Parallel Linear Regression on Encrypted Data

Toufique Morshed
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada
morshed@cs.umanitoba.ca

Dima Alhadidi
Faculty of Computer Science
University of New Brunswick
Fredericton, New Brunswick, Canada
dalhadid@unb.ca

Noman Mohammed
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada
noman@cs.umanitoba.ca

***Abstract*—In recent years, the advent of machine learning models on private data has been remarkable. However, incorporating machine learning techniques to healthcare data is pretty challenging due to the privacy issues of sensitive data which restricts data sharing in plaintext. Ensuring the privacy of individuals in healthcare datasets while constructing a machine learning model is a challenging research problem today. This paper proposes an approximate mathematical model utilizing linear regression on homomorphically encrypted data to predict the disease association of an individual. Furthermore, as these encryption schemes are not efficient considering computation time, we incorporate the multi-core parallelism to make the framework realistic. We experimentally evaluate the performance of the proposed methods and report on the experimental results.**

***Index Terms*—Parallel Homomorphic Encryption, Secure Linear Regression, Secure Machine Learning.**

I. Introduction

Different machine learning techniques are used to build classification or prediction models for the unknown samples based on the given samples. Some machine learning techniques can be applied to healthcare data. However, building machine learning classifier/prediction models for healthcare data becomes complex due to privacy issues as this kind of data is very sensitive. In this paper, we considered Parkinson's disease (*PD*). It is a neurodegenerative disease which affects the brain cells. An early detection can pacify the disease growth and reduce the cost. In the UK, the annual cost has been estimated to be between 449 million and 3.3 billion pounds [1].

As mentioned earlier, computing machine learning models for healthcare data is quite challenging for two reasons. First, we should protect the privacy of patients when sharing their data with a third party (e.g., for research). Second, we should protect the privacy of patients when conducting the required computation (i.e., developing prediction model). The first challenge is tackled with encryption. The resultant data after encryption is protected from any adversarial attack such that the data can be shared for further investigation with any research organization. The second challenge is to develop a prediction model using encrypted data.

In this paper, we protect the privacy of Parkinson's patients while sharing and constructing a prediction model on PD patient data that has been outsourced to a cloud server. The main contributions of this paper are summarized as follow:

- We develop a linear regression framework to generate a model that can classify the unknown samples based on the training samples. For generating the linear regression model, we use the gradient descent algorithm [7] for the convergence of the model. The proposed framework has the following features:
  - The framework considers encrypted sensitive voice samples of Parkinson's patients. The samples are encrypted using the Homomorphic Encryption (HE) scheme proposed by Fan and Vercauteren (FV) [8]. This scheme exhibits both additive and multiplicative homomorphism.
  - The framework approximates the values using the gradient descent algorithm. The approximation is done due to the limitations of the HE scheme [8].
- We demonstrate the effectiveness of our approach through experiments and evaluations. These results clearly exhibit similar performance for both plaintext and ciphertext. In addition to that, we demonstrate significant improvement in terms of runtime using parallel approach.

II. Related Work

A. Evolution of Homomorphic Encryption (HE)

Homomorphic Encryption (HE) is a form of encryption that allows computation over ciphertexts. Basic RSA encryption scheme [9] preserves multiplicative homomorphism. Paillier *et al.* [10] proposed an encryption scheme in 1999, which has the additive homomorphic property. After the Gentry's breakthrough [11] in 2009, there has been some momentum in this research domain. In the same year, Regev's encryption scheme [12] was published regarding the proof of concept of HE scheme using learning with errors (LWE) and random linear codes. In 2010, Lyubashevsky *et al.* extended learning with error problem over rings [13]. Since then a number of HE schemes have been proposed with the improvement of bootstrapping time [14, 8]. In 2016, Chillotti *et al.* [15] improved HE scheme and brought the bootstrapping time to less than 0.1 second.

B. HE and Machine learning

Hall *et al.* [2] demonstrated that a fully secure approach to linear regression based on the HE is practical for use

978-1-5386-7493-2/18/$31.00 © 2018 IEEE

TABLE I
SUMMARY OF RELATED WORKS OF MACHINE LEARNING ALGORITHMS ON ENCRYPTED DATA

Name	Year	Machine Learning Algorithm	Homomorphic Encryption	Data Model	Garbled Circuit	Parallel
Hall *et al.* [2]	2011	Regression Analysis	Paillier	Federated	×	×
Graepel *et al.* [3]	2012	Linear Means Classifier, Fisher's Linear Discriminant Classifier	Levelled	Federated	×	×
Bost *et al.* [4]	2015	Hyperplane Decision, Nave Bayes, and Decision Trees	Fully	Federated	✓	×
Sadat *et al.* [5]	2018	Linear Regression, Logistic Regression	Somewhat	Federated (SGX)	×	×
Jiang *et al.* [6]	2018	Logistic Regression	Somewhat	Federated (SGX)	×	✓
Our work	2018	Linear Regression	Somewhat	Centralized	×	✓

on moderately large datasets shared between several parties (federated model). For their regression analysis, they used Paillier HE scheme [10]. In our work, we considered a non-federated data model where the data is outsourced to the cloud. Later in 2012, Graepel *et al.* [3] showed the possibility of implementing machine learning algorithms on encrypted data where they considered a leveled HE scheme for non-federated data model. Besides, they demonstrated the HE scheme based on the level (depth) of mathematical operations.

In addition to that a number of approaches had been considered for generating classification models using Yao's garbled circuits (GC) [16] (secure two-party computation). In 2015, Bost *et al.* [4] proposed different methods for executing machine learning algorithms on encrypted data using HE and GC schemes. They provided two models for building the classifier, one using the data from an HE scheme and another one using Yao's protocol. Recently in 2016, Gilad *et al.* [17] showed that we can apply neural network on encrypted data. They restricted their work on low degree polynomials due to the complexity and limitation of the considered HE scheme. In 2018, Hesamifard *et al.* proposed a framework for machine learning as service in which they used client-server interaction while computing the model. In addition to that, Sadat *et al.* [5] and Jiang *et al.* [6] proposed their framework for linear and logistic regression using the somewhat HE scheme. However, in their proposal, they also used the secure hardware SGX [18]. On the other hand, in our work, we designed the framework to be non-interactive. A brief summary of related works in the field of machine learning and HE has been illustrated in Table I.

III. BACKGROUND

A. Linear Regression

For a given set of ordered pair (x_i, y_i) where $i = 0, 1, 2, \ldots (n-1)$, simple linear regression model finds a regression line to fit the given data. However, in case of multidimensional data, this regression line becomes a plane and the regression model is called multiple linear regression [7] and the corresponding model is:

$$y_i = w_0 + w_1 x_0 + w_2 x_0 + w_3 x_0 + \ldots + w_{n+1} x_n + \epsilon_i \quad (1)$$

where w_0 is this the bias, w_i is the coefficient of $x_i \epsilon X_i$, and ϵ_i is the amount of error (residual error) while predicting y_i for a given X_i. In vector format, this can be written as $Y = WX^T + \epsilon$.

For having best fitted straight line/plane, ϵ needs to be minimized. To calculate the overall error we will be using the total Residual Sum of Squares (RSS) defined as, $RSS = \sum_{i=0}^{i=n-1} \epsilon_i^2$, which is strongly convex [19]. For reducing the error, we need to find the solution for $\left\|\nabla RSS = \gamma\right\|$, where $\nabla RSS = [\frac{\delta RSS}{\delta w_0}, \frac{\delta RSS}{\delta w_1}, \ldots, \frac{\delta RSS}{\delta w_{n+1}}]^T$ and $\gamma \to 0$, which will provide us with the optimal values of W. In our case, we will be using iterative gradient descent algorithm. Iterative gradient descent algorithm updates the model parameter in each iteration to fit the model using Eqn 2 where α represents the learning rate.

$$w^{t+1} = w^t + \alpha \frac{\sum_{i=0}^{i=n-1} \frac{\delta RSS}{\delta w_i}}{n} \quad (2)$$

B. Homomorphic encryption

In our work, we depend on the HE scheme proposed in [8]. They used Ring Learning with Error (RLWE) for the HE scheme. The main object is the polynomial ring, $R = \mathbb{Z}[x]/f(x)$, where $f(x)$ is an irreducible polynomial of degree d. The distribution on R is defined by χ.

a) Encryption Scheme: Let $q > 0$ is an integer representing the ciphertext modulus. Given the security parameter λ and sampling $\mathbf{x}$ from the distribution χ, the secret key will be s. Given the secret key s, sampling $a \in R_q$, and $e \in \chi$ we generate the following public key, $pk = ([(a.s + e)]_q, a)$ The plaintext space is taken as R_t for some integer $t > 1$ (plaintext modulus). Let $\Delta = \lfloor q/t \rfloor$ and the remainder is $r_q = q \bmod t$. Then we can say, $q = \Delta.t + r_q$. For the encryption of the message $m \epsilon R_t$ we get the ciphertext ct using the following equation used in [8], $ct = ([p_0.u + e_1 + \Delta.m]_q, [p_1.u + e_2]_q)$ where $p_0 = pk[0], p_1 = pk[1]$ and u, e_1, e_2 are samples from χ.

b) Decryption Scheme: To decrypt the ciphertext ct, $[\lfloor t.[c_0 - c_1.sk]_q / q \rfloor]_t$ is used, where $c_0 = ct[0]$ and $c_1 = ct[1]$.

IV. MODELS AND DESIGN GOALS

A. System Model

The system requires only two types of entities:

a) Data Provider: Data provider possesses the data containing sensitive information of patients. It may be an institution, which targets to generate a classifier model on patients' data but lacks the required computational power. For generating a classifier model, data providers encrypt patient data and send it to a cloud server.

b) Cloud Server: Cloud server receives encrypted the data from the data provider and computes on encrypted data to generate a model for classification.

B. Service Flow Model

As the underlying healthcare data is sensitive, the data provider needs to encrypt the plaintext data and send it to the cloud for storage. After storing the encrypted data, the cloud server will run the linear regression algorithm to generate a classification model. The generated model will be encrypted and the cloud server will not have any information about this model. Thus, it satisfies our requirement where the semi-honest adversaries will not have any information not only about data but also about the generated model. After the model generation, the cloud server will send the encrypted model to the data provider and the data provider will decrypt (retrieve) the model. Fig 1 shows the interactions between the data provider and cloud server.

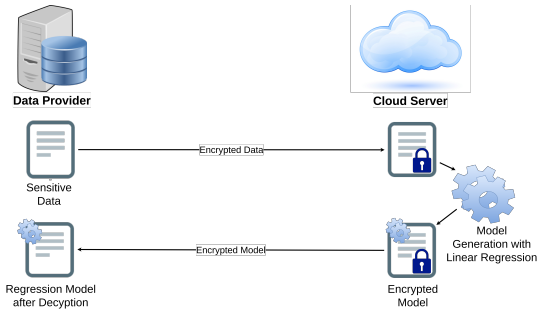

Fig. 1. Data Provider shares the encrypted data with a public cloud server which computes the model on encrypted data and sends back the encrypted model to the data provider. The final decryption is done by the data provider.

C. Threat Model

In our architecture, we will be considering the cloud as a semi-honest adversary. In the semi-honest adversarial model, the cloud server will run the protocol properly but may try to gain more information from the data. Additionally, our main target is to withhold the sensitivity of the sensitive data. Attacks on the encryption scheme like ciphertext attack or chosen plaintext attack lie on the hardness of the mathematical problem Learning With Errors (LWE) [8]. The security parameters of the encryption scheme will be discussed later in section VI-D.

V. Methodology

The main goal of this paper is to develop a predictive model to predict whether an individual is a PD patient or not using encrypted data. We divided the methodology into four parts which are described below:

A. Data Preprocessing

The above mentioned encryption scheme (discussed in Section III-B) has some limitations which are:

a) No floating point calculation,
b) Does not provide divisional homomorphism, and
c) Larger inputs require larger ciphertext modulus size (degree of the ring to achieve $80+$ bits of security) (we further discuss this on section VI-D).

Based on these limitations we preprocess our data as follows:

1) Attributes with larger values: For attributes with larger values, we calculated the variance of the attribute and checked its impact on the accuracy of the regression model. If the attribute has a negative impact on the accuracy, we considered the attribute as an outlier and removed it. If the impact is positive, we reduced the length of the value to two digits. The reason behind the two-digit reduction is to fit within the ciphertext modulus and to avoid overflow.

2) Attributes with smaller values: For attributes with smaller values (less than one), we also calculated the variance of the attribute and checked its relation with accuracy. If the attribute has a positive impact, we scaled it up so that the minimum value of that attribute becomes an integer; otherwise, we removed it.

3) Rounding to nearest integer: The last operation of data preprocessing is to round values to nearest integer. The reason is that the HE [8] does not support floating point values.

B. Approximate Linear Regression Model Generation

The main challenge of this work is conducting the required computation using encrypted data. The dataset is homomorphically encrypted, where the encryption scheme supports only addition and multiplication. Therefore, the linear regression model has to be developed in a way such that it can handle encrypted data. For this purpose, we need to alter the gradient descent (Eqn 2) as it requires division and floating point arithmetic. Due to this, we reformulated the Eqn 2 in the following manner:

$$w^{t+1} = w^t.n.(round(\alpha^{-1})) + \sum_{i=0}^{i=n-1} \frac{\delta RSS}{\delta w_i} \tag{3}$$

Here n is the total number of the training samples and $round(\alpha^{-1})$ is rounding $(1/\alpha)$, which is an integer value. Moreover, as the values get larger in each iteration and as we can not compare encrypted values, it is not feasible to use $\left\|\nabla RSS = \gamma\right\|$, where $\gamma \rightarrow 0$. This forces us to change the terminating condition. In our case, we used a fixed number of iteration as a terminating condition.

C. Classification using OneR algorithm

OneR algorithm [20] (short form of One Rule) is a classification algorithm that generates one rule for each predictor that is available in the dataset. It tries to select the rule that has the smallest total error. In our case, the data provider uses this approach to define a split point that can give the data provider a margin to distinguish between two classes.

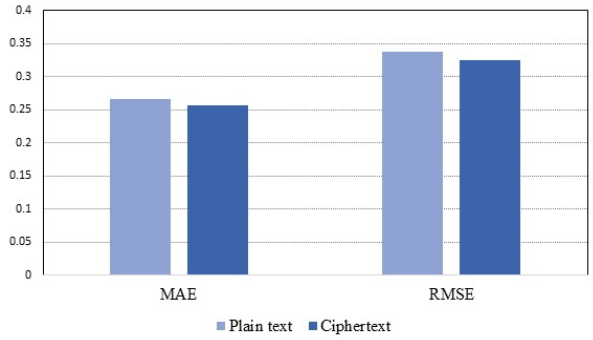

Fig. 2. Error Analysis between plaintext and ciphertext.

D. Parallelization

In the vector format of Eqn 1, we have one matrix-vector multiplication and subtraction for calculating the error. The matrix-vector multiplication can be parallelized by dividing the rows of matrices among different cores and vector addition/subtraction can be parallelized via Single Instruction Multiple Data (SIMD) operations. We took the advantage of parallel computation by partitioning the matrix among different cores (for multiplication) and partitioning the vectors for SIMD operations. Both strategies for multiplication and SIMD operations improved the runtime which has been discussed in Sec VI-E.

VI. Experiments and Analysis

A. Data Description

For computing machine learning models, we used PD dataset available in the UCI repository [21]. The dataset contains 195 instances composed of the voice sample measurements retrieved from 31 people (23 PD patients). The data representation has been shown in Table II.

TABLE II
PARKINSON'S DISEASE DATA REPRESENTATION.

Independent				Dependent
i	*MDVP:Fo(Hz)*	*MDVP:Fhi(Hz)*	...	*Disease Status*
1	119.992	157.302	...	+
2	122.4	148.65	...	+
...	...	...	...	...
195	214.289	260.277	...	−

B. Error Analysis

a) Mean Absolute Error (MAE): The mean absolute error is a quantity to measure the closeness of predictions to the actual outcome. It is calculated as: $MAE = 1/n \sum_{i=1}^{i=n} |e_i|$

b) Root Mean Squared Error (RMSE): Root mean squared error measures the differences between values predicted by a predictor and the values actually observed. It gives the standard deviation of the residuals (i.e., prediction errors), which is calculated as: $RMSE = \sqrt{\sum_{i=1}^{i=n} e_i^2/n}$

Fig. 2 shows the error analysis between cipher and plaintext computations. In case of ciphertext, the value of RMSE is 0.325, whereas MAE is 0.257. For plaintext, the value of RMSE and MSE are 0.337 and 0.266, respectively. Therefore, we can see that in both cases our encrypted text has less error compared to the plaintext.

C. Accuracy Analysis Based on ROC

Receiver Operating Characteristic (ROC) area is a graphical instrument to illustrate the performance of a binary classifier [22]. The area is defined by plotting the true positive rate against the false positive rate of the proposed classifier at various thresholds (0 to 1). The performance of the classifier increases if it can provide more true positive rates than the false positive rates.

Fig 3 shows ROC area for our classification problem. In our classification problem, we found 0.803 (baseline .5) for the area under ROC curve.

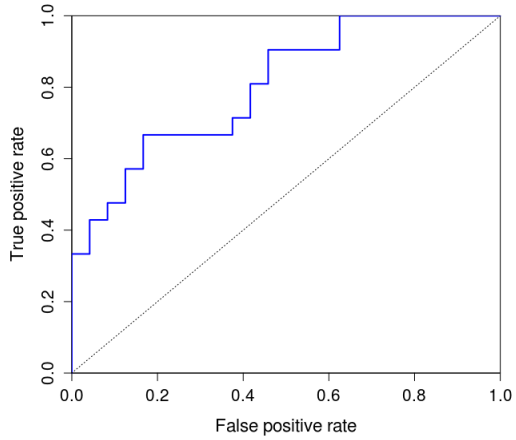

Fig. 3. Area under ROC curve for linear regression in ciphertext

D. Security Analysis

A widely used measure to evaluate the security of a lattice cryptosystem is the root-hermit factor δ. Linder and Peikert [23] derived the relationship between the root-hermit factor and the security level λ (bit) of lattice-based cryptosystem as follows:

$$\lambda = 1.8 \times \frac{1}{\log_2 \delta} - 110 \tag{4}$$

In Eqn 4, $\frac{1}{\log_2 \delta} = \frac{4n(\log_2 q)}{(\log_2(\frac{cq}{s}))^2}$, where $c = \sqrt{\frac{\ln \frac{1}{\epsilon}}{\pi}}$ and $s = \sigma\sqrt{2\pi}$. Here n, q, and s represent the degree of the polynomial ring, ciphertext modulus, and the scale parameter of the error distribution, respectively. σ denotes the standard deviation of the error (of LWE) distribution that is used for the advantage of an attacker. In our case, we used $n = 2^{12}, q = 2^{124}, \sigma = 3$, and $\epsilon = 2^{-32}$, which results in approximately 132-bit security.

E. Runtime Analysis

For generating our model, we used a 4 core virtual machine with 8 GB RAM hosted in the Amazon cloud. Our method took around 75 minutes for one iteration without any parallelism. However, utilizing the 4 cores and our parallel framework, we reduced the time to only 10 minutes for the computation.

VII. Conclusion

The above framework has been developed to protect the privacy of PD patients when computing a predictive model using linear regression. However, one of the major limitations of our work is the use of somewhat HE that does not allow infinite amount of computations. This is a major drawback of Fan *et al.* scheme [8] and can be addressed by utilizing a fully HE like TFHE [15]. Also, in order to test the scalability and practicality of our proposed model, we plan to evaluate this framework using other disease data.

Acknowledgments

This research was supported in part by the NSERC Discovery Grants (RGPIN-2015-04147), University Research Grants Program (URGP) from the University of Manitoba, and Amazon Research Credits

References

[1] L. J. Findley, "The economic impact of parkinson's disease," *Parkinsonism & related disorders*, vol. 13, pp. S8–S12, 2007.

[2] R. Hall, S. E. Fienberg, and Y. Nardi, "Secure multiple linear regression based on homomorphic encryption," *Journal of Official Statistics*, vol. 27, no. 4, p. 669, 2011.

[3] T. Graepel, K. Lauter, and M. Naehrig, "Ml confidential: Machine learning on encrypted data," in *International Conference on Information Security and Cryptology*. Springer, 2012, pp. 1–21.

[4] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data." in *NDSS*, 2015.

[5] M. N. Sadat, X. Jiang, M. M. Al Aziz, S. Wang, and N. Mohammed, "Secure and efficient regression analysis using a hybrid cryptographic framework: Development and evaluation," *JMIR medical informatics*, vol. 6, no. 1, 2018.

[6] Y. Jiang, J. Hamer, C. Wang, X. Jiang, M. Kim, Y. Song, Y. Xia, N. Mohammed, M. N. Sadat, and S. Wang, "Securelr: Secure logistic regression model via a hybrid cryptographic protocol," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2018.

[7] T. M. Mitchell *et al.*, "Machine learning. wcb," 1997.

[8] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption." *IACR Cryptology ePrint Archive*, vol. 2012, p. 144, 2012.

[9] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[10] P. Paillier *et al.*, "Public-key cryptosystems based on composite degree residuosity classes," in *Eurocrypt*, vol. 99. Springer, 1999, pp. 223–238.

[11] C. Gentry *et al.*, "Fully homomorphic encryption using ideal lattices." in *STOC*, vol. 9, no. 2009, 2009, pp. 169–178.

[12] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM (JACM)*, vol. 56, no. 6, p. 34, 2009.

[13] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2010, pp. 1–23.

[14] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) lwe," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 831–871, 2014.

[15] 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 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22*. Springer, 2016, pp. 3–33.

[16] A. C.-C. Yao, "How to generate and exchange secrets," in *Foundations of Computer Science, 1986., 27th Annual Symposium on*. IEEE, 1986, pp. 162–167.

[17] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *International Conference on Machine Learning*, 2016, pp. 201–210.

[18] M. Hoekstra, R. Lal, P. Pappachan, V. Phegade, and J. Del Cuvillo, "Using innovative instructions to create trustworthy software solutions." *HASP@ ISCA*, vol. 11, 2013.

[19] A. A. Ahmadi, "Characterizations of convex functions, strict and strong convexity, optimality conditions for convex problems." 2017.

[20] "Gerardnico one rule," https://gerardnico.com/wiki/data_mining/one_rule, accessed: 2017-11-17.

[21] M. A. Little, P. E. McSharry, E. J. Hunter, J. Spielman, L. O. Ramig *et al.*, "Suitability of dysphonia measurements for telemonitoring of parkinson's disease," *IEEE transactions on biomedical engineering*, vol. 56, no. 4, pp. 1015–1022, 2009.

[22] T. Fawcett, "Roc graphs: Notes and practical considerations for researchers," *Machine learning*, vol. 31, no. 1, pp. 1–38, 2004.

[23] R. Lindner and C. Peikert, "Better key sizes (and attacks) for lwe-based encryption." in *CT-RSA*, vol. 6558. Springer, 2011, pp. 319–339.