( S \) subsets: \( \{Y^1, \ldots, Y^S\} \), where \( S = [N/1000] \).
2. for \( s = 1 \) to \( S \) do
   3. for \( r = r^1_\text{max} \) to \( r = r^N_\text{max} \) do
      4. Apply Algorithm 2 on \( Y^s \) to build a single-layered CDMMMA, \( M^r = \{M^1_r, \{P^1_r\}\} \) where \( M^1_r = \{\alpha^1_r, a^1_r, M^1_r, \sigma^1_r, w^1_r, \beta^1_r\} \), taking \( n \) as the subspace dimension; maximum number of auxiliary points as equal to \( r \times \#Y^s \) (where \( \#Y^s \) is the number of data points in \( Y^s \)); and \( L = 1 \).
   5. end for
3. Set \( r_\text{max} = \arg \max \{r^1_\text{max}, \ldots, r^N_\text{max}\} \beta^1_r \).
5. Build a CDMMMA, \( M^* \), by applying Algorithm 2 on \( Y^s \) taking \( n \) as the subspace dimension; maximum number of auxiliary points as equal to \( r_\text{max} \times \#Y^s \) (where \( \#Y^s \) is the number of data points in \( Y^s \)); and \( L \) as the number of layers.
8. end for
9. return the parameters set \( \mathcal{P} = \{M^*\}_{s=1}^S \).

\[
s^* = \arg \min_{s \in \{1, 2, \ldots, S\}} \|y - \hat{D}(y; M^s)\|^2
\]

where \( \hat{D}(y; M^s) \) is the output of \( s \)th CDMMMA estimated using (53).

**C. Robustness Analysis**

Consider that the data samples \( \{(x^i, y^i) \mid i \in \{1, \ldots, N\}\} \) are subject to deterministic perturbations, and thus, the matrix \( K_{xa}(23) \) and vector \( y_j, (18) \) are subject to perturbations. Let \( \Delta K_{xa} \in \mathbb{R}^{N \times M} \) and \( \Delta y_j \in \mathbb{R}^N \) be the unknown (but bounded) perturbations in \( K_{xa} \) and \( y_j \), respectively, due to the perturbations in the data samples. The data model assumes that there exists some parameters vector \( \alpha_j^* \in M^r \) such that

\[
y_j + \Delta y_j = (K_{xa} + \Delta K_{xa})\alpha_j^*.
\]

The modeling problem is concerned with the estimation of \( \alpha_j^* \) in the presence of unknown perturbations \( \Delta K_{xa} \) and \( \Delta y_j \). To show that Algorithm 1 provides a robust estimation of \( \alpha_j^* \), define

\[
\Delta x := \Delta K_{xa}K_{aa}^{-1/2}K_{aa}^{-1/2}
\]

where \( K_{aa}^{-1/2} \) is the unique square root of positive definite matrix \( K_{aa} \). The set of (57) can be expressed as

\[
y_j + \Delta y_j = (K_{xa} + \Delta K_{xa})K_{aa}^{-1/2}\alpha_j^*.
\]

The perturbation matrix \( \Delta x \Delta y_j \) is unknown, however, is assumed bounded. That is, there exists a scalar \( \delta_m > 0 \) such that

\[
\|\Delta x \Delta y_j\|_F \leq \delta_m,
\]

where \( \|\cdot\|_F \) denotes the Frobenius norm. A robust solution to the estimation of parameters seeks to alleviate the worst-case effect of perturbations. For example, the worst-case residual error can be minimized via solving a min–max estimation problem as in Result 3.

**Result 3 (Robustness)**: Algorithm 1 provides a robust estimation of \( \alpha_j^* \) via solving the following min-max estimation problem:

\[
\hat{\alpha}_j = \arg \min_{\alpha_j} \max_{\|\Delta x \Delta y_j\| \leq \delta_m} \| (K_{xa} + \Delta xK_{aa}^{1/2})\alpha_j^* - (y_j + \Delta y_j) \|
\]

where the upper bound on the norm of perturbation matrix is given as

\[
\delta_m = \sqrt{1 + \|K_{aa}^{1/2}(K_{xa}^T K_{xa} + \tau K_{aa} + \beta^{-1}K_{aa}^{-1}K_{xa}^T y_j)^{-1}K_{xa}^T y_j\|
\]

(61)

where \( \beta^{-1} \) is the unique fixed point of \( R(\beta^{-1}) \) to which the iterations (42) converge.

**Proof**: The proof is provided in Appendix C.

**IV. SECURE DISTRIBUTED DEEP LEARNING**

**A. Autoencoder Induced Fuzzy Attributes for Classification**

The membership-mappings-based autoencoders can be applied for classification. For this, a distance function and a fuzzy attribute are defined in Definitions 9 and 10, respectively.

**Definition 9 (Distance Function Induced by Autoencoder)**: A distance function \( d_\mathcal{P} : \mathbb{R}^p \to \mathbb{R}_+ \), associated to a wide conditionally deep membership-mapping autoencoder \( \mathcal{P} \) (that has been learned using Algorithm 3 on dataset \( Y \subset \mathbb{R}^p \)), can be defined as

\[
d_\mathcal{P}(y) := \|y - \hat{W}D(y; \mathcal{P})\|
\]

(62)

where \( \hat{W}D(y; \mathcal{P}) \) is through autoencoder filtered output (55).

**Definition 10 (A Fuzzy Attribute Induced by Autoencoder)**: A fuzzy attribute \( A_\mathcal{P} \), associated to a wide conditionally deep membership-mapping autoencoder \( \mathcal{P} \) (that has been learned using Algorithm 3 on dataset \( Y \subset \mathbb{R}^p \)), can be defined on a universe of discourse \( \mathbb{R}^p \) as

\[
A_\mathcal{P} := \{(y, \mu_{A_\mathcal{P}}(y)) \mid y \in \mathbb{R}^p\}
\]

(63)

where \( \mu_{A_\mathcal{P}}(y) : \mathbb{R}^p \to [0, 1] \) is a \( p \)-variate membership function that can be defined without the loss of generality of, e.g., Gaussian type

\[
\mu_{A_\mathcal{P}}(y) = \exp\left(-\frac{1}{2p}d_\mathcal{P}(y)^2\right)
\]

(64)

where \( d_\mathcal{P} \) is the distance function defined by (62).
Fig. 2 provides three different examples of distance functions and fuzzy attributes induced by the autoencoders. It is demonstrated through color plots in Fig. 2 that as the distance of a point from data samples increases, the value of distance function \( \Delta \) decreases, and thus, the value of membership function \( \mu \) decreases. That is, the value \( \mu_{\mathcal{A}}(y) \) represents the degree of belongingness of \( y \) to dataset \( Y \) (which is modeled by means of autoencoder \( \mathcal{P} \)). Thus, fuzzy attributes learn a representation of the data samples. This motivates to define a fuzzy classifier based on the following if–then rules:

\[
\text{If } y \in \mathcal{A}_{\mathcal{P}_1}, \text{ then the class is 1} \\
\vdots \\
\text{If } y \in \mathcal{A}_{\mathcal{P}_C}, \text{ then the class is } C. \quad (65)
\]

Here, \( \mathcal{P}_c \) learns the representation of data samples of \( c \)-th class and \( \mu_{\mathcal{A}_{\mathcal{P}_c}}(y) \) represents the degree of belongingness of \( y \) to \( c \)-th class labeled data samples. The class-label associated to a data point \( y \) is predicted based on fuzzy rules (65) as

\[
\mathcal{C}(y; \{ \mathcal{P}_c \}_{c=1}^C) = \arg \max_{1 \leq c \leq C} \mu_{\mathcal{A}_{\mathcal{P}_c}}(y). \quad (66)
\]

The classifier (66), \( \mathcal{C} : \mathbb{R}^p \rightarrow \{1, 2, \ldots, C\} \), assigns to an input vector the label of the class to which the data point has highest degree of matching. Using (64), (66) can be alternatively expressed as

\[
\mathcal{C}(y; \{ \mathcal{P}_c \}_{c=1}^C) = \arg \min_{1 \leq c \leq C} \Delta_{\mathcal{P}_c}(y). \quad (67)
\]

Hence, the classifier assigns to \( y \) the label of the class, the autoencoder of which has minimum distance function value at \( y \), i.e., \( y \) is nearest to the samples of the predicted class. For practical purpose, Algorithm 4 is provided for building the classifier from a given labeled dataset.

**Algorithm 4: A Globally Convergent Algorithm for the Variational Learning of Membership-Mappings-Based Classifier.**

**Require:** Labeled dataset \( Y = \{ Y_c | c \in \{1, \ldots, C\} \} \) where \( Y_c = \{ y^{i,c} \in \mathbb{R}^p | i \in \{1, \ldots, N_c\} \} \).

1: Choose the subspace dimension as \( n = \min(20, p) \), array of possible \( r_{\text{max}} \) values as \( \{0.1, 0.2, \ldots, 0.5\} \), and number of layers as \( L = 5 \).

2: for \( c = 1 \) to \( C \) do

3: Build a wide CDMA, \( \mathcal{P}_c \), by applying Algorithm 3 on \( Y_c \) for the given \( n \), possible \( r_{\text{max}} \) values, and \( L \).

4: end for

5: return the parameters set \( \{ \mathcal{P}_c \}_{c=1}^C \).

**B. Distributed Learning**

We consider a scenario that data are distributed amongst different parties. Assume that there are \( K \) different datasets, \( \{ Y^1, \ldots, Y^K \} \), owned locally by \( K \) different parties. We consider the multiclass classification problem assuming that each local dataset, say \( Y^k \), can be partitioned into \( C \) different classes, i.e.,

\[
Y^k = \{ Y^k_1, \ldots, Y^k_C \} \quad (68)
\]

where \( Y^k_c \) refers to the \( c \)-th class labeled data samples owned locally by the \( k \)-th party. Let \( \mathcal{P}^k_c \) be the wide conditionally deep membership-mapping autoencoder learned from \( Y^k_c \) and \( \mathcal{A}_{\mathcal{P}^k_c} \) be the corresponding fuzzy attribute.
can be extended for distributed setting as follows:

If \( y \) is \( A_{p_1} \) OR \( A_{p_2} \) OR \( \cdots \) OR \( A_{p_K} \), then the class is 1.

\[ \vdots \]

If \( y \) is \( A_{p_1} \) OR \( A_{p_2} \) OR \( \cdots \) OR \( A_{p_K} \), then the class is C.

The class-label associated to a data point \( y \) is predicted based on fuzzy rules (69) as

\[ 
\hat{c} = \arg \max_{1 \leq c \leq C} \left( \max_{1 \leq k \leq K} \mu_{A_{c,k}}(y) \right) 
\]

\[ 
= \arg \min_{1 \leq c \leq C} \left( \min_{1 \leq k \leq K} \mu_{A_{c,k}}(y) \right), \text{ where } K = \text{arg max}_{1 \leq k \leq K} \mu_{A_{c,k}}(y). 
\]

(72)

C. Secure Homomorphic Evaluation of Global Classifier

For a secure distributed learning based on fully homomorphic encryption, define \( cl_k \) as the class-label predicted by the \( k \)-th local classifier, i.e.,

\[ cl_k = C(y; \{ p^h_c \}_{c=1}^C) \]  

where \( C(\cdot) \) is defined as in (66). Now, (71) can be alternatively expressed as

\[ \hat{c} = cl_k^* \]  

(74)

\[ k^* = \arg \min_{1 \leq k \leq K} \bar{\mu}_{A_{c,k}}(y). \]  

(75)

A practical method using the TFHE scheme [13, 14] is provided for secure distributed deep learning. For this, a few variables are defined in the following. Let \( \delta(m_1, m_2) \) be the Kronecker delta function of two variables \( m_1, m_2 \in [0, 1] \) defined as

\[ \delta(m_1, m_2) = \begin{cases} 1, \text{ if } m_1 = m_2 \\ 0, \text{ if } m_1 \neq m_2. \end{cases} \]  

(76)

It follows from (74) and (75) that

\[ \hat{c} = \sum_{k=1}^{K} cl_k \delta \left( \bar{\mu}_{A_{c,k}}(y), \bar{\mu}_{A_{c,k}^*}(y) \right). \]  

(77)

For a given positive integer \( N_b \in \mathbb{Z}_{>0} \), let \( pt_{N_b} : [0, 1] \rightarrow \{0, 1, \ldots, 2^{N_b} - 1\} \) be a function defined as

\[ pt_{N_b}(m) := \lfloor (2^{N_b} - 1)m \rfloor, \text{ m } \in [0, 1]. \]  

(78)

In our setting, \( pt_{N_b}(m) \) is the plaintext that encodes a message \( m \in [0, 1] \) as unsigned \( N_b \)-bit integer. Let \( \text{BitDec}_{N_b} : \{0, 1, \ldots, 2^{N_b} - 1\} \rightarrow \{0, 1\}^{N_b} \) be the binary representation of a \( N_b \)-bit unsigned integer. That is,

\[ (\text{bt}_1(m), \ldots, \text{bt}_{N_b}(m)) = \text{BitDec}_{N_b}(pt_{N_b}(m)). \]  

(79)

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

\[ c_{\text{st}}(\text{bt}) = \text{TFHE.Encryption}(\text{bt}; st). \]  

(80)

Let \( c_{\text{pt}, N_b} : [0, 1] \rightarrow \mathbb{T}^{N_b(N_b+1)} \) be a function defined as

\[ c_{\text{pt}, N_b}(m) := (c_{\text{st}}(\text{bt}_1(m)), \ldots, c_{\text{st}}(\text{bt}_{N_b}(m))) \]  

(81)

where \( c_{\text{st}}(\text{bt}_k(m)) \) is the TFHE encryption of bit \( \text{bt}_k(m) \). Thus, \( c_{\text{pt}, N_b}(m) \) homomorphically encrypts the message \( m \in [0, 1] \) with \( N_b \)-bit precision.

The proposed approach to homomorphically evaluate the global classifier (69) is illustrated in Fig. 3 and Algorithm 5 provides a step-by-step procedure to implement the method.

V. EXPERIMENTS

The proposed method was implemented using MATLAB R2017b and TFHE C/C++ library [15] on a MacBook Pro laptop with a 2.2-GHz Intel Core i7 processor and 16 GB of memory. Parallel Computing Toolbox of MATLAB was used to compute the outputs of local classifiers on a parallel pool of eight workers. The homomorphic evaluation of the global classifier was done on the same MacBook Pro laptop, however, no parallel computing was done for evaluating the global classifier. Algorithm 5 requires the local classifiers and a secret key. Algorithm 4 is

---

**Algorithm 5:** An Algorithm for Implementing Secure Distributed Deep Learning Based on Membership-Mappings and Fully Homomorphic Encryption.

**Require:** Input data vector \( y \), \( K \) different parties participating in collaborative learning each of which owns a local classifier (such that \( k \)-th party’s local classifier, \( \{ p^k_c \}_{c=1}^C \), is built with private dataset \( \mathbf{Y}^k = \{ \mathbf{Y}^k_1, \ldots, \mathbf{Y}^k_K \} \) using Algorithm 4); and a secret key \( st \).

1: Choose bits precision \( N_b \in \mathbb{Z}_{>0} \), e.g., \( N_b = 16 \).
2: The output of each local classifier to the input \( y \) is locally computed, encrypted, and then, exported to the cloud. That is, the \( k \)-th party exports \( \{ c_{\text{pt}, N_b}(cl_k), c_{\text{st}}(\bar{\mu}_{A_{c,k}}(y)) \} \) to the cloud.
3: The global classifier is homomorphically evaluated in the cloud from the encrypted data sent by all parties and the resulting output (which remains encrypted) is returned to the owner of input data. That is, data \( \{ c_{\text{pt}, N_b}(cl_k), c_{\text{st}}(\bar{\mu}_{A_{c,k}}(y), \bar{\mu}_{A_{c,k}^*}(y)) \mid k = 1, \ldots, K \} \) are returned by the cloud to the owner of input data vector \( y \).
4: The user (i.e., owner of the input data) decrypt the encrypted data provided by the cloud. That is, user obtains after decryption \( \{ cl_k, \bar{\delta}(\bar{\mu}_{A_{c,k}}(y), \bar{\mu}_{A_{c,k}^*}(y)) \mid k = 1, \ldots, K \} \).
5: The user determines the class-label \( \hat{c} \) associated to the input \( y \) using (77).
6: **return** \( \hat{c} \).
Fig. 3. Practical method for secure distributed deep learning based on membership mappings and fully homomorphic encryption.

used to build local classifiers and secret key is generated using TFHE library for 128-bits of security. Furthermore, the precision of homomorphic evaluation needs to be chosen at step 1 of Algorithm 5. Experiments are performed with the precision of 16-bits and also 8-bits.

The previous works [16], [17], [18], [19], [29] have already verified the competitive performance of variational membership-mappings-based deep models over a wide range of data dimensions and data types. In this study, the aims of experiments are as follows.

1) Compare the proposed method with the state-of-art differentially private distributed deep learning approach [31], [32] where fuzzy rules were used to aggregate the local deep models as is done in the proposed method.

2) Evaluate the proposed approach by comparing with the state-of-art study [33] on homomorphic inference of deep neural networks.

3) Study the practicality and scalability of the proposed method in-terms of computational time.

A. Comparison With Differential Privacy-Based Approach

For comparing with the state-of-art differential privacy-based approach [31], the local deep models are trained using Algorithm 3 instead with the noisy training data obtained via optimal noise adding mechanism [31], [32] (taking adjacency parameter \(d = 1\), failure probability \(\delta = 1 e^{-5}\), and privacy-loss bound \(\epsilon = 0.1\) and also \(\epsilon = 1\)), and differentially private local deep models are combined through fuzzy rules (69) to build the global classifier.

1) MNIST Dataset: The first experiment is on the widely used MNIST digits dataset containing 28 × 28 sized images divided into training set of 60 000 images and testing set of 10 000 images. The images’ pixel values were divided by 255 to normalize the values in the range from 0 to 1. 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.

2) Freiburg Groceries Dataset: The second experiment is on “Freiburg Groceries Dataset” considered previously [31] for privacy-preserving distributed learning experiments. The dataset contains 4 947 labeled images of grocery products categorized into 25 different classes. A feature vector is created from each image by extracting features from “AlexNet” and “VGG-16” networks, which are pretrained convolutional neural networks (CNNs). The activations of the fully connected layer “fc6” in AlexNet constitute a 4096-D feature vector. Similarly, the activations of the fully connected layer “fc6” in VGG-16 constitute another 4096-D feature vector. The features extracted by both networks are joined together to form a 8192-D vector. The feature vectors are normalized to have zero-mean and unity-variance along each dimension. The set of normalized feature vectors is split into a training set containing around 80% of data points and a testing set containing remaining data points. A three-party scenario is created such that Party-A owns all the training data of first ten grocery categories, Party-B owns all the training data of second ten grocery categories, and Party-C owns all the training data of rest five grocery categories.

3) Real Biomedical Dataset: As an application example, the mental stress detection problem is considered. The dataset from [17], consisting of heart rate interval measurements of different subjects, is considered for the study of individual stress detection problem. The problem is concerned with the detection of stress on an individual based on the analysis of recorded sequence of R-R intervals, \(\{R_i\}_i\). The R-R data vector at ith time
index, \( y^i \), is defined as
\[
 y^i = [R \ R R^{i-1} \cdots R R^{i-d}]^T.
\]
That is, the current interval and history of previous \( d \) intervals constitute the data vector. Assuming an average heartbeat of 72 beats per minute, \( d \) is chosen as equal to \( 72 \times 3 = 216 \) so that R-R data vector consists of on an average 3-min long R-R intervals sequence. Following [17], a dataset, say \( \{y^i\} \), is built via preprocessing the R-R interval sequence \( \{RR^i\} \) with an impulse rejection filter for artifacts detection, and excluding the R-R data vectors containing artifacts from the dataset. The dataset contains the stress score on a scale from 0 to 100. A label of either "no-stress" or "under-stress" is assigned to each \( y^i \) based on the stress score. Thus, we have a binary classification problem. A two-party collaborative learning scenario is considered where a randomly chosen subject is considered as Party-A. While keeping Party-A fixed, the distributed learning experiments are performed independently on every other subject being considered as Party-B. For each subject, 50% of the data samples serve as training data, while remaining as test data. The subjects, with data containing both the classes and at least 60 samples, are considered for experimentation. There are in total 48 such subjects, and thus, the results are averaged over 48 independent experiments.

4) Results and Discussion: The test data accuracy and average 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) are considered as performance indices. Tables II–IV report the experimental results obtained on the MNIST dataset, Freiburg groceries dataset, and real biomedical dataset respectively.

We draw following inferences from the obtained experimental results.

1) As expected, the proposed approach leads to better accuracy (observed in Tables II–IV) than the differential privacy-based approach [31], since differential privacy requires contaminating data with noise. Remarkably, to attain a higher level of differential privacy (i.e., a lower value of privacy-loss bound \( \epsilon \)), the amount of added noise is so high that accuracy is severely affected. The proposed method to preserve privacy on the other hand delivers high accuracy. The proposed method achieved the accuracies of 0.9762, 0.9872, and 0.9880 in comparison to 0.0892, 0.1356, and 0.5123 on the MNIST dataset, Freiburg groceries dataset, and real biomedical dataset, respectively.

2) The proposed membership-mappings-based approach alleviates the large computational overhead issue of fully homomorphic encryption, since the computational time on a MacBook Pro machine with a 2.2-GHz Intel Core i7 processor and 16 GB of memory (as reported in Tables II–IV) is practical.

3) As observed in Table III, the proposed method not only preserves privacy but also performs better than non-private membership-mappings-based conditionally deep autoencoder [29] and the classical machine learning methods including support vector machine (SVM), \( k \)-nearest neighbor (\( k-\)NN), deep neural network, random forest, naive Bayes, ensemble learning, and decision tree.

B. Comparison With Fully Homomorphic Inference of Deep Neural Networks

To evaluate the gains achieved in accuracy and computational time as a result of the proposed fuzzy-based method to secure distributed deep learning (as illustrated in Fig. 3), a comparison is made with the state-of-art study [33] on fully homomorphic inference of deep neural networks. The study in [33] reports the experiments on MNIST dataset for evaluating the neural networks with different depths (referred to as NN-20, NN-50, and NN-100) over TFHE fully homomorphic encrypted data.

1) Results and Discussion: Table V compares the performance of the proposed method with the results of [33]. Following inferences are drawn.

1) The proposed method is more accurate than homomorphically evaluated deep neural networks.
The proposed method is computationally highly efficient. The computational time required by the proposed method is 4.98 s in comparison to the computational time of 115.52 s, which is required by the method of [33].

3) Computational time for the homomorphic evaluation of deep neural networks grows as the size of network grows. As the depth of network increases from 20 to 100 layers, the computational time grows from 115.52 to 481.61 s. On the other hand, the computational time required by the proposed method does not depend on the size of local deep models, since local models are not homomorphically evaluated. The proposed method requires the homomorphic evaluation of the global classifier, which can be efficiently homomorphically evaluated using Algorithm 5.

C. Practicality and Scalability

The computational time required for the homomorphic evaluation of global classifier depends on the number of parties (i.e., \( K \)) participating in collaborative learning. Therefore, experiments are performed to study the computational time as the number of parties is varied from \( K = 2 \) to \( K = 100 \).

1) Results and Discussion: Fig. 4 plots the computational time required for secure homomorphic computations on a MacBook Pro machine with a 2.2-GHz Intel Core i7 processor and 16 GB of memory. Following inferences are made from the experimental results.

1) A linear increase of the computational time with increasing number of parties indicates that the proposed approach is scalable using parallel computing for homomorphic inference of the global classifier.

2) Remarkably, the computational time required for secure homomorphic evaluation of the global model in the cloud is independent of the sizes of distributed deep models and dimension of the input data, and thus, the approach is practical. The computational time depends only on the number of parties and the chosen precision.

3) The ratio of computational time to the number of parties is observed to be approximately equal to 3 s for 16-bit precision and 1.75 s for 8-bit precision. This verifies the application potential of the proposed method.

D. Robust Inference

To verify the robustness, MNIST Dataset is reconsidered with test images contaminated by zero-mean Gaussian additive noise with varying level of standard deviation. For a reference, the results obtained by a CNN in classifying noisy MNIST test images are reported. A CNN with the patch size of 5 × 5, first convolutional layer of 32 features, second convolutional layer of 64 features, and densely connected layer of 1024 neurons is considered. The convolutions use a stride of one and are zero padded so that the output is the same size as the input. The pooling is max pooling over 2 × 2 blocks. The CNN was trained for 10 000 iterations with the batch size of 100 in each iteration. Tables VI lists the accuracies obtained by two methods. It is observed from Table VI that the proposed method is far more robust than the CNN. Thus, the proposed method is not only privacy preserving but also results in a robust inference.

VI. CONCLUDING REMARKS

This study has outlined a membership-mappings-based approach to secure distributed deep learning. An algorithm for the variational learning of membership mappings is provided together with establishing convergence and robustness properties. The study has demonstrated the application potential of membership mappings for accurate, practical, and scalable privacy-preserving distributed deep learning. The main contribution is to introduce a method that relies on defining fuzzy attributes such that fuzzy attributes allow combining local models by means of a rule-based fuzzy system and the global model can be homomorphically evaluated efficiently.

The proposed fuzzy-based method offers the practical advantages via reducing computational time for homomorphic inference of the global classifier. This is achieved by reducing the problem of homomorphic inference of the global classifier to the problem of homomorphic evaluation of the “arg min” function over \( K \) encrypted values, where \( K \) is the number of parties participating in distributed learning. Thus, the computational time remains independent of the sizes of distributed

| Method | Security | Machine | Accuracy | Time |
|--------|----------|---------|----------|------|
| Proposed | 128-bits | 2.2 GHz Intel Core i7 | 0.987 | 4.98 s |
| NN-20 [33] | 128-bits | 2.5 GHz Intel Core i7 | 0.971 | 115.52 s |
| NN-50 [33] | 128-bits | 2.5 GHz Intel Core i7 | 0.947 | 233.55 s |
| NN-100 [33] | 128-bits | 2.5 GHz Intel Core i7 | 0.830 | 481.61 s |

Table V: Comparison with homomorphically evaluated deep neural networks on MNIST dataset.

**Table VI: Performance on Noisy MNIST Digits**

| Noise standard deviation | Proposed (16-bits precision of homomorphic computations) | Non private CNN |
|--------------------------|--------------------------------------------------------|-----------------|
| 0                        | 0.9872                                                 | 0.9897          |
| 0.2                      | 0.9817                                                 | 0.9608          |
| 0.4                      | 0.9649                                                 | 0.8198          |
| 0.6                      | 0.8987                                                 | 0.6227          |
| 0.8                      | 0.7986                                                 | 0.4482          |
| 1                        | 0.6815                                                 | 0.3203          |
deep models and dimension of the input data. This feature is of practical significance, since it results into a linear increase of the computational time with increasing number of parties, and thus, the proposed approach becomes scalable using parallel computing for homomorphic inference of the global classifier.

The method protects the training data privacy in a distributed learning scenario from an aggressive aggregator. The limitation of the method is that the user (i.e., input data owner) sends data without encryption to all parties (who own training datasets) for evaluating local deep models, and thus, the privacy of input data is not protected from training data owners. The future research will extend the method in the following three directions.

1) The privacy of input data will be protected from training data owners by means of homomorphic encryption.
2) Instead of classification, the privacy-preserving distributed regression problem will be considered.
3) An application of the membership mappings to privacy-preserving federated learning will be considered.

APPENDIX A
PROOF OF RESULT 1

The proof is split into three parts.

1) \textbf{Part 1:} Since $K_{aa} > 0$, there exists the unique square root, $K_{aa}^{-1/2} > 0$. Thus,

$$SV^TK_{aa}^{-1}VS = (K_{aa}^{-1/2}VS)^T(K_{aa}^{-1/2}VS) \quad (82)$$

> 0. \quad (83)

Since $\tau > 0$ and $\beta > 0$,

$$\min_{\text{eigen}} \left( I + \frac{1}{(\tau + \beta^{-1})}SV^TK_{aa}^{-1}VS \right) > 1 \quad (84)$$

where \(\min_{\text{eigen}}(.)\) denotes the minimum eigenvalue. Thus,

$$\max_{\text{eigen}} \left( \left( I + \frac{1}{(\tau + \beta^{-1})}SV^TK_{aa}^{-1}VS \right)^{-2} \right) < 1 \quad (85)$$

where \(\max_{\text{eigen}}(.)\) denotes the maximum eigenvalue. As a result of (85),

$$\mathcal{R}(\beta^{-1}) < \frac{1}{pN} \sum_{j=1}^{p} (\|b_j^1\|^2 + \|b_j^2\|^2). \quad (86)$$

As $U$ is orthogonal, it follows from (31) that $\|b_j^1\|^2 + \|b_j^2\|^2 = \|y_j\|^2$, and thus,

$$\mathcal{R}(\beta^{-1}) < \beta^{-1}_{\text{up}}. \quad (87)$$

It follows immediately from (37) that $\mathcal{R}(\beta^{-1}) > \beta^{-1}_{\text{low}}$. Hence, $\mathcal{R}(\beta^{-1}) \in (\beta^{-1}_{\text{low}}, \beta^{-1}_{\text{up}})$.

2) \textbf{Part 2:} The derivative of $\mathcal{R}$ w.r.t. $\beta^{-1}$ is given as

$$\frac{d\mathcal{R}(\beta^{-1})}{d\beta^{-1}} = \frac{2}{pN} \sum_{j=1}^{p} (\tau + \beta^{-1}) (b_j^1)^T \times \left( (\tau + \beta^{-1})I + \frac{1}{(\tau + \beta^{-1})}SV^TK_{aa}^{-1}VS \right)^{\frac{1}{2}} b_j^1. \quad (88)$$

Consider

$$\langle (\tau + \beta^{-1})^2 (b_j^1)^T \left( (\tau + \beta^{-1})I + \frac{1}{(\tau + \beta^{-1})}SV^TK_{aa}^{-1}VS \right)^{-3} b_j^1 \rangle \quad (89)$$

$$\leq (\tau + \beta^{-1}) \left\| \left( (\tau + \beta^{-1})I + \frac{1}{(\tau + \beta^{-1})}SV^TK_{aa}^{-1}VS \right)^{-1} b_j^1 \right\|_2 \quad (90)$$

$$\times (b_j^1)^T \left( (\tau + \beta^{-1})I + \frac{1}{(\tau + \beta^{-1})}SV^TK_{aa}^{-1}VS \right)^{-2} b_j^1 \quad (91)$$

where \(\min_{\text{sing}}(\cdot)\) denotes the minimum singular value. Observing that $\tau > 0$, $\beta^{-1} > 0$, and $SV^TK_{aa}^{-1}VS > 0$ [i.e., (83)], we have

$$\min_{\text{sing}} \left( I + \frac{1}{(\tau + \beta^{-1})}SV^TK_{aa}^{-1}VS \right)$$

$$= 1 + \min_{\text{sing}} \left( I + \frac{1}{(\tau + \beta^{-1})}SV^TK_{aa}^{-1}VS \right)$$

$$> 1. \quad (92)$$

Combining (91) and (92), we have

$$\langle (\tau + \beta^{-1})^2 (b_j^1)^T \left( (\tau + \beta^{-1})I + \frac{1}{(\tau + \beta^{-1})}SV^TK_{aa}^{-1}VS \right)^{-3} b_j^1 \rangle$$

$$< (\tau + \beta^{-1}) (b_j^1)^T \left( (\tau + \beta^{-1})I + \frac{1}{(\tau + \beta^{-1})}SV^TK_{aa}^{-1}VS \right)^{-2} b_j^1. \quad (93)$$

Using (93) in (88), we get

$$\frac{d\mathcal{R}(\beta^{-1})}{d\beta^{-1}} > 0. \quad (94)$$

Observing that $\tau > 0$, $\beta^{-1} > 0$, and $SV^TK_{aa}^{-1}VS > 0$ [i.e., (83)], we also have

$$\langle (\tau + \beta^{-1})^2 (b_j^1)^T \left( (\tau + \beta^{-1})I + \frac{1}{(\tau + \beta^{-1})}SV^TK_{aa}^{-1}VS \right)^{-3} b_j^1 \rangle > 0. \quad (95)$$

Using (95) in (88), we get

$$\frac{d\mathcal{R}(\beta^{-1})}{d\beta^{-1}} < \frac{2}{pN} (\tau + \beta^{-1}) \sum_{j=1}^{p} (b_j^1)^T \times \left( (\tau + \beta^{-1})I + \frac{1}{(\tau + \beta^{-1})}SV^TK_{aa}^{-1}VS \right)^{\frac{1}{2}} b_j^1. \quad (96)$$
Inequality (96), using (37), can be expressed as
\[
\frac{dR(\beta^{-1})}{d\beta^{-1}} < \frac{2 \ pN R(\beta^{-1})}{\tau + \beta^{-1}} \sum_{j=1}^{p} ||b_j^2||^2 \quad (97)
\]
where (98) follows using (86). Inequalities (94) and (97) lead to (40).

3) Part 3: Introduce \( h(\beta^{-1}) = R(\beta^{-1}) - \beta^{-1} \), and observe that \( h(\beta^{-1})_{\text{low}} > 0 \) and \( h(\beta^{-1})_{\text{up}} < 0 \). By the intermediate value theorem, there is a \( \hat{\beta}^{-1} \in (\beta^{-1}_{\text{low}}, \beta^{-1}_{\text{up}}) \) such that \( h(\hat{\beta}^{-1}) = 0 \), i.e., \( \hat{\beta}^{-1} = R(\beta^{-1}) \). Thus, \( \hat{\beta}^{-1} \) is a fixed point of \( R(\beta^{-1}) \).

**Appendix B**

**Proof of Result 2**

As \( \beta^{-1}_{\text{it}} > \beta^{-1}_{\text{low}} \), it follows from (41) that
\[
\tau + \beta^{-1}_{\text{it}} > \frac{2}{pN} \sum_{j=1}^{p} ||y_j||^2
\]
and thus,
\[
\frac{dR(\beta^{-1}_{\text{it}})}{d\beta^{-1}} < \frac{2}{pN} \sum_{j=1}^{p} ||b_j^2||^2 = \frac{2}{pN} \sum_{j=1}^{p} (||b_j^2||^2 + ||b_j^2||^2).
\]
That is, there exists a constant \( k \) such that
\[
0 < \frac{dR(\beta^{-1}_{\text{it}})}{d\beta^{-1}} < k < 1 \quad \forall \ \beta^{-1}_{\text{it}} \in \{0, 1, 2, \ldots \}.
\]

Let \( \hat{\beta}^{-1} \) be a fixed point of \( R(\beta^{-1}) \). Now, consider
\[
\beta_{\text{it}} - \hat{\beta}^{-1} = R(\beta_{\text{it}} - \hat{\beta}^{-1}) - R(\beta^{-1})
\]
\[
\leq k \beta_{\text{it}} - \hat{\beta}^{-1} - \hat{\beta}^{-1}
\]
\[
\leq k^t \beta_{\text{it}} - \hat{\beta}^{-1}
\]
that leads to
\[
\lim_{it \to \infty} \beta_{\text{it}} - \hat{\beta}^{-1} = \lim_{it \to \infty} k^t \beta_{\text{it}} - \hat{\beta}^{-1} = 0. \quad (106)
\]

The uniqueness of the fixed point can be seen via assuming by contradiction that there exists another fixed point, say \( \tilde{\beta}^{-1} \). Now consider
\[
\tilde{\beta}^{-1} - \hat{\beta}^{-1} = R(\tilde{\beta}^{-1}) - R(\hat{\beta}^{-1}) \leq k \tilde{\beta}^{-1} - \hat{\beta}^{-1}
\]
\[
< \tilde{\beta}^{-1} - \hat{\beta}^{-1}.
\]
This implies that \( \tilde{\beta}^{-1} = \hat{\beta}^{-1} \). Hence, the result follows.

**Appendix C**

**Proof of Result 3**

According to the triangle inequality
\[
\left\| (K_{xa} + \Delta_x K_{aa}^{1/2}) \alpha_j^* - (y_j + \Delta y_j) \right\|
\leq \left\| K_{xa} \alpha_j^* - y_j \right\| + \left\| \Delta_x K_{aa}^{1/2} \alpha_j^* - \Delta y_j \right\|
\leq \left\| K_{xa} \alpha_j^* - y_j \right\| + \left\| \Delta_x \Delta y_j \right\|_{\mathcal{F}} \left\| K_{aa}^{1/2} \alpha_j^* \right\|
\leq \left\| K_{xa} \alpha_j^* - y_j \right\| + \delta_m \sqrt{1 + (\alpha_j^*)^T K_{aa} \alpha_j^*}
\]
and hence,
\[
\hat{\alpha}_j = \left( K_{xa}^T K_{xa} + \delta_m \frac{||K_{xa} \alpha_j^* - y_j||}{1 + \alpha_j^* K_{aa} \alpha_j^*} \right)^{-1} K_{xa}^T y_j.
\]

Algorithm 1 estimates \( \alpha_j(\hat{\beta}^{-1}) \) using (24). It can be seen using (24) that
\[
\left( \tau + \hat{\beta}^{-1} \right)^{-1} \frac{1 + (\alpha_j(\hat{\beta}^{-1}))^T K_{aa} \alpha_j(\hat{\beta}^{-1})}{||K_{xa} \alpha_j(\hat{\beta}^{-1}) - y_j||} = \delta_m. \quad (111)
\]

As a result of (111), it follows from (24) that
\[
\alpha_j(\hat{\beta}^{-1}) = K_{xa}^T K_{xa}
\]
\[
+ \delta_m \frac{||K_{xa} \alpha_j(\hat{\beta}^{-1}) - y_j||}{\sqrt{1 + (\alpha_j(\hat{\beta}^{-1}))^T K_{aa} \alpha_j(\hat{\beta}^{-1})}} K_{xa}^T y_j.
\]

As equalities (112) and (110) are identical, \( \alpha_j(\hat{\beta}^{-1}) \) which is the solution of (112) must be equal to \( \hat{\alpha}_j \) which is the solution of (110), i.e.,
\[
\alpha_j(\hat{\beta}^{-1}) = \hat{\alpha}_j. \quad (113)
\]

Hence, Algorithm 1 solves the min–max problem (60).

**References**

[1] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in *Proc. 41st Annu. ACM Symp. Theory Comput.*, New York, NY, USA, 2009, pp. 169–178.

[2] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, “Fully homomorphic encryption over the integers,” in *Advances in Cryptology – EUROCRYPT*, H. Gilbert, Ed. Berlin, Germany: Springer, 2010, pp. 24–42.

[3] J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi, “Fully homomorphic encryption over integers with shorter public keys,” in *Advances in Cryptology*, P. Rogaway, Ed. Berlin, Germany: Springer, 2010, pp. 487–504.

[4] J. H. Cheon et al., “Batch fully homomorphic encryption over the integers,” in *Advances in Cryptology*, T. Johansson and P. Q. Nguyen, Eds. Berlin, Germany: Springer, 2013, pp. 315–335.

[5] K. Nuida and K. Kurosawa, “(Batch) fully homomorphic encryption over integers for non-binary message spaces,” in *Advances in Cryptology*, E. Oswald and M. Fischlin, Eds. Berlin, Germany: Springer, 2015, pp. 537–555.
[6] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” in Proc. 37th Annu. ACM Symp. Theory Comput., Symp. Theory Comput., New York, NY, USA, 2005, pp. 84–93.

[7] Z. Brakerski and V. Vaikuntanathan, “Efficient fully homomorphic encryption from (standard) LWE,” in Proc. IEEE 52nd Annu. Symp. Found. Comput. Sci., 2011, pp. 97–106.

[8] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, “Leveled fully homomorphic encryption without bootstrapping,” in Proc. 3rd Innov. Theor. Comput. Sci. Conf., New York, NY, USA, 2012, pp. 309–325.

[9] Z. Brakerski, “Fully homomorphic encryption without modulus switching from classical GAPSVP,” in Advances in Cryptology, R. Safavi-Naini and R. Canetti, Eds. Berlin, Germany: Springer, 2012, pp. 868–886.

[10] J. Fan and F. Vercauteren, “Somewhat practical fully homomorphic encryption,” IACR ePrint Arch., vol. 2012, p. 144, 2012. [Online]. Available: http://eprint.iacr.org/2012/144

[11] C. Gentry, A. Sahai, and B. Waters, “Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based,” in Advances in Cryptology, R. Canetti and J. A. Garay, Eds. Berlin, Germany: Springer, 2013, pp. 75–92.

[12] L. Ducas and D. Micciancio, “FHEW: Bootstrapping homomorphic encryption in less than a second,” in Advances Cryptology—EUROCRYPT, E. Oswald and M. Fischlin, Eds. Berlin, Germany: Springer, 2015, pp. 617–640.

[13] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, “Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds,” in Advances in Cryptology—ASIACRYPT, J. H. Cheon and T. Takagi, Eds., Berlin, Germany: Springer, 2016, pp. 3–33.

[14] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, “Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE,” in Advances in Cryptology—ASIACRYPT, T. Takagi and T. Peyrin, Eds., Cham, Switzerland: Springer, 2017, pp. 377–408.

[15] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, “TFHE: Fast fully homomorphic encryption library,” Aug. 2016. [Online]. Available: https://tfhe.github.io/tfhe/

[16] M. Kumar and B. Freudenthaler, “Fuzzy membership functional analysis for nonparametric deep models of image features,” IEEE Trans. Fuzzy Syst., vol. 28, no. 12, pp. 3345–3359, Dec. 2020.

[17] 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 Syst., vol. 29, no. 12, pp. 3873–3886, Dec. 2021.

[18] M. Kumar, S. Singh, and B. Freudenthaler, “Gaussian fuzzy theoretic analysis for variational learning of nested compositions,” Int. J. Approx. Reasoning, vol. 131, pp. 1–29, 2021.

[19] W. Zhang, M. Kumar, W. Ding, X. Li, and J. Yu, “Variational learning of deep fuzzy theoretic nonparametric model,” Neurocomputing, vol. 506, pp. 128–145, 2022.

[20] S. Park, S. J. Lee, E. Weiss, and Y. Motai, “Intra- and inter-fractional variation prediction of lung tumors using fuzzy deep learning,” IEEE J. Transl. Eng. Health Med., vol. 4, pp. 1–12, 2016, Art. no. 4300112.

[21] Y. Deng, Z. Ren, Y. Kong, F. Bao, and Q. Dai, “A hierarchical fused fuzzy deep neural network for data classification,” IEEE Trans. Fuzzy Syst., vol. 25, no. 4, pp. 1006–1012, Aug. 2017.

[22] S. Zhou, Q. Chen, and X. Wang, “Fuzzy deep belief networks for semi-supervised sentiment classification,” Neurocomputing, vol. 131, pp. 312–322, May 2014.

[23] C. E. Hatri and J. Boumhidi, “Fuzzy deep learning based urban traffic incident detection,” in Proc. Intell. Syst. Comput. Vis., Apr. 2017, pp. 1–6.

[24] S. Bonanno, K. Nock, L. Smith, P. Elmore, and F. Petry, “An approach to differentially private learning of distributed deep fuzzy models,” in Database and Expert Systems Applications—DEXA 2021, G. Kotsis et al., Eds., Cham, Switzerland: Springer, 2021, pp. 138–147.

[25] M. Kumar, B. Moser, L. Fischer, and B. Freudenthaler, “Membership-mappings for data representation learning: A Bregman divergence based conditionally deep autoencoder,” in Database and Expert Systems Applications—DEXA 2021, G. Kotsis et al., Eds., Cham, Switzerland: Springer, 2021, pp. 127–137.

[26] M. Kumar, R. Canetti, and J. A. Garay, Eds. Berlin, Germany: Springer, 2012, pp. 868–886.

[27] M. Kumar, B. Moser, L. Fischer, and B. Freudenthaler, “Membership-mappings for data representation learning: A Bregman divergence based conditionally deep autoencoder,” in Database and Expert Systems Applications—DEXA 2021, G. Kotsis et al., Eds., Cham, Switzerland: Springer, 2021, pp. 127–137.

[28] M. Kumar, B. Moser, L. Fischer, and B. Freudenthaler, “Membership-mappings for data representation learning: Measure theoretic conceptualization,” in Database and Expert Systems Applications—DEXA 2021, G. Kotsis et al., Eds., Cham, Switzerland: Springer, 2021, pp. 127–137.

[29] M. Kumar, B. Moser, L. Fischer, and B. Freudenthaler, “Membership-mappings for data representation learning: A Bregman divergence based conditionally deep autoencoder,” in Database and Expert Systems Applications—DEXA 2021, G. Kotsis et al., Eds., Cham, Switzerland: Springer, 2021, pp. 127–137.

[30] M. Kumar, “Differentially private transferrable deep learning with membership-mappings,” Adv. Comput. Intell., vol. 3, no. 1, pp. 1–27, 2023.

[31] M. Kumar, M. Rossbory, B. A. Moser, and B. Freudenthaler, “An optimal (ε, δ)–differentially private learning of distributed deep fuzzy models,” Inf. Sci., vol. 546, pp. 87–120, 2021.

[32] M. Kumar, M. Rossbory, B. A. Moser, and B. Freudenthaler, “Differentially private learning of distributed deep models,” in Proc. Adjunct Pub. 29th ACM Conf. User Model., Adapt. Personalization, New York, NY, USA, 2020, pp. 193–200.

[33] I. Chillotti, M. Joye, and P. Paillier, “Programmable bootstrapping enables efficient homomorphic inference of deep neural networks,” in Proc. 5th Int. Symp. Cyber Secur. Cryptogr. Mach. Learn., 2021, pp. 1–19.

[34] M. Kumar, B. Moser, L. Fischer, and B. Freudenthaler, “Towards practical secure privacy-preserving machine (deep) learning with distributed data,” in Database and Expert Systems Applications—DEXA Workshops, G. Kotsis et al., Eds., Cham, Switzerland: Springer, 2022, pp. 55–66.

Mohit Kumar received the B.Tech. degree in electrical engineering from the National Institute of Technology Hamirpur, Hamirpur, India, in 1999, the M.Tech. degree in control engineering and instrumentation from the Indian Institute of Technology Delhi, Delhi, India, in 2001, and the Ph.D. and Habilitation degrees in automation engineering from the University of Rostock, Rostock, Germany, in 2004 and 2009, respectively. He is currently a Professor of “computational intelligence in automation” with the Faculty of Computer Science and Electrical Engineering, University of Rostock, Rostock, Germany. He also holds the position of Key Researcher with Software Competence Center Hagenberg, Austria. His research interests include computational intelligence and privacy-preserving machine learning.

Weiping Zhang received the Ph.D. degree in computer science from the University of Rostock, Rostock, Germany, in 2014. He is currently an Associate professor with the School of Software, Northwestern Polytechnical University, Taicang, China. His research interests include machine learning algorithms and their practical applications.

Lukas Fischer received the M.Sc. degree in medical informatics from TU Wien, Vienna, Austria, in 2010, and the Ph.D. degree in medical imaging from the Medical University of Vienna, Vienna, Austria, in 2016. He is a Research Manager with Data Science at Software Competence Center, Hagenberg, Austria. His research interests include machine learning and computer vision.

Bernhard Freudenthaler studied business informatics with Johannes Kepler University Linz, Linz, Austria, and received the Ph.D. degree in business informatics from the Institute for Application Oriented Knowledge Processing, Johannes Kepler University Linz, in 2009. He is a Chief Operating Officer with the Software Competence Center, Hagenberg, Austria. His main research interests include case-based reasoning, decision support systems, as well as predictive analytics/maintenance.