Information Sciences 526 (2020) 166–179

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Privacy preservation for machine learning training and classification based on homomorphic encryption schemes

Jing Li[a,b], Xiaohui Kuang[c,*], Shujie Lin[a], Xu Ma[d], Yi Tang[e]

[a] School of Computer Science, Guangzhou University, Guangzhou, China
[b] State Key Laboratory of Integrated Service Networks, Xidian University, Xian, China
[c] National Key Laboratory of Science and Technology on Information System Security, Beijing, China
[d] School of Software, Qufu Normal University, Jinan, China
[e] Department of Mathematics, Guangzhou University, Guangzhou, China

ARTICLE INFO

Article history:
Received 18 November 2019
Revised 3 February 2020
Accepted 15 March 2020
Available online 4 April 2020

Keywords:
Privacy preservation
Homomorphic encryption
Machine learning

ABSTRACT

In recent years, more and more machine learning algorithms depend on the cloud computing. When a machine learning system is trained or classified in the cloud environment, the cloud server obtains data from the user side. Then, the privacy of the data depends on the service provider, it is easy to induce the malicious acquisition and utilization of data. On the other hand, the attackers can detect the statistical characteristics of machine learning data and infer the parameters of machine learning model through reverse attacks. Therefore, it is urgent to design an effective encryption scheme to protect the data's privacy without breaking the performance of machine learning.

In this paper, we propose a novel homomorphic encryption framework over non-abelian rings, and define the homomorphism operations in ciphertexts space. The scheme can achieve one-way security based on the Conjugacy Search Problem. After that, a homomorphic encryption was proposed over a matrix-ring. It supports real numbers encryption based on the homomorphism of 2-order displacement matrix coding function and achieves fast ciphertexts homomorphic comparison without decrypting any ciphetexts operations' intermediate result. Furthermore, we use the scheme to realize privacy preservation for machine learning training and classification in data ciphertexts environment. The analysis shows that our proposed schemes are efficient for encryption/decryption and homomorphic operations.

1. Introduction

Machine learning is a data-driven model. The development mode of machine learning has gradually changed from "closed training style" to a more complex intelligent system. At the same time, the open cloud environment provides rich data sources and computing resources for machine learning, then large-scale information processing can be effectively realized [1,2]. However, while the cloud computing brings convenience to machine learning, its security problems also face severe challenges. Since users store machine learning data in the storage center of the cloud servers, then the users will lose the ability of physical isolation and data protection. In addition, some complex computing of machine learning is usually out-

* Corresponding author.
E-mail address: xhkuang@bupt.edu.cn (X. Kuang).

https://doi.org/10.1016/j.ins.2020.03.041
0020-0255/

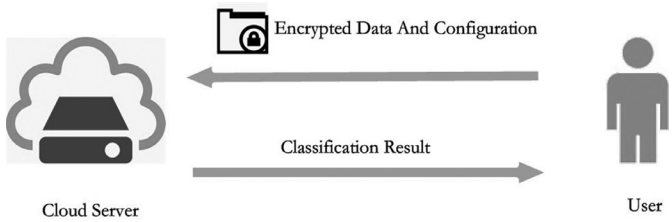

Fig. 1. Private machine learning mode.

sourced to the cloud server, which will cause data damage, illegal intrusion and many other security risks [3–5]. Therefore, privacy preservation has become a key factor for the further improvement of machine learning in cloud environment [6–8].

With the development of homomorphic encryption theory, the research on privacy preservation of machine learning data is increasing. To illustrate private machine learning mode based on encryption schemes with homomorphism, we take a private Logistic Regression classification as an example. The corresponding network architecture involves a cloud server and an user, where the user has large data source and wants to build a machine learning classifier with the help of the cloud server without revealing any sensitive information (see Fig. 1). In the interaction process, the user randomly initializes model parameters (weights $\vec{w}$, intercept b) and encrypts the data points, model parameters and relevant configuration in the light of special model. Then, the user sends the ciphertexts and relevant necessary configuration to the cloud server. Finally, the server employs optimized model parameters to classify under ciphertexts environment and returns the corresponding label to the user without privacy disclosure.

The private machine learning algorithms have been constantly designed based on homomorphic encryptions during these ten years. In 2011, for the security problem in multi-layer neural networks, Chen et al. [9] designed a private two-party distributed back propagation algorithm based on ElGamal multiplicative homomorphism encryption scheme. This algorithm allows training neural networks without requiring to display encrypted data to the other party. In 2012, Graepel et al. [10] combined layered homomorphic encryption with some gradient descent methods to realize binary classification, linear mean and Fisher linear discrimination. In 2014, based on Support Vector Machine (SVM), Rahulamathavan et al. [11] proposed a novel classification scheme for an user-server system, and its security depends on that of Paillier homomorphic encryption. In 2015, Bost et al. [12] employed additive homomorphism encryption to design some classification protocols in term of hyperplane decision, Naive Bayes and decision tree. In the same year, Wong et al. [13] constructed an asymmetric encryption based on vector scalar product, which supports the calculation of k-nearest neighbor of ciphertexts. In addition, Aslett et al. [14] proposed a Complete Random Forest (CRF) algorithm, Naive Bayes algorithm and Logical Regression algorithm for classifying ciphertexts by using a FHE scheme. In 2016, Liu et al. [15] presented an additive homomorphism agent aggregation algorithm, and the authors employed Naive Bayes classifier and a Top-k disease name retrieval protocol to protect the security of patient information and effectively evaluate the disease risk of patients. In addition, Dowlin et al. [16] gave an approximate neural network CryptoNets for ciphertexts data, and the accuracy is 99% for MNIST handwritten recognition data set. In addition, Baryalai et al. [17] established a non-collusive dual cloud model to decentralize the cloud power to enhance the security of neural networks system, at the same time, the authors improved the operation speed by using Paillier encryption scheme. In 2018, Li et al. [18] presented a private SVM classification technique based on a homomorphic encryption algorithm. In the same year, Phong et al. [19] built a deep learning privacy protection system to protect gradients by using an additive homomorphism encryption. In addition, some scholars studied homomorphic encryption algorithms [20,21] for the real number field, where Kim et al. [20] implemented a fast logic regression algorithm based on homomorphic encryption with parallelization techniques. Besides, the homomorphic comparison in ciphertexts space has been proposed in [22]. However, the given method needs to decrypt some intermediate results of ciphertext operations with the help of a secret key, namely, the ciphertexts comparison cannot be done by any untrusted third party.

Motivation. Throughout the above work, we summarize three urgent problems in private machine learning algorithms: (1) how to construct an efficient homomorphic encryption algorithm; (2) how to design a coding method for real numbers and define ciphertexts comparison operation; (3) how to realize complete machine learning training and classification algorithms by using a homomorphic encryption.

Contribution. In this paper, we put forward a novel homomorphic encryption framework based on non-abelian conjugate transformation, and then define the homomorphism addition and multiplication operations in ciphertexts space. This scheme achieves one-way security based on the Conjugacy Search Problem. Furthermore, a homomorphic encryption was proposed over a matrix-ring, which supports real numbers encryption. At the same time, we give the definitions of addition, subtraction, multiplication, division homomorphisms and define ciphertexts homomorphic comparison based on the homomorphic subtraction homomorphic operation. Finally, we realize privacy preservation for machine learning training and classification in data ciphertexts environment.

In the designing process, we also encounter some obstacles. The corresponding methods for solving these problems are given as below.

- For achieving secure homomorphic encryption over (real) number field, we adopt a homomorphic mapping from a random number to a 2-order displacement matrix, in which the sum of two elements in the first row of matrix is equal to the chosen random number. Such a coding method can keep some operations' homomorphism between (real) numbers and 2-order displacement matrices. Based on the non-abelian property for matrix multiplications and Conjugacy Search Problem, the encryption scheme is one-way scheme.
- To realize ciphertexts homomorphic comparison, we firstly define the homomorphic subtraction operation. However, based on the above coding method, each homomorphic operation result for two ciphertexs must keep the form of a "fresh" ciphertext. Then, unlike the homomorphic addition or multiplication, the homomorphic subtraction is not the direct subtraction for two ciphertex-matrices. To solve the intractability, we introduce the displacement matrix of 2-order identity matrix into the homomorphic subtraction to keep the "secondhand" ciphertext's form.
- In the process of realizing machine learning training and classification algorithms, some complex functions cannot be run for ciphertexts. For example, in private Logistic Regression classification, since the ciphertexts are matrices, then the non-linear *Sigmoid* function is not well-defined in ciphertexts space. Hence, we utilize Taylor Approximation to convert *Sigmoid* to a linear expression, where the Taylor expansion is approximately chosen the first eight terms to keep the accuracy of the algorithm.

The rest of this paper is organized as below: In Section 2, some related definitions are reviewed. A framework of homomorphic encryption (HE) scheme and the one-way security are given in Section 3. In Section 4, a HE scheme over (real) number field is proposed and then the homomorphic comparison in ciphertexts space is achieved. In Section 5, the private machine learning training and classification algorithms are presented. In Section 6, the performance of the HE scheme is discussed. Finally, conclusion is provided in Section 7.

2. Preliminaries

Some basic preliminaries are reviewed in this section.

Homomorphic encryption-HE [23,24] In the cryptography, a homomorphic encryption scheme involves an encryption algorithm *Enc*, a decryption algorithm *Dec* and ciphertexts homomorphic operations "$\oplus, \otimes$", where

$$Dec(Enc(m) \oplus Enc(\tilde{m})) = m + \tilde{m},$$

$$Dec(Enc(m) \otimes Enc(\tilde{m})) = m \cdot \tilde{m}$$

hold. Note that, the homomorphism of an encryption algorithm is different from the definition of homomorphic function in the algebra. That is, the homomorphism in the HE scheme does necessarily satisfy that $Enc(m) \oplus Enc(\tilde{m}) = Enc(m + \tilde{m})$ and $Enc(m) \otimes Enc(\tilde{m}) = Enc(m \cdot \tilde{m})$.

To achieve privacy preservation for machine learning training and classification based on homomorphic encryptions, the underlying hard problem is essential.

Conjugacy Search Problem-CSP. [25] Given a non-abelian algebraic structure Γ and $\Gamma_0 \subseteq \Gamma$, $b \in \Gamma$, where $b = hah^{-1}$ and $a \in \Gamma_0$. It is intractable to solve h, $a \in \Gamma$.

The CSP hard assumption is come from post-quantum cryptography and it can resist the existing quantum algorithm attacks [26]. In fact, CSP is a special form for GFP (Group Factorization Problem). A new study [27] proved that the GFP problem over general linear group $GL_d(R)$ is unsolvable when $d \geq 4$. Thus, the degree of the matrices used in our scheme should be at least 4 to ensure the security.

3. Warming-up

In this section, we firstly propose a symmetric HE scheme based on the intractability of the non-abelian Conjugacy Search Problem-CSP. Now we give the construction of this basic symmetric HE framework over non-abelian rings.

3.1. The framework of HE scheme over non-abelian rings

Setup: For a given non-abelian ring R, randomly select $h_1, h_2, h_3, h_4 \in R$ such that $H = \begin{pmatrix} h_1 & h_2 \\ h_3 & h_4 \end{pmatrix}$ is invertible. Then, H is the symmetric key.

Encryption: Let the message be $m \in R$. The user randomly chooses $r_1, r_2 \in R$ and constructs a matrix as $M = \begin{pmatrix} m & r_1 \\ 0 & r_2 \end{pmatrix}$. Then the corresponding ciphertext is $C = Enc_H(m) = HMH^{-1}$.

Decryption: The message is $m = Dec_H(C) = (H^{-1}CH)_{11}$, where $(H^{-1}CH)_{11}$ denotes the top left corner element of matrix $H^{-1}CH$.

Operations: Let C_1 and C_2 be the ciphertexts of m_1 and m_2, respectively. Now define "$\oplus$" gate and "$\otimes$" gate as $C_1 \oplus C_2 = C_1 + C_2$, $C_1 \otimes C_2 = C_1 \cdot C_2$ are matrix addition and matrix multiplication, respectively.

The homomorphic property can be obtained as below.

Homomorphic property. The addition homomorphism holds since

$$\begin{aligned} C_1 \oplus C_2 &= Enc_H(m_1) + Enc_H(m_2) \\ &= H\begin{pmatrix} m_1 & r_1 \\ 0 & r_2 \end{pmatrix}H^{-1} + H\begin{pmatrix} m_2 & r_1' \\ 0 & r_2' \end{pmatrix}H^{-1} \\ &= H\begin{pmatrix} m_1+m_2 & r_1+r_1' \\ 0 & r_2+r_2' \end{pmatrix}H^{-1}. \end{aligned}$$

Then we have that $Dec_H(C_1 \oplus C_2) = m_1 + m_2$. Meanwhile, the encryption has the multiplication homomorphism since

$$\begin{aligned} C_1 \otimes C_2 &= Enc_H(m_1) \cdot Enc_H(m_2) \\ &= H\begin{pmatrix} m_1 & r_1 \\ 0 & r_2 \end{pmatrix}H^{-1} \cdot H\begin{pmatrix} m_2 & r_1' \\ 0 & r_2' \end{pmatrix}H^{-1} \\ &= H\begin{pmatrix} m_1m_2 & r_1'' \\ 0 & r_2r_2' \end{pmatrix}H^{-1}. \end{aligned}$$

Then $Dec_H(C_1 \otimes C_2) = m_1m_2$.

3.2. Security analysis

The security of the above HE framework depends on the non-abelian property, and the scheme achieves the one-way security. That is, an adversary fails to obtain messages from the given ciphertexts. Actually, the privacy of message is based on the intractability of Conjugacy Search Problem and solving eigenvalues of the given ciphertext matrix for non-abelian rings.

- Due to the intractability of Conjugacy Search Problem, an adversary cannot get M by factoring ciphertext $C = HMH^{-1}$ [27]. Thus, the message m won't be revealed by factoring ciphertext C.
- According to the encryption algorithm, the message can be regarded as one eigenvalue of ciphertext matrix C, where $C = HMH^{-1} = H\begin{pmatrix} m & r_1 \\ 0 & r_2 \end{pmatrix}H^{-1}$. Since $m, r_1, r_2 \in R$ for non-abelian ring R, any attacker cannot get m by solving C' eigenvalue equation.

To better illustrate the importance of the non-abelian algebraic structure in data privacy, we assume R is an abelian ring, in this solution, the adversary can obtain C' eigenvalues by solving $det(\lambda E - C) = 0$, where $det(X)$ denotes the determinant of matrix X. Fortunately, we use a non-abelian ring R in our construction, the above eigenvalue-attack is unworkable. Since $det(A \cdot B) \neq det(A) \cdot det(B) \in R$ for non-abelian ring (The detailed reason is given in Remark 1), then the eigenvalue equation is

$$det(\lambda E - C) = det(H\begin{pmatrix} \lambda - m & r_1 \\ 0 & \lambda - r_2 \end{pmatrix}H^{-1})$$

$$\neq det(H) \cdot det\begin{pmatrix} \lambda - m & r_1 \\ 0 & \lambda - r_2 \end{pmatrix} \cdot det(H^{-1}) = (\lambda - m) \cdot (\lambda - r_2).$$

That is, $det(\lambda E - C) \neq (\lambda - m) \cdot (\lambda - r_2)$. In summary, the eigenvalues of the matrices over non-abelian ring are hard to be solved.

Remark 1. Let $A = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$, $B = \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix}$ for a_i, b_i $(i = 1, \ldots, 4)$ in non-abelian ring R, then $A \cdot B = \begin{pmatrix} a_1b_1 + a_2b_3 & a_1b_2 + a_2b_4 \\ a_3b_1 + a_4b_3 & a_3b_2 + a_4b_4 \end{pmatrix}$. Hence,

$$det(A \cdot B) = (a_1b_1 + a_2b_3) \cdot (a_3b_2 + a_4b_4) - (a_1b_2 + a_2b_4) \cdot (a_3b_1 + a_4b_3)$$

$$= a_1b_1a_3b_2 + a_2b_3a_4b_4 - a_1b_2a_3b_1 - a_2b_4a_4b_3 + \underbrace{a_1b_1a_4b_4 + a_2b_3a_3b_2 - a_1b_2a_4b_3 - a_2b_4a_3b_1}_{},$$

$$det(A) \cdot det(B) = (a_1a_4 - a_2a_3) \cdot (b_1b_4 - b_2b_3) = \underbrace{a_1a_4b_1b_4 + a_2a_3b_2b_3 - a_1a_4b_2b_3 - a_2a_3b_1b_4}_{}.$$

Note that, since $a_ib_j \neq b_ja_i$, then $det(A \cdot B) \neq det(A) \cdot det(B)$.

4. Two new homomorphic encryption schemes

Now we propose two new homomorphic encryption schemes for private machine learning based on the above framework.

4.1. Homomorphic encryption over number fields

To achieve the HE over any number field (especially, real number field), we propose a homomorphic number-coding method and give the symmetric HE scheme over some number field based on the above HE framework. Meanwhile, the scheme supports ciphertexts' addition, subtraction, multiplication and division homomorphic operations.

Setup: For any number field $\mathbb{F}$, let Ω denote a 2×2 matrix-ring over $\mathbb{F}$.

Key Generation: The user randomly selects $h_1, h_2, h_3, h_4 \in \Omega$ and constructs invertible matrix $H = \begin{pmatrix} h_1 & h_2 \\ h_3 & h_4 \end{pmatrix}$. Then the user keeps the symmetric key H.

Encryption: For a message $m \in \mathbb{F}$, the user randomly chooses $m_1, m_2 \neq 0$ such that $m = m_1 + m_2$ and constructs a matrix $m_0 = \begin{pmatrix} m_1 & m_2 \\ m_2 & m_1 \end{pmatrix}$. Meanwhile, the user selects $r_1, r_2 \in \Omega$. Then the ciphertext is

$$C = H \cdot \begin{pmatrix} m_0 & r_1 \\ \mathbf{0} & r_2 \end{pmatrix} \cdot H^{-1}.$$

Note that, (1) "**0**" in ciphertext matrices is a 2×2 zero matrix; (2) The ciphertext C is actually a 4×4 matrix over $\mathbb{F}$.

Decryption: For a given ciphertext C, the receiver computes

$$u = (H^{-1}CH)_{11}, \qquad m = u_{11} + u_{12}.$$

Here, we regard ciphertext matrix C as a nested matrix.

Operations: The definitions of addition homomorphism and multiplication homomorphism are the same as that of Scheme 1. That is, $C^*_{\oplus} = C + \bar{C}$, $C^*_{\otimes} = C \cdot \bar{C}$ for two ciphertexts $C, \bar{C}$. In addition, the homomorphic division is defined as $C^*_{\oslash} = C \cdot \bar{C}^{-1}$, and the homomorphic subtraction is $C^*_{\ominus} = C - \bar{C}$.

Security. On one hand, the scheme's security still relies on the non-abelian algebraic structure, namely, the adversary cannot factor ciphertext matrix C. On the other hand, the new scheme introduces random coding for the message, where the message $m = m_1 + m_2$ and $\begin{pmatrix} m_1 & m_2 \\ m_2 & m_1 \end{pmatrix} = (H^{-1}CH)^{-1}_{11}$ for random $m_1, m_2 \in \mathbb{F}$. That is, m_1, m_2 increase the randomness of the encryption algorithm.

4.2. Homomorphic property

Since the encryption algorithm uses the conjugate transformation on an upper triangular matrix, thus this scheme's homomorphic property is determined by **message-coding** and the homomorphic property is only discussed with respect to message-coding function: $m \to m_0 = \begin{pmatrix} m_1 & m_2 \\ m_2 & m_1 \end{pmatrix}$, where $m = m_1 + m_2$.

(1) The **addition homomorphism** holds since

$$m_0 + \bar{m}_0 = \begin{pmatrix} m_1 & m_2 \\ m_2 & m_1 \end{pmatrix} + \begin{pmatrix} \bar{m}_1 & \bar{m}_2 \\ \bar{m}_2 & \bar{m}_1 \end{pmatrix} = \begin{pmatrix} m_1 + \bar{m}_1 & m_2 + \bar{m}_2 \\ m_2 + \bar{m}_2 & m_1 + \bar{m}_1 \end{pmatrix}$$

Then, $m + \bar{m} = m_1 + m_2 + \bar{m}_1 + \bar{m}_2 = (m_0 + \bar{m}_0)_{11} + (m_0 + \bar{m}_0)_{12}$. Hence, $Dec(C^*_{\oplus}) = m + \bar{m}$.

(2) The **subtraction homomorphism** holds since

$$m_0 - \bar{m}_0 = \begin{pmatrix} m_1 & m_2 \\ m_2 & m_1 \end{pmatrix} - \begin{pmatrix} \bar{m}_1 & \bar{m}_2 \\ \bar{m}_2 & \bar{m}_1 \end{pmatrix} = \begin{pmatrix} m_1 - \bar{m}_1 & m_2 - \bar{m}_2 \\ m_2 - \bar{m}_2 & m_1 - \bar{m}_1 \end{pmatrix}$$

Then, $m - \bar{m} = m_1 + m_2 - (\bar{m}_1 + \bar{m}_2) = (m_1 - \bar{m}_1) + (m_2 - \bar{m}_2) = (m_0 - \bar{m}_0)_{11} + (m_0 - \bar{m}_0)_{12}$. Hence, $Dec(C^*_{\ominus}) = m - \bar{m}$.

(3) The **multiplication homomorphism** holds since

$$m_0 \cdot \bar{m}_0 = \begin{pmatrix} m_1 & m_2 \\ m_2 & m_1 \end{pmatrix} \cdot \begin{pmatrix} \bar{m}_1 & \bar{m}_2 \\ \bar{m}_2 & \bar{m}_1 \end{pmatrix} = \begin{pmatrix} m_1\bar{m}_1 + m_2\bar{m}_2 & m_1\bar{m}_2 + m_2\bar{m}_1 \\ m_1\bar{m}_2 + m_2\bar{m}_1 & m_1\bar{m}_1 + m_2\bar{m}_2 \end{pmatrix}$$

Here, $m_i, \bar{m}_j \in \mathbb{F}$ are pairwise commutative. Furthermore, $m = m_1 + m_2$, $\bar{m} = \bar{m}_1 + \bar{m}_2$ and $m \cdot \bar{m} = (m_1\bar{m}_1 + m_2\bar{m}_2) + (m_1\bar{m}_2 + m_2\bar{m}_1) = (m_0 \cdot \bar{m}_0)_{11} + (m_0 \cdot \bar{m}_0)_{12}$. Hence, $Dec(C^*_{\otimes}) = m\bar{m}$.

(4) The **division homomorphism** holds since

$$\bar{m}_0^{-1} = \begin{pmatrix} \bar{m}_1 & \bar{m}_2 \\ \bar{m}_2 & \bar{m}_1 \end{pmatrix}^{-1} = \frac{1}{(\bar{m}_1 - \bar{m}_2)\cdot(\bar{m}_1 + \bar{m}_2)} \cdot \begin{pmatrix} \bar{m}_1 & -\bar{m}_2 \\ -\bar{m}_2 & \bar{m}_1 \end{pmatrix}$$

and $\bar{m}_1 + \bar{m}_2 = \bar{m}$ for $\bar{m}_1 \neq \bar{m}_2$. Therefore,

$$(\bar{m}_0^{-1})_{11} + (\bar{m}_0^{-1})_{12} = \frac{1}{(\bar{m}_1 - \bar{m}_2)\cdot(\bar{m}_1 + \bar{m}_2)} \cdot \bar{m}_1 + \frac{1}{(\bar{m}_1 - \bar{m}_2)\cdot(\bar{m}_1 + \bar{m}_2)} \cdot (-\bar{m}_2)$$

$$= \frac{\bar{m}_1 - \bar{m}_2}{(\bar{m}_1 - \bar{m}_2)\cdot(\bar{m}_1 + \bar{m}_2)} = \frac{1}{\bar{m}_1 + \bar{m}_2} = \frac{1}{\bar{m}}.$$

That is, $\bar{m}_0^{-1}$ is one random message-coding of $\bar{m}^{-1}$. Furthermore, based on the multiplication homomorphism, we have that $m_0 \cdot \bar{m}_0^{-1}$ is the corresponding message-coding of $m \cdot \bar{m}^{-1}$, then $(m_0 \cdot \bar{m}_0^{-1})_{11} + (m_0 \cdot \bar{m}_0^{-1})_{12} = m \cdot \bar{m}$. Hence, $Dec(C^*_{\oslash}) = m \cdot \bar{m}^{-1}$.

Remark 2. The homomorphisms for ciphertexts not only depend on the correctness of decrypting "secondhand" ciphertexts, but also depend on the form of these ciphertexts, especially the form of message-codings. For example,

$$m_0 + \bar{m}_0 = \begin{pmatrix} m_1 + \bar{m}_1 & m_2 + \bar{m}_2 \\ m_2 + \bar{m}_2 & m_1 + \bar{m}_1 \end{pmatrix},$$

except $(m_0 + \bar{m}_0)_{11} = (m_0 + \bar{m}_0)_{12} = m + \bar{m}$, we also have $(m_0 + \bar{m}_0)_{11} = (m_0 + \bar{m}_0)_{22}$ and $(m_0 + \bar{m}_0)_{12} = (m_0 + \bar{m}_0)_{21}$. Then, $m_0 + \bar{m}_0$ has the same form as the fresh message-codings m_0 and $\bar{m}_0$.

4.3. Homomorphic encryption scheme with ciphertexts comparison

We present a homomorphic comparison scheme that requires no trusted third party, in particular, the comparison process does not need to decrypt some intermediate result for ciphertexts operations. Namely, the comparison between two ciphertexts can be efficiently done by any third party without secret key. The scheme lightly extends the dimension of ciphertext based on the HE scheme in Section 4.1.

Setup: Suppose that $\mathbb{F}$ denotes a number field and Ω is a 2×2 matrix-ring over $\mathbb{F}$.

Key Generation: The user randomly selects $h_1, \ldots, h_9 \in \Omega$ such that matrix

$$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}$$

is invertible. Then the symmetric key is H (In fact, H is a 6×6 matrix over $\mathbb{F}$).

Encryption: Let the message be $m \in \mathbb{F}$. The user randomly chooses m_1, $m_2 (m_1 > m_2)$, m_3, $m_4 (m_3 > m_4)$, m_5, $m_6 (m_5 > m_6)$ such that $m = m_1 + m_2 = m_3 + m_4 = m_5 + m_6$ and constructs matrices

$$M_1 = \begin{pmatrix} m_1 & m_2 \\ m_2 & m_1 \end{pmatrix}, \quad M_2 = \begin{pmatrix} m_3 & m_4 \\ m_4 & m_3 \end{pmatrix}, \quad M_3 = \begin{pmatrix} m_5 & m_6 \\ m_6 & m_5 \end{pmatrix}.$$

Meanwhile, the user randomly selects $r_1, r_2, r_3 \in \Omega$. Then the ciphertext is

$$C = H \cdot \begin{pmatrix} M_1 & r_1 & r_2 \\ \mathbf{0} & M_2 & r_3 \\ \mathbf{0} & \mathbf{0} & M_3 \end{pmatrix} \cdot H^{-1}.$$

Note that, "$\mathbf{0}$" in ciphertext is still a 2×2 zero matrix over $\mathbb{F}$.

Decryption: The receiver computes

$$M_1 = (H^{-1} C H)_{11}.$$

Then, the message is $m = (M_1)_{11} + (M_1)_{12}$.

Operations: Suppose that C and $\bar{C}$ are the ciphertexts of m and $\bar{m}$, respectively. Then "$\oplus$" gate, "$\otimes$" gate and "$\oslash$" gate are defined as $C^*_{\oplus} = C + \bar{C}$, $C^* = C \times \bar{C}$ and $C^*_{\oslash} = C \cdot \bar{C}^{-1}$.

To keep the form of ciphertext $C^*_{\ominus}$, the **homomorphic subtraction** between $C, \bar{C}$ is re-defined as

$$C^*_{\ominus} = C - T \cdot \bar{C},$$

where a public matrix $T = H \cdot \begin{pmatrix} \delta & r_4 & r_5 \\ \mathbf{0} & \delta & r_6 \\ \mathbf{0} & \mathbf{0} & \delta \end{pmatrix} \cdot H^{-1}$ for $\delta = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and $r_4, r_5, r_6 \in \Omega$.

Comparison: To compare the messages $m, \bar{m}$, the cloud server carries $C, \bar{C}$ and computes $t = det(C - T \cdot \bar{C}) \in \mathbb{F}$. If $t > 0$, then it means that $m > \bar{m}$. If $t > 0$, then $m = \bar{m}$. If $t = 0$, then $m < \bar{m}$.

In the following proof, we only give the illustration of the subtraction homomorphism. The other homomorphic operations can be easily obtained from Section 4.2.

Subtraction Homomorphism.

$$C^*_{\ominus} = C - T \cdot \bar{C} = H \cdot \begin{pmatrix} M_1 & r_1 & r_2 \\ \mathbf{0} & M_2 & r_3 \\ \mathbf{0} & \mathbf{0} & M_3 \end{pmatrix} \cdot H^{-1} - H \cdot \begin{pmatrix} \delta \cdot \bar{M}_1 & r'_4 & r'_5 \\ \mathbf{0} & \delta \cdot \bar{M}_2 & r'_6 \\ \mathbf{0} & \mathbf{0} & \delta \cdot \bar{M}_3 \end{pmatrix} \cdot H^{-1},$$

where

$$\delta \cdot \bar{M}_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \bar{m}_1 & \bar{m}_2 \\ \bar{m}_2 & \bar{m}_1 \end{pmatrix} = \begin{pmatrix} \bar{m}_2 & \bar{m}_1 \\ \bar{m}_1 & \bar{m}_2 \end{pmatrix},$$

and

$$(\delta \cdot \bar{M}_1)_{11} + (\delta \cdot \bar{M}_1)_{12} = \bar{m}_2 + \bar{m}_1 = \bar{m}.$$

Actually, δ is the displacement matrix of identity matrix, in which only the places of $\bar{m}_1, \bar{m}_2$ are changed. Then, $\delta \cdot \bar{M}_1$ is also a random coding of message $\bar{m}$. Similarly, $\delta \cdot \bar{M}_2$ and $\delta \cdot \bar{M}_3$ are still message-codings of $\bar{m}$. That is, $T \cdot \bar{C}$ is a rational ciphertext of message $\bar{m}$. Furthermore, the correctness of the homomorphic subtraction can be obtained from the proof in Section 4.2.

Now we show the correctness of "secondhand" ciphertexts' forms under homomorphic operations.

Correctness of ciphertexts comparison.

(1) We firstly prove the homomorphic operation can keep the form of the ciphertext.

Addition: $C + \bar{C} = H \cdot \begin{pmatrix} M_1 + \bar{M}_1 & r_1 + \bar{r}_1 & r_2 + \bar{r}_2 \\ \mathbf{0} & M_2 + \bar{M}_2 & r_3 + \bar{r}_3 \\ \mathbf{0} & \mathbf{0} & M_3 + \bar{M}_3 \end{pmatrix} \cdot H^{-1}$, where

$$M_1 + \bar{M}_1 = \begin{pmatrix} m_1 + \bar{m}_1 & m_2 + \bar{m}_2 \\ m_2 + \bar{m}_2 & m_1 + \bar{m}_1 \end{pmatrix}.$$

Since $m_1 > m_2$, $\bar{m}_1 > \bar{m}_2$, then $m_1 + \bar{m}_1 > m_2 + \bar{m}_2$. That is, $(M_1 + \bar{M}_1)_{11} > (M_1 + \bar{M}_1)_{12}$. Similarly, $(M_2 + \bar{M}_2)_{11} > (M_2 + \bar{M}_2)_{12}$ and $(M_3 + \bar{M}_3)_{11} > (M_3 + \bar{M}_3)_{12}$. Thus, the homomorphic addition operation does not break the form of the secondhand ciphertext $C + \bar{C}$.

Multipication: $C \cdot \bar{C} = H \cdot \begin{pmatrix} M_1 \cdot \bar{M}_1 & r^*_1 & r^*_2 \\ \mathbf{0} & M_2 \cdot \bar{M}_2 & r^*_3 \\ \mathbf{0} & \mathbf{0} & M_3 \cdot \bar{M}_3 \end{pmatrix} \cdot H^{-1}$, where

$$M_1 \cdot \bar{M}_1 = \begin{pmatrix} m_1\bar{m}_1 + m_2\bar{m}_2 & m_1\bar{m}_2 + m_2\bar{m}_1 \\ m_1\bar{m}_2 + m_2\bar{m}_1 & m_1\bar{m}_1 + m_2\bar{m}_2 \end{pmatrix}.$$

Since

$$\begin{aligned} &(M_1 \cdot \bar{M}_1)_{11} - (M_1 \cdot \bar{M}_1)_{12} \\ &= m_1\bar{m}_1 + m_2\bar{m}_2 - (m_1\bar{m}_2 + m_2\bar{m}_1) \\ &= (m_1\bar{m}_1 - m_1\bar{m}_2) + (m_2\bar{m}_2 - m_2\bar{m}_1) \\ &= (m_1 - m_2) \cdot (\bar{m}_1 - \bar{m}_2) > 0 \end{aligned}$$

then $(M_1 \cdot \bar{M}_1)_{11} > (M_1 \cdot \bar{M}_1)_{12}$. Similarly, $(M_2 \cdot \bar{M}_2)_{11} > (M_2 \cdot \bar{M}_2)_{12}$ and $(M_3 \cdot \bar{M}_3)_{11} > (M_3 \cdot \bar{M}_3)_{12}$. Hence, the multiplication operation can also keep the ciphertext's form.

Subtraction: The homomorphic subtraction can be done by using the public matrix T. Based on the definition, $C - T \cdot \bar{C} = H \cdot \begin{pmatrix} M_1 & r_1 & r_2 \\ \mathbf{0} & M_2 & r_3 \\ \mathbf{0} & \mathbf{0} & M_3 \end{pmatrix} \cdot H^{-1} - H \cdot \begin{pmatrix} \delta \cdot \bar{M}_1 & r^*_4 & r^*_5 \\ \mathbf{0} & \delta \cdot \bar{M}_2 & r^*_6 \\ \mathbf{0} & \mathbf{0} & \delta \cdot \bar{M}_3 \end{pmatrix} \cdot H^{-1} = H \cdot \begin{pmatrix} M_1 - \delta \cdot \bar{M}_1 & r'_1 & r'_2 \\ \mathbf{0} & M_2 - \delta \cdot \bar{M}_2 & r'_3 \\ \mathbf{0} & \mathbf{0} & M_3 - \delta \cdot \bar{M}_3 \end{pmatrix} \cdot H^{-1}$, where

$$M_1 - \delta \cdot \bar{M}_1 = \begin{pmatrix} m_1 & m_2 \\ m_2 & m_1 \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \bar{m}_1 & \bar{m}_2 \\ \bar{m}_2 & \bar{m}_1 \end{pmatrix} = \begin{pmatrix} m_1 - \bar{m}_2 & m_2 - \bar{m}_1 \\ m_2 - \bar{m}_1 & m_1 - \bar{m}_2 \end{pmatrix}.$$

It can be seen that $(M_1 - \delta \cdot \bar{M}_1)_{11} - (M_1 - \delta \cdot \bar{M}_1)_{12} = m_1 - \bar{m}_2 - (m_2 - \bar{m}_1) = (m_1 - m_2) + (\bar{m}_1 - \bar{m}_2) > 0$, then $(M_1 - \delta \cdot \bar{M}_1)_{11} > (M_1 - \delta \cdot \bar{M}_1)_{12}$. Similarly, $(M_2 - \delta \cdot \bar{M}_2)_{11} > (M_2 - \delta \cdot \bar{M}_2)_{12}$ and $(M_3 - \delta \cdot \bar{M}_3)_{11} > (M_3 - \delta \cdot \bar{M}_3)_{12}$. Hence, the homomorphic subtraction can keep the ciphertext's form.

Division: $C \cdot \bar{C}^{-1} = H \cdot \begin{pmatrix} M_1 \cdot \bar{M}_1^{-1} & r'_1 & r'_2 \\ \mathbf{0} & M_2 \cdot \bar{M}_2^{-1} & r'_3 \\ \mathbf{0} & \mathbf{0} & M_3 \cdot \bar{M}_3^{-1} \end{pmatrix} \cdot H^{-1}$, where

$$\bar{M}_1^{-1} = (\bar{m}_1^2 - \bar{m}_2^2)^{-1} \cdot \begin{pmatrix} \bar{m}_1 & -\bar{m}_2 \\ -\bar{m}_2 & \bar{m}_1 \end{pmatrix} = \begin{pmatrix} (\bar{m}_1^2 - \bar{m}_2^2)^{-1} \cdot \bar{m}_1 & (\bar{m}_1^2 - \bar{m}_2^2)^{-1} \cdot (-\bar{m}_2) \\ (\bar{m}_1^2 - \bar{m}_2^2)^{-1} \cdot (-\bar{m}_2) & (\bar{m}_1^2 - \bar{m}_2^2)^{-1} \cdot \bar{m}_1 \end{pmatrix}.$$

To prove the correctness of the form of $\bar{M}_1^{-1}$, we only need to show $(\bar{m}_1^2 - \bar{m}_2^2)^{-1} \cdot \bar{m}_1 > (\bar{m}_1^2 - \bar{m}_2^2)^{-1} \cdot (-\bar{m}_2)$.

When $\bar{m}_1 > \bar{m}_2 > 0$, we have $\bar{m}_1^2 - \bar{m}_2^2 = (\bar{m}_1 - \bar{m}_2) \cdot (\bar{m}_1 + \bar{m}_2) > 0$ and $\bar{m}_1 > -\bar{m}_2$, then $(\bar{m}_1^2 - \bar{m}_2^2)^{-1} \cdot \bar{m}_1 > (\bar{m}_1^2 - \bar{m}_2^2)^{-1} \cdot (-\bar{m}_2)$;

When $0 > \bar{m}_1 > \bar{m}_2$, we have $\bar{m}_1^2 - \bar{m}_2^2 < 0$ and $\bar{m}_1 < -\bar{m}_2$, then $(\bar{m}_1^2 - \bar{m}_2^2)^{-1} \cdot \bar{m}_1 > (\bar{m}_1^2 - \bar{m}_2^2)^{-1} \cdot (-\bar{m}_2)$;

When $\bar{m}_1 > 0 > \bar{m}_2$, this case is discussed according to the absolute values of $|\bar{m}_1|$, $|\bar{m}_2|$.

(a) If $|\bar{m}_1| > |\bar{m}_2|$, then $\bar{m}_1^2 - \bar{m}_2^2 > 0$ and $\bar{m}_1 > -\bar{m}_2$, hence $(\bar{m}_1^2 - \bar{m}_2^2)^{-1} \cdot \bar{m}_1 > (\bar{m}_1^2 - \bar{m}_2^2)^{-1} \cdot (-\bar{m}_2)$.

(b) If $|\bar{m}_1| < |\bar{m}_2|$, then $\bar{m}_1^2 - \bar{m}_2^2 < 0$ and $\bar{m}_1 < -\bar{m}_2$, $(\bar{m}_1^2 - \bar{m}_2^2)^{-1} \cdot \bar{m}_1 > (\bar{m}_1^2 - \bar{m}_2^2)^{-1} \cdot (-\bar{m}_2)$ also holds.

Therefore, $(\bar{M}_1^{-1})_{11} > (\bar{M}_1^{-1})_{12}$, $(\bar{M}_2^{-1})_{11} > (\bar{M}_2^{-1})_{12}$ and $(\bar{M}_3^{-1})_{11} > (\bar{M}_3^{-1})_{12}$. Hence, the homomorphic division can keep the ciphertext's form.

(2) We will present the correctness of ciphertexts comparison. The cloud server computes

$$t = det(C - T \cdot \bar{C}) = det(H) \cdot det\begin{pmatrix} M_1 - \delta \cdot \bar{M}_1 & r'_1 & r'_2 \\ \mathbf{0} & M_2 - \delta \cdot \bar{M}_2 & r'_3 \\ \mathbf{0} & \mathbf{0} & M_3 - \delta \cdot \bar{M}_3 \end{pmatrix} \cdot det(H^{-1})$$

$$= det(M_1 - \delta \cdot \bar{M}_1) \cdot det(M_2 - \delta \cdot \bar{M}_2) \cdot det(M_3 - \delta \cdot \bar{M}_3)$$

for $t \in \mathbb{F}$. At the same time, based on the message-coding method, let $A = \begin{pmatrix} a & b \\ b & a \end{pmatrix}$ for $a > b$ and $a + b = m$, then

$$det(A) = a^2 - b^2 = (a+b) \cdot (a-b) = m \cdot (a-b).$$

Since $a > b$, we have $a - b > 0$. Hence, $det(A) = m \cdot (a-b)$ and m have the same sign. Thus, $det(M_1 - \delta \cdot \bar{M}_1) = (m - \bar{m}) \cdot (x - y)$ for $x, y \in \mathbb{F}$ and $x - y > 0$. Then,

$$t = det(M_1 - \delta \cdot \bar{M}_1) \cdot det(M_2 - \delta \cdot \bar{M}_2) \cdot det(M_3 - \delta \cdot \bar{M}_3) = \kappa \cdot (m - \bar{m})^3 = \kappa \cdot (m - \bar{m})^2 \cdot (m - \bar{m})$$

for some $\kappa \in \mathbb{F}$ and $\kappa > 0$. Thus, if $t > 0$, then it means $m > \bar{m}$; if $t < 0$, then $m < \bar{m}$. And if $t = 0$, then $m = \bar{m}$.

Equivalence Test for HE. Based on the above analysis, if $t = 0$, it means that $m = \bar{m}$. Thus, we get an equivalence test method for two ciphertexts C and $\bar{C}$. That is, the cloud server can decide whether the messages for ciphertexts C and $\bar{C}$ are equal or not. This fact provides an efficient ciphertext-search method in cloud environment.

5. Private machine learning training and classification

This section will present private machine learning training and classification for some classical algorithms [28] based on the proposed HE scheme in Section 4.3. In the following description, we uniformity denote the ciphertext of message m as c_m.

5.1. Private training and classification for logistic regression

Now we take private Logistic Regression for instance and give the processes of training algorithm-parameters and classifying data points on ciphertexts. Logistic Regression is a prevailed classification algorithm, where the user can utilize train data set to establish regression formulas. In the training algorithm, each feature of data points multiplies a special regression coefficient, as $z = x_1w_1 + x_2w_2 + \cdots + x_nw_n$, where z will be substituted into *Sigmoid* function. If *Sigmoid*(z) is greater than 0.5, the label is 1, else, the label is 0. To protect sensitive training data, we hope this process can be done in ciphertext environment. Based on the homomorphism of our proposed HE scheme in Section 4.3, we adopt analogous treatment to train parameters on the ciphertexts.

Let $\mathbb{F}$ be the real number field and k train data points be $\vec{x}^{(i)} = (x_1^{(i)}, \ldots, x_n^{(i)}) \subseteq \mathbb{F}^n$ for $i = 1, \ldots, k$. Then the user can encrypt each feature of the data points and the corresponding labels based on the HE scheme. At the same time, the user randomly initializes model parameters (weights $\vec{w} = (w_1, \ldots, w_n)$, intercept b and comparison parameter 0.5, etc) and encrypts them for private Logistic Regression training process. The user can get $c_{\vec{x}^{(i)}}$ $(i = 1, \ldots, k)$, $(c_{label^{(i)}})$ $(i = 1, \ldots, k)$, $c_{\vec{w}}$, c_b, $c_{0.5}$, where $c_{\vec{x}^{(i)}} = (c_{x_1^{(i)}}, \ldots, c_{x_n^{(i)}})$ and $c_{\vec{w}} = (c_{w_1}, \ldots, c_{w_n})$. Then, he sends these encrypted data and configuration to cloud server, and the cloud server trains model parameters over ciphertext-data without revealing any secret information.

5.1.1. Private logistic regression training

In training process, we use stochastic gradient descent method and cross entropy loss function, then update model parameters by the following formulas.

$$c_{z^{(rand)}} = \sum_{j=1}^{n} c_{w_j} \cdot c_{x_j^{(rand)}} + c_b,$$

$$c_{w_j} = c_{w_j} + \eta \cdot (c_{label^{(rand)}} - Sigmoid(c_{z^{(rand)}})) \cdot c_{x_j^{(rand)}},$$

$$c_b = c_b + \eta \cdot (c_{label^{(rand)}} - Sigmoid(c_{z^{(rand)}})),$$

where η denotes the learning rate and *rand* is a random number in $\{1, \ldots, k\}$. Since $c_{z^{(rand)}}, c_{w_j}, c_b$ are 6×6 matrices, which means that we cannot directly compute $Sigmoid(c_{z^{(rand)}})$. Fortunately, we can utilize *Taylor Approximation* to solve *Sigmoid* function values for matrices. As we know,

$$Sigmoid(a) = \frac{1}{\mathbf{1} + e^{-a}} = (\mathbf{1} + e^{-a})^{-1}.$$

If a is a 6×6 matrix in ciphertext space, then "$\mathbf{1}$" is replaced with identity matrix I and e^{-a} can be expressed as matrix linear operations based on *Taylor Approximation*. That is,

$$e^{-a} = I + \frac{a}{1!} + \frac{(-a)^2}{2!} + \frac{(-a)^3}{3!} + \cdots + o(-a) = I + a + \frac{1}{2!} \cdot (-a)^2 + \frac{1}{3!} \cdot (-a)^3 + \cdots + o(-a)$$

Then, e^{-a} is converted to a matrix and $Sigmoid(a) = (I + e^{-a})^{-1}$ is computable. The detailed training process is shown in Algorithm 1.

Algorithm 1 Private logistic regression training.

Require: Ciphertext random weight: $c_{\vec{w}}$
Ciphertext random intercept: c_b
Ciphertext dataset: $c_{\vec{x}^{(1)}}, \ldots, c_{\vec{x}^{(k)}}$
Public matrix: T
Iteration times: *epochs*
Learning rate: η
Ensure: $c_{\vec{w}}$, c_b
for *time* = 1 to *epochs* **do**
for i = 1 to m **do**
$rand \leftarrow Rand(1, k)$
Execute $c_z^{(rand)} = \sum_{j=1}^{n} c_{w_j} \cdot c_{x_j}^{(rand)} + c_b$
$A \leftarrow Taylor\_Approximation(1 + e^{-c_z^{(rand)}})$
Homomorphic subduction in our scheme in Section 4.3
$\#^{-1}$ is equivalent to $Sigmoid(c_z^{(rand)})$
$error \leftarrow c_{label}^{(rand)} - T \cdot A^{-1}$
# *error*, c_{w_j}, c_b are 6×6 matrices
for j = 1 to n **do**
Update $c_{w_j} = c_{w_j} + \eta * error \cdot c_{x_j}^{(rand)}$
end for
Update $c_b = c_b + \eta \cdot error$
end for
end for

5.1.2. Private logistic regression classification

According to the training process in ciphertext environment, the cloud server can get optimized ciphertext parameters. If an user wants to classify data point $\vec{x}$, he encrypts the data point and sends $c_{\vec{x}} = (c_{x_1}, \ldots, c_{x_n})$ to the cloud server. The cloud server utilizes ciphertext model optimized parameters to classify data points and returns corresponding label to the user. In the private classification process, *Taylor Approximation* is also used to obtain the value of *Sigmoid* function. The detailed process is given in Algorithm 2.

Algorithm 2 Private logistic regression classification.

Require: Ciphertext weights : $c_{\vec{w}}$
Ciphertext intercept: c_b
Ciphertext token: $c_{0.5}$
Ciphertext input vector: $c_{\vec{x}}$
Matrix T
Ensure: The classify result
Execute $c_z \leftarrow \sum_{j=0}^{n-1} c_{w_j} \cdot c_{x_j} + c_b$
$A \leftarrow Taylor\_Approximation(1 + e^{-c_z})$
# A^{-1} is equivalent to $Sigmoid(c_z)$
if $det(A^{-1} - T \cdot c_{0.5}) > 1$ **then**
# it means $z > 0.5$
result $\leftarrow 1$
else
# it means $z < 0.5$
result $\leftarrow 0$
end if

5.2. Other private machine learning classification algorithms

In this section, we resume to give four other machine learning classify algorithms: Naive Bayes, Support Vector Machine, Decision Tree and Random Forest, respectively.

5.2.1. Private naive bayes classification

We apply the bags-of-words model to generate feature words vector. The user has prior class probability $P(Y = i)_{i=0,1}$, conditional probability $P(X_j = w_j|Y = i)_{i=0,1,j=1,\ldots,n}$ and words vector $\vec{x}$ ($\vec{x} \in \mathbb{F}^n$). Based on the encryption scheme, the user can get ciphertexts $c_{P(Y=i)_{i=0,1}}$, $c_{P=(X_j=w_j|Y=i)_{i=0,1,j=1,\ldots,n}}$, $c_{\vec{x}}$ and send these ciphertexts to the cloud server. In ciphertexts Naive Bayes classification, the server will return labels to the user without revealing any secret information. In our settings, the cloud server will calculate $c_{P_0} = c_{P(Y=0)} \cdot \prod_{j=1}^{n} c_{P(X_j=w_j|Y=0)} \cdot c_{x_j}$ and $c_{P_1} = c_{P(Y=1)} \cdot \prod_{j=1}^{n} c_{P(X_j=w_j|Y=1)} \cdot c_{x_j}$, where the equations

$$c_{P(Y=0)} \cdot \prod_{j=1}^{n} c_{P(X_j=w_j|Y=0)} \cdot c_{x_j} = c_{P(Y=0)\cdot\prod_{j=1}^{n} P(X_j=w_j|Y=0)\cdot x_j},$$

$$c_{P(Y=1)} \cdot \prod_{j=0}^{n} c_{P(X_j=w_j|Y=1)} \cdot c_{x_j} = c_{P(Y=1)\cdot\prod_{j=1}^{n} P(X_j=w_j|Y=1)\cdot x_j}$$

hold based on the multiplication homomorphism of HE scheme. Furthermore, to avoid result probability's gradual underflowing, we take the natural logarithm of the prior class probability and conditional probability. That is, $P(X_j = w_j|Y = i)$ can be transformed to $\log P(X_j = w_j|Y = i)$, then, $\prod_{j=1}^{n} c_{P(X_j=w_j|Y=i)} \cdot c_{x_j}$ can be written as $\sum_{j=1}^{n} c_{\log P(X_j=w_j|Y=i)} \cdot c_{x_j}$. The private classification is shown in Algorithm 3.

Algorithm 3 Private naive bayes classification.

Require: Ciphertexts $c_{logP(Y=i)_{i=0,1}}$
$c_{logP(X_j=w_j|Y=i)_{j=1,\cdots,n,i=0,1}}$
Bag-of-words ciphertext input vector $c_{\vec{x}}$
Public matrix T
Ensure: The classify result.
Execute $c_{P_0} = c_{logP(Y=0)} + \sum_{j=1}^{n} c_{logP(X_j=w_j|Y=0)} \cdot c_{x_j}$
Execute $c_{P_1} = c_{logP(Y=1)} + \sum_{j=1}^{n} c_{logP(X_j=w_j|Y=1)} \cdot c_{x_j}$
if $det(c_{P_1} - T \cdot c_{P_0}) > 0$ **then**
# It means $P_1 > P_0$ and the label is 1
result $\leftarrow 1$
else
# It means $P_0 > P_1$ and the label is 0
result $\leftarrow 0$
end if

5.2.2. Private support vector machine classification

Let $\vec{w}$ be a weight vector, $\vec{x}$ be the input feature vector and b denote intercept, where $\vec{w}, \vec{x} \in \mathbb{F}^n$, $b \in \mathbb{F}$. Based on the encryption scheme, the user can get ciphertexts $c_{\vec{w}}$, $c_{\vec{x}}$, c_b and send them to the cloud server. In the private Support Vector Machine, the cloud server returns labels by classifying in ciphertext space. In detailed, this server calculates $c_z = \langle c_{\vec{w}}, c_{\vec{x}} \rangle$ based on the definitions of homomorphic addition and multiplication operations for ciphertexts matrices, hence c_z is a 6×6 matrix. Then, in the classification, sign function $sign(c_z)$ can be replaced with $det(c_z)$. If $det(c_z) > 0$, it means $sign(c_z) > 0$ and the label is 1, otherwise, the label is 0. The detailed process is shown in Algorithm 4.

Algorithm 4 Private support vector machine classification.

Require: Ciphertext weights: $c_{\vec{w}}$
Ciphertext intercept: c_b
Ciphertext input vector: $c_{\vec{x}}$
Ensure: The classify result
Execute $c_z = \sum_{j=1}^{n} c_{w_j} \cdot c_{x_j} + c_b$
if $det(c_z) > 0$ **then**
# It means $z > 0$ and the label is 1
result $\leftarrow 1$
else
# It means $z < 0$ and the label is 0
result $\leftarrow 0$
end if

5.2.3. Private decision tree classification

The user carries a decision tree model and inputs a feature vector $\vec{x}$. Based on the encryption scheme, the user can encrypt each feature splitting value in a tree model and send the corresponding ciphertexts c_{tree}, $c_{\vec{x}}$ to the cloud server. In ciphertexts classification, the cloud server will return labels for given c_{tree}, $c_{\vec{x}}$. In detailed, the cloud server will recursively calculate $det(c_{x_{split\_index}} - T \cdot c_{split\_value})$ to decide to enter left child tree or right child tree until arriving the leaf node, and then the cloud server returns the node label. The detailed process is shown in Algorithm 5, where $tree\_predict()$ is the

Algorithm 5 Private decision tree classifcation.

Require: Ciphertext model c_{tree}
Ciphertext input vector $c_{\vec{x}}$
Ensure: The classify result
if c_{tree} is a node **then**
result $\leftarrow c_{tree\_label}$
else
Execute $token = det(c_{x_{feature\_index}} - T \cdot c_{split\_value})$
end if
if $token < 1$ **then**
# It means $x_{feature\_index} < split\_value$ and enters $leftsubtree$.
result $\leftarrow tree\_predict(c_{tree\_leftsubtree}, c_{\vec{x}})$
else
# It means $x_{feature\_index} > split\_value$ and enters $rightsubtree$.
result $\leftarrow tree\_predict(c_{tree\_rightsubtree}, c_{\vec{x}})$
end if

private Decision Tree classification function that is called recursively.

5.2.4. Private random forest classification

The user has a random forest model that consists of several decision tree models. Like the above private decision tree classification, the user can encrypt each decision tree model in the random forest model and send the ciphertexts c_{random_forest}, $c_{\vec{x}}$ to the cloud server. In the private Random Forest classification, the cloud server utilizes private decision tree classification into each tree model and records labels from the tree model. Finally, the cloud server returns a most common label. The detailed process is presented in Algorithm 6.

Algorithm 6 Private random forest classification.

Require: Ciphertext model: c_{random_forest}
Ciphertext input vector: $c_{\vec{x}}$
Tree number : n_trees
Ensure: The classify result
for i = 1 to n_trees **do**
Execute label = $tree\_predict(c_{random\_forest[i]}, c_{\vec{x}})$
Update $Votes[label] = Votes[label] + 1$
end for
# Get the most commol label as final result
Execute result = $\arg\max_i Votes[i]$

Table 1
Computation cost of HE scheme.

Basic operations	Enc	Dec	H-Add	H-Sub	H-Mul	H-Div	H-Comp
Matrix-Mul	2	2	0	1	1	2	1
Determinant	0	0	0	0	0	0	1

6. Performance discussion

6.1. Efficiency of the HE scheme

Now we present the efficiency (computation cost) of the HE scheme in Section 4.3. Table 1 shows the numbers of basic operations in the scheme.

- Computation cost for encryption (Enc): In the encryption stage, the user chooses random numbers from $\mathbb{F}$ and constructs message matrix M for m. Then, the user carries the symmetric key H to compute ciphertext $C = H \cdot M \cdot H^{-1}$, where M, H, C are 6×6 matrices. Thus, the user mainly runs two matrix-multiplication operations over $\mathbb{F}$.
- Computation cost for decryption (Dec): The user computes $M = H^{-1} \cdot C \cdot H$ to recover the message m. Therefore, he also does two matrix-multiplication operations in the decryption stage.
- Computation cost for homomorphic addition operation (H-Add): Since $C \oplus \bar{C} = C + \bar{C}$, then the computation cost of this operation can be negligible.
- Computation cost for homomorphic subtraction operation (H-Sub): Since $C \ominus \bar{C} = C - T \cdot \bar{C}$, then the ciphertext homomorphic subtraction operation needs one matrix-multiplication.
- Computation cost for homomorphic multiplication operation (H-Mul): Since $C \otimes \bar{C} = C \cdot \bar{C}$, then this operation needs one matrix-multiplication.
- Computation cost for homomorphic division operation (H-Div): Since $C \oslash \bar{C} = C \cdot \bar{C}^{-1}$, the homomorphic division operation needs two matrix-multiplication operations, where the computation cost for calculating $\bar{C}^{-1}$ is approximately equal to that of one matrix-multiplication.
- Computation cost for homomorphic comparison (H-Comp): To compare the messages m and $\bar{m}$ by implementing homomorphic ciphertexts comparison, anyone can compute $t = det(C - T \cdot \bar{C})$, where $C, \bar{C}$ are ciphertexts of m and $\bar{m}$. Thus, the comparison needs one matrix multiplication (Mul) and one determinant calculation.

6.2. HE's homomorphism and security

Our proposed HE scheme is only one-way secure, but it has an obvious advantage that the scheme supports basic homomorphic operations for ciphertexts and homomorphic comparison. At the same time, the "secondhand" ciphertexts obtained from homomorphic operations still keep the form of "fresh" ciphertexts, and any "secondhand" ciphertext can be correctly decrypted. Thus, the number of homomorphic operation layers is not limited. Hence, due to the strong homomorphism, the HE scheme can be well adapted to private deep learning environment with ciphertexts's deep iterations.

In addition, to better illustrate the security of the encryption scheme in Section 4.3, we will use Quantile-quantile plot (QQ-plot) [29] to show the random distribution for each component of ciphertext matrix c_{ij} $(1 \le i, j \le 6)$. Since QQ-plot is a graphic method to compare two probability distributions by drawing quantiles. If the two distributions are similar, then QQ graph is approximately located on line $y = x$. Since the real number is an infinite number field, the amount of quantiles cannot be displayed in the graph. Then, we set $\mathbb{F}$ is a finite field Z_p, where let $p = 307$ is a prime. Here, we only show the distribution of one component c_{11} for ciphertext C, let s be the comparison object for c_{11}, where s is randomly sampled from Z_p. Hence, the message space is Z_p, and M, C are 6×6 matrices over Z_p. Let the encrypted message is $m = 1$. We randomly select numbers and construct matrix M for 10,000 times, then get the corresponding ciphertext matrices and

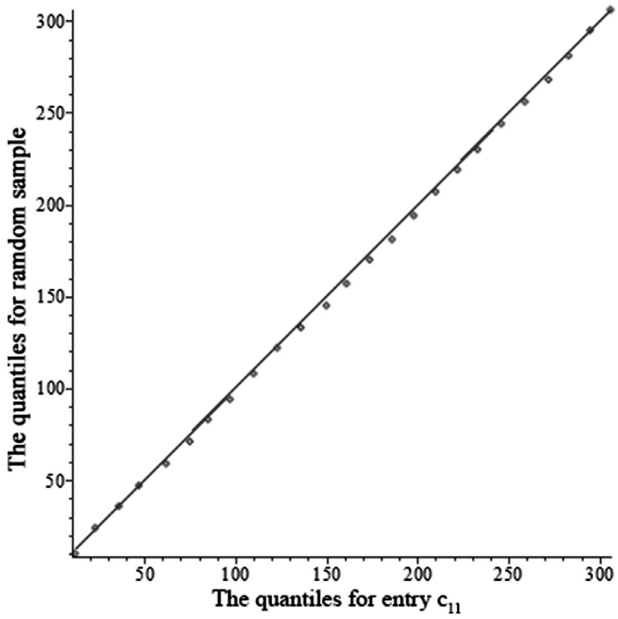

Fig. 2. QQ-plot for ciphertext's random distribution.

$c_{11} \in Z_p$. Suppose that sequence X is obtained from c_{11} and sequence Y is sampled for s. The distribution image is given in Fig. 2, the figure shows that the distributions of c_{11} and random sample s are indistinguishable.

7. Conclusion

This paper presents a novel homomorphic encryption framework over a non-abelian ring, where its one-way security depends on the Conjugacy Search Problem. Based on this framework, a homomorphic encryption was proposed over matrix-ring. Then, we define the basic homomorphic operations in ciphertexts space. This homomorphic encryption scheme supports real number encryption and ciphertexts homomorphic comparison. Finally, by using Taylor approximation to solve some non-linear function's computation for ciphertexts, we achieve privacy preservation for machine learning training and classification based on the proposed encryption scheme. In the future work, we will improve the efficiency of our encryption scheme, especially reducing the ciphertext expansion rate. In addition, we will consider the privacy preservation for machine learning in more complex environment [30] and study other new methods [31,32].

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Jing Li: Methodology, Formal analysis, Writing - original draft, Writing - review & editing, Data curation. **Xiaohui Kuang:** Conceptualization, Project administration, Supervision, Funding acquisition, Resources. **Shujie Lin:** Software, Validation, Formal analysis. **Xu Ma:** Visualization, Investigation. **Yi Tang:** Formal analysis, Investigation.

Acknowledgements

This work was supported by National Natural Science Foundation of China for Outstanding Youth Foundation (No. 61722203), National Natural Science Foundation of China for Joint Fund Project (No. U1936218) and Open Research Project of State Key Laboratory of Integrated Service Networks (No. ISN20-10).

References

[1] G. Sun, T. Cui, J. Yong, J. Shen, S. Chen, MLaaS: a cloud-based system for delivering adaptive micro learning in mobile MOOC learning, IEEE Trans. Serv. Comput. 11 (2) (2015) 292–305.
[2] H. Kim, J. Kim, Y. Kim, I. Kim, K.J. Kim, Design of network threat detection and classification based on machine learning on cloud computing, Cluster Comput. 22 (1) (2019) 2341–2350.

[3] B. Hitaj, G. Ateniese, F. Perez-Cruz, Deep models under the GAN: information leakage from collaborative deep learning, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2017, pp. 603–618.
[4] H. Xiao, B. Biggio, B. Nelson, H. Xiao, C. Eckert, F. Roli, Support vector machines under adversarial label contamination, Neurocomputing, 160 (2015) 53–62.
[5] S. Mei, X. Zhu, Using machine teaching to identify optimal training-set attacks on machine learners, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
[6] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, K. Chen, Multi-key privacy-preserving deep learning in cloud computing, Future Gener. Comput. Syst. 74 (2017) 76–85.
[7] T. Li, Z. Huang, P. Li, Z. Liu, C. Jia, Outsourced privacy-preserving classification service over encrypted data, J. Netw. Comput. Appl. 106 (2018) 100–110.
[8] T. Li, J. Li, Z. Liu, P. Li, C. Jia, Differentially private naive bayes learning over multiple data sources, Information Sciences (2018). S0020025518301415
[9] A. Bansal, T. Chen, S. Zhong, Privacy preserving back-propagation neural network learning over arbitrarily partitioned data, Neural Comput. Appl. 20 (1) (2011) 143–150.
[10] T. Graepel, K. Lauter, M. Naehrig, Ml confidential: machine learning on encrypted data, in: International Conference on Information Security and Cryptology, Springer, 2012, pp. 1–21.
[11] Y. Rahulamathavan, R.C.-W. Phan, S. Veluru, K. Cumanan, M. Rajarajan, Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud, IEEE Trans. Dependable Secure Comput. 11 (5) (2013) 467–479.
[12] R. Bost, R.A. Popa, S. Tu, S. Goldwasser, Machine learning classification over encrypted data., in: NDSS, vol. 4324, 2015, p. 4325.
[13] W.K. Wong, D.W.-l. Cheung, B. Kao, N. Mamoulis, Secure kNN computation on encrypted databases, in: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, ACM, 2009, pp. 139–152.
[14] L.J. Aslett, P.M. Esperança, C.C. Holmes, Encrypted statistical machine learning: new privacy preserving methods, arXiv:1508.06845 (2015).
[15] X. Liu, R. Lu, J. Ma, L. Chen, B. Qin, Privacy-preserving patient-centric clinical decision support system on naive bayesian classification, IEEE J. Biomed. Health Inform. 20 (2) (2015) 655–668.
[16] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, J. Wernsing, CryptoNets: applying neural networks to encrypted data with high throughput and accuracy, in: International Conference on Machine Learning, 2016, pp. 201–210.
[17] M. Baryalai, J. Jang-Jaccard, D. Liu, Towards privacy-preserving classification in neural networks, in: 2016 14th Annual Conference on Privacy, Security and Trust (PST), IEEE, 2016, pp. 392–399.
[18] X. Li, Y. Zhu, J. Wang, Z. Liu, Y. Liu, M. Zhang, On the soundness and security of privacy-preserving SVM for outsourcing data classification, IEEE Trans. Dependable Secure Comput. 15 (5) (2018) 906–912.
[19] L.T. Phong, Y. Aono, T. Hayashi, L. Wang, S. Moriai, Privacy-preserving deep learning via additively homomorphic encryption, IEEE Trans. Inform. Forensics Secur. 13 (5) (2018) 1333–1345.
[20] M. Kim, Y. Song, S. Wang, Y. Xia, X. Jiang, Secure logistic regression based on homomorphic encryption: design and evaluation, JMIR Med. Inform. 6 (2) (2018) e19.
[21] X. Liu, K.-K.R. Choo, R.H. Deng, R. Lu, J. Weng, Efficient and privacy-preserving outsourced calculation of rational numbers, IEEE Trans. Dependable Secure Comput. 15 (1) (2016) 27–39.
[22] X. Sun, P. Zhang, J.K. Liu, J. Yu, W. Xie, Private machine learning classification based on fully homomorphic encryption, IEEE Trans. Emerging Top. Comput. (2018) 1.
[23] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM 21 (2) (1978) 120–126.
[24] C. Gentry, et al., Fully homomorphic encryption using ideal lattices., in: Stoc, vol. 9, 2009, pp. 169–178.
[25] L. Gu, S. Zheng, Conjugacy systems based on nonabelian factorization problems and their applications in cryptography, J. Appl. Math. 2 (2014) (2014) 1–10.
[26] L. Gu, L. Wang, K. Ota, M. Dong, Z. Cao, Y. Yang, New public key cryptosystems based on non-abelian factorization problems, Secur. Commun. Netw. 6 (7) (2013) 912–922.
[27] E. Begelfor, S.D. Miller, R. Venkatesan, Non-abelian analogs of lattice rounding, Groups Complexity Cryptol. 7 (2) (2015) 117–133.
[28] P. Harrington, Machine Learning in Action, Manning Publications Co., 2012.
[29] G. Blom, Statistical Estimates and Transformed Beta-Variables, Almqvist & Wiksell, 1958 Ph.D. thesis.
[30] X. Wang, J. Li, X. Kuang, Y.-a. Tan, J. Li, The security of machine learning in an adversarial setting: a survey, J. Parallel Distrib. Comput. 130 (2019) 12–23.
[31] T. Li, X. Li, X. Zhong, N. Jiang, C.-z. Gao, Communication-efficient outsourced privacy-preserving classification service using trusted processor, Inf. Sci. 505 (2019) 473–486.
[32] A. Hassan, R. Hamza, H. Yan, P. Li, An efficient outsourced privacy preserving machine learning scheme with public verifiability, IEEE Access 7 (2019) 146322–146330.