International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Privacy Preserving Inference Over Encrypted Data

Suhel Sayyad[1], Dinesh Kulkarni[2]
[1] *Annasaheb Dange College of Engineering and Technology, Ashta*
ORCID ID: 0000-0002-5208-0829
[2] *Walchand College of Engineering, Sangli*
ORCID ID: 0000-0002-3448-3173
* *Corresponding Author Email: suhelsayyad2006@gmail.com*

Submitted: 22/07/2022 **Accepted**: 25/09/2022

Abstract: Machine learning and deep learning techniques provide solution to various medical applications relevant to disease detection. Many a time's these learning algorithms and their inferences are generated in untrusted environments like cloud. Medical practitioners would like to protect their data in such cases in untrusted environments but would like to generate an inference on their data. Our work provides a solution to generate privacy preserving inference over encrypted data in such untrusted environment like public cloud. HELib based fully homomorphic encryption approach is used to provide security to the trained model and the data of the owner. Our results shows the effectiveness of using the technique to generate inference on encrypted data without comprising the accuracy of the system. Our work demonstrates on benchmark datasets like MNIST for prototyping and heart disease detection that are used by many machine learning applications for benchmarking.

Keywords: *Privacy preserving, homomorphic encryption, secured learning, federated learning.*

1. Introduction

Neural network and deep learning methods are mathematical models that can be used to solve problems related to classification, regression, clustering, etc. Many of the medical diagnosis problems have been effectively solved using machine learning and deep learning approach. Medical practitioners would like to learn an inference on the diseases based on the available patient data with them. Federated learning, Secure multiparty learning is another area which has enabled the users to learn on encrypted data in untrusted environments like cloud. In such scenarios, a medical practitioner owning a data of a patient would like to generate the inference on the trained model in untrusted environments like cloud. In such situations, a medical practitioners would like to send an encrypted record of a patient on cloud and generate an encrypted result which he can decrypt at his end and learn the inference. The model on the cloud will also be in encrypted form. In this case, the service provider will also not learn anything about the model and also about the data of the medical practitioners on which inference is carried out. In order to achieve this security, use of techniques like homomorphic encryption, secure multiparty computation, and perturbation techniques can be done.

2. Related Work

Neural network is a mathematical model which provides us to implement machine learning technique which models higher level of abstraction for the data elements. Neural network learning makes use of a network that utilizes multiple processing layers. These processing layers involve multiple nonlinear transformations. Neural network learning can be used to solve different types of problems like speech recognition, image recognition and face detection.

Schlitter [1] has proposed the privacy preserving back propagation neural network scheme that provides learning between two or more parties. In this scheme the data is horizontally partitioned between the participating parties. This technique does not provide any mechanism that will help to protect the intermediate results during the learning process. Chen and Zhong [2] has proposed a method involving privacy preserving technique for back propagation neural network learning that solves two party problems. In this technique data is considered to be vertically partitioned. This technique provides solution for data security and it also provides security to the intermediate results obtained during the training process. This approach present lightweight distributed algorithm between two parties for privacy-preserving propagation neural network learning and is assumed to work on vertically partitioned data set. Piecewise linear approximation for the sigmoid function is being carried out in order to security compute the sigmoid function in this approach. This technique fails to address the issue of security to the model that is being trained. Nathan [3] has proposed a privacy preserving learning model in which the security issues of the data of participating parties is addressed. Theo Ryffel at al. [4] has proposed generic Framework that helps to solve the problem of privacy preserving deep learning. This framework provides the secure processing of data and helps to represent the information in the form of tensors and chain of commands. This way of representing the data helps the user to implement complex privacy-preserving like secure multiparty computation, federated learning and differential hiring with help of an API. Jiawei Yuan and Shucheng Yu [5] have proposed a technique for privacy-preserving algorithm on a cloud environment. In this research work solution to the neural network learning problem with arbitrary partition data sets in horizontal and vertical data set is carried out. CrypTen [7] is a new framework that is built on PyTorch to help research in privacy-preserving machine learning and address the security issues. CrypTen enables machine learning researchers those who are not experts in cryptography, to easily implement the machine learning models by applying secure computing techniques. The complexity barrier is reduced by CrypTen by enabling the PyTorch API to end researchers. Pysyft [6] is a framework built over Pytorch to provide security to data owners in Federated learning environment. Li Li et al.[10] has reviewed federated learning. Federated learning is privacy preserving technique that is decentralized in collaborative way to overcome various issues of data sensibility. It reviews various application that can be guided using federated learning.

3. Proposed Work

3.1 Neural Network Model

Consider an example of simple neural network containing a output nodes, b hidden nodes and c input nodes. Iteration m, learning rate η, target value ti, sigmoid function f(x) = 1/1+e^{-x}. Weight vector V_{jk} (representing weights between hidden and output nodes) and V_{ij} (representing weights between input and hidden nodes).

```
Steps for Neural Network learning
Initialize all weights randomly $V^h{}_{jk}$, $V^o{}_{ij}$ .
for iteration = 1, 2 ··· , $iteration_m$ do
for sample = 1, 2 ··· , N do
for j = 1, 2 ··· , b do
    $h_j = f( \sum x_k * V^h{}_{jk})$
    for i = 1, 2 ··· , c do
     $o_i = f( \sum a\ j=1\ h_j * V^o{}_{ij} )$
    if Error = $1/2 \sum c\ i=1(t_i - o_i)^2$ > threshold then
            $\Delta V^o{}_{ij} = -(t_i - o_i) * h_j$
            $\Delta V^h{}_{jk} = -h_j(1 - h_j)x_k \sum^c{}_{i=1}[(t_i - o_i) * V^o{}_{ij} )]$
            $V_{ij} = V_{ij} - \eta \Delta V_{ij}$
            $V^h{}_{jk} = V^h{}_{jk} - \eta \Delta V^h{}_{jk}$
    else
    break
```

When all the training samples completes one round of training then it is said to complete one epoch. Batch size indicates number of samples used in one cycle of training. For example, if there are 1000 samples and batch size is 100 then, 1 epoch will be completed when all 1000 samples will complete one round of training. For this 10 iterations of 100 batch size will be executed.

3.2 HE Lib

An encryption scheme is said to homomorphic, if for a given Enc (P) and Enc (Q) it is possible to calculate the value for Enc (F (P, Q)), where F can be: addition, multiplication and without making use of the private key of the encryption.

- e_k represents algorithm with key k used for encryption purpose.
- d_k represents algorithm used by decryption purpose
- Multplicative d_k (e_k (P) × e_k (Q)) = P × Q.
- Additive d_k (e_k (P) + e_k (Q)) = P + Q.

Fully homomorphic encryption – the one that supports multiplication and addition both and arbitrary number of times the operation can be repeated. Helib [9] is an open source library. This library implements BGV and CKKS fully Homomorphic encryption schemes.

$\mathbb{Z}$ Represents ring of integers, $\mathbb{Q}$ represents field of rational numbers, $\mathbb{R}$ represents the field of real numbers, and $\mathbb{C}$ represents the field of complex numbers.

For a given positive integer m, $\mathbb{Z}_m$ represents the quotient ring$\mathbb{Z}/(m)$, the ring of integers modulo$m$. $\mathbb{Z}_m^*$ Denotes the group of units in $\mathbb{Z}_m$. $\mathbb{Z}_m^*$ Comprises of those residue classes of which representative are relatively prime to$m$. $|\mathbb{Z}_m^*| = \phi(m)$, where ϕ is representing Euler totient function.

For a given positive integer, $[m]$ represents the set of integers $\{0,..,\mathrm{m}-1\}$. $\mathbb{Z}_m$ And $[m]$ are not the same terms: the $\mathbb{Z}_m$ is representing set of residue classes, which builds a ring, whereas the [m] is representing subset of the integers, which does not build a ring.

3.3 Secret keys and cipher texts: basic structure and operations

Secret keys and cipher texts in the BGV and CKKS encryption schemes are vectors of elements over rings $\mathcal{A}$ or $\mathcal{A}_q$, and BGV and CKKS decryption scheme is representing an inner-product between these entities followed by rounding operations. HElib implements a flexible structure, in which cipher text objects comprises of a changing set of ring $\mathcal{A}_q$-elements, and every element contains a descriptor for which the secret-key element it should use for decryption purpose.

A secret-key object is a family of elements over the ring $\mathcal{A}$, which is indexed by index set I

$\boldsymbol{S} := \{s_i\}_{i \in I}$ with each $s_i \in \mathcal{A}$.

A cipher text object, for this secret key, comprises of corresponding family of elements of the ring $\mathcal{A}_q$ for an integer $q > 1$ and is relatively prime to m, that is indexed by same index set I

$C := \{\bar{c}_i\}_{i \in I}$, with each$\bar{c}_i \in \mathcal{A}_q$.

3.4 Homomorphic Addition

Suppose there are two given cipher texts to be added, for which, $\ell = 1, 2$, contains the following entities:

- $P_\ell = p^{r_\ell}$, indicates plaintext modulus
- q_ℓ, cipher text modulus
- enciphering family $\boldsymbol{C}_\ell$ that is relative to a secret key $\boldsymbol{S}_\ell$,
- correction factor $\kappa_\ell \in \mathbb{Z}^*_{P_\ell}$,
- bound ϵ_ℓ on the noise.

Before these two cipher texts can be added, provided they are adjusted so that the plaintext modulus, cipher text modulus, and correction factors match.

1 First, the plaintext modulus are matched by making them both equal to $:= \gcd(P_1, P_2) = p^{\min(r_1, r_2)}$.

2 Second, the cipher text modulus are matched by making them both equal to $:= \operatorname{lcm}(Q_1, Q_2)$. For performing this upscaling is carried out.

3 For matching the correction factors, choose integers e_1, e_2, that are relatively prime to P, such that

$$[e_1 \bmod P] \cdot \kappa_1 = \kappa = [e_2 \bmod P] \cdot \kappa_2.$$

3.4 Homomorphic Multiplication

Suppose there are two given cipher texts to be multiplied, for $\ell = 1{,}2$, contains the following entities:

- $P_\ell = p^{r_\ell}$, indicates plaintext modulus
- q_ℓ, indicates cipher text modulus
- enciphering family $\boldsymbol{C}_\ell$ that is relative to a secret key $\boldsymbol{S}_\ell$,
- correction factor $\kappa_\ell \in \mathbb{Z}^*_{P_\ell}$,
- bound ϵ_ℓ on the noise.

Before these two cipher texts can be multiplied, provided they are adjusted so that the plaintext moduli and cipher text moduli match.

The plaintext modulus is been matched by keeping them equal to

$P := \gcd(P_1, P_2) = p^{\min(r_1, r_2)}$.

Match the cipher text modulus by applying up scaling and mod switching method to convert them to common cipher text modulus$Q$. In selecting$Q$, an effort is to be made for reducing the noise in every cipher text somewhat.

Hence, both cipher texts have same plaintext modulus P and same cipher text modulus Q. Suppose that

$$\boldsymbol{S}_1 = \{\boldsymbol{s}_i\}_{i\in I} \text{ and } \boldsymbol{S}_2 = \{\boldsymbol{s}_j\}_{j\in J}.$$

Assume that secret keys for two cipher texts are indexed in an uniform way, so that if two indices are found equal then components are equal. Secret key for resulting cipher text is

$$\boldsymbol{S} := \{s_i s_j\}_{(i,j)\in I\times J}.$$

Assume

$$C_1 = \{\bar{c}_i\}_{i\in I} \text{ and } C_2 = \{\bar{c}'_j\}_{j\in J}.$$

The enciphering family of the cipher text is

$$\boldsymbol{C} := \{\bar{c}_i \bar{c}'_j\}_{(i,j)\in I\times J}.$$

Correction factor for the resulting cipher text is$\kappa := \kappa_1\kappa_2$. The noise bound of the resulting cipher text is$\epsilon := \epsilon_1\epsilon_2$.

3.5 Conceptual Framework of Privacy Preserving Inference using fully homomorphic encryption

Figure 1 represents the conceptual idea of privacy preserving inference. The model is trained initially in plaintext form and then using fully homomorphic encryption scheme the model is encrypted. Further, the user who wish to perform the inference will provide the encrypted data for testing and the inference generated will be in encrypted form as shown in the figure 1. The user can further decrypt and get the classification result. This approach helps us to resolve the security issues of the data item and also the model that is kept on the untrusted environment. Fully homomorphic encryption technique implementation based on HELib [9] is used in the proposed system.

4. Experimental Results

We have training the plaintext model in this experimentation using Keras library. The objective is to classify samples using neural network model. This model is created and is trained using Keras library. In the first step, the plain model is created for the MNIST. Then the model is been encrypted. Further the inference is executed on the encrypted model using Fully Homomorphic encryption technique. For building the encrypted model, it is required to define the configuration of the library and the layout of each of the cipher texts.

Table 1. Inference on Encrypted Data and Encrypted Model for MNIST Dataset

Batch Size	***8***	***16***	***32***	***64***	***128***
Tile layout	16 * 64 * 8	8 * 64 * 16	8 * 32 * 32	4 * 32 * 64	4 * 16 * 128
Predict time (sec)	3.49	4.37	6.21	8.27	12.41
Init model time (sec)	3.67	6.62	12.33	22.52	43.35
Encrypt input time (sec)	0.8	1.61	3.1	5.42	10.35
Model memory (MB)	213.14	384.07	721.47	1313.7	2540.63

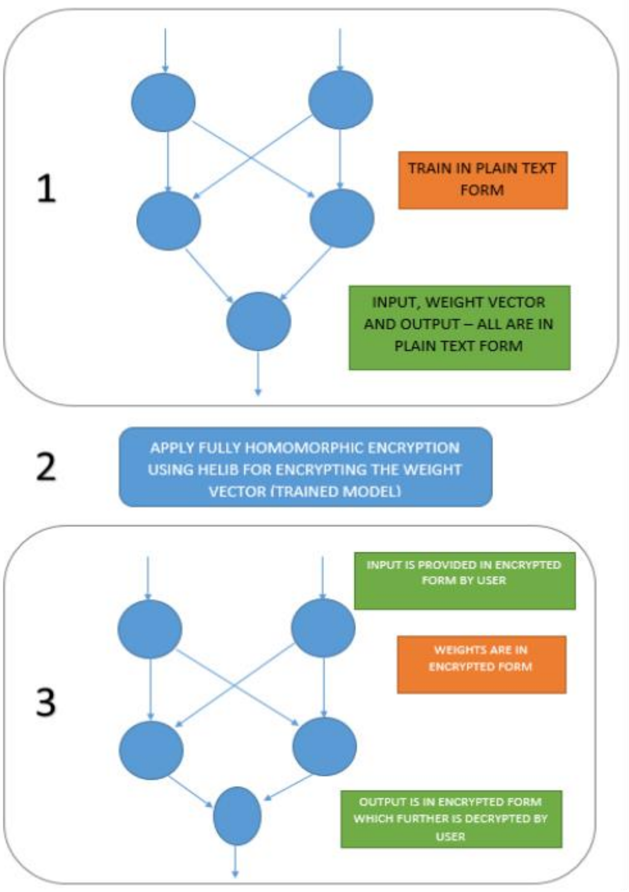
1
TRAIN IN PLAIN TEXT FORM
INPUT, WEIGHT VECTOR AND OUTPUT – ALL ARE IN PLAIN TEXT FORM
2
APPLY FULLY HOMOMORPHIC ENCRYPTION USING HELIB FOR ENCRYPTING THE WEIGHT VECTOR (TRAINED MODEL)
3
INPUT IS PROVIDED IN ENCRYPTED FORM BY USER
WEIGHTS ARE IN ENCRYPTED FORM
OUTPUT IS IN ENCRYPTED FORM WHICH FURTHER IS DECRYPTED BY USER

Figure 1: Conceptual idea of privacy preserving inference

This is referred to as a Tile in this work. Table 1 indicates results for various Batch size applied for generating encrypted inference based on encrypted data provided by end user to the untrusted environment. The model is in encrypted form. The results also shows the increase in Model memory size, encrypt input time and increase in predict time with the increasing value of batch size. The experimentation was carried out on Dell Optiplex 3910 Desktop with 8 GB RAM. Figure 2 represents the graphical representation of the time required in seconds for different activities like init model time, encrypt model time and encrypt input time against different batch size. The figure also represents the model memory required for the training of MNIST for different batch size. We obtain an accuracy of 99% for non-privacy preserving and privacy preserving approach using HELib with model and data security. We make use of HELib fully Homomorphic encryption technique to solve the security issues of the medical practitioners and get results using inference on encrypted model. For experimentation, dataset in [8] is used.

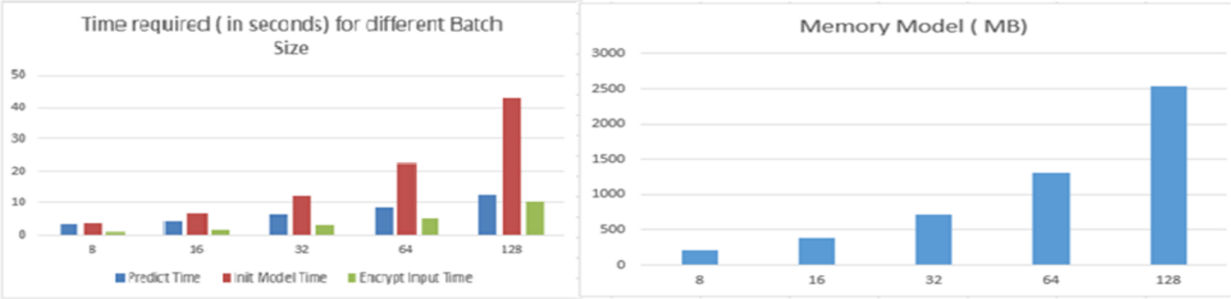

Figure 2: Comparison of time required and memory size against different batch

It comprises of 14 attributes that are used for various machine learning algorithms.

Case Study: Medical Practitioner using the application
Following steps are carried out for testing it on heart disease detection dataset.

1. Load data and train the model in plain text form.
2. Perform optimization for deciding the best parameters for security
3. Create encrypted network
4. By concerned medical practitioner who wish to detect for his samples- Encrypt the samples.
5. Perform inference on encrypted network for encrypted samples.
6. By concerned medical practitioner: Decrypt the results and observe classification.

Below Table 2 represents the test classification results for model trained with batch size of 16.

Table 2. Analysis of Heart Disease dataset

Number of Training Samples	16
Correctly Classified Samples	15
Accuracy (%)	93.75
Time duration for prediction (sec)	0.062
Time duration for prediction per sample (sec)	0.004

A comparative study is carried out to understand the amount of time required to predict the inference in non-privacy preserving, privacy preserving with data encryption and privacy preserving with data and mode encryption. Accuracy is also compared across all the three parameters for different batch size in MNIST Dataset training. For this experimentation, Dell OptiPlex 3910 Desktop with 8 GB RAM system is used. Table 3 indicates the results for this comparison. It is observed that accuracy of the system is not comprised and the predict time required is also almost the same for different batch size of the training and security issues of the data and the model has taken care of.

Table 3. Comparison of Privacy and Non Privacy Preserving technique for different batch size on MNIST.

Batch Size		**Non PP**	**PP-Data Encrypt**	**PP-Data and Model**
8	**Predict Time(sec)**	3.49	3.47	3.49
	Accuracy (%)	98.23	98.23	98.23
16	**Predict Time(sec)**	4.37	4.35	4.37
	Accuracy (%)	99.12	99.12	99.12
32	**Predict Time(sec)**	6.21	6.22	6.21
	Accuracy (%)	100	100	100

5. Conclusion and Future Work

Our proposed method provides solution to concerns related to security issues of the data owners who wish to get an inference on their data in untrusted environments. Our proposed method provides encrypted inference on the encrypted data provided by a medical practitioner and is tested on an encrypted model present on the untrusted environment. Our benchmarked results on MNIST and heart dieases dataset from UCI also shows the comparitive study of time required towards encryption, decryption, memory requirement for th emodel and predict time for different batch size. Experimental results also demonstrates that there is no loss of accuracy on the encyprted inference.

Acknowledgement
This work is carried out as part of research work carried at Walchand College of Engineering, Sangli as research center.

Author contributions

Suhel Sayyad: Conceptualization, Methodology, Writing-Original draft preparation, Software, Validation

Dinesh Kulkarni: Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

1. N. Schlitter, A Protocol for Privacy Preserving Neural Network Learning on Horizontal Partitioned Data, Proc. Privacy Statistics in Databases (PSD 08), Sept. 2008
2. Nouby M. Ghazaly, A. H. H. . (2022). A Review of Using Natural Gas in Internal Combustion Engines. International Journal on Recent Technologies in Mechanical and Electrical Engineering, 9(2), 07–12. https://doi.org/10.17762/ijrmee.v9i2.365
3. T. Chen and S. Zhong,Privacy-Preserving Backpropagation Neural Network Learning, IEEE Trans. Neural Network, vol. 20, no. 10, pp. 1554-1564, Oct. 2009.
4. Sudhakar, C. V., & Reddy, G. U. . (2022). Land use Land cover change Assessment at Cement Industrial area using Landsat data-hybrid classification in part of YSR Kadapa District, Andhra Pradesh, India. International Journal of Intelligent Systems and Applications in Engineering, 10(1), 75–86. https://doi.org/10.18201/ijisae.2022.270
5. Avhankar, M. S. ., D. J. A. . Pawar, S. . Majalekar, and S. . Kedari. “Mobile Ad Hoc Network Routing Protocols – Using OPNET Simulator”. International Journal on Recent and Innovation Trends in Computing and Communication, vol. 10, no. 1, Jan. 2022, pp. 01-07, doi:10.17762/ijritcc.v10i1.5513.
6. Ghazaly, N. M. . (2022). Data Catalogue Approaches, Implementation and Adoption: A Study of Purpose of Data Catalogue. International Journal on Future Revolution in Computer Science &Amp; Communication Engineering, 8(1), 01–04. https://doi.org/10.17762/ijfrcsce.v8i1.2063
7. Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig and John Wernsing, CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy 29 December 2015
8. A generic framework for privacy preserving deep learning, Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, Jonathan Passerat-Palmbach, 13 Nov 2018.
9. Privacy Preserving Back-Propagation Neural Network Learning Made Practical with Cloud Computing. IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 1, January 2014.
10. PySyft. [Online]. Available: https://github.com/OpenMined/PySyft
11. Sharma, V. . (2022). Unique Functors of Everywhere Connected Homomorphisms and the Countability of Groups. International Journal on Recent Trends in Life Science and Mathematics, 9(2), 01–09. https://doi.org/10.17762/ijlsm.v9i2.130
12. Knott, Brian and Venkataraman, Shobha and Hannun, Awni and Sengupta, Shubho and Ibrahim, Mark and van der Maaten, Laurens, CrypTen: Secure Multi-Party Computation Meets Machine Learning, https://doi.org/10.48550/arxiv.2109.00984, 2021
13. UCI Machine Learning Heart Disease Dataset: https://archive.ics.uci.edu/ml/datasets/heart+disease
14. HElib [Online]. Available: https://github.com/homenc/HElib
15. N. A. Libre. (2021). A Discussion Platform for Enhancing Students Interaction in the Online Education. Journal of Online Engineering Education, 12(2), 07–12. Retrieved from http://onlineengineeringeducation.com/index.php/joee/article/view/49
16. Li Li, Yuxi Fan, Mike Tse, Kuo-Yi Lin, A review of applications in federated learning, Computers & Industrial Engineering, Volume 149, 2020