Received February 10, 2019, accepted February 19, 2019, date of publication February 25, 2019, date of current version March 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2901219

MSCryptoNet: Multi-Scheme Privacy-Preserving Deep Learning in Cloud Computing

OWUSU-AGYEMANG KWABENA, ZHEN QIN, TIANMING ZHUANG, AND ZHIGUANG QIN
School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

Corresponding author: Zhen Qin (qinzhen@uestc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672135, in part by the National Science Foundation of China–Guangdong Joint Foundation under Grant U1401257, in part by the Sichuan Science-Technology Support Plan Program under Grant 2018GZ0236 and Grant 2017FZ0004, and in part by the Fundamental Research Funds for the Central Universities under Grant 2672018ZYGX2018J057.

ABSTRACT Privacy in the Internet of Things is a fundamental challenge for the Ubiquitous healthcare systems that depend on the data aggregated and collaborative deep learning among different parties. This paper proposes the MSCryptoNet, a novel framework that enables the scalable execution and the conversion of the state-of-the-art learned neural network to the MSCryptoNet models in the privacy-preservation setting. We also design a method for approximation of the activation function basically used in the convolutional neural network (i.e., Sigmoid and Rectified linear unit) with low degree polynomials, which is vital for computations in the homomorphic encryption schemes. Our model seems to target the following scenarios: 1) the practical way to enforce the evaluation of classifier whose inputs are encrypted with possibly different encryption schemes or even different keys while securing all operations including intermediate results and 2) the minimization of the communication and computational cost of the data providers. The MSCryptoNet is based on the multi-scheme fully homomorphic encryption. We also prove that the MSCryptoNet as a privacy-preserving deep learning scheme over the aggregated encrypted data is secured.

INDEX TERMS Internet of Things, privacy-preserving, fully homomorphic encryption.

I. INTRODUCTION

With the emergence of internet of things (IoT), deep neural network is gradually subsumed by the more challenging goal of training models in the private domain [1], [2]. More specifically, the concept of data privacy is a broad field concerned with accountability, secrecy, fairness, correctness and availability regarding user data [3]. These characteristics are all applied to deep neural network (DNN) to various extent depending on the application domain. There is no compromise or tradeoff with these data processing requirements. Since they are mandatory EU General Data Protection Regulation [4] and Health Insurance Portability and Accountability Act of 1996 [5] HIPAA, and also adds constraints that principally cannot be trivialized. Potential privacy and accuracy losses arises during computations [6] or training of the deep learning model.

There are two major existing settings to protect data privacy or its aftermath: Perturbation-based and cryptography-based method. Perturbation-based [6]–[8] approach modifies data with noise to provide protection: summary of statistics remains approximately the same to aid avoidance of inferring information about any specific record. However, the alteration requires careful calibration to put model usability and privacy at equilibrium. Nonetheless, privacy of the attributes are confronted with security threats since they still remain in the plaintext. Arguably, cryptography-based approach which is based on fully homomorphic encryption (FHE) [9] has recently been touted as a promising solution. It enables encrypted data with the public keys to be uploaded to the cloud service provider for *secure multi-party computations* (SMC) to generate encrypted intermediate results. At this point, the data services provider cannot access any user record since it does not have the secrete keys but rather it only serve as a computational platform. This powerful recent advances in cloud server has made it a preferred choice for the integration of deep learning [10]–[12] which is capable of building a convolutional neural network (CNN) models on FHE to process inference queries.

Obviously, DNN model generalization performance is highly stimulated by the quality and volume of datasets used

The associate editor coordinating the review of this manuscript and approving it for publication was Kuo-Hui Yeh.

in the training process [13]. Collaborative DNN learning has been the preferred option in improving the model with the incorporation of more datasets which cannot be provided by a sole local datasets.

A. CHALLENGES

For example, several medical institutions deploy internet of things (IoT) devices in a Ubiquitous Healthcare (U-Health) System to collect health data through wearables or mobile devices to produce new diagnostic models with the ability to find the correlation between symptoms and diagnosis from these encrypted patient records. A frequent concern in such markets are confronted with three major problems:

Q1 The application of multi-key fully homomorphic encryption (MK-FHE) [15]–[17] in [18] have the somewhat undesirable property of being "*single-hop for key*". All relevant keys must be known at the beginning of the homomorphic computations and the output cannot be usefully combined with ciphertexts encrypted under other keys(unless an expensive bootstrapping step is performed) therefore increasing the cost of computing, thus making it very undesirable for large scale neural network.

Q2 Another interesting open challenge in [18] is whether every FHE scheme can be made multikey in MK-FHE during data aggregation for the collaborative deep learning.

Q3 The accuracy, privacy and stability of the training model are drastically affected since the activation functions are not cryptographically computable and it requires polynomial approximation of activation function.

In this paper, we propose a novel framework dubbed *Multi-Scheme Crypto Deep Neural Network* (*MSCryptoNet*). This is to resolve the three outlined challenges in the existing approaches [18]. The intrinsic strategy is to demonstrate that even if the aggregated ciphertexts for the training of the collaborative DNN model are encrypted with *different encryption schemes* or *even with different keys*, the existence of algorithm ***Evaluate*** enable us to produce the ciphertext ψ of the desired result of classifier $\boldsymbol{C}$. This system offers security guarantee in the *Honest-but-curious* by the underlying cryptosystem, i.e. adversaries only have negligible chance to disclose the confidential data since the records are *encrypted globally*. It provides more practicable and efficiency than the application of MK-FHE in [18]. MSCryptoNet, solves the Q1. Note that it will be practically impossible to convert all FHE keys of schemes to multi-key during data aggregation for the training of DNN model. We design methods for approximation of the cryptographically incompatible action functions which is commonly used in CNN (i.e ReLU (Rectified Linear Unit) Sigmoid) with low degree polynomials which is essential for efficient FHE schemes. This is to resolve the issues in Q2 and Q3.

By carefully exploring the unique characteristics of deep neural networks and examining the existing cryptographic methods, we present a collaborative privacy-preserving protocol for securing multiple datasets from different entities. Our main contributions are summarized in three-fold:

- We propose MSCryptoNet, a novel collaborative privacy-preserving, computationally efficient, Multi-Scheme Fully homomorphic encryption-based architecture which is capable of resolving the major challenges in [18] and other similar existing challenges.
- We apply a theoretical foundation to find the lowest degree polynomial approximation of a function within a certain error range in the decomposed CNN.
- MSCryptoNet will be subjected to security analysis to guarantee the privacy-preservation of the scheme

B. ORGANIZATION OF THE PAPER

The paper is organized as follows. Section II provides a literature review over privacy-preserving deep learning. Section III presents some notations and definitions on cryptographic primitives and deep learning. In Section IV, we present the system architecture, the problem statement and the adversary model. In Section V, we provide the privacy-preserving scheme and security analysis. Then, we present performance evaluation in Section VI. Finally, the conclusions and directions for future work are presented in Section VII.

II. RELATED WORKS

In cloud computing, Deep learning has progressively demonstrated its success in domains such as speech recognition [29] fraud detection, medical data analysis [30], image recognition [29], [31] and signal processing [32], [33]. Deep learning is capable of transforming a high dimensional data into a more abstract expression. It means that, deep learning uses hierarchical layers of latent features to reconstruct high multidimensional predictors in input-output models from low-dimensional data. Through these transformation, complex computational functions can be learned. Zhang [34] and Hinton and Salakhutdinov [35] demonstrated the excellent learnability of the multiple hidden layers of the deep neural network. The generalization performance of deep learning is highly dependent on the quality and volume of data used in the training process. The feature extraction by learning are more intrinsic characterization of the data to accelerates and improve classification or visualization of the data. They also demonstrated that optimizing the weights in the nonlinear auto-encoders is extremely difficult and can be overcome by the layer-by-layer 'pre-training' process. Basically, the deep neural network framework are constructed as a multi-layer neural networks. There are diverse state-of-the-art neural architectures and training algorithms that has been used in training the plethora of dataset generated from medical internet of things devices. Some of such frameworks are feed-forward neural network, backpropagation, recurrent neural network, convolutional neural networks etc. However, these neural network models and their training datasets are exposed to privacy concerns. It is therefore important to consider the privacy of these two distinguished area of study in our work.

A. PRIVACY-PRESERVING DEEP LEARNING

With the advancement in cloud computing, there has been numerous existing works considering the problems confronting the security of the cloud architecture [36], [37], cloud storage [38]–[40], data mining [30], [41], authentication of protocols [53], security robustness [54], flexible access control of sharing files [55]and privacy-preserving model [7], [42] of the deep learning model and the training dataset in the cloud. Existing works on privacy-preserving cryptographic schemes in the cloud are basically based on secure multi-party computations. The cryptographic algorithms are based on two different domains i.e. (a) training on the cloud server (b) training without the help of the cloud server. For example, Chen and Zhong [43] proposes a two party distributed privacy-preserving algorithm based on back-propagation called BPNN. Whiles Bansal *et al.* [44] presented a similar BPNN over arbitrarily portioned data between two parties. They used ElGamal scheme to support the secure two-party computational operations. These algorithm enables the neural network to be trained without any data providers sensitive information leakage. They applied fully homomorphic encryption scheme to preserve the privacy of the data and intermediate results. Their proposed algorithms conducts the vertically partitioned data, which means each data provider has a subset of the feature vector. However, there are lack of solutions for multiple parties, each with each with an arbitrarily partitioned dataset to conduct collaborative deep learning. Yuan and Yu [45] proposes a practical solution to this problem by utilizing the cloud computing resources. Generally speaking, the application of [43]–[45] algorithms to the multi-party scenario might lead to the expensive communication overhead. Bos *et al.* [46] presented a novel privacy-preserving model called CryptoNet, which allows the data provider to outsource a homomorphically encrypted data to ensure that the datasets remains private. CryptoNet is a feedforwarding pretrained model applied on encrypted data outsourced to the cloud by data providers. CryptoNet which used fully homomorphic encryption scheme [47] to evaluate deep convolutional neural network with two convolutional layers and two fully connected layers. The continues quest to preserve the privacy of data lead to the proposal of CryptoNet [10] which allow the deep neural network on fully homomorphic encrypted data allowing the collaborative parties to receive these services without exposure of the sensitive data to neither the cloud service provider and other data providers. CryptoDL is a combination of the convolutional neural network with leveled homomorphic encryption based on Shamir's secret sharing model. Li *et al.* [18] also presented a new and efficient protocols for privacy-preserving neural network training using the stochastic gradient descent method called SecureML. This algorithm falls in the two-server model where data providers distribute their private data among two non-colluding servers. Training of the various deep neural network models are performed on the collaborative data using the secured two-party computations. SecureML also support secure arithmetic computations on shared decimal numbers. The goal of MSCryptoNet in this paper differs from that of [10], [50], and [54], as our protocol and [49] aims at training the deep neural network model on the encrypted aggregated dataset from multiple data providers, while CryptoNet is an already trained model. However, none of the existing privacy-preserving schemes are able to evaluate deep neural networks whose inputs are encrypted with possibly different encryption schemes. In this scenario we consider the cloud service provider as the adversary whiles the learning data providers are seen as honest-but-curious entities. In our scenario the data providers are organizations that generate huge among of data from their internet of things devices such as hospitals, insurance companies and financial institutions whose operations are highly governed by law. In this paper, we propose solutions to overcome this security and privacy challenge. We demonstrate procedures in achieving seemingly impossible problem of these privacy-preserving deep learning for training over encrypted datasets with possibly different scheme and /or different public keys limited to a small value. In MSCryptoNet, the data providers encrypt dataset before outsourcing it to the cloud server. The neural network computations are performed by the cloud service provider whiles data providers are only capable of obtaining intermediate result from the training model.

III. PRELIMINARIES

TABLE 1. Notations of our protocol.

Acronym	Description
DP	Data Providers
DSP	Data Service Provider
CP	Computational Party
(pk_i, sk_i)	Data providerÂŠs public and secret key
(pk_0, sk_0)	Crypto service providerÂŠs public and secret key
$\mathcal{C}$	Classifier
$\mathcal{C}(x, \theta)$	Prediction of $\mathcal{C}$ with input x and θ

A. DEEP LEARNING

Backpropagation in DNN is one of the most widely used learning models based on gradient descent. For simplicity, the following discussion is on fully connected network. The main objective is to classify data samples x into multiple classes. For an n-layer network, w_i and b_i are the weights and biases corresponding to the i-th layer. The weighted-sum $W_i a_{i-1}$ is calculated by the feedforward propagation for the $i-$th layer, where a_{i-1} is the activation function from the $(i-1)$-th layer and $a_0 = x$. The softmax function is therefore adopted by the output layer to map the high-dimensional vectors into prediction probabilities, i.e y. Immediately the feedforward propagation generates the y, the cost E is then calculated which is the distance between the prediction and the true label t. The weight and the biases are then updated based on the backpropagation: $W_{ij}^{(l)} = W_{ij}^{(l)} - \eta \Delta W_{ij}^{(l)}$ and

$b_i^{(l)} = b_i^{(l)} - \eta \Delta b_i^{(l)}$ where η is the learning rate. The gradients $\Delta W_{ij}^{(l)}$ and $\Delta b_i^{(l)}$ are calculated as follows:

$$\Delta W_{ij}^{(l)} = a_{i-1}\delta_i, \quad \Delta b_i^{(l)} = \delta_i \tag{1}$$

where $\delta_i = \delta_{i-1} W_{i+1} a_1 (1 - a_1)$ for sigmoid activation and $\delta_{n-1} = y - t$ for cross-entropy loss.

B. FULLY HOMOMORPHIC ENCRYPTION

Fully homomorphic encryption (FHE) supports meaningful [19] unlimited addition and multiplication computations over encrypted data with results of addition and multiplications of plaintext by the application of the corresponding ciphertext directly without decryption. To cope with the computations in DNN, we adopt a well-known Fully homomorphic encryption system called *Multi-Scheme Fully Homomorphic Encryptions* (MS-FHE) [20]. MS-FHE support a given circuit C (classifier) for the evaluation algorithm $Eval_F$, a public key pk_F and any ciphertext ψ_i is generated by $Enc_{pk^F}(m_i)$, it outputs a refreshed ciphertext ψ_* such that $Dec_{sk_F}(\psi_*) = C$ $(m_1, \ldots m_n)$. A key concept in our cryptosystem is designed by the algorithm *ciphertext tree*. A *ciphertext tree* is a representation of a ciphertext that has been encrypted by multiple scheme or even different keys. The model parameters in the deep learning training are typically implemented in the floating point numbers for high accuracy.

IV. MSCryptoNet ARCHITECTURE

In this paper, we adopt the honest-but-curious model, which is a standard adversary model serving as the basis for many state-of-the-art privacy designs [21], [22]. The honest-but-curious cloud server is curious to derive information from the encrypted data, but only by taking a deterministic step not by performing a brute-force attacks. The goal of MSCryptoNet is to develop a collaborative privacy-preserving learning method where multiple parties holding sensitive data can collaboratively learn a model across all of their datasets whiles minimizing exposure and leakage of their data. In this scenario, the Data Service Provider (DSP), who stores the encrypted records, are considered not trustworthy; also, each data provider may not trust each other, they consider all other parties (including the central server) may collude together.

Let $Db_i = U_{i-1}^n Db_i$ i.e $(Db_1 \bigcap Db_2 \ldots, \bigcap)$ that has records $r_i = pub(r) || sec(r)$, which is the concatenation of public fields $pub(r)$ and secret fields $sec(r)$. Each of this data entities is horizontally partitioned among n parties, $P_i(i \in [1, n])$. The data owners independently generates a pair of public and private keys $(pk_i, sk_i) \longleftarrow KeyGen(1^\lambda)$ to encrypt their private fields to generate a ciphertext: $\psi_i = pub(r_i) || Enc(pk_i sec(r_i))$. The records in the database is encrypted using *different* fully homomorphic encryption scheme or *even different keys*.

Given a relevant scenario, consider Alice using ideal lattice scheme [23] to encrypt $r_{(i)}$ in $Db_{(1)}$ whiles bob is using different scheme from [24] to encrypt $r_{(i)}$ in $Db_{(2)}$. Our solution (MSCryptoNet) provides an evaluation of a classifier C over concatenated ciphertext $Enc(\boldsymbol{X})$ from $(\psi_1, \psi_2, \ldots.., \psi_n)$. $Enc(\boldsymbol{X})$ is an encrypted database constructed from the public keys and ciphertext tuples $\langle pk_i, \psi_i \rangle$ from each data owner. The overview of MSCryptoNet is illustrated in Figure 1.

In MSCryptoNet, the overall collaborative neural network model is still implemented on the DSP. However, the nonlinear activation function(Rectified Linear Unit - ReLU) is

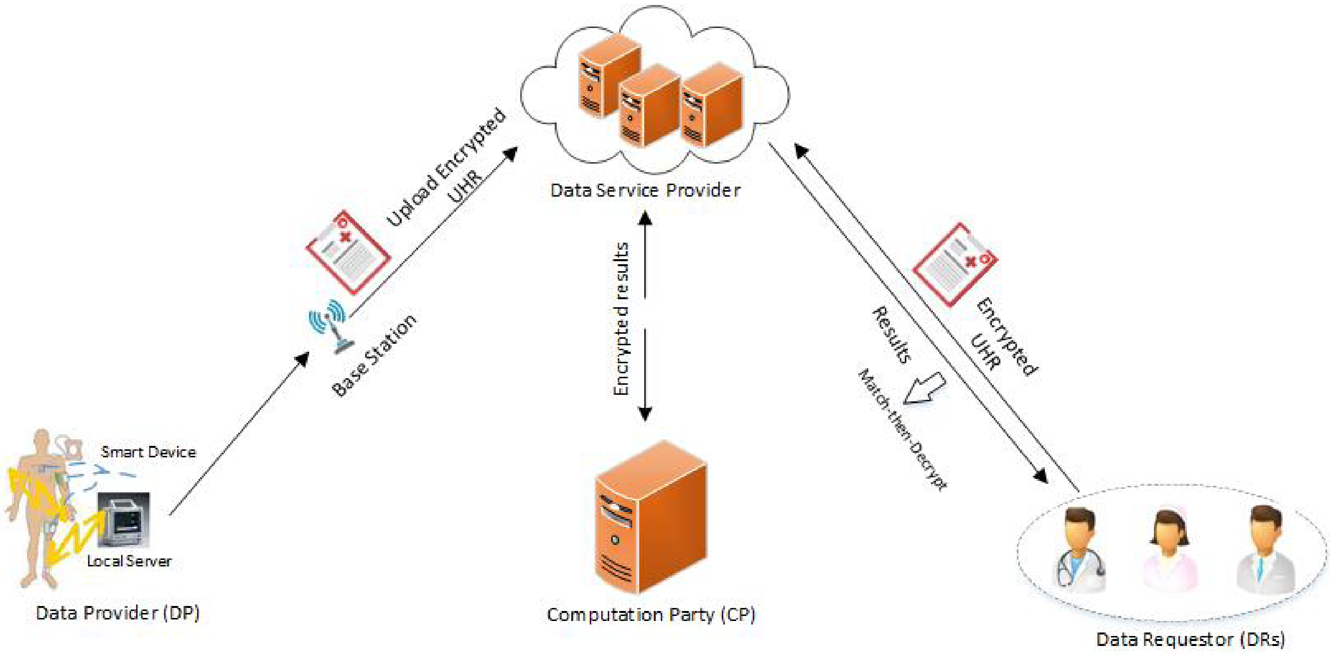

FIGURE 1. MSCryptoNet architecture.

approximated with low degree polynomials which is essential for efficient homomorphic encryption schemes. To this end, data providers jointly negotiate and setup the target vector $t = \{t_i\}_{i=1}^{n}$ and weight matrix $W_{ij}^{(i)}$ in advance, where $t_i = t_1^{(i)}, t_2^{(i)}, \ldots t_{n_i}^{(i)}, j = $1,2 before training the DNN model. There is no need for the existence of a trusted authority.

The proposed MSCryptoNet has two prominent advantages as summarized below. The first advantage of this design is to simulate the structure of derivation of the ReLU function a step function therefor achieving the best approximation of the ReLU function and it will be used for the approximation of MSCryptoNet. This will also improve the accuracy of the model, thus addressing problems Q2 and Q3 as introduced in Section I. The second is the drastic improvement in computational accuracy. Our results demonstrated that, computations of an activation function in MSCryptoNet is 12 times faster than the sigmoid approximation used in [18] and 190 faster than the 5-th order approximation. The overall performance gain is even higher and it enjoys significantly lower complexity.

V. PRIVACY-PRESERVING LEARNING

In this section, we discuss the proposed collaborative privacy-preserved learning algorithm. For a coherent presentation, the following description is based on training on the Data Service Provider with the provision of encrypted datasets by the Data Providers. These process is repeated by all parties.

Algorithm 1 Overall Scheme of MSCryptoNet

```
Input: $\{Db_1, Db_2, \ldots Db_n\}$, with data record
       $r_i = pub(r_i)||sec(r_i)$ $\phi = \{W^1, W^2, b^1, b^2\}$;
       $iteration_{max}$ learning rate $\eta$
Output: $\phi = \{W^1, W^2, b^1, b^2\}$
Data Provider $P_i$ $(i \in [1, k])$ does:;
Initialize the Models parameters randomly;
Sample a key tuple:$(pk_i, sk_i, ek_i) \leftarrow KeyGen(1^k)$ ;
Encrypt private data objects: $Db_i$ $W_i^{(j)}$
$\psi_i \leftarrow pub(r_i)||Enc(pk_i, sec(r_i))$ $d_i \leftarrow Enc(pk_i, W_i^{(j)})$,
$g \leftarrow Enc(pk_i, t_i)$;
Crowdsource the private data to Data Service Provider:
$(pk_i, ek_i, \psi_1, d_i, g_i)$;
Data Service Provider: $Enc(\mathcal{X}) \triangleq \{\langle pk_i, \psi_i \rangle\}$ an
encrypted database which is the concatenation of public
keys and ciphertext tuples. Execute Algorithm 2 and 3
over the ciphertext domain;
Update the encrypted parameters: $\phi =$
$\{W^1, W^2, b^1, b^2\}$;;
Send the learning results $\gamma^*$ to the Data providers:;
The Data providers $P_i, P_2, \ldots P_k$ do:;
The Data providers $P_i, P_2, \ldots P_k$ jointly run a secure
multiparty protocols to calculate ;
Evaluate $(\mathcal{C})$, $\{\langle pk_i, \psi_i \rangle\}$
```

Algorithm 2 Privacy-Preserved Forward Propagation

```
Input: $\{Db_1, Db_2, \ldots Db_n\} = Enc(\mathcal{X})$, with data record
       $r_i = pub(r_i)||Enc(pk_i sec(r_i)$ $\phi =$
       $\{W^1, W^2, b^1, b^2\}$;
Output: $z^2, z^3; a^2, a^3$
for $i = 1,2,\ldots i_{max}$ do
    for $i = 1,2,\ldots (n-1)$ do
        Compute;
        $z_{j_1 j_2 \ldots j_n}^{(2)} = W_{\alpha}^{(1)} \cdot X + b_{j_1 j_2 \ldots j_n}^{(1)}$;
        $a_{j_1 j_2 \ldots j_n}^{(2)} = f(z_{j_1 j_2 \ldots j_n}^{(2)})$;
        $z_{i_1 i_2 \ldots i_n}^{(3)} = W_{\beta}^{(2)} \cdot a^{(2)} + b_{i_1 i_2 \ldots i_n}^{(2)}$;
        $h_{(i_1 i_2 \ldots i_n) W, b}(X) = a_{i_1 i_2 \ldots i_n}^{(3)} = f(z_{i_1 i_2 \ldots i_n}^{(3)})$;
    call Algorithm 3 for backpropagation
```

In this process, given a public key pair (pk_i, sk_i), a vector of ciphertext from Db_i with records $r_i = pub(r_i)||sec(r_i)$ is denoted as ψ_i encrypted by public key pk_i. Initially, the client publish their individual encrypted records to the DSP. DSP then aggregate all the Db_1 with encrypted records $r_i = pub(r_i)||Enc(pk_i sec(r_i)$ into a concatenated database $Enc(\mathcal{X})$. The DSP needs to perform an α-input collaborative deep learning algorithm over the concatenated database domain : $\{Enc_{pk_i}(\mathcal{X})\}, Enc_{pk_i}(W_i^{(k)}), T_i\}$, where $k = 1, 2, \colon \alpha = \Sigma_{s=1}^{n} I_s$

The proposed scheme consist of privacy-preserved feedforward propagation Algorithm 2) and backpropagation (Algorithm 3) with the details below.

A. PRIVACY-PRESERVED FORWARD PROPAGATION

The feedforward propagation is summarized in *Algorithm 2*. All the parties first encrypt the data with pk_i and sends it to the Data Service Provider(DSP). The weighted sum is homomorphically computed by the DSP. $\|\tilde{z}_i\|_s = (\tilde{w}_i \bigotimes \|a_{i-1}\|_s) \bigoplus \tilde{b}_i$. In each layer i, the server then accumulate a_{i-1} and z_i over several iterations to solve the linear equation $z_i = w_i a_i + b_i$ for w_i and b_i.

In an iteration, a number of n linear equations can be established (where n is the number of neuron in the layer). A number of n linear equations can be established (where n is the number of neuron in the layer). There are n^2+n unknowns including n^2 weighted connections and n biases. In the next iteration, an additional n equations are established while there is only one more unknown, i.e. the learning rate η.

B. PRIVACY-PRESERVED BACKPROPAGATION

As illustrated in *Algorithm 3*, DSP performs the privacy-preserving backpropagation algorithm over the concatenated ciphertext for updating the parameters. According to algorithm 1, DSP is required to complete the following computations.

The DSP is required to compute the values of z^2, z^3, a^2 and a^3 over the ciphertext from the last layer $i = n - 1$ to compute the error δ_i. The secure computations of the

Algorithm 3 Privacy-Preserved Backpropagation

```
Input : $(\mathcal{X})$, with data record
        $r_i = pub(r_i)||Enc(pk_i sec(r_i)\ \phi = \{W^1, W^2, b^1, b^2\}; z^2, z^3; a^2, a^3; \eta$
Output: $\phi = \{W^1, W^2, b^1, b^2\}$
for $i$ = 1,2,... $i_{max}$ do
    for example = 1,2, ...,N do
        for $i_n = 1, 2, \ldots, I_1 \times \ldots, I_N$ do
            Calculate;
            $\sigma_i^{(3)} = (a_i^{(3)} \cdot (1 - a_i^{(3)})) \cdot \sum_{j=1}^{I_1 \times I_2 \times \cdots \times I_N} g_{ij}(a_j^{(3)} - y_j)$
            $\sigma_{j_1 j_2 \ldots j_n}^{(2)} = (\sum_{i_1=1}^{I_1} \cdots \sum_{i_n}^{I_n} w_{\lambda j_1 j_2 \ldots j_n}^{(2)} \cdot \sigma_{i_1 i_2 \ldots i_n}^{(3)}) f'(z_{j_1 j_2 \ldots j_n}^{(2)})$;
            $i_n = 1, 2, \ldots, I_N$
            $\Delta b_{i_1 i_2 \ldots i_n}^{(2)} = \Delta b_{i_1 i_2 \ldots i_n}^{(2)} + \sigma_{i_1 i_2 \ldots i_n}^{(3)}$
            $\Delta w_{i_1 i_2 \ldots i_n j_1 j_2 \ldots j_n}^{(2)} = \Delta w_{i_1 i_2 \ldots i_n j_1 j_2 \ldots j_n}^{(2)} + a_{j_1 j_2 \ldots j_n}^{(2)} \cdot \sigma_{i_1 i_2 \ldots i_n}^{(3)}$
            $b_{j_1 j_2 \ldots j_n}^{(1)} = \Delta b_{j_1 j_2 \ldots j_n}^{(1)} + \sigma_{j_1 j_2 \ldots j_n}^{(2)}$
            $\Delta w_{j_1 j_2 \ldots j_n i_1 i_2 \ldots i_n}^{(1)} = \Delta w_{j_1 j_2 \ldots j_n i_1 i_2 \ldots i_n}^{(1)} + x_{i_1 i_2 \ldots i_n} \cdot \sigma_{j_1 j_2 \ldots j_n}^{(2)}$
    $W = W - \eta \cdot (\frac{1}{N} \Delta w)$;
    $b = b - \eta \cdot (\frac{1}{N} \Delta b)$;
```

TABLE 2. Operations used in back-propagation.

Operation	Homomorphic	Examples
Multiplication(×)	yes	$a^2_{j_1 j_2 \cdots j_n} \times \sigma^3_{i_1 i_2 \ldots i_n}$
Division (÷)	no	$\frac{1}{(1+e^{-x})}$
Subtraction (-)	yes	$a^3_j - y_j$
Addition (+)	yes	$\sum_{k=1}^{a} x_k \times w^h_j k$
Exponent (e^x)	no	e^{-x}

values z^2 and z^3 requires addition operations and multiplication operations which the DSP can complete this task with aid of [20] MS-FHE scheme. The DSP afterwards applied Algorithm 4 to calculate the values of a^2 and a^3 which are the results of the activation function of z^2 and z^3.

C. SECURE COMPUTATION OF ACTIVATION FUNCTION

Simulating the activation functions with polynomials are the commonly applied methods of approximating these functions. Due to the impact on the error calculation and updating the weights of the neural network, our approach considers the simulation of the structure of derivative of the ReLU function instead of simulating the Activation function (i.e. ReLU function). The derivative of ReLU is like a step function and is non-differentiable at point 0. It is important to use a function that is continuous and infinitely derivative, since it can approximate more accurately that a non-continuous or non-infinitely differentiable function.

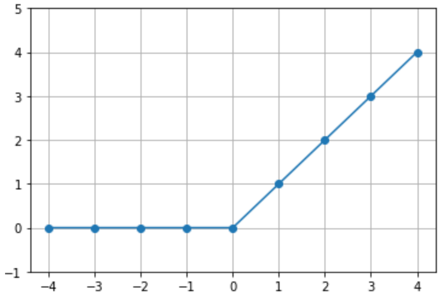

FIGURE 2. ReLU function.

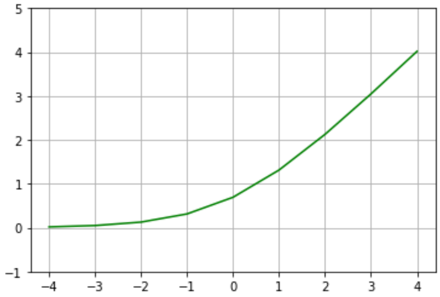

FIGURE 3. Approximation of the ReLU function with our method.

Algorithm 4 Secure Computation of Activation Function

```
Input : Ciphertext $[\mathcal{X}]$,$[a^2]$,$[a^3]$, where $a^0$, $a^1$ and $a^2$ are constants
Output : [y]
1. Compute $c_0$ with $Enc(pk_i)$, i.e., $c_0 = [a_0]$
2. Compute $c_1$ with secure multiplication $Multi_{FN}$, i.e $c_1 = [a_1] \times_{(FN)} [x]$;
3. Compute $c_3$ with secure multiplication $Multi_{FN}$, i.e $c_3 = [a_3] \times_{(FN)} [x] \times_{(FN)} [x] \times_{(FN)} [x]$;
4. Compute [y] with secure addition $Add_{FN}$, $i.e[y] = c_0 +_{FN} c_1 +_{FN} c_3$
5. return [y]
```

Since Sigmoid function is bounded, continuous and infinitely differentiable function which has a structure like the derivative of the ReLU function in the large interval, we therefore approximate the sigmoid function with the polynomial whiles calculating the integral of the polynomial and applied it as the activation function.

D. SECURITY ANALYSIS

Our system performs security analysis of MSCryptoNet against the well-known equation solving attack in the honest-but-curious [21], [26] model. Neural networks have lately been subjected numerous attacks such as adversarial [13], [28] and membership attacks [7] targeting their vulnerabilities. In this settings all data providers are assumed to be semi-honest. We therefore describe the definition of the semantic security, i.e. security against polynomially indistinguishable chosen-plaintext attack (IND-CPA security).

Definition 1.1 [Semantic Security (SS), IND-CPA]: A public-key encryption scheme ε = (KeyGen, Enc, Dec) is semantically secure, if for any stateful PPT adversary $A = (A_1, A_2)$, its advantage $Adv^{SS}_{\varepsilon.A}(K) := |Pr[Exp^{SS}_{\varepsilon.A}(K) = 1] - \frac{1}{2}]$ is negligible, where the experiment $Exp^{SS}_{\varepsilon.A}(K)$ is defined as follows:

$Exp^{SS}_{\varepsilon.A}(K)$
$b \leftarrow 0, 1$
$(pk, sk) \leftarrow keyGen(1^k)$
$(m_0, m_1) \leftarrow A_1(pk)$
$c \leftarrow Enc_{pk}(m_b)$
$b' \leftarrow A_2(c)$
if $b' = b$. return 1:
else, return 0

Padding messages can be applied in case the plaintext have different length since having the same length for the plaintext is part of the requirements.

1) DATA PRIVACY

To protect the privacy and security of the Data Providers (DPs), we apply the MS-FHE to concatenate all the datasets $Enc(\mathcal{X})$ from DPs to support the secure multi-party computation(SMC). We therefore draw the following conclusions based on the definition of semantic security (SS)

Lemma 4.1 [Multi-Scheme FHE Semantic Security (SS)]: If the encryption scheme is semantically secure, then the multi-scheme fully homomorphic encryption is semantically secure.

Proof: Let us assume the public key encryption scheme $\varepsilon = \{KeyGen, Enc, Dec\}$ is semantically secure. Based on this scheme ε, the challenger constructs a evaluate algorithm Eval, such that the new public key encryption scheme $\varepsilon' = \{KeyGen, Enc, Dec, Eval\}$ keeps homomorphic of addition and multiplication operations. If the evaluation key ek is public, then the adversary can compute $Eval$ directly according to the public key pk, the ciphertext ψ and the evaluation key ek. Therefore, the MS-FHE scheme is semantically secure.

Let Recall in This Domain: DPs do not communicate with each other until the decryption phase. Each DPs P_i $(i \in [1, k])$ apply their own schemes or generate their own keys (pk_i, sk_i and encrypts its input Db_i with data record $r_i = pub(r_i)||sec(r_i)$ under the public key pk_i of MS-FHE.Based on the Definition of 4.1 MSCryptoNet is guaranteed to be secured against corrupt DPs since in a honest-but-curious settings, data provider P might collude with the same data provider and want to reveal a sample vector Db outsourced by DPs. The privacy of the DPs is therefore confidential and assured.

Since the MS-FHE is semantically secured, DSP should be probabilistic polynomially bounded whiles sending the ciphertext to CP for computation. The computational power of DSP does not have to be bounded, sine it only receives and concatenate the ciphertext from all the DP. At this level, DSP and DPs cannot obtain the learning results. Therefore, privacy of the learning results in confidential. The following lemma is obtained.

Lemma 4.2. There are no collusion, Algorithm 2, 3 and 4 is privacy preserving for the a^2, a^3, z^2, z^3 and w^1, w^2.

2) PRIVACY OF MODEL

The honest-but-curious CP can privately train a collaborative deep learning model. There is no information leakage, since the addition and multiplication operations are both semantically secured. Weights w^1, w^2 in the forward and backpropagation during the training process by the CP can compute the Addition and Multiplication operations, therefore guaranteeing the privacy of the training process.

VI. PERFORMANCE EVALUATION

This section demonstrates the application of MSCryptoNet to predict diabetes progression in patients.

A. PRIVACY-PRESERVING PREDICTION OF DIABETES PROGRESSION

Prediction of diabetes progression in patients has gradually been applied in the medical diagnosis and has also gain prominence in research. Outsourcing the computations of this technique to the Data Service Providers is gradually gaining widespread privacy concerns. Assuming sensitive medical data from multiple hospitals as sample set, which collects $n \times m$ datasets, where $n = 50$ DPs P_i $(i \in [1, k])$ in various records. This medical records includes variables such as (ages, gender, body mass index, average blood pressure and six blood serum measurements). Our main task is to learn the target function from this securely sampled records. The target variable is a quantitative measurement of the disease progression. Suppose we choose a network with a $(\alpha - \beta - \gamma)$ configuration, i.e. one input layer with ρ nodes, one hidden layer with β and an output layer with γ nodes. This system include three stages, which include input layer, learned hidden representations and output layer. Below are the details described:

B. INPUT STAGE

Prior to crowdsourcing the medical data to the DSP for collaborative deep learning, $P_i(i \in [1, k])$ should encrypted their datasets with their choice of encryption scheme whiles preprocessing the data for feature extraction. The deep learning network is executed with variables as input. This feature as input can reduce the number of input and weights, and cut down the computation and also keep the classification accurately.

C. HIDDEN REPRESENTATIONS

The DSP has obtained the encrypted datasets with different schemes or even different public keys, after the concatenation of all the datasets to $\mathcal{X}$ CP runs Algorithm 2 and 3 to securely train the neural network model.

D. OUTPUT STAGE

Let assume the deep learning network has two input each other representing the status of the patient. These outputs can be viewed as a two-dimension vector and we apply two real numbers to represent the possibilities of the status of the patient. The type of component value, states the target value of the deep learning prediction. For example, $(0, 1)$ if the target vector is 1 then this indicate that the patient has diabetes. The results are also encrypted since the all computations are performed in the encrypted settings. The DSP then sends the encrypted results with pk_i where $\{i \in n\}$.

1) SIMULATION RESULTS

This section present results of implementing an adopted version of convolutional neural network, replacing the Rectified Linear Unit with our approximated polynomial over the collaborative encrypted data. We train our model and measure the accuracy of the model for classification over encrypted data. We used pyseal [52] for the implementation and all other computations were carried on a computer with NVIDIA GeForce GTX 780M 4096 MB @ 3.4 GHz, Intel Core i5 and 16 GB 1600 MHz DDR3 RAM with Ubuntu 16.04 operating system. The encrypted inputs are given to the trained network and the accuracy of the output is measured. The running time for the encryption and sending data from data providers to the cloud server is measured. On the other hand, the running time for the classification of the encrypted batch of data is measured as well as the amount of data transferred in the process. In our simulation, the encrypted data along with the public keys and the encryption parameters are sent to the cloud server, where the cloud service provider runs the deep neural network model over the encrypted data. In our simulation, we classify a batch of the aggregated ciphertext with size 8192 with same size as applied in [12] and provide the running time for the classification. In Table 3 we demonstrate the time it takes to apply MSCryptoNet using the deep neural network model. Furthermore we provide the time required for encryption, transfer and decryption time as depicted in Table 3. It is obvious that MSCryptoNet is much faster.

TABLE 3. Data transfer running time.

	MSCryptoNet	[11]
Encryption	14.8	133
Decryption	1	4
Communication	290	697

2) COMPARISON EVALUATION

In this section, we show the comparison of MSCryptoNet with the existing state-of-the-art privacy-preserving deep neural networks in terms of computational cost and communication overhead. These approaches are based on secure multi-party computations and fully homomorphic encryption.

In order to demonstrate the fair performance comparisons, we simulate [12], [10], and [54] which are the four closest schemes to our MSCryptoNet scheme based on Pyseal. Pyseal [52] is python wrapper for the Microsoft Research's homomorphic encryption implementation, the Simple Encrypted Arithmetic Library (SEAL) homomorphic encryption library. The implementation and all the privacy-preserving deep learning computations were run on a computer with NVIDIA GeForce GTX 780M 4096 MB @ 3.4 GHz, Intel Core i5 and 16 GB 1600 MHz DDR3 RAM with Ubuntu 16.04 operating system. CryptoNets [12] as a pre-trained model trained on an unencrypted dataset uses fully homomorphic encryption to implement convolutional neural network with two convolutional layers and fully connected layers. Dowlin *et al.* [12] applied the sigmoid activation function whiles replacing the max pooling with the scaled mean-pool square function. Table 3 therefore demonstrate that MSCryptoNet outperforms CryptoNet in every aspect of our comparison. Furthermore, [48] identified the ineffectiveness of existing primitives since they become ineffective for deeper neural networks therefore creating the open privacy-preserving challenges in this frameworks. Chabanne *et al.* [48] addresses these problems with the combination of the original ideas proposed by Dowlin et al. and Ioffe and Szegedy [51], using Taylor's series for the approximation of the ReLU activation function and the batch normalization layers to improve the performance of the model. This algorithm was evaluated on a non-secure datasets and none of their experiments were conducted over the encrypted datasets, therefore making it impossible to compare the measurement of the performance of their model over encrypted data. In this instance MSCryptoNet provide better accuracy.

Recently proposed schemes based on secure multiparty computations algorithm are [10] and [48]. Mohassel et al presented SecureML which enables data providers distribute the private data among the two non-colluding servers in a distributed setting therefore falling under the two-server model where diverse neural network models are evaluated on the collaborative data using secure two-party computation. The authors use Yao's Garbled Circuit to securely perform deep learning. Two of the most important advantages of this setting are that, (a) data owners can distribute their data inputs among the two non-colluding servers in the setup phase without engaging in any future computations. (b) It also benefit from a combination of efficient methods for Boolean computations such as the garbled circuits and oblivious Transfer extension and arithmetic computations. They perform experiments on Arcene and MNIST datasets and report the results. As presented in Table 3 and Fig 4, MSCryptoNet significantly outperforms [48] and [10] in all aspects. It is important to note that, [10] decided to keep the computational cost and the communication cost of the data providers at a minimal level. The proposed algorithm classifies one instance at each prediction round. MSCryptoNet on the other hand is capable of classifying a bath of instances in each round with size 8192MB or larger.

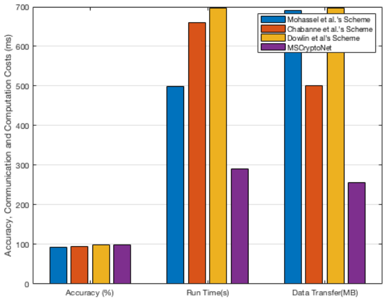

FIGURE 4. Comparison of accuracy, communication costs and computation overheads.

TABLE 4. Deep neural network over encrypted data.

	Times
Convolutional Layer	10.014
Average Pooling Layer	6.501
Convolutioal layer(50 feature maps)	71.524
Average Pooling Layer	5.985
Activation Layer	9.239
2 Fully Connected layer(250 and 10 neurons)	33.91

TABLE 5. Comparison with the state-of-the-art framework.

Dataset	Criteria	MSCryptoNet	[10]	[11]	[12]
[56]	Accuracy(%)	99.64	93.42	95.52	98.95
	Run Time(s)	290	NA	NA	697
	Data Transfer(MB)	257.4	NA	NA	595.55

To provide a fair comparison, we implemented the deep neural network with 2 hidden layers with 128 neurons in each of the layers without any convolutional layers using MSCryptoNet. In a 100 instances they reported 16 seconds as their running time whiles MSCryptoNet reported 10 seconds. By increasing the input batch size, the running time increases sub-linearly in [10] and [48] whereas in MSCryptoNet, the running time remains the same even when the input batch size becomes larger. Furthermore, MSCryptoNet does not require any communication between data providers and the cloud server for the provision of privacy-preserving predictions.

Generally, existing secure multi-party deep learning privacy-preserving solutions have one of the biggest communication overheads since such schemes requires an interaction between the data providers and cloud server for the secure computations. Reference [12] has a huge communication cost of 595.55MB for a relatively small convolutional neural network where as MSCryptoNet is only 257.4MB for the same work. Note that since the data providers participate in the training of the model and the computations, information about the deep neural network model may possibly be compromised. For example, the data providers can learn information such as the structure of the layers, activation functions and the number of the layers therefore demonstrating the potential for security breaches.

VII. CONCLUSION AND FUTURE WORK

We have proposed a collaborative privacy-preserved deep neural network architecture dubbed *MSCryptoNet* based on a fully homomorphic cryptosystem. This novel design has ensured the neural network are free from loss of accuracy and efficiency whiles training on different dataset encrypted with *different schemes or even different keys*. The complexities and security of our scheme is analyzed, and has demonstrated better performance as compared to other state-of-the-art solutions.

ACKNOWLEDGMENT

We are greatly grateful to the associate editor and reviewers for their invaluable suggestions.

REFERENCES

[1] D. L. Goroff, "Balancing privacy versus accuracy in research protocols," *Science*, vol. 347, no. 6221, pp. 479–480, 2015.

[2] M. Enserink and G. Chin, "The end of privacy," *Science*, vol. 347, no. 6221, pp. 490–491, 2015.

[3] R. Canetti, "Security and composition of cryptographic protocols: A tutorial," *ACM SIGACT News*, vol. 37, no. 3, pp. 67–92, 2006.

[4] C. Tikkinen-Piri, A. Rohunen, and J. Markkula, "EU general data protection regulation: Changes and implications for personal data collecting companies," *Comput. Law Secur. Rev.*, vol. 34, no. 1, pp. 135–153, 2018.

[5] J. M. Kiel, "HIPAA and its effect on informatics," *Comput. Inform. Nursing*, vol. 30, no. 1, pp. 1–5, 2012.

[6] N. H. Phan, X. Wu, and D. Dou, "Preserving differential privacy in convolutional deep belief networks," *Mach. Learn.*, vol. 106, nos. 9–10, pp. 1681–1704, 2017.

[7] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. 53rd Annu. Allerton Conf. Commun.*, 2016, pp. 1310–1321.

[8] Differential Privacy Team, Apple, "Learning with privacy at scale," *Mach. Learn. J.*, vol. 1, no. 8, pp. 1–25, 2017.

[9] C. Gentry, S. Halevi, and V. Vaikuntanathan, "A simple BGN-type cryptosystem from LWE," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2010, pp. 506–522.

[10] P. Mohassel and Y. Zhang, "SecureML: A system for scalable privacy-preserving machine learning," in *Proc. IEEE Symp. Secur. Privacy*, May 2017, pp. 19–38.

[11] H. Chabanne, A. de Wargny, J. Milgram, C. Morel, and E. Prouff, "Privacy-preserving classification on deep neural network," *IACR Cryptol. ePrint Arch.*, vol. 2017. pp. 1–35, 2017.

[12] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: Applying neural networks to Encrypted data with high throughput and accuracy," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1–10.

[13] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," in *Proc. ACM Commun. Secur.*, 2017, pp. 603–618.

[14] Y. Zhang, L. Sun, H. Song, and X. Cao, "Ubiquitous WSN for healthcare: Recent advances and future prospects," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 311–318, Aug. 2014.

[15] A. Lpez-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proc. 44th Annu. ACM Symp. Theory Comput.*, 2012, pp. 1219–1234.

[16] P. Mukherjee and D. Wichs, "Two round multiparty computation via multi-key FHE," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, vol. 5763. Berlin, Germany: Springer, 2016, pp. 735–763.

[17] O. Goldreich, "Secure multi-party computation," Found. Cryptogr., Tech. Rep., vol. 2, pp. 1–109, 1998.

[18] P. Li *et al.*, "Multi-key privacy-preserving deep learning in cloud computing," *Future Gener. Comput. Syst.*, vol. 74, pp. 76–85, Sep. 2017.

[19] B. Wang, Y. Zhan, and Z. Zhang, "Cryptanalysis of a symmetric fully homomorphic encryption scheme," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 6, pp. 1460–1467, Jun. 2018.

[20] Z. Li and T.-H. Lai, ''On evaluating circuits with inputs encrypted by different fully homomorphic encryption schemes,'' IACR Cryptol. ePrint Arch., vol. 2013, pp. 1–19, 2013.

[21] D. Liu, ''Efficient processing of encrypted data in honest-but-curious clouds,'' in *Proc. IEEE 9th Int. Conf. Cloud Comput. (CLOUD)*, San Francisco, CA, USA, Jun./Jul. 2016, pp. 970–974.

[22] C.-Z. Gao, Q. Cheng, X. Li, and S.-B. Xia, ''Cloud-assisted privacy-preserving profile-matching scheme under multiple keys in mobile social network,'' *Cluster Comput.*, pp. 1–9, 2018.

[23] D. Stehlé and R. Steinfeld, ''Faster fully homomorphic encryption,'' in *Proc. Advances in Cryptology—ASIACRYPT*. 2010, pp. 377–394.

[24] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, ''(Leveled) fully homomorphic encryption without bootstrapping,'' *ACM Trans. Comput. Theory*, vol. 6, no. 3, p. 13, 2014.

[25] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, ''GAZELLE: A low latency framework for secure neural network inference,'' in *Proc. USENIX Secur.*, 2018, pp. 1651–1669.

[26] J. Šeděnka, S. Govindarajan, P. Gasti, and K. S. Balagani, ''Secure outsourced biometric authentication with performance evaluation on smartphones,'' *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 384–396, Feb. 2015.

[27] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, ''Membership inference attacks against machine learning models,'' in *Proc. IEEE Symp. Secur. Privacy*, May 2017, pp. 3–18.

[28] A. Nguyen, J. Yosinski, and J. Clune, ''Deep neural networks are easily fooled: High confidence predictions for unrecognizable images,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 427–436.

[29] A. Graves and N. Jaitly, ''Towards end-to-end speech recognition with recurrent neural networks,'' in *Proc. Conf. Mach. Learn.*, 2014, pp. 1764–1772.

[30] J. W. Bos, K. Lauter, and M. Naehrig, ''Private predictive analysis on encrypted medical data,'' *J. Biomed. Inform.*, vol. 50, pp. 234–243, Aug. 2014.

[31] T. H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, ''PCANet: A simple deep learning baseline for image classification?'' *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5017–5032, Dec. 2015.

[32] T. G. Kolda and B. W. Bader, ''Tensor decompositions and applications,'' *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[33] A. Adler, M. Elad, and M. Zibulevsky. (2016). ''Compressed learning: A deep neural network approach.'' [Online]. Available: https://arxiv.org/abs/1610.09615

[34] X. L. Zhang, ''Nonlinear dimensionality reduction of data by deep distributed random samplings,'' in *Proc. Asian Conf. Mach. Learn.*, 2014, pp. 1–18.

[35] G. E. Hinton and R. R. Salakhutdinov, ''Reducing the dimensionality of data with neural networks,'' *Science*, vol. 313, pp. 504–507, Jul. 2006.

[36] V. Chang, Y.-H. Kuo, and M. Ramachandran, ''Cloud computing adoption framework: A security framework for business clouds,'' *Futur. Gener. Comput. Syst.*, vol. 57, pp. 24–41, Apr. 2016.

[37] M. O. Alassafi, A. Alharthi, R. J. Walters, and G. B. Wills, ''A framework for critical security factors that influence the decision of cloud adoption by Saudi government agencies,'' *Telematics Inform.*, vol. 34, no. 7, pp. 996–1010, 2017.

[38] V. Chang and M. Ramachandran, ''Towards achieving data security with the cloud computing adoption framework,'' *IEEE Trans. Services Comput.*, vol. 9, no. 1, pp. 138–151, Jan. 2016.

[39] J. Li, X. Chen, M. Li, J. Li, P. P. C. Lee, and W. Lou, ''Secure deduplication with efficient and reliable convergent key management,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 6, pp. 1615–1625, Jun. 2014.

[40] J. Li, Y. K. Li, X. Chen, P. P. C. Lee, and W. Lou, ''A hybrid cloud approach for secure authorized deduplication,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 5, pp. 1206–1216, 2015.

[41] D. Talia, P. Trunfio, and F. Marozzo, *Data Analysis in the Cloud: Models, Techniques and Applications*, vol. 1. Amsterdam, The Netherlands: Elsevier, 2016, pp. 1–150.

[42] X. Ma, F. Zhang, X. Chen, and J. Shen, ''Privacy preserving multi-party computation delegation for deep learning in cloud computing,'' *Inf. Sci.*, vol. 459, pp. 103–116, Aug. 2018.

[43] T. Chen and S. Zhong, ''Privacy-preserving backpropagation neural network learning,'' *IEEE Trans. Neural Netw.*, vol. 20, no. 10, pp. 1554–1564, Oct. 2009.

[44] A. Bansal, T. Chen, and S. Zhong, ''Privacy preserving back-propagation neural network learning over arbitrarily partitioned data,'' *Neural Comput. Appl.*, vol. 20, no. 1, pp. 143–150, 2011.

[45] J. Yuan and S. Yu, ''Privacy preserving back-propagation neural network learning made practical with cloud computing,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 212–224, Jan. 2014.

[46] J. J. W. Bos, K. K. Lauter, J. Loftus, and M. Naehrig, ''Improved security for a ring-based fully homomorphic encryption scheme,'' *Cryptography and Coding*, vol. 8308. New York, NY, USA: Springer, 2013, pp. 45–64.

[47] M. Hesamifard, E. Takabi, H. Ghasemi, E. Hesamifard, H. Takabi, and M. Ghasemi. (2017). ''CryptoDL: Deep neural networks over encrypted data.'' [Online]. Available: https://arxiv.org/abs/1711.05189

[48] H. Chabanne, A. de Wargny, J. Milgram, C. Morel, and E. Prouff, ''Privacy-preserving classification on deep neural network,'' IACR Cryptol. ePrint Arch., Tech. Rep., 2017.

[49] B. D. Rouhani, M. S. Riazi, and F. Koushanfar, ''DeepSecure: Scalable provably-secure deep learning,'' in *Proc. 55th Annu. Design Autom. Conf. (DAC)*, San Francisco, CA, USA, Jun. 2018, pp. 2:1–2:6.

[50] W. S. Hong, A. D. Haimovich, and R. A. Taylor, ''Predicting hospital admission at emergency department triage using machine learning,'' *PLoS ONE*, vol. 13, no. 7, 2018, Art. no. e0201016.

[51] S. Ioffe and C. Szegedy, ''Batch normalization: Accelerating deep network training by reducing internal covariate shift,'' in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, Lille, France, Jul. 2015, pp. 448–456.

[52] A. J. Titus, S. Kishore, T. Stavish, S. M. Rogers, and K. Ni. (2018). ''PySEAL: A Python wrapper implementation of the SEAL homomorphic encryption library.'' [Online]. Available: https://arxiv.org/abs/1803.01891

[53] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, ''A secure authentication protocol for Internet of vehicles,'' *IEEE Access*, vol. 7, pp. 12047–12057, 2019.

[54] K.-H. Yeh, ''A secure transaction scheme with certificateless cryptographic primitives for IoT-based mobile payments,'' *IEEE Syst. J.*, vol. 12, no. 2, pp. 2027–2038, Jun. 2018.

[55] H. Xiong, H. Zhang, and J. Sun, ''Attribute-based privacy-preserving data sharing for dynamic groups in cloud computing,'' *IEEE Syst. J.*, to be published.

OWUSU-AGYEMANG KWABENA received the M.Sc. degree from Coventry University. He is currently pursuing the Ph.D. degree with the School of Information and Software Engineering, University of Electronic Science and Technology of China. His research interests include machine learning, data mining, big data analysis, applied cryptography, blockchain technology, and medical image processing.

ZHEN QIN received the Ph.D. degree from the University of Electronic Science and Technology of China, in 2012, where he is currently an Associate Professor with the School of Information and Software Engineering. He was a Visiting Scholar with the Department of Electrical Engineering and Computer Science, Northwestern University. His research interests include network measurement, mobile social networks, wireless sensor networks, and medical image processing.

TIANMING ZHUANG is currently pursuing the bachelor's degree with the Glasgow College, University of Electronic Science and Technology of China. His research interests include big data analysis and data mining.

ZHIGUANG QIN is currently a Full Professor with the School of Information and Software Engineering, University of Electronic Science and Technology of China (UESTC), where he is also the Director of the Key Laboratory of New Computer Application Technology and the UESTC-IBM Technology Center. His research interests include medical image processing, computer networking, information security, cryptography, information management, intelligent traffic, electronic commerce, distribution, and middleware.