Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 202 (2022) 324–329

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Identification, Information and Knowledge in the internet of Things, 2021

Implementing Linear Regression with Homomorphic Encryption

Bijiao Chen[a,b], Xianghan Zheng[a,b]

[a] *College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China*
[b] *The Academy of Digital China, Fuzhou University, Fuzhou 350108, China*

Abstract

Benefiting from the computing power and storage power of cloud computing, machine learning tasks usually choose to upload data and models to a third-party server. However, third-party servers may face data privacy leaks in the process of data acquisition, storage, and use, especially in the fields of finance, medical treatment, and biometrics. Homomorphic encryption technology supports ciphertext calculation without decryption, and can be used for machine learning model training and prediction in the ciphertext domain. We use the homomorphic encryption library SEAL to reconstruct the linear regression protocol, use six data sets to conduct experiments, and analyze its performance and security.

Keywords: Homomorphic Encryption; Linear Regression; Cloud Computing.

1. Introduction

Machine learning models often appear in the medical, financial, automation and biological fields. Cloud-based machine learning can improve training capabilities and reduce costs. For example, the Epidemiology Research Center hopes to train high-precision epidemic prediction models through the cloud with high-performance storage and computing capabilities to provide patients with appropriate treatment recommendations, reduce hospital service costs, and improve diagnosis quality. However, when machine learning tasks are delegated to a third-party cloud server, there is a risk of privacy leakage. Because the intermediate results or data of the calculation may involve user-related information, such as financial information, personal pictures and other sensitive data. Therefore, third-party cloud servers need to perform machine learning without violating privacy. The existing secure multi-party computing protocol is

* Xianghan Zheng. Tel.: +86-150-8002-5921.
E-mail address: xianghan.zheng@fzu.edu.cn

10.1016/j.procs.2022.04.044

used to solve this problem. However, this solution has the problems of high deployment cost and multiple parties needing to stay online for all computing time.

Sensitive data involves privacy and is stored in a third-party cloud server. There is a possibility of leakage. The usual solution is to encrypt the data and upload it to an untrusted third party. However, it cannot realize the special features on the ciphertext domain. calculate. Compared with other security technologies, a major feature of homomorphic encryption is the computability of ciphertexts.

Although there are several effective implementations of homomorphic encryption, there are relatively few machine learning algorithms implemented using homomorphic encryption. Therefore, we want to explore the use of homomorphic encryption to implement secure machine learning algorithms. In this work, we use homomorphic encryption to implement a secure linear regression machine learning algorithm, reconstruct a linear regression protocol, and implement regression training and prediction under the premise of protecting data privacy.

The main contributions of this paper can be summarized as follows:

- A linear regression scheme based on homomorphic encryption is proposed, and the linear regression protocol is reconstructed to realize linear regression training and prediction that protects data privacy.
- Our experimental results on six sets of data show that the accuracy of the algorithm in the ciphertext domain is close to that of the plaintext domain. Security analysis shows that this scheme can guarantee data privacy.

2. Related Work

2.1. Homomorphic Encryption

Gentry designed the first fully homomorphic encryption scheme using lattice-based encryption for the first time in 2009. Gentry subsequently added bootstrapping, turning a certain homomorphic encryption scheme booted into a fully homomorphic encryption scheme [1]. Many researchers are working on the design of second-generation schemes [2] [3]. These new schemes rely on learning problems with errors. During this period, two optimizations were discovered: ciphertext packing [4] and analog-to-digital conversion [5]. Later, these two improvements were combined into a single program, the BGV program [6]. The BFV scheme uses the Chinese remainder theorem to express and process the large coefficients of the ciphertext polynomial, and the noise introduced is lower [7]. The previous scheme designs were all based on integers, while the third-generation scheme design was mainly aimed at the performance optimization of bootstrapping and the homomorphic encryption that supports floating-point numbers. Lee et al. proposed to improve the bootstrapping process and optimize performance [8]. Cheon et al. proposed a CKKS scheme that supports real and complex numbers [9].

2.2. HE Linear Regression

Linear regression can use the least squares method to find the linear relationship between the dependent variable and the independent variable. When there is only one independent variable, it is called simple regression, and when there are two or more independent variables, it is called multiple linear regression. Regression analysis focuses on the distribution relationship between the independent variable and the dependent variable. However, when the sample has many features, it will take more time to solve the parameters one by one using the algebraic derivation method. At this time, the matrix method can be introduced to solve the problem to speed up the solution, that is, the normal equation of regression is solved. Various previous studies have proposed to train a linear regression model from homomorphic ciphertext data, such as using matrix inversion to obtain a linear regression model. However, their proposed method is only suitable for data with small dimensions (ie less than 6) [10]. The Cholesky decomposition in the current scheme requires a square root, which cannot be achieved by SEAL at present [11]. Yoshinori et al. use gradient descent, but the solution is an approximate solution [12]. Nikolaenko et al. used an iterative matrix inversion of an undivided variable, but relied on pre-known assumptions [13]. Kikuchi et al. used Paillier additive homomorphism to achieve multiple regression of stroke medical data sets [14]. However, the currently selected homomorphic scheme only supports one operation scheme.

3. System and Threat Model

In this section, we provide a brief overview of the system and consider the use of homomorphic encryption to solve machine learning algorithm problems. We provide an overview of the functional implementation of the proposed privacy protection linear regression scheme. Finally, we describe the threat model of the problem.

3.1. System Model

This scheme includes two sub-processes: linear regression training for privacy protection and linear regression prediction for privacy protection. Whether in the training or prediction phase, the privacy data involved (such as training data, prediction data, model parameters, prediction results, and intermediate calculation results) cannot be stolen.

- Privacy-preserving linear regression training: The solution is to solve the regular equation of linear regression, without iterative training, and the algorithm trains the model based on the training data set. In order to achieve data privacy, the training process is carried out in the ciphertext domain. The program will also give a safe normal equation algorithm for linear regression. When the linear regression training of privacy protection is completed, the algorithm will return the model parameters of the ciphertext.
- Privacy protection linear regression prediction: After receiving the user's ciphertext prediction data set and ciphertext model parameters, the cloud performs privacy protection linear regression prediction calculations on the server. The final ciphertext model prediction result is returned, and only the user is allowed to decrypt the corresponding plaintext value by himself.

3.2. Threat Model

Assume that in the scenario, all entities are honest and curious. Usually, they will receive the message and respond to the message in full accordance with the provisions of the interactive protocol. Therefore, they can honestly perform the system's corresponding protocol calculations, but once they have the opportunity in the process, they will not refuse to learn anything that is beneficial to them, that is, try to obtain data from other entities. Define adversary $\mathcal{A}^*$, which has the following abilities:

(1) $\mathcal{A}^*$ may eavesdrop on all transmission channels in the system to obtain encrypted data.

(2) $\mathcal{A}^*$ may buy the cloud and try to derive the corresponding plaintext data from the ciphertext training data and other ciphertext intermediate results.

(3) $\mathcal{A}^*$ may buy the cloud and try to derive the corresponding plaintext data from the sent ciphertext prediction data, ciphertext model parameters, and corresponding calculation logic.

4. Protocol Description

The input of the linear regression training protocol is the training data set, and the output is the parameters of the model. In the preparation and processing stage of the Client, the data set is standardized and normalized, and the ciphertext data set is obtained through the homomorphic encryption algorithm. Assuming that the training data set is D_{T-N}, the encrypted ciphertext training data set is $[D_{T-N}]_{pk}$. In order to simplify the representation, pk will be omitted in the following description, that is, $[\cdot]_{pk}$ is represented as $[\cdot]$.

At the stage of the ciphertext regression protocol in the Cloud, the model parameters are updated according to the agreement of the regression protocol. Perform linear regression under ciphertext according to the specified calculation logic. However, at present, the solution of $([X^T][X])^{-1}$ cannot be solved, that is, the ciphertext inverse operation cannot be achieved. Therefore, first, keep cycling, use each ciphertext data sample to update and calculate the $[X^T X]$ and $[X^T Y]$ under the ciphertext.

After the ciphertext linear regression protocol is completed, the Cloud will obtain the ciphertext $[X^T X]$ and $[X^T Y]$. The Cloud sends the ciphertext result to the user who has the private key. The user decrypts it with the corresponding

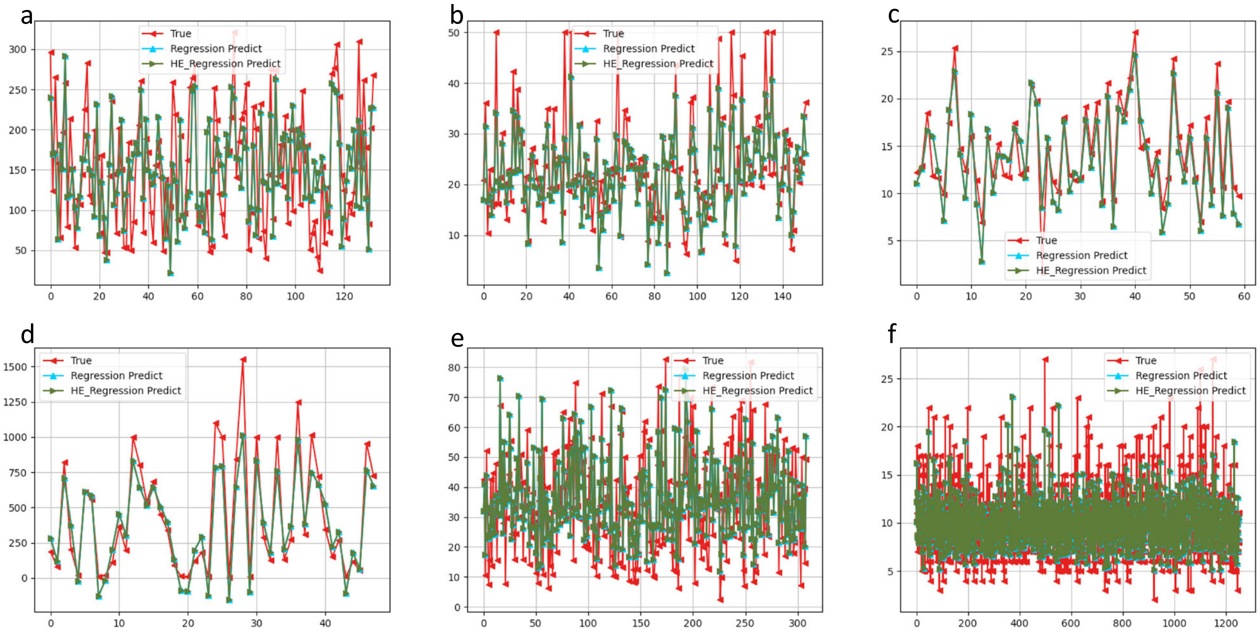

Fig. 1. (a) diabetes; (b) boston; (c) advertising; (d) fish; (e) ccsds; (f) abalone.

private key, and obtains $X^T X$ and $X^T Y$ in the plaintext domain. Execute $\Theta = (X^T X)^{-1} X^T Y$ to obtain the model parameter Θ.

5. Implementations and Analysis

5.1. Performance Analysis

We use six datasets to evaluate the linear regression scheme based on homomorphic encryption. Fig. 1 shows the comparison curve between the predicted value and the label value of Homomorphic Encryption Regression and Regression. From the prediction results, it is found that the two almost overlap, and the prediction results are very similar, which indicates that good results have been achieved.

As shown in Table 1, the four regression evaluation indexes are almost the same, indicating Homomorphic Encryption Regression and Regression achieve relatively consistent results. From the results of R-squared, the regression effect of solving the normal equation of linear regression is not very ideal.

After removing the constraints of data sample number n and sample attribute p, the average calculation time of encryption, training and decryption of each dataset can be found that the calculation time of single encryption, decryption and training is similar. Through further calculation, the average time of encryption is 21.70 ms, the average time of ciphertext training is 31.737411 ms, and the average time of decryption solution is 5.817356 ms, as shown in Table 2.

5.2. Safety Analysis

In the encryption phase of the scheme, when the homomorphic encryption algorithm is secure and the Cloud is not capable to obtain user's private key, it is not capable to recover the plaintext data from the ciphertext data, so the plaintext data is secure. Of course, it should be assumed that the private key used for decryption is stored privately by the user, and the Cloud would not obtain the key information. Therefore, we need to pay attention to the security of approximate homomorphic encryption itself.

Table 1. Evaluation index of linear regression scheme.

Dataset	MSE	RMSE	MAE	R Squared
diabetes	3400.930923	58.317501	48.02731253	0.396233196
homomorphic encryption diabetes	3400.930928	58.31750104	48.02731258	0.396233196
boston	27.64640196	5.257984591	3.541550836	0.719556074
homomorphic encryption boston	27.64640196	5.257984591	3.541550836	0.719556074
advertising	4.4454982	2.108435012	1.746329629	0.817133359
homomorphic encryption advertising	4.445498202	2.108435012	1.746329629	0.817133359
fish	22639.59072	150.464583	118.9595453	0.86240084
homomorphic encryption fish	22639.59076	150.4645831	118.9595456	0.86240084
ccsds	104.8272235	10.23851667	8.062372072	0.613556445
homomorphic encryption ccsds	104.8272235	10.23851667	8.062372071	0.613556445
abalone	5.295819705	2.301264805	1.667359685	0.532935949
homomorphic encryption abalone	5.295819704	2.301264805	1.667359685	0.53293595

Table 2. Average time of linear regression scheme.

Dataset	Encryption(*ms*)	Ciphertext training(*ms*)	Decrypt(*ms*)
homomorphic encryption diabetes	22.055174	33.368956	5.107749
homomorphic encryption boston	18.732836	26.732423	3.969222
homomorphic encryption advertising	22.666514	34.105915	5.311965
homomorphic encryption fish	25.396422	39.319192	6.335934
homomorphic encryption ccsds	22.152491	28.243144	4.23428
homomorphic encryption abalone	19.198156	28.654834	9.944977

The difficulty of the homomorphic encryption problem can be summarized as Ring Learning with Errors Problem (RLWE). The difficulty of the RLWE problem is based on the difficulty of the underlying lattice. Therefore, the difficulty of the RLWE problem guarantees the security of nearly homomorphic encryption.

6. Conclusion

The data collection and calculation of machine learning tasks in the cloud environment will cause concerns about the leakage of sensitive data. We implement a secure linear regression scheme based on homomorphic encryption technology and reconstruct the regression protocol. Through a large number of analyses, experiments and evaluations, we can conclude that it is feasible and effective to apply homomorphic encryption to machine learning without affecting accuracy. It is suitable for application scenarios with high security requirements.

References

[1] Gentry, Craig, Shai Halevi, and Nigel P. Smart. (2012) "Better bootstrapping in fully homomorphic encryption." *International Workshop on Public Key Cryptography* **2012** : 1–16.

[2] Brakerski, Zvika. (2012) "Fully homomorphic encryption without modulus switching from classical GapSVP." *Annual Cryptology Conference* **2012** : 868–886.

[3] Gentry, Craig, Amit Sahai, and Brent Waters. (2013) "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based." *Annual Cryptology Conference* **2013** : 75–92.

[4] Brakerski, Zvika, Craig Gentry, and Shai Halevi. (2013) "Packed ciphertexts in LWE-based homomorphic encryption." *International Workshop on Public Key Cryptography* **2013** : 1–13.

[5] Coron, Jean-Sébastien, David Naccache, and Mehdi Tibouchi. (2012) "Public key compression and modulus switching for fully homomorphic encryption over the integers", *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, **2012** : 446–464.

[6] Brakerski, Zvika, Craig Gentry, and Vinod Vaikuntanathan. (2014) "(Leveled) fully homomorphic encryption without bootstrapping", *ACM Transactions on Computation Theory (TOCT)*, **6** (3): 1–36.

[7] Halevi, Shai, Yuriy Polyakov, and Victor Shoup. (2019) "An improved RNS variant of the BFV homomorphic encryption scheme", *Cryptographers' Track at the RSA Conference*, **2019** : 83–105.

[8] Lee, Jae Woo, et al. (2011) “0 to 10k in 20 seconds: Bootstrapping Large-scale DHT networks”, *2011 IEEE International Conference on Communications (ICC)*, **2011** : 1–6.

[9] Cheon, Jung Hee, et al. (2017) “Homomorphic encryption for arithmetic of approximate numbers”, *International Conference on the Theory and Application of Cryptology and Information Security*, **2017** : 409–437.

[10] Wu, David, and Jacob Haven. (2012) “Using homomorphic encryption for large scale statistical analysis”, *FHE-SI-Report*, **2012**.

[11] Nikolaenko, Valeria, et al. (2013) “Privacy-preserving ridge regression on hundreds of millions of records”, *2013 IEEE Symposium on Security and Privacy*, **2013** : 334–348.

[12] Aono, Yoshinori, et al. (2015) “Fast and Secure Linear Regression and Biometric Authentication with Security Update”, *IACR Cryptol. ePrint Arch*, **2015** : 692.

[13] Lu, Wenjie, Shohei Kawasaki, and Jun Sakuma. (2016) “Using Fully Homomorphic Encryption for Statistical Analysis of Categorical”, *Ordinal and Numerical Data*, **2016** : 1163.

[14] Kikuchi, Hiroaki, et al. (2018) “Privacy-preserving multiple linear regression of vertically partitioned real medical datasets”, *Journal of Information Processing*, **26** : 638–647.