A Survey on Deep Learning Techniques for Privacy-Preserving

Harry Chandra Tanuwidjaja, Rakyong Choi, and Kwangjo Kim(✉)

School of Computing, Korea Advanced Institute of Science and Technology (KAIST), 291 Gwahak-ro, Yuseong-gu, Daejeon 34141, Korea
kkj@kaist.ac.kr

Abstract. There are challenges and issues when machine learning algorithm needs to access highly sensitive data for the training process. In order to address these issues, several privacy-preserving deep learning techniques, including Secure Multi-Party Computation and Homomorphic Encryption in Neural Network have been developed. There are also several methods to modify the Neural Network, so that it can be used in privacy-preserving environment. However, there is trade-off between privacy and performance among various techniques. In this paper, we discuss state-of-the-art of Privacy-Preserving Deep Learning, evaluate all methods, compare pros and cons of each approach, and address challenges and issues in the field of privacy-preserving by deep learning.

Keywords: Secure Multi-Party Computation · Homomorphic encryption · Trade-Off · Privacy-Preserving Deep Learning

1 Introduction

The invention of machine learning, *i.e.*, Artificial Intelligence (AI) brings a new era to human life. We can train a machine to do decision making like human beings. In general, machine learning consists of training phase and testing phase. In order to get better result by using machine learning, huge dataset is required during the training phase. There is a trend to utilize machine learning in the field of social engineering [1], image recognition [2], healthcare service [3], *etc.* In order to get a satisfying result in machine learning, one of the main challenges is the dataset collection. Since the data will be scattered upon individuals, lots of efforts to collect them are required.

Sensitive users tend to reluctantly submit their private data to a third party. A risk of data leakage will happen due to compromised server-side, e.g., when

This work was partly supported by Indonesia Endowment Fund for Education (LPDP) and Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00555, Towards Provable-secure Multi-party Authenticated Key Exchange Protocol based on Lattices in a Quantum World).

X. Chen et al. (Eds.): ML4CS 2019, LNCS 11806, pp. 29–46, 2019.
https://doi.org/10.1007/978-3-030-30619-9_4

we use cloud computing. Users choose not to store their confidential data in cloud because they worry about that somebody can look at their private data. In order to convince users for their data security and privacy, an approach to use privacy-preserved data is required to input training process in deep learning. For this, the data sent to server must be encrypted and it should be kept encrypted during the training phase, too. The challenge here is to modify the current deep learning technique, so that it can process encrypted data. In this paper, we will discuss state-of-the-art of Privacy-Preserving Deep Learning (PPDL) techniques, evaluate them, compare pros and cons of each technique, and suggest the issues and challenges in PPDL.

The remainder of this paper is organized as follows: Sect. 2 discusses classical privacy-preserving technology in brief. We examine the original structure of Neural Network and modification needed for privacy-preserving environment in Sect. 3. Section 4 presents state of the art of PPDL techniques. Furthermore, Sect. 5 discusses about the analysis of the surveyed methods. Finally, conclusion and future work are provided in Sect. 6. The main contribution of this work is to give detailed analysis about state-of-the-art of PPDL method and show which method has the best performance based on our metrics described on Sect. 4.

2 Classical Privacy-Preserving Technology

Privacy-preserving technique is classified as a special tool that enables the processing of encrypted data [4]. The importance of privacy-preserving technique is to enable computation on data, without revealing the original content. So, it can ensure the privacy of highly confidential data. Directive 95/46/EC [5] on the protection of individuals with regard to the processing of personal data is a European Union directive that regulates the processing of personal data based on human rights law. The directive states that *"[The data] controller must implement appropriate technical and organizational measures to protect personal data against accidental or unlawful destruction or accidental loss, alteration, unauthorized disclosure or access, in particular where the processing involves the transmission of data over a network, and against all other unlawful forms of processing."* The goal of privacy-preserving is based on this regulation.

2.1 Homomorphic Encryption

In 1978, Rivest *et al.* [6] questioned whether any encryption scheme exists to support the computation on encrypted data without the knowledge of the secret information. For example, the textbook RSA encryption supports multiplication on encrypted data without its private secret key and we call such a system as multiplicative Homomorphic Encryption (HE). Likewise, we call a system as an additive HE [7] if it supports addition on encrypted data without its secret key.

Fully Homomorphic Encryption (FHE) means that it supports any computation on encrypted data without the knowledge of the secret key, *i.e.,* for any operation o and two plaintexts m_1, m_2, $Enc(m_1)\ o\ Enc(m_2) = Enc(m_1\ o\ m_2)$.

It was remained as an interesting open problem in cryptography for decades till Gentry [8] suggested the first FHE in 2009.

Afterwards, there are a number of research on HE schemes based on lattices with Learning With Errors (LWE) and Ring Learning With Errors (Ring-LWE) problems [9–13] and schemes over integers with approximate Greatest Common Divisor (GCD) problem [14,15]. Early work on HE was impractical but for now, there are many cryptographic algorithm tools that supports HE efficiently such as HElib, FHEW, and HEEAN [16–18].

HE can be applicable to various areas. For example, it can improve the security of cloud computing system since it delegates processing of user's data without giving access to the original data. It is also applicable to machine learning methods for encrypted data by outsourcing computation of simple statistics like mean and variance of all original data.

2.2 Secure Multi-Party Computation

The concept of secure computation was formally introduced as secure two-party computation in 1986 by Yao [19] with the invention of Garbled Circuit (GC). In GC, all functions are described as a Boolean circuit and an oblivious transfer protocol is used, to transfer the information obliviously.

Then, Goldreich *et al.* [20] extended the concept to Secure Multi-Party Computation (MPC) in 1987. The purpose of MPC is to solve the problem of collaborative computing that keeps privacy of a user in a group of non-trusted users, without using any trusted third party.

Formally, in MPC, for a given number of participants, $p_1, p_2, \cdots, p_n$, each has his private data, $d_1, d_2, \cdots, d_n$, respectively. Then, participants want to compute the value of a public function f on those private data, $f(d_1, d_2, \cdots, d_n)$ while keeping their own inputs secret.

Compared to HE schemes, in secure MPC, parties jointly compute a function on their inputs using a protocol instead of a single party. During the process, information about parties' secret must not be leaked.

In secure MPC, each party has almost no computational cost with a huge communication cost, while the server has a huge computational cost with almost no communication cost in HE scheme.

To apply secure MPC to deep learning, we must handle the cost of calculating non-linear activation functions like sigmoid or softmax since its cost during training is too large.

2.3 Differential Privacy

Differential privacy was first proposed by Dwork *et al.* in 2005 [21], to treat the problem of privacy-preserving analysis of data.

From the definition in [22], a randomized function $\mathcal{K}$ gives ϵ-differential privacy if for all datasets D_1 and D_2 differing on at most one element, and for all $S \subseteq Range(\mathcal{K})$,

$$\Pr[\mathcal{K}(D_1) \in S] \geq \exp(\epsilon) \times \Pr[\mathcal{K}(D_2) \in S]$$

Differential privacy deals with the case that a trusted data manager wants to release some statistics over his/her data without revealing any information about the data. Thus, an adversary with access to the output of some algorithm learns almost the same information whether user's data is included or not.

Applying differential privacy, there are a number of researches on machine learning algorithms like decision trees, support vector machines, or logistic regressions [23–25].

3 Deep Learning in Privacy-Preserving Technology

This section describes the original structure of deep learning technique and the modification needed for privacy-preserving environment.

3.1 Deep Neural Network (DNN)

Activation Layer. Activation layer, as shown in Fig. 1, decides whether the data is activated (value one) or not (value zero). The activation layer is a non-linear function that applies mathematical process on the output of convolutional layer. There are several well-known activation function, such as Rectified Linear Unit (ReLU), sigmoid, and tanh. Since those functions are not linear, the complexity becomes really high if we use the functions to compute the HE encrypted data. So, we need to find a replacement function that only contains multiplication and addition. The replacement function will be discussed later.

Pooling Layer. Pooling layer, as shown in Fig. 2, is a sampling layer whose purpose is to reduce the size of data. There are two kinds of pooling: max and average poolings. In HE, we cannot use max pooling function, because we are not able to search for the maximum value of encrypted data. As a result, average pooling is the solution to be implemented in HE. Average pooling calculates the sum of values, so there is only addition operation here, which is able to be used over HE encrypted data.

Fig. 1. Activation layer

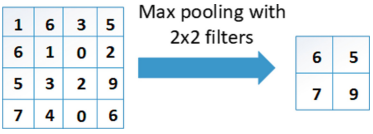

Fig. 2. Pooling layer

Fully Connected Layer. The illustration of fully connected layer is shown in Fig. 3. Each neuron in this layer is connected to neuron in previous layer, so it is called fully connected layer. The connection represents the weight of the feature like a complete graph. The operation in this layer is dot product between the value of output neuron from the previous layer and the weight of the neuron. This function is similar to hidden layer in Neural Network. There is only dot product function that consists of multiplication and addition function, so we can use it over HE encrypted data.

Dropout Layer. Dropout layer, which is shown in Fig. 4, is a layer created to solve over-fitting problem. Sometimes, when we train our machine learning model, the classification result will be too good for some kind of data, which shows bias to the training set. This situation is not good, resulting in huge error during the testing period. Dropout layer will drop random data during training and set the data to zero. By doing this iteratively during the training period, we can prevent over-fitting during the training phase.

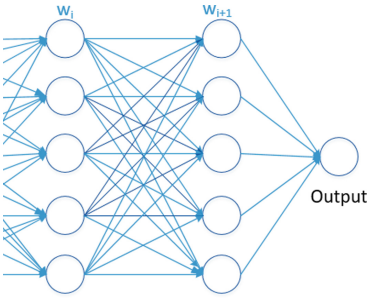

Fig. 3. Fully connected layer

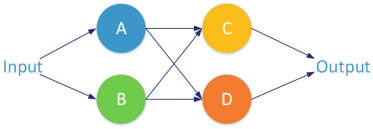

Fig. 4. Dropout layer

3.2 Convolutional Neural Network (CNN)

CNN [26] is a class of DNN, which is usually used for image classification. The characteristic of CNN is convolutional layer whose purpose is to learn features which are extracted from the dataset. The convolutional layer has $n \times n$ size, which we will do dot product between neighbor values in order to make convolution. As a result, there are only addition and multiplication in convolutional layer. We do not need to modify this layer as it can be used for HE data, which is homomorphically encrypted.

3.3 Modification of Neural Network in Privacy-Preserving Environment

Batch Normalization Layer. Batch Normalization (BN) layer was proposed by Ioffe and Szegedy [28]. The main purpose of BN layer is to fasten the training process by increasing the stability of NN. This layer receives the output from activation layer, then do re-scaling process, resulting in a value between zero and one. BN layer computes the subtraction of each input with the batch mean value, then divides it by the average value of the batch.

Approximation of Activation Function. There have been several researches [4,29,30] to do polynomial approximation for activation function. Some well-known methods include numerical analysis, Taylor series, and polynomial based on the derivative of the activation function. Numerical analysis generates some points from ReLU function, then uses the points as the input of approximation function. Taylor series uses polynomials of different degrees to approximate the activation function.

Convolutional Layer with Increased Stride. This architecture is proposed by Liu *et al.* [30] to replace the pooling layer. They leverage convolutional layer with increased stride as a substitution of pooling layer. They use BN layer between the fully connected layer and ReLU. By doing this, the depth of the data stays the same but the dimension is reduced.

4 State of the Art of PPDL Techniques

In this section, we will discuss state of the art of current PPDL techniques. We divide PPDL method into three: HE-based PPDL, Secure MPC-based PPDL, and Differential Privacy-based PPDL. Figure 5 shows the classification of privacy-preserving method, to the best of our knowledge. The methods are divided into classical and Hybrid PPDL. Classical privacy-preserving method does not contain any deep learning technique, whereas Hybrid PPDL is the combination of classical privacy-preserving method with deep learning. In this paper, we focus on Hybrid PPDL technique since the classical privacy-preserving technique has been already outdated.

In order to compare the performance of each surveyed paper, we use five metrics including accuracy, run time, data transfer, Privacy of Client (PoC), and Privacy of Model (PoM). Figure 6 shows the metrics for surveyed PPDL works in this paper. Accuracy means the percentage of correct prediction made by PPDL model. Run time is the time needed by the model to do encryption, sending data from client to server, and doing classification process. Data transfer is the amount of data transferred from client to server. PoC means that neither the server or any other party knows about client data. PoM means that neither the client or any other party knows about the model classifier in server. We measure the average of accuracy, run time, and data transfer of each method. Then, we set

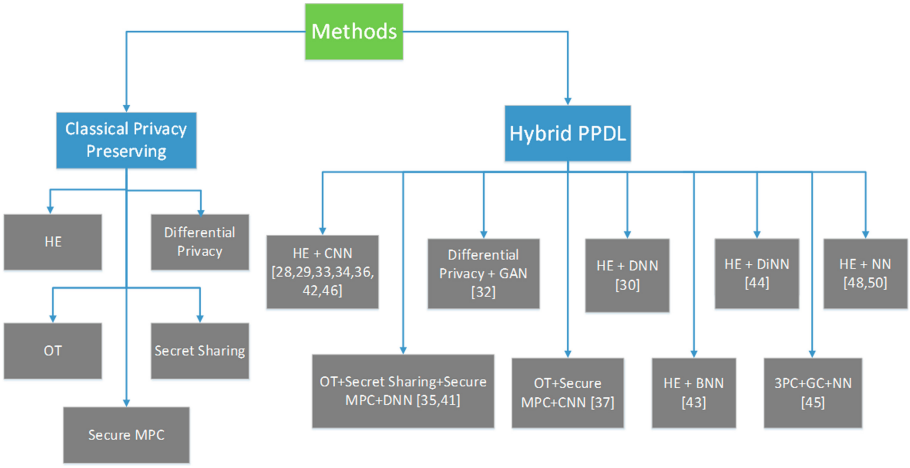

Fig. 5. Classification of privacy-preserving (PP)

the average value as the standard. If the accuracy value is higher than average, it means that the accuracy of the proposed method is good. Furthermore, if the run time and data transfer are lower than average, it means that the run time and data transfer of proposed method are good. We take the comparison data from the respective papers as we believe it is the best result that is possible to achieve. We do not re-execute their codes since not all of the codes are open to public. We focus our paper to Hybrid PPDL method which combines classical privacy-preserving with various deep learning practices.

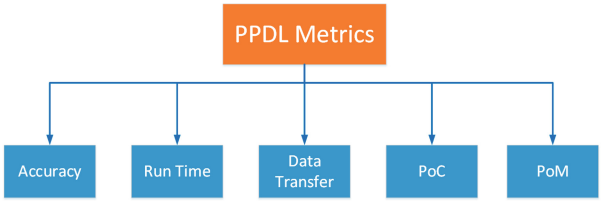

Fig. 6. Metrics for surveyed PPDL works

4.1 HE-Based PPDL

In this section, we discuss PPDL method that leverages HE to ensure the privacy of the data.

ML Confidential [31]**,** developed by Graepel *et al.*, is a modified CNN that works on HE scheme. They use polynomial approximation to substitute non-linear activation function. They use cloud service based scenario, and utilize

their proposed method to ensure the privacy of data during transfer period between client and server. At first, they do key generation, producing public key and private key for each client. Then, client data is encrypted using HE and transferred to the server. The cloud server will do training process using the encrypted data, and use the training model to do classification on testing dataset.

Cryptonets [34]**,** proposed by Gilad-Bachrach *et al.*, applies CNN to homomorphically encrypted data. They propose Cryptonets to protect data exchange between user and cloud service. They show that cloud service can apply encrypted prediction based on the encrypted data, then give back the encrypted prediction to user. Later, a user can use his own private key to decrypt it, and finally get the prediction result. This scheme can be implemented for hospital service, for example, when a doctor needs to predict the health condition of a patient and take care of an outpatient. The weakness of Cryptonets is its performance limitation on the number of non-linear layer. If the number of non-linear layer is large, which we can find from deeper Neural Network, the error rate will increase and its accuracy drops.

PP on DNN [35]**,** proposed by Chabanne *et al.*, is a privacy-preserving technique on DNN. For the methodology, they combine HE with CNN. Their main idea is to combine Cryptonets [34] with polynominal approximation for activation function and batch normalization layer proposed by Ioffe and Szegedy [28]. They want to improve the performance of Cryptonets, which is only good when the number of non-linear layer in the model is small. The main idea of this paper is changing the structure of regular Neural Network that consists of convolutional layer, pooling layer, activation layer, and fully connected layer into convolutional layer, pooling layer, batch normalization layer, activation layer, and fully connected layer. Max pooling is not a linear function. As a result, in pooling layer they use average pooling, instead of max pooling to provide the homomorphic part with linear function. The batch normalization layer gives contribution to restrict the input of each activation layer, resulting in stable distribution. Polynomial approximation with low degree gives small error, which is very suitable to be used in this model. The training phase is done using the regular activation function, and the testing phase is done using the polynomial approximation, as a substitution to non-linear activation function. Their experiment shows that their model achieves 99.30% accuracy, which is better than Cryptonets (98.95%). The pros of this model is its eligibility to work in Neural Network with high number of non-linear layers, but still gives accuracy more than 99%, unlike Gilad-Bachrach *et al.* [34] approach that experiences accuracy drop when the number of non-linear layers are increased.

CryptoDL [29]**,** proposed by Hesamifard *et al.*, is a modified CNN for encrypted data. They change the activation function part of CNN with low degree polynomial. This paper shows that the polynomial approximation is indispensable

for Neural Network in HE environment. They try to approximate three kinds of activation function; ReLU, sigmoid, and tanh. The approximation technique is based on the derivative of activation function. Firstly, during training phase, CNN with polynomial approximation is used. Then, the model produced during the training phase is used to do classification over encrypted data. The authors apply their method to MNIST dataset [41], and achieve 99.52% accuracy. The weakness of this scheme is not covering privacy-preserving training in deep Neural Network. They use the privacy-preserving for classification process only. The pros of this work is it can classify many instances (8,192 or larger) for each prediction round, unlike Rouhani *et al.* [40] that classifies one instance per round. So we can say that CryptoDL works more effective compared to DeepSecure [40].

PP All Convolutional Net [30]**,** proposed by Liu *et al.*, is a privacy-preserving technique on convolutional network using HE. They use MNIST dataset [41] that contains handwritten number. They encrypt the data using HE, then use the encrypted data to train CNN. Later, they do classification and testing process using the model from CNN. Their idea is adding batch normalization layer before each activation layer and approximate activation layer using Gaussian distribution and Taylor series. They also change the non-linear pooling layer with convolutional layer with increased stride. By doing this, they have successfully modified CNN to be compatible with HE, and achieve 98.97% accuracy during the testing phase. We can see that the main difference between regular CNN and modified CNN in privacy-preserving technology is the addition of batch normalization layer and the change of non-linear function in activation layer and pooling layer into linear function.

Distributed PP Multi-Key FHE [39]**,** proposed by Xue *et al.*, is a PPDL method using multi-key FHE. They do some modification to conventional CNN structure, such as changing max pooling into average pooling, adding batch normalization layer before each activation function layer, and replacing ReLU activation function with low degree approximation polynomial. Their method is beneficial for classifying large scale distributed data, for example, in order to predict the future road condition, we need to train Neural Network model from traffic information data which are collected from many cars. The security and privacy issue during data collection and training process can be solved using their approach.

Gazelle [43]**,** proposed by Juvekar *et al.*, is a new framework for PPDL. They combine HE with GC to ensure privacy in Prediction-as-a-Service (PaaS) environment. The goal of this paper is to facilitate a client to do classification process without revealing his input to the server and also preserve the privacy of model classifier in server. They try to improve the encryption speed of HE using Single Instruction Multiple Data (SIMD). They also propose new algorithm to accelerate convolutional and matrix vector multiplication process. Finally, Gazelle is

also able to switch protocol between HE and GC, so it successfully combines secret-sharing and HE for privacy preserving environment. For the deep learning part, they leverage CNN that consists of two convolutional layers, two ReLU layers as activation layers, one pooling layer, and one fully connected layer. In order to ensure the privacy of the Neural Network model, they hide the weight, bias, and stride size in the convolutional layer. Furthermore, they also limit the number of classification queries from client to prevent linkage attack. The experiment shows that Gazelle fully outclasses another popular technique such as MiniONN [42] and Cryptonets [34] in terms of runtime.

TAPAS [44]**,** proposed by Sanyal *et al.*, is a new framework to accelerate parallel computation using encrypted data in PaaS environment. They want to address the main drawback of HE to do a prediction service, which is the large amount of processing time required. The main contribution here is a new algorithm to speed up binary computation in Binary Neural Network (BNN). The algorithm firstly transforms all data into binary. Then, it computes the inner product by doing XNOR operation between encrypted data and unencrypted data. After that, they count the amount of 1's from the result of previous step. Finally, they check whether two times of the counted amount is bigger than the difference between the number of bits and the bias. If yes, then they assign value 1 to activation function and if no, they assign -1 to the activation function. They also show that their technique can be parallelized by evaluating gates at the same level for three representations at the same time. By doing this, the time needed for evaluation step will be improved drastically. They compared their approach with and without parallelization. The result shows that using MNIST dataset, non-parallel process needs 65.1 h while the parallelized process only takes 147 s to complete.

FHE DiNN [45]**,** proposed by Bourse *et al.*, stands for Fast HE Discretized Neural Network technique, which is used for PPDL. They want to address complexity problem in common HE technique when it is used in Neural Network. The deeper the network is, the higher the complexity, resulting in more computational cost. They use bootstrapping technique to achieve linear complexity to the depth of the Neural Network. When we compare to standard Neural Network, there is one main difference, the weight, bias value, and the domain of activation function in the proposed method needs to be discretized. They use sign activation function to limit the growth of signal in the range of $-1, 1$, showing its characteristic of linear scale invariance for linear complexity. The activation function will be computed during bootstrapping process, in order to refresh neuron's output. They successfully show that BNN can accomplish accuracy close to regular NN by gaining more network size. During the experiment, FHE-DiNN achieves more than 96% accuracy in less than 1.7 s. Overall, the processing time of FHE-DiNN is much faster than Cryptonets [34], but their accuracy is slightly worse (2.6% less).

E2DM [47], proposed by Jiang *et al.*, stands for Encrypted Data and Encrypted Model, which is a PPDL framework that performs matrices operation on HE system. E2DM encrypts a matrix homomorphically, then do arithmetic operations on it. The main contribution of E2DM is less complexity needed during computation process. It has $O(d)$ complexity to do dot product between two encrypted $d \times d$ matrices, instead of $O(d^2)$ complexity. They leverage CNN with one convolutional layer, two fully connected layers, and a square activation function. During the experiment, they use plain text whose size is less than 212 and can predict 64 images during one circle of processing. E2DM achieves 20 fold latency reduction and 34 fold size reduction compared to Cryptonets [34]. They also show that compared to MiniONN [42] and Gazelle [43], E2DM has less bandwidth usage because it does not require interaction between protocol participants.

As a summary, Table 1 illustrates the comparison of each HE-based PPDL method based on our metrics.

Table 1. Comparison of HE-Based PPDL techniques

Scenario	Proposed schemes	DL technique	Accuracy (%)	Run time (s)	Data transfer (Mbytes)	PoC	PoM
Cloud Service	ML Confidential [31]	DNN	Bad (95.00)	Bad (255.7)	–	Yes	No
	Cryptonets [34]	CNN	Good (98.95)	Bad (697)	Bad (595.5)	Yes	No
	PP on DNN [35]	CNN	Good (99.30)	–	–	Yes	No
	E2DM [47]	CNN	Good (98.10)	Good (28.59)	Good (17.48)	Yes	Yes
	PPDL via Additively HE [48]	CNN	Good 97.00	Good (120)	–	Yes	Yes
Image Recognition	CryptoDL [29]	CNN	Good (99.52)	Bad (320)	Bad (336.7)	Yes	No
	PP All Convolutional Net [30]	CNN	Good (98.97)	Bad (477.6)	Bad (361.6)	Yes	No
Content Sharing	Distributed PP Multi-Key FHE [39]	CNN	Good (99.73)	–	–	Yes	No
PaaS	Gazelle [43]	CNN	–	Good (0.03)	Good (0.5)	Yes	Yes
	Tapas [44]	BNN	Good (98.60)	Good (147)	–	Yes	Yes
	FHE-DNN [45]	DiNN	Bad (96.35)	Good (1.64)	–	Yes	Yes

PPDL via Additively HE [48]**,** proposed by Phong *et al.*, is a PPDL system based on a simple NN structure. The author shows that there is a weakness in Shokri and Shmatikov paper [49] that leaks client data during training process. The weakness is called Gradients Leak Information. It is an adversarial method to get input value by calculating the gradient of corresponding truth function to weight and the gradient of corresponding of truth function to bias. If we divide the two results, we will get the input value. Because of that reason, Phong *et al.* propose their revised PPDL method to overcome this weakness. The key idea of is letting cloud server updating deep learning model by accumulating gradient value from users. However, actually there is a weakness too on this approach because it does not prevent attacks between participants. Proper authentication to participants should be done by the cloud server to prevent this vulnerability.

Secure Weighted Possibilistic C-Means (PCM) Algorithm for PP [50]**,** proposed by Zhang *et al.*, is a secure clustering method to preserve data privacy in cloud computing. They combine C-Means Algorithm with BGV encryption scheme [12] to produce a HE based big data clustering on a cloud environment. The main reason of choosing BGV in this scheme is because of its ability to ensure correct result on the computation of encrypted data. They also address PCM weakness, which is very sensitive and need to be initialized properly. To solve this problem, the authors combine fuzzy clustering and probabilistic clustering. During the training process, there are two main steps: calculating the weight value and updating the matrix. In order to do it, Taylor approximation is used here, as the function is polynomial with addition and multiplication operation only.

4.2 Secure MPC-Based PPDL

In this section, we will talk about PPDL method that leverages Secure MPC to ensure the privacy of the data.

SecureML [36]**,** proposed by Mohassel and Zhang, is a new protocol for privacy-preserving machine learning. They use Oblivious Transfer (OT), Yao's GC, and Secret Sharing. OT is a security protocol proposed by Rabin [37], in which the sender of message remains oblivious whether the receiver has got the message or not. Secret sharing becomes one of basic cryptographic tools to distribute a secret between parties since the introduction of secret sharing by Shamir [38] in 1979. For deep learning part, they leverage linear regression and logistic regression in DNN environment. They propose addition and multiplication algorithm for secretly shared values in linear regression. The authors leverage Stochastic Gradient Descent (SGD) method in order to calculate the optimum value of regression. The weakness of this scheme is that they can only implement a simple Neural Network, without any convolutional layer, so the accuracy is quite low.

DeepSecure [40], proposed by Rouhani *et al.*, is a framework that enables the use of deep learning in privacy-preserving environment. The authors use OT and Yao's GC protocol [19] with CNN to do the learning process. DeepSecure enables a collaboration between client and server to do learning process on cloud server using data from client. They do security proof of their system by using semi-honest, honest-but-curious adversary model. It has been successfully shown that the GC protocol keeps the client data private during the data transfer period. The cons of this method is its limitation of number of instance processed each round. They are only able to classify one instance during each prediction round.

MiniONN [42], proposed by Liu *et al.*, is a privacy preserving framework to transform a Neural Network into an oblivious Neural Network. The transformation process in MiniONN include the nonlinear functions, with a price of negligible accuracy lost. There are two kinds of transformation provided by MiniONN, including oblivious transformation for piecewise linear activation function and oblivious transformation for smooth activation function. A smooth function can be transformed into a continuous polynomial by splitting the function into several parts. Then, for each part, polynomial approximation is used for the approximation, resulting in a piecewise linear function. So, MiniONN supports all activation functions that have either monotonic range, piecewise polynomial, or can be approximated into polynomial function. During the experiment, they show that MiniONN beats Cryptonets [34] and SecureML [36] in terms of message size and latency.

Table 2. The comparison of secure MPC-Based PPDL techniques

Scenario	Proposed schemes	DL technique	Accuracy (%)	Run time (s)	Data transfer (Mbytes)	PoC	PoM
Cloud Service	DeepSecure [40]	CNN	Good (98.95)	Bad (10,649)	Bad (722,000)	No	Yes
Image Recognition	SecureML [36]	DNN	Bad (93.40)	–	–	No	Yes
PaaS	MiniONN [42]	NN	Good (98.95)	Good (1.04)	Good (47.60)	No	Yes
	ABY3 [46]	NN	Bad (94.00)	Good (0.01)	Good (5.20)	No	Yes

ABY3 [46], proposed by Mohassel *et al.*, is a protocol for privacy-preserving machine learning based on three-party computation (3PC). This protocol can switch between arithmetic, binary, and Yao's 3PC, depending on processing needs. The usual machine learning process works on arithmetic operation. As a result, it cannot do polynomial approximation for activation function. ABY3

can be used to train linear regression, logistic regression, and Neural Network model. They use arithmetic sharing when training linear regression model. On the other hand, for computing logistic regression and Neural Network model, they use binary sharing on three party GC. During experiment, they show that ABY3 outperforms MiniONN [42] by four order of magnitude faster, when it runs on the same machine. Table 2 summarizes the comparison of each Secure MPC-Based method.

4.3 Differential Privacy-Based PPDL

PATE [33]**,** proposed by Papernot *et al.*, stands for Private Aggregation of Teacher Ensembles. PATE learning process consists of teacher phase and student phase based on differential privacy in GAN (Generative Adversarial Network) [27]. In PATE, firstly, during teacher phase, the model is trained using subset of data. Then, the student model will learn from the teacher model. The key of privacy is in teacher model [32], which is not made public. The advantage of this model is due to the distinguished model, when an adversary can get a hold on student model, it will not give them any confidential information. They also show that there is possible failure that reveals some part of training data to the adversary. As a result, notification to the failure is really important, aside from developing cryptography technique for privacy protection.

5 Analysis of the Surveyed Methods

After we have surveyed all papers mentioned above, we can see that E2DM [47] gives the best performance based on our metrics defined here. It is indicated by getting good accuracy, good run time, good data transfer, and ensure both PoC and PoM. E2DM is the only work that satisfies all parameters that we define, which indicates the best PPDL method for this time. Furthermore, from our analysis above, we believe that main challenge in privacy-preserving machine learning technique regards to the trade-off between accuracy and complexity. If we use high degree polynomial approximation for activation function, the accuracy will become better, but in cost for high complexity. On the other hand, low degree polynomial approximation for activation function gives low complexity with worse accuracy compared to high degree polynomial. Choosing correct approximation method for each privacy-preserving scenario is the main challenge here.

6 Conclusion and Future Work

In this paper, we have discussed state of the art of PPDL. We analyze the original structure of Neural Network and the modification needed to use it in privacy-preserving environment. We also address the trade-off between accuracy and complexity during the substitution process of non-linear activation function

as the main challenge. An open problem regarding privacy-preserving machine learning technique is to reduce computational burden. How to divide the burden between a client and a server optimally, to get the best performance is a big challenge that needs to be addressed in the future. Another challenge is to ensure the PoC and PoM at the same time, while maintaining the computation performance. Ensuring the PoC and PoM requires two extra computation from client's and model's point of view, respectively. Our survey shows that only E2DM has successfully fulfill those requirements, even though its accuracy still lower than CryptoDL [29], DeepSecure [40], and MiniONN [42]. However, those three methods only satisfy one of PoC or PoM, not both of them. Achieving more than 99% accuracy with PoC and PoM properties becomes the main challenge of the future PPDL method. Lightweight PPDL with fast and cheap cost is also an interesting challenge for future work.

References

1. Lazer, D., Pentland, A.S., Adamic, L., Aral, S., Barabasi, A.L.: Life in the network: the coming age of computational social science. Science **323**, 721 (2009)
2. Nasrabadi, N.M.: Pattern recognition and machine learning. J. Electron. Imaging **16**, 049901 (2007)
3. Chen, M., Hao, Y., Hwang, K., Wang, L.: Disease prediction by machine learning over big data from healthcare communities. IEEE Access **5**, 8869–8879 (2017)
4. Zhang, D., Chen, X., Wang, D., Shi, J.: A survey on collaborative deep learning and privacy-preserving. In: IEEE Third International Conference on Data Science in Cyberspace, pp. 652–658 (2018)
5. Meints, M., Moller, J.: Privacy-preserving data mining: a process centric view from a european perspective (2004)
6. Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. Found. Secure Comput. **4**(11), 169–180 (1978)
7. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
8. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Annual ACM on Symposium on Theory of Computing, pp. 169–178. ACM (2009)
9. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (Standard) LWE. SIAM J. Comput. **43**(2), 831–871 (2014)
10. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_29
11. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_5
12. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) Fully homomorphic encryption without bootstrapping. ACM Transact. Comput. Theory (TOCT) **6**(3), 13 (2014)

13. Clear, M., McGoldrick, C.: Multi-identity and multi-key leveled FHE from learning with errors. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 630–656. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_31
14. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_2
15. Cheon, J.H., et al.: Batch fully homomorphic encryption over the integers. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 315–335. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_20
16. Halevi, S., Shoup, V.: Algorithms in HElib. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 554–571. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_31
17. Ducas, L., Micciancio, D.: FHEW: bootstrapping homomorphic encryption in less than a second. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 617–640. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_24
18. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10624, pp. 409–437. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_15
19. Yao, A.C.-C.: How to generate and exchange secrets. In: Foundations of Computer Science 27th Annual Symposium, pp. 162–167. IEEE (1986)
20. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game. In: Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, pp. 218–229. ACM (1987)
21. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
22. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_1
23. Chaudhuri, K., Monteleoni, C., Sarwate, A.D.: Differentially private empirical risk minimization. J. Mach. Learn. Res. **12**, 1069–1109 (2011)
24. Kifer, D., Smith, A., Thakurta, A.: Private convex empirical risk minimization and high-dimensional regression. In: Conference on Learning Theory, pp. 1–25 (2012)
25. Jagannathan, G., Pillaipakkamnatt, K., Wright, R.N.: A practical differentially private random decision tree classifier. In: IEEE International Conference on Data Mining Workshops 2009, ICDMW 2009, pp. 114–121. IEEE (2009)
26. LeCun, Y., Haffner, P., Bottou, L., Bengio, Y.: Object recognition with gradient-based learning. Shape, Contour and Grouping in Computer Vision. LNCS, vol. 1681, pp. 319–345. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-46805-6_19
27. Goodfellow, I.: Generative adversarial nets. In: Advances in neural information processing systems, pp. 2672–2680 (2014)
28. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167 (2015)
29. Hesamifard, E., Takabi, H., Ghasemi, M.: CryptoDL: deep neural networks over encrypted data. arXiv:1711.05189 (2017)

30. Liu, W., Pan, F., Wang, X.A., Cao, Y., Tang, D.: Privacy-preserving all convolutional net based on homomorphic encryption. In: International Conference on Network-Based Information Systems, pp. 752–762 (2018)
31. Graepel, T., Lauter, K., Naehrig, M.: ML confidential: machine learning on encrypted data. In: International Conference on Information Security and Cryptology, pp. 1–21 (2012)
32. Abadi, M., Erlingsson, U., Goodfellow, I.: On the protection of private information in machine learning systems: two recent approches. In: Computer Security Foundations Symposium, pp. 1–6 (2017)
33. Papernot, N., Abadi, M., Erlingsson, U.: Semi-supervised knowledge transfer for deep learning from private training data. arXiv:1610.05755 (2016)
34. Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., Wernsing, J.: Cryptonets: applying neural networks to encrypted data with high throughput and accuracy. In: International Conference on Machine Learning, pp. 201–210 (2016)
35. Chabanne, H., de Wargny, A., Milgram, J., Morel, C., Prouff, E.: Privacy-preserving classification on deep neural network. IACR Cryptology ePrint Archive (2017)
36. Mohassel, P., Zhang, Y.: SecureML: a system for scalable privacy-preserving machine learning, pp. 19–38 (2017)
37. Rabin, M.O.: How to exchange secrets with oblivious transfer. IACR Cryptology ePrint Archive, p. 187 (2005)
38. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
39. Xue, H., et al.: Distributed large scale privacy-preserving deep mining. In: IEEE Third International Conference on Data Science in Cyberspace, pp. 418–422 (2018)
40. Rouhani, B., Riazi, M., Koushanfar, F.: DeepSecure: scalable provably-secure deep learning. In: 55th ACM/ESDA/IEEE Design Automation Conference, pp. 1–6 (2018)
41. Deng, L.: The MNIST database of handwritten digit images for machine learning research. IEEE Signal Process. Mag. **29**, 141–142 (2012)
42. Liu, J., Juuti, M., Lu, Y., Asokan, N.: Oblivious neural network predictions via MiniONN transformations. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 619–631 (2017)
43. Juvekar, C., Vaikuntanathan, V., Chandrakasan, A.: GAZELLE: a low latency framework for secure neural network inference. In: 27th USENIX Security Symposium, pp. 1651–1669 (2018)
44. Sanyal, A., Kusner, M.J., Gascón, A., Kanade, V.: TAPAS: tricks to accelerate (Encrypted) prediction as a service. arXiv preprint, arXiv:1806.03461 (2018)
45. Bourse, F., Minelli, M., Minihold, M., Paillier, P.: Fast homomorphic evaluation of deep discretized neural networks. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10993, pp. 483–512. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96878-0_17
46. Mohassel, P., Rindal, P.: ABY 3: a mixed protocol framework for machine learning. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 35–52. ACM (2018)
47. Jiang, X., Kim, M., Lauter, K., Song, Y.: Secure outsourced matrix computation and application to neural networks. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 1209–1222. ACM (2018)
48. Phong, L.T., Aono, Y., Hayashi, T., Wang, L., Moriai, S.: Privacy-preserving deep learning via additively homomorphic encryption. In: IEEE Transactions on Information Forensics and Security, pp. 1333–1345. IEEE (2018)

49. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1310–1321. ACM (2015)
50. Zhang, Q., Yang, L.T., Castiglione, A., Chen, Z., Li, P.: Secure weighted possibilistic C-means algorithm on cloud for clustering big data. Inf. Sci. **479**, 515–525 (2019)