South Ural State University

National Research University

South Ural State University

School of Electronic Engineering and Computer Science

Problem-Oriented Cloud Computing Environment International Laboratory

Seminar

Privacy-Preserving Machine Learning as a Service

Speaker

Jorge Mario Cortés-Mendoza

Russia, December 2020.

Content

- Motivation
- Machine Learning as a Service
- Homomorphic Encryption
- Privacy-Preserving Neural Networks
- Privacy-Preserving Logistic Regression
- Future work

Motivation

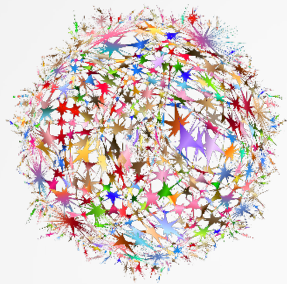
Cloud computing has been widely adopted because it allows acquiring on-demand computing resources

Machine Learning as a Service (MLaaS) has emerged as a flexible and scalable solution in cloud environments

Motivation

MLaaS offers different types of resources and tools to train and deploy ML models

Neural Networks



Deep Learning

Natural Language Processing

Machine Learning process

The training process can consume many computational resources and time

- The high-performance computing resources in the cloud can reduce training and testing time

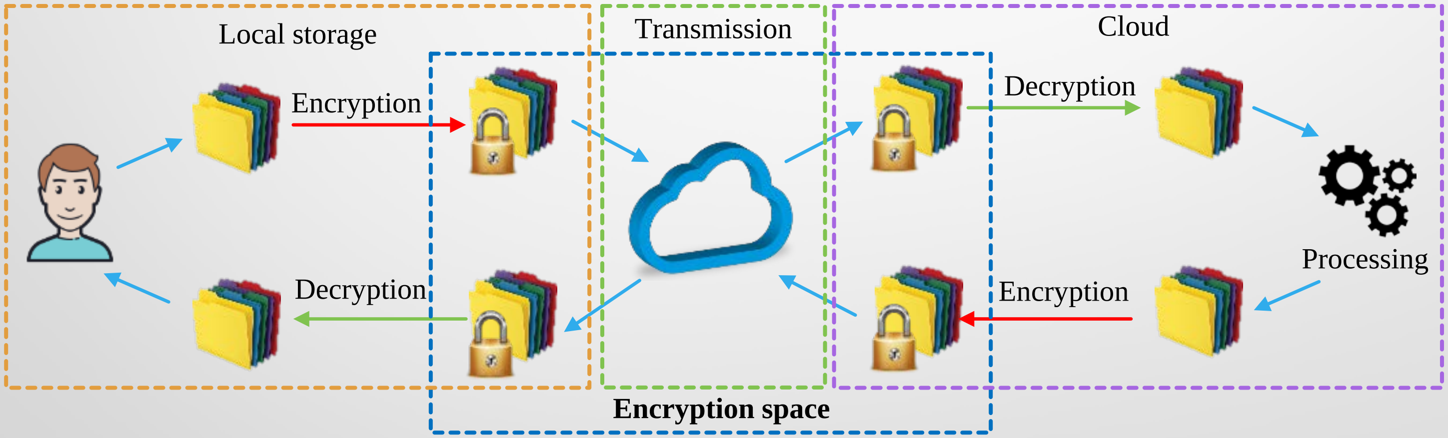
The remote infrastructure of the cloud reduces the problems of resources and implementation, but it introduces several privacy concerns in sensitive information

Machine Learning as a Service

Data security in cloud computing offers data protection from theft, leakage, deletion at levels of firewalls, penetration testing, obfuscation, tokenization, Virtual Private Networks (VPN), etc.

The use of third-party services can bring several cybersecurity risks

- Traditional encryption does not solve the problem because ML model requires full access to confidential data

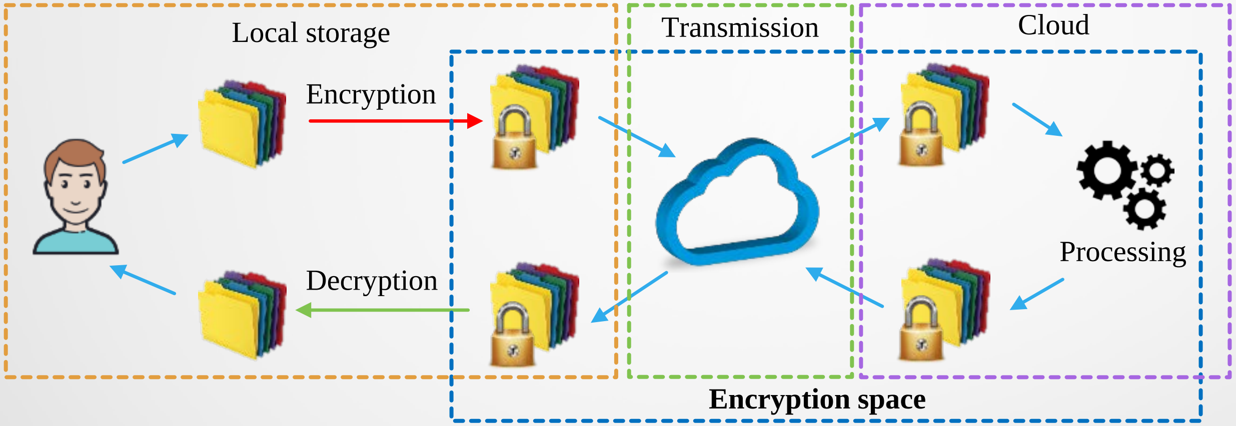


Security and privacy are significant challenges because data must be decrypted for analytics

Homomorphic Encryption

Homomorphic Encryption (HE) and Secure Multi-party Computation (SMC) are ways to address vulnerabilities of data processing

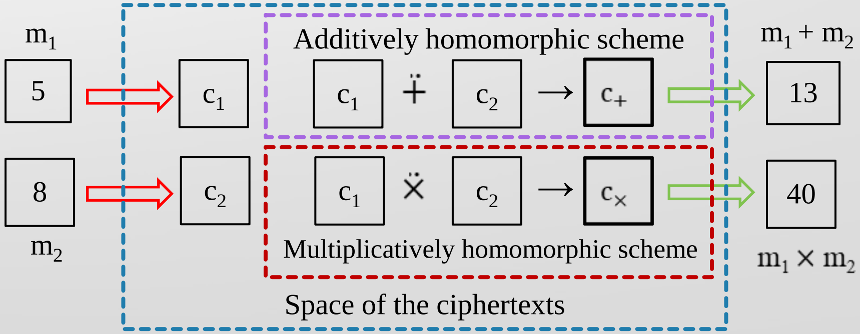
HE is an encryption system that enables the processing of information on ciphertexts



HE schemas strengthen several aspects of security in the cloud

Ciphertexts c_1 and c_2 encrypt the content of messages m_1 and m_2

- c_+ is created using c_1 and c_2, and its decryption produces $m_1 + m_2$
- $c_\times$ encrypts $m_1 \times m_2$



Homomorphic Encryption

"Homomorphic" refers to a mapping between functions on the space of messages and ciphertexts

- A function applied to ciphertexts provides the same (encrypted) result than its homomorphic function used in the messages they encrypt

The system only uses publicly available information without risks of the data breach

- No access to information in the ciphertext or any secret key

HE implementation exhibits several limitations, the three main directions in this field are:

Low efficiency

Small number of primitives

Neural Network

Deep Learning

ML models

South Ural State University
National Research University

Homomorphic Encryption for ML

HE surveys consolidate significant contributions focusing on performance improvement, new approaches, applications, among others

They provide knowledge foundation and general panorama to researchers interested in applying and extending HE approaches

Table 1. Main topics of HE reviews

Reference \ Topic	Technical	Limitations	Applications	Tools	Cloud-based	Implementations
Vaikuntanathan [1]	●		●			
Armknecht et al. [2]	●	●	●	●		●
Naehrig et al. [3]	●	●	●		●	●
Archer et al. [4]			●		●	
Acar et al. [5]	●	●	●	●		
Martins et al. [6]	●	●	●			
Parmar et al. [7]	●		●			
Shunmuganathan [8]	●		●			
Gentry [9]	●	●				
Aguilar-Melchor [10]	●		●			●
Hrestak and Picek [11]	●			●	●	
Moore et al. [12]	●					●

Table 2. Comparative of HE approaches

Year	Operations			ML approach				Scheme						Objective	
	Addition	Multiplication	Other	LR	NN	DNN	Decision Trees	Ideal Lattice	Integer-based	(R) LWE	NTRU	Two-party	Multi-party	Security	Efficiency
1978		•										•		•	
1985		•										•		•	
1999	•											•		•	
2009	•	•						•				•		•	
2011	•	•		•						•		•		•	
2014	•	•									•	•			•
	•	•		•					•			•		•	
2015	•	•					•			•		•		•	
	•	•					•			•		•		•	
2016	•	•				•			•				•	•	•
	•	•			•				•			•		•	
	•	•		•						•		•		•	
	•	•				•			•				•	•	
2017	•	•								•		•		•	•
	•	•				•			•			•		•	
	•	•								•		•		•	
2018	•	•		•						•		•		•	
	•					•			•				•	•	•
	•	•				•			•			•		•	
	•	•				•			•			•		•	•
	•	•				•			•			•		•	
2019	•	•			•				•				•	•	
	•	•	•									•		•	•

Homomorphic Encryption for ML

- A small number of primitives have been developed for predicting and classifying confidential information using HE schemas
- The main goal is to enrich the MLaaS paradigm

Homomorphic Encryption for ML

Theoretical research in HE should be complemented with high-quality implementations

Industrial and academic groups have been released several HE libraries in recent years

Table 3. Comparison of commonly general-purpose HE libraries across their pros and cons

Tool	Support	Pros	Cons
SEAL	Microsoft	Well-documented Easy security parameters setting	Poor flexibility Limited number of supported schemes
HElib	IBM	Efficient homomorphic operations	Low bootstrapping performance Complicated security parameter setting
TFHE		Fast bootstrapping	Poor performance for simple tasks
PALISADE	DARPA, MIT, UCSD, etc.	Multiple HE schemes Cross-platform	Poor documentation and support
cuHE		Mass parallelism and high memory bandwidth of GPUs	
HEAAN	Seoul National University	Operations between rational numbers	
HE-transformer	Intel	Integration with deep learning libraries	Extension of SEAL

Homomorphic Encryption for ML

An analysis shows the emerging interest of the research community in the construction of HE in handling highly sensitive data

- Machine learning models to process over encrypted data

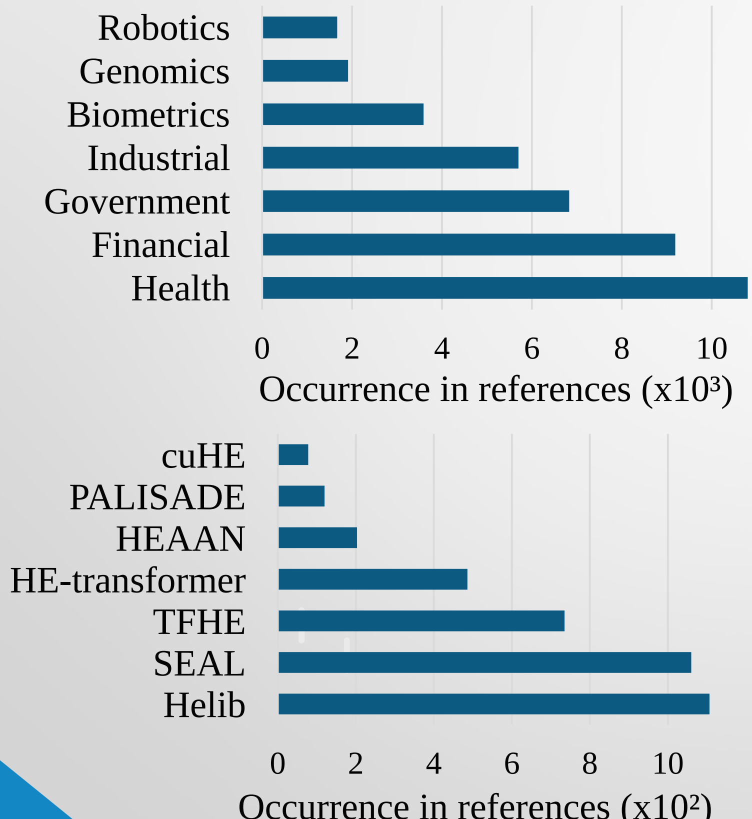


Fig. 1. Keywords related to HE concepts and specific applications in the HE area (five years)

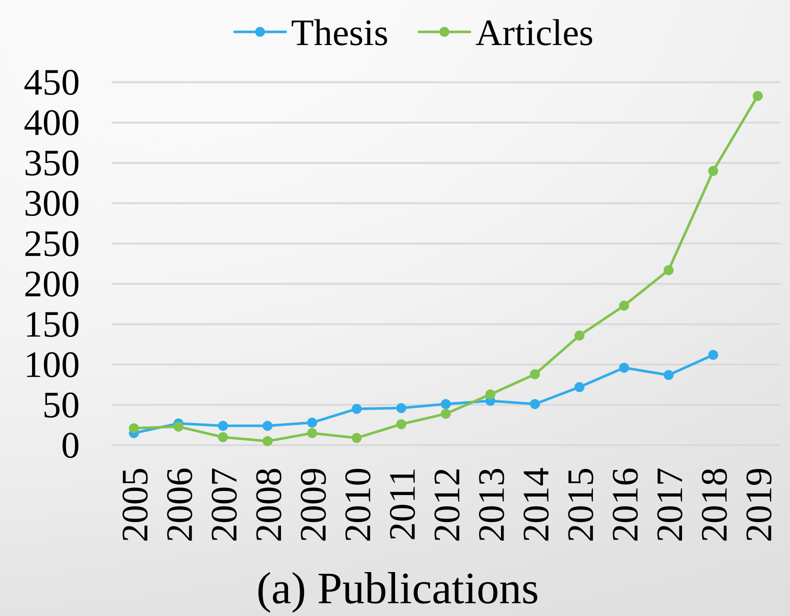


Fig. 2. Number of publications in the literature related to HE

South Ural State University
School of Electronic Engineering and Computer Science
Problem-Oriented Cloud Computing Environment International Laboratory

Privacy-Preserving Neural Networks

Jorge M. Cortés-Mendoza
Andrei Tchernykh
Mikhail Babenko
Luis B. Pulido-Gaytán
Gleb Radchenko
Arutyun Avetisyan
Alexander Yu. Drozdov

Russia, December 2020.

Privacy-Preserving Neural Networks

Each neuron consists of n_I inputs $x = (x_1, \ldots, x_{n_I})$ and an output y

$$y = f\left(\sum_{i=1}^{n_I} w_i \times x_i + \beta\right)$$

The value of y defines a weighted sum of the inputs considering the weights $w = (w_1, \ldots, w_{n_I})$, a bias β and the non-linear activation function f

The HE version of a neuron (NN-HE) substitutes $+, \times$, and f

$$\bar{y} \leftarrow \ddot{f}\left(\sum_{i=1}^{n_I} (\bar{w}_i \ddot{\times} \bar{x}_i) \ddot{+} \bar{\beta}\right)$$

where $\bar{x}, \bar{w}$, and $\bar{\beta}$ are the corresponding ciphertexts of x, w, and β, and $\ddot{f}$ is the homomorphic version of f

$\ddot{f}$ is a polynomial approximation that only consists of operations $\ddot{+}$ and $\ddot{\times}$

$\bar{y}$ contains the encrypted output of the neuron computation, it guarantees the privacy of the result even if it is disclosed

The network structure defines the interaction between layers (sets of neurons)

- The NN-HE does not apply any modification in the structure of the NN

Privacy-Preserving Neural Networks

The activation function is essential in the construction of a NN model

- The definition of $\ddot{f}$ is an open problem (standard activation functions use operations not supported by HE)

A polynomial approximation have to balance between complexity and accuracy

- High-degree polynomials provide high accuracy with slow computations
- Low-degree polynomials provide fast computations with low accuracy

Table 4. Summary of activation function approximations

Function	n	Model LR	Model NN	Approximation Method
Sigmoid	2, 3		●	Chebyshev polynomials
	2	●		Taylor series, area
	1	●		Taylor series
	3, 5, 7	●		Taylor series
	9		●	Taylor series, Padé
Tanh	2, 3		●	Chebyshev polynomials
	9		●	Taylor series, Padé
	3, 4		●	Chebyshev polynomials
ReLU	2,3,4,5,6		●	Least squares polynomial fit (soft.)
	2, 3		●	Derivative of ReLU
	1		●	Taylor series, Padé
	3, 4		●	Chebyshev polynomials
	2		●	Polytope-based method
Swish	3,4		●	Chebyshev polynomials
	2		●	Polytope-based method

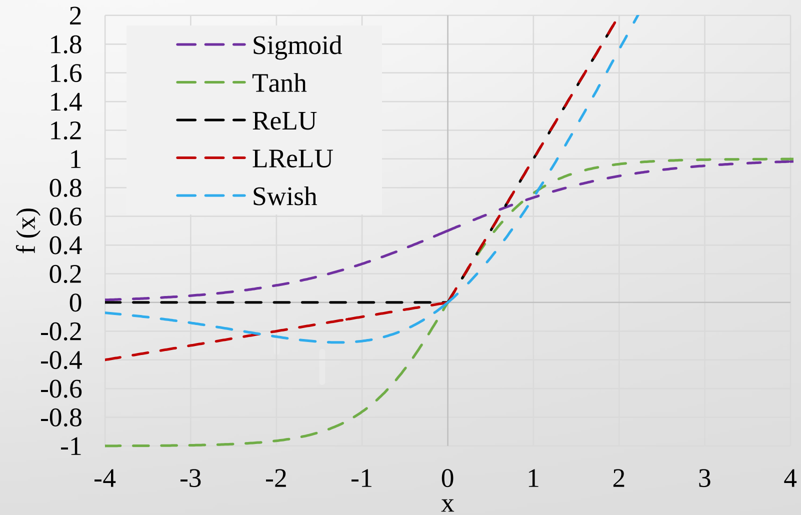


Fig. 3. Activation functions

Privacy-Preserving Neural Networks

The training process consists of developing a mapping from the input to the output space based on the modification of *w* of each neuron

- In the HE domain, the training process implies *large encrypted messages* and *several bootstrapping executions*

1. The bootstrapping reduces the noise in the ciphertext
2. Noise guarantees certain level of security
3. Each operation increases the underlying noise
4. Message decryption fails when noise overpasses a certain threshold

“The computational cost of seven-layer CNN training is around ***one hour*** *with a conventional CPU, while to train the same CNN with HE requires around* ***a year*** *[13]”*

Privacy-Preserving Neural Networks

- Two options are common to deal with the bootstrapping during NN-HE training

1. Acceleration of bootstrapping. The use of high-performance, distributed, and parallel computing provide tools for training over large encrypted datasets
   - Hardware accelerators (GPU, FPGA, etc.) and customized chips (ASIC)

2.1 Avoiding bootstrapping operations focuses on decrypting the ciphertext inside a secure entity (client-server, secured HPC, etc.)

- An hybrid model between HE and SMC

2.2 Public weight of pre-trained NNs. The training phase is performed over unencrypted data to avoid overhead (the evaluation is done over encrypted data)

- Current practice

The NN-NE evaluation involves efficient implementations of *weighted-sum and f*

- Multiplication operation is slower and adds large amounts of noise
- A bootstrapping operation is necessary when a ciphertext contains too much noise

South Ural State University
School of Electronic Engineering and Computer Science
Problem-Oriented Cloud Computing Environment International Laboratory

Privacy-Preserving Logistic Regression as a Cloud Service based on Residue Number System

Jorge M. Cortés-Mendoza
Andrei Tchernykh
Mikhail Babenko
Luis B. Pulido-Gaytán
Gleb Radchenko
Franck Leprevost
Xinheng Wang
Arutyun Avetisyan
Sergio Sergio Nesmachnow

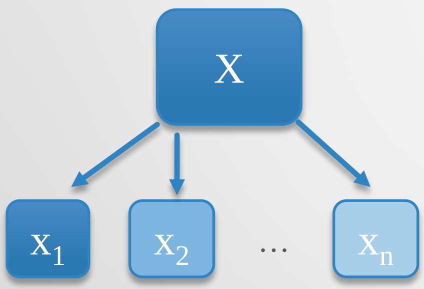
Russia, December 2020.

Residue Number System

Residue Number System (RNS) is a variation of finite ring isomorphism where original numbers are represented as residues

- A moduli set of pairwise co-prime numbers $\{p_1, p_2, \ldots, p_n\}$ defines the representation of the values in the range of $P = \prod_1^n p_i$

An integer number $X \in [0, P-1)$ is defined in RNS as a tuple $(x_1, x_2, \ldots, x_n)$ where x_i represents the remainder of the division of X by p_i



$p_1 = 7, p_2 = 5, p_3 = 3, p_4 = 2,$

$8_{10} = (1, 3, 2, 0)_{RNS}$

1 ← quotient

divisor→ 7 | 8 ← dividend

1 ← remainder

$$X \otimes Y = Z$$

$$\text{4-moduli}\begin{cases} x_1 \otimes y_1 \bmod m_1 = z_1 \\ x_2 \otimes y_2 \bmod m_2 = z_2 \\ x_3 \otimes y_3 \bmod m_3 = z_3 \\ x_4 \otimes y_4 \bmod m_4 = z_4 \end{cases}$$

$\otimes$ denotes $+, -, \times$

Logistic Regression

Logistic Regression (LR) is a statistical method for analyzing information where:

- A dataset $X^{(i)} \in \mathbb{R}^d$ and their labels $Y^{(i)} \in \{0,1\}$ for $i = 1, 2, \ldots, n$ are used to model a binary dependent variable
- The predict of a binary outcome considers the logistic function

The inference of LR considers the hypothesis $h_\theta\left(X^{(i)}\right) = g\left(\theta^T X^{(i)}\right)$ where

- Logistic function: $g(z) = \frac{1}{1+e^{-z}}$
- Weights: $\theta^T = [\theta_0, \theta_1, \ldots, \theta_d]^T$
- Data: $X^{(i)} = [1, X_1{}^{(i)}, X_2{}^{(i)}, \ldots, X_d{}^{(i)}]^T$

The training phase of LR focuses on finding θ^*, the values of θ that minimizes the number of errors in the prediction

- θ^* is used to estimate the binary classification of new data
- For $X' = [1, X_1, \ldots, X_d] \in \mathbb{R}^{d+1}$ is possible to guess its binary value $Y' \in \{0,1\}$ by

$$Y' = \begin{cases} 1 & if\ h_{\theta^*}(X') \geq \tau \\ 0 & if\ h_{\theta^*}(X') < \tau \end{cases}$$

- τ defines a variable threshold in $0 < \tau < 1$, typically with value equal to 0.5

Privacy-Preserving Logistic Regression

- Gradient Descent (GD) is an optimization algorithm to minimize the error function or objective function $J(\theta)$
- The optimization process updates θ according to $\nabla_\theta J(\theta)$, a partial derivate of $J(\theta)$
- The learning rate α defines the dimension of the steps

Algorithm 1. Batch Gradient Descent

Input: $X, Y, \theta, \alpha, and\ maxIter$
Output: θ
1 For $i \leftarrow 1$ to $maxIter$
2 $\theta \leftarrow \theta - \alpha \nabla_\theta J(\theta, X, Y)$
3 Return θ

$$J(\theta) = -\frac{1}{\mathrm{n}} \sum_{i=1}^{n} y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)$$

We propose a data confidentiality LR for cloud service with HE based on RNS

Table 5. Main characteristics of HE schemas for logistic regression

Encryption	Degree of polynomial approximation	Gradient descent	Metrics	Library	Datasets	Ref.
Paillier, LWE, Ring-LWE	2	BGD	F-score, AUC	-	Pima, SPECTF	[14]
Ring-LWE	1	GD-FHN	ROC, accuracy	NFLlib	iDASH, financial data	[15]
Ring-LWE	3,5, 7	NAG	AUC, accuracy	HEAAN	iDASH, lbw, mi, nhanes3, pcs, uis	[16]
Ring-LWE, RNS	7	NAG	AUC, accuracy	HEAAN	Lbw, uis	[17]
Ring-LWE	5	NAG	AUC	HEAAN	MNIST, credit	[18]
-	-	BGD	AUC	-	NIDDK	[19]

Privacy-Preserving Logistic Regression

Four main variants of the original GD are commonly used in the literature:

- Batch Gradient Descent (BGD)
- Stochastic Gradient Descent (SGD)
- Momentum Gradient Descent (MGD)
- Nesterov Accelerated Gradient (NAG)

Algorithm 2. Stochastic Gradient Descent

Input: $X, Y, \theta, \alpha, and\ iters.$

Output: $\theta.$

```
For i ← 1 to iters
  Shuffle (X, Y)
  For j ← 1 to length(X)
    θ ← θ − α ∇_θ J(θ, x^(j), y^(j))
Return θ
```

Algorithm 3. Momentum Gradient Descent

Input: $X, Y, \theta, \alpha, \beta, and\ iters.$

Output: $\theta.$

```
For i ← 1 to iters
  Shuffle (X, Y)
  For j ← 1 to length(X)
    v_t ← β v_{t−1} − α ∇_θ J(θ, x^(j), y^(j))
    θ ← θ + v_t
Return θ
```

Algorithm 4. Nesterov Gradient Descent

Input: $X, Y, \theta, \alpha, \beta, and\ iters$

Output: $\theta.$

```
For i ← 1 to iters
  Shuffle (X, Y)
  For j ← 1 to length(X)
    v_t ← β v_{t−1} − α ∇_θ J(θ − β v_{t−1}, x^(j), y^(j))
    θ ← θ + v_t
Return θ
```

Privacy-Preserving Logistic Regression

Each iteration of the algorithm, all records in the training set are used to update the values of θ

- **hTheta** provides a polynomial approximation to the logistic function

HE.rescale eliminates the accumulated scaling factor generated after each multiplication

Algorithm 5. HE-RNS Batch Gradient Descent

Input: $X, Y, theta, alpha, maxIter$
Output: $theta$
For $iter$ ← 1 to $maxIter$
    For i ← 1 to X.size
        parcialCost ← **HE.sub** (**hTheta** (X[i], $theta$), Y[i]))
        For j ← 1 to $theta$.size
            cost[j] ← **HE.add** (cost[j], **HE.mul** (parcialCost, X[i][j]))
    For i ← 1 **to** $theta$.size
        cost[i] ← **HE.rescale** (~~**HE.mul**~~(average, ~~**HE.mul**~~ (cost[i], $alpha$)))
        $theta$[i] ← **HE.sub** ($theta$[i], cost[i])
Return $theta$

We propose a data confidentiality LR for cloud service with HE based on RNS

Configuration setup

Experimental analysis considers 30 configurations for each dataset to compare the performance and quality of our solution with the state of the art algorithms

- Six datasets from medicine and genomics
- Polynomial approximation of logistic function
- 5-fold cross-validation
- A scalar factor of 16 bits
- Seven pair-wise relatively primes
- Iterations: 5, 10, 15, 20, 25
- Learning rate: 1.6, 1.1, 0.6, 0.1, 0.06, 0.01, 0.006, 0.001, 0.0006, 0.0001

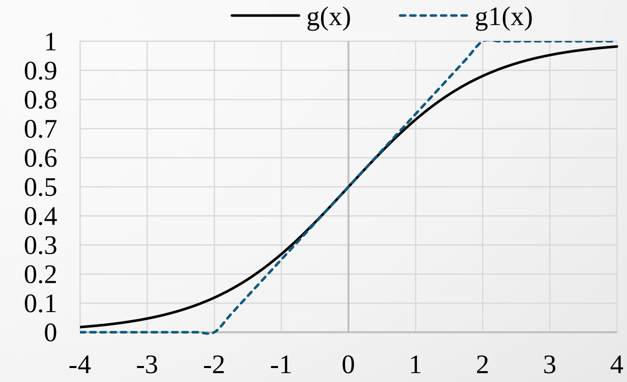


Fig. 5. Sigmoid and approximation functions

Metrics

- Accuracy (A) expresses the systematic error to estimate a value
- The Area Under the ROC Curve (AUC) is a performance indicator of classifiers

Configuration setup

We consider six datasets widely used in the literature

1. Low Birth Weight (Lbw) dataset consists of information about births to women in an obstetrics clinic
2. Myocardial Infarction (Mi) is a heart disease dataset
3. National Health and Nutrition Examination Survey (Nhanes3) includes a database of human exposomes and phenomes
4. The Indian's diabetes dataset (Pima)
5. Prostate Cancer Study (Pcs) dataset of patients with and without cancer of prostate
6. Umaru Impact Study (Uis) dataset stores information about resident treatment for drug abuse

Table 6. Datasets characteristics and size of sets

Dataset	N	Features	N-Training	N-Testing
Lbw	189	9	151	38
Mi	1,253	9	1,002	251
Nhanes3	15,649	15	12,519	3,130
Pima	768	8	614	154
Pcs	379	9	303	76
Uis	575	8	460	115

Results

Table 7 presents the best average values of *UAC* and *A* for all GD versions, each value represents the average of 30 execution with different initial values of θ

Table 7. Average AUC and A

Name	*AUC*							*A*(%)						
	Lbw	Mi	Nhanes3	Pcs	Pima	Uis	Average	Lbw	Mi	Nhanes3	Pcs	Pima	Uis	Average
HE-BGD-RNS	0.7358	0.9388	0.8112	0.7445	0.6983	0.5483	0.7557	71.84	88.87	78.89	66.05	67.79	74.43	76.02
BGD	0.7353	0.9357	0.7961	0.7406	0.6964	0.5458	0.7507	71.84	89.02	78.86	66.14	67.65	74.35	76.04
HE-SGD-RNS	0.7541	0.9421	**0.9029**	**0.8151**	0.8505	**0.6118**	0.8052	**73.42**	88.9	84.51	**66.32**	**74.7**	**74.81**	77.59
SGD	0.7618	**0.9445**	**0.903**	0.8162	0.8487	**0.6158**	0.8083	73.86	89.39	84.3	66.32	74.7	**74.75**	77.72
HE-MGD-RNS	**0.7552**	**0.9445**	0.902	0.8143	**0.8508**	0.6116	0.8055	72.89	**88.95**	**84.53**	66.01	74.66	74.72	77.42
MGD	**0.7634**	**0.9445**	**0.903**	0.8169	0.8488	0.6152	0.8086	73.86	**89.42**	**84.33**	**66.36**	74.77	74.72	77.74
HE-NAG-RNS	**0.7552**	**0.9445**	0.902	0.8143	**0.8508**	0.6115	0.8055	72.81	**88.95**	**84.53**	66.01	**74.7**	74.72	77.40
NAG	0.763	**0.9445**	**0.903**	**0.817**	**0.8489**	0.6154	0.8086	**74.04**	**89.42**	**84.33**	**66.36**	**74.79**	74.72	77.77
HE-NA-LR [16]	0.689	0.958	0.717	0.74	-	0.603	0.7414	69.19	91.04	79.22	68.27	-	74.44	76.43
HE-SS-LR [14]	-	-	-	-	0.8763	-	-	-	-	-	-	80.7	-	-

For *AUC*, HE-SGD-RNS, HE-MGD-RNS, and HE-NAG-RNS provide the best-found solutions in three datasets

For *A*, HE-SGD-RNS and HE-NAG-RNS found three times the best values of θ

The maximal difference between RNS and non-homomorphic algorithms are:

- 1.51 % for *AUC* with HE-BGD-RNS and Nhanes3
- 1.23 % for *A* with HE-NAG-RNS and Lbw

Results

Table 8 present the best values of *AUC* and *A*, each value represents the best θ of 1,500 execution: learning rates × iters × initial values

Table 8. Best AUC and A

Name	*AUC*							*A*(%)						
	Lbw	Mi	Nhanes3	Pcs	Pima	Uis	Average	Lbw	Mi	Nhanes3	Pcs	Pima	Uis	Average
HE-BGD-RNS	0.7981	0.9485	0.8509	0.8045	0.795	0.585	0.7974	78.95	90.44	79.74	77.63	74.66	76.52	80.66
BGD	0.8013	0.947	0.8317	0.8061	0.7946	0.5846	0.7941	78.95	90.84	79.36	**77.63**	73.97	76.52	80.66
HE-SGD-RNS	0.7949	0.9536	**0.9039**	**0.8357**	0.8602	0.6604	0.8297	**81.58**	**91.24**	**86.01**	**78.95**	**79.45**	76.52	82.86
SGD	0.7949	0.9557	**0.9039**	0.8341	0.86	0.66	0.8297	81.58	**91.24**	**86.17**	**77.63**	**80.14**	76.52	82.63
HE-MGD-RNS	**0.8125**	**0.9541**	0.9033	0.8341	**0.8608**	**0.6632**	0.8334	**81.58**	90.84	85.88	**78.95**	**79.45**	**79.13**	83.28
MGD	0.8045	0.9562	**0.9039**	**0.8518**	0.8627	0.6596	0.8352	81.58	**91.24**	85.88	**77.63**	78.77	**77.39**	82.74
HE-NAG-RNS	0.8013	0.9536	0.9033	0.8349	0.8596	0.6584	0.8303	**81.58**	**91.24**	85.94	**78.95**	**79.45**	76.52	82.85
NAG	**0.8077**	**0.9574**	**0.9039**	0.8486	**0.8631**	**0.6616**	0.8358	**84.21**	**91.24**	85.97	**77.63**	79.45	76.52	83.11
HE-NA-LR	0.689	0.958	0.717	0.740	-	0.603	0.7414	69.19	91.04	79.22	68.27	-	74.44	76.43
HE-SS-LR	-	-	-	-	0.8763	-	-	-	-	-	-	80.7	-	-

For *AUC*, HE-MGD-RNS provides the best θ in four of the six datasets, it is followed by HE-SGD-RNS with only two

For *A*, HE-SGD-RNS outperforms HE-MGD-RNS and HE-NAG-RNS in five datasets

The maximal difference between RNS and non-homomorphic algorithms are

- 1.92 % for HE-BGD-RNS with Nhanes3 dataset in *AUC*
- 2.63 % for HE-NAG-RNS with respect to Lbw dataset in *A*

Future work

Privacy-Preserving Neuronal Networks

1. Polynomial approximation of activation function $\ddot{f}$
2. Bootstrapping
   - Acceleration
   - Secure Multi-party Computation
   - Pre-trained NNs models

Privacy-Preserving Logistic Regression with RNS

1. Level of security
2. Polynomial approximation of logistic function

Publications

1. Luis Bernardo Pulido-Gaytan, Andrei Tchernykh, **Jorge M. Cortés-Mendoza**, Mikhail Babenko, Gleb Radchenko, Arutyun Avetisyan, and Alexander Yu. Drozdov. Privacy-Preserving Neural Networks via Homomorphic Encryption: Challenges and Opportunities. ***Peer-to-Peer Networking and Applications: Special Issue on Advances in Privacy-Preserving Computing***, Springer. IF 2.793, Q2. July 2020 (under review).
2. Andrei Tchernykh, Luis Bernardo Pulido-Gaytan, Mikhail Babenko, **Jorge M. Cortés-Mendoza**, Gleb Radchenko, Arutyun Avetisyan, Alexander Yu. Drozdov. Privacy-Preserving Toward Fast and Accurate Polynomial Approximations for Practical Homomorphic Evaluation of Neural Network Activation Functions. ***International Workshop on Security, Privacy and Performance of Cloud Computing*** (SPCLOUD 2020), Barcelona, Spain. December 2020 (accepted).
3. Luis Bernardo Pulido-Gaytan, Andrei Tchernykh, **Jorge Mario Cortés-Mendoza**, Mikhail Babenko, Gleb Radchenko. A Survey on Security-Preserving of Machine Learning with Fully Homomorphic Encryption. ***CARLA 2020 -The Latin America High Performance Computing Conference***. Cuenca, Ecuador. September 2020 (accepted).

Publications

4. **Jorge M. Cortés-Mendoza**, Gleb Radchenko, Andrei Tchernykh, Luis Bernardo Pulido-Gaytan, Mikhail Babenko, Arutyun Avetisyan, Alexander Yu. Drozdov, and Sergio Nesmachnow. Privacy-Preserving Logistic Regression Solutions based on Residue Number System: Design and Analysis. ***2nd Workshop on Secure IoT, Edge and Cloud systems (SIoTEC) 2021***, Melbourne, Australia. May 2021 (under submission).
5. Mikhail Babenko, Andrei Tchernykh, Bernardo Pulido-Gaytan, Elena Golimblevskaia, **Jorge M. Cortés-Mendoza**, Arutyun Avetisyan. Experimental Evaluation of Homomorphic Comparison Methods. ***ISPRAS OPEN 2020 - Ivannikov ISP RAS Open Conference***, Moscow, Russia, December 10-11, 2020 (under review)
6. **Jorge M. Cortés-Mendoza**, Andrei Tchernykh, Mikhail Babenko, Luis Bernardo Pulido-Gaytán, and Gleb Radchenko. Privacy-Preserving Logistic Regression with Residue Number System as a Cloud Service. ***RuSCDays'20 - The Russian Supercomputing Days***. Moscow, Russia. September 2020 (accepted).

References

[1] Vaikuntanathan, V.: Computing Blindfolded: New Developments in Fully Homomorphic Encryption. In: IEEE 52nd Annual Symposium on Foundations of Computer Science, Palm Springs. pp. 5–16 (2011).

[2] Armknecht, F., Boyd, C., Carr, C., Gjøsteen, K., Jäschke, A., Reuter, C.A., Strand, M.: A Guide to Fully Homomorphic Encryption, IACR Cryptology ePrint Archive, (2015).

[3] Naehrig, M., Lauter, K., Vaikuntanathan, V.: Can Homomorphic Encryption Be Practical? In: 3rd ACM Workshop on Cloud Computing Security Workshop - CCSW '11. pp. 113– 124 (2011)

[4] Archer, D., Chen, L., Cheon, J.H., Gilad-Bachrach, R., Hallman, R.A., Huang, Z., Jiang, X., Kumaresan, R., Malin, B.A., Sofia, H., Song, Y., Wang, S.: Applications of Homomorphic Encryption. (2017)

[5] Acar, A., Aksu, H., Selcuk Uluagac, A., Aksu, H., Uluagac, A.S.: A Survey on Homomorphic Encryption Schemes: Theory and Implementation. ACM Comput. Surv. 51, (2018). https://doi.org/10.1145/3214303

[6] Martins, P., Sousa, L., Mariano, A.: A Survey on Fully Homomorphic Encryption: An Engineering Perspective. ACM Comput. Surv. 50, 33 (2017). https://doi.org/10.1145/3124441

[7] Parmar, P. V, Padhar, S.B., Patel, S.N., Bhatt, N.I., Jhaveri, R.H., S'ad Vidya, S., Shri S'ad, M., Mandal, V.: Survey of Various Homomorphic Encryption Algorithms and Schemes. Int. J. Comput. Appl. 91, (2014)

[8] Sobitha Ahila, S., Shunmuganathan, K.L.: State Of Art in Homomorphic Encryption Schemes. Int. J. Eng. Res. Appl. 4, 37–43 (2014)

[9] Gentry, C.: Computing on the Edge of Chaos: Structure and Randomness in Encrypted Computation. In: Proceedings of the International Congress of Mathematicians (2014)

[10] Aguilar-Melchor, C., Fau, S., Fontaine, C., Gogniat, G., Sirdey, R.: Recent Advances in Homomorphic Encryption: A Possible Future for Signal Processing in the Encrypted Domain. IEEE Signal Process. Mag. 30, 108–117 (2013). https://doi.org/10.1109/MSP.2012.2230219

References

[11] Hrestak, D., Picek, S.: Homomorphic Encryption in the Cloud. In: 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO'14). pp. 1400–1404 (2014)

[12] Moore, C., O'Neill, M., Hanley, N., O'Sullivan, E.: Accelerating integer-based fully homomorphic encryption using Comba multiplication. In: IEEE Workshop on Signal Processing Systems, SiPS. pp. 1–6. IEEE (2014)

[13] Rondeau, T.: Data Protection in Virtual Environments (DPRIVE). (2020)

[14] Aono, Y., Hayashi, T., Trieu Phong, L., Wang, L.: Scalable and Secure Logistic Regression via Homomorphic Encryption. In: Proceedings of the Sixth ACM on Conference on Data and Application Security and Privacy - CODASPY '16. pp. 142–144. ACM Press, New York, New York, USA (2016). https://doi.org/10.1145/2857705.2857731.

[15] Bonte, C., Vercauteren, F.: Privacy-preserving logistic regression training. BMC Med. Genomics. 11, 86 (2018). https://doi.org/10.1186/s12920-018-0398-y.

[16] Kim, A., Song, Y., Kim, M., Lee, K., Cheon, J.H.: Logistic regression model training based on the approximate homomorphic encryption. BMC Med. Genomics. 11, 83 (2018).

[17] Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y.: A Full RNS Variant of Approximate Homomorphic Encryption. Presented at the (2019). https://doi.org/10.1007/978-3-030-10970-7_16.

[18] Cheon, J.H., Kim, D., Kim, Y., Song, Y.: Ensemble Method for Privacy-Preserving Logistic Regression Based on Homomorphic Encryption. IEEE Access. 6, 46938–46948 (2018).

[19] Yoo, J.S., Hwang, J.H., Song, B.K., Yoon, J.W.: A Bitwise Logistic Regression Using Bi-nary Approximation and Real Number Division in Homomorphic Encryption Scheme. Presented at the (2019). https://doi.org/10.1007/978-3-030-34339-2_2.

Thank you

Questions?

South Ural State University
School of Electronic Engineering and Computer Science
Problem-Oriented Cloud Computing Environment International Laboratory

Seminar

Privacy-Preserving Machine Learning as a Service

Speaker

Jorge Mario Cortés-Mendoza

Russia, December 2020.