A Survey on Collaborative Deep Learning and Privacy-Preserving

Dayin Zhang
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
Email: zhangdayin@iie.ac.cn

Xiaojun Chen
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
Email: chenxiaojun@iie.ac.cn

Dakui Wang
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
Email: wangdakui@iie.ac.cn

Jinqiao Shi
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
Email: shijinqiao@iie.ac.cn

***Abstract*—In the era of big data, the amount of data that individuals and enterprises hold is increasing, and the efficiency and effectiveness of data analysis are increasingly demanding. Collaborative deep learning, as a machine learning framework that can share users' data and improve learning efficiency, has drawn more and more attention and started to be applied in practical problems. In collaborative deep learning, data sharing and interaction among multi users may lead data leakage especially when data are very sensitive to the user. Therefore, how to protect the data privacy when processing collaborative deep learning becomes an important problem. In this paper, we review the current state of art researches in this field and summarize the application of privacy-preserving technologies in two phases of collaborative deep learning. Finally we discuss the future direction and trend on this problem.**

Keywords—Big data, Collaborative deep learning, Privacy-preserving, Secure multi-party computing, Homomorphic encryption, Differential privacy

I. INTRODUCTION

In recent years, deep learning proves extremely effective at learning nonlinear features and functions from complex data, and has been widely used in text recognition [1], social networking [2], biomedical [3] and other fields. The effect of the deep learning model is often related to the size of the model and the training data set. Under a reasonable learning mechanism, if there is more training data, the model will have a better effect. However, in the era of big data, data is often scattered among individuals and cannot be brought together because of policy, competition, or privacy. Users can only learn based on only a part of data and cannot benefit from the whole data. In order to solve this problem, it is the current trend to apply collaborative learning to the deep learning. Collaborative deep learning is a situation in which two or more users learn a deep learning model together. The generation of collaborative deep learning avoids the problem of the long acquisition cycle of traditional deep learning data and the low accuracy of the model caused by the use of only a portion of the data.

Although collaborative deep learning applications are becoming more and more widely used, due to the variety of collaborative deep learning application scenarios, privacy exposure methods are more diversified. For example, smart bracelets can record information such as the user's heart rate and motion trajectory throughout the day. Smart homes can record the user's diet, routines and other laws. These data can be collected to provide users with high-quality personalized services such as recommendation and identification. However, they also face unavoidable problems such as non-trusted third parties and untrusted users.

How to ensure the utility of collaborative deep learning without revealing the privacy of users and models is a key issue in the field of deep learning. In this paper, we first introduce the architecture of collaborative deep learning and the issue of privacy leakage. Secondly, we introduce privacy-preserving technology commonly used in the applications and analyze their advantages and disadvantages when using in the two phases of collaborative deep learning. Finally, we summarize its development direction and trend.

II. COLLABORATIVE DEEP LEARNING

A. Architecture

As the same as traditional machine learning algorithms, collaborative deep learning algorithms are composed of two phases: the training phase and the using phase. Training phase essentially aims at inferring the algorithm parameters from a labelled data set by optimizing some training objective. The using phase aims at getting a prediction or classification result based on the user's input.

1) Training phase: Based on the special environment of collaborative deep learning, this paper divides the training phase of collaborative deep learning into two modes: direct collaborative deep learning and indirect collaborative deep learning.

a) Direct collaborative deep learning: There is a central server and multiple users. The central server maintains a global model. Each user has a local data set. During the training process, each user uploads local data to the server. The central server collects user data and run the deep learning algorithm centrally to get the model.

This work was supported by the National Key Research and Development Program of China under Grant No. 2016YFB0801003

978-1-5386-4210-8/18/$31.00 ©2018 IEEE
DOI 10.1109/DSC.2018.00104

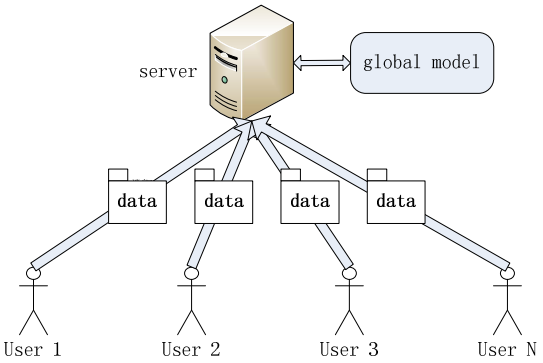

Fig. 1. Direct collaborative deep learning in training phase

b) Indirect Collaborative Deep Learning: There is a central server and multiple users. The central server maintains a global model. Each user has a local data set and maintains a local model. During the model training process, the user first downloads the global model from the central server. The local training set is trained to obtain an improved local model, and updated information is uploaded to the server. The central server aggregates all user information to form an updated global model, and multiple iterations are required in the training process to converge to the optimal model. Data do not leave the user's device from beginning to end.

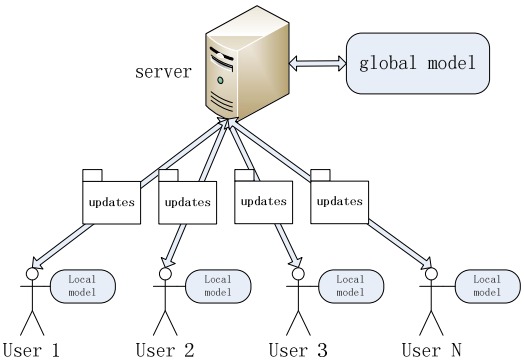

Fig. 2. Indirect collaborative deep learning in training phase

2) Using phase: The using phase of collaborative deep learning mainly takes the user's input according to the model operation to obtain an output category. In general, the server provides the user with API. The user uploads his own input data and gets a output after the server runs the deep learning algorithm.

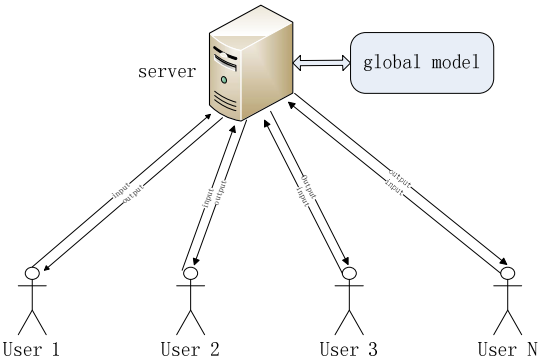

Fig. 3. Collaborative deep learning in using phase

B. Problems

In this paper, a malicious adversary refers to an honest but curious user or server. It honestly follows the protocol but is curious about the privacy information of other honest users and may attempt to learn or infer sensitive information from the data of honest user.

1) The server is a malicious adversary: In direct collaboration and deep learning, although the server is generally not malicious, it may be semi-honest. When the server uses the user's data, it may leave the user's supervision. If the server is attacked, the user's data will face great risks. For example, Strava made a heat map of the user's motion trajectory around the world based on the GPS position information of the user's fitness tracker, and the heat map exposed the trajectory of the US soldiers and the outline of some sensitive military bases [5]. iHearFood is an app that helps users improve their food taste, but it analyzes the user's chewing voice through Bluetooth headsets and mobile phones, unobtrusively monitoring and detecting the user's diet [50]. Facebook is an open social networking platform and allows applications to collect information such as the user's age, location, preferences, and friends. Then these applications can use these targeted ads to change user's minds [38].

2) The user is a malicious adversary: In indirect collaborative deep learning, each user has a local training data set that will not be uploaded to the server. Although each user in the training process transmits the model update information calculated through the local training set as short as possible, there is still the risk of privacy leakage. Reference [6] designes a malicious adversary disguised as an internal honest user to generate false gradient information generated by the anti-network during the training of the model to encourage other honest users to release more sensitive information, disturb model training to infer information about labels it doesn't have. In the course of deep learning training, it is unavoidable to overfit on specific training examples in the sense that some of these examples are implicitly memorized. The inversion attack [7] can reconstruct the victim's face based on the label information provided by the model or learn the identity of the corresponding individual based on an image

containing a blurred-out face. Reference [8] proposes a member reasoning attack model in which malicious adversary call the model interface multiple times to construct an attack model to recognize differences in the target model's predictions on the inputs that it trained on versus the inputs that it did not train on.

3) Servers and users are malicious adversaries: Compared to a single user or server as a malicious adversary, when a malicious server collaborates with a malicious user in a collaborative learning environment, even if a single honest user can achieve anonymity, multiple malicious adversaries with complementary attributes also will collectively speculate on honest user privacy. Reference [23] simulates two situations. One is that some users are malicious and collusion. The other is that the server is malicious and colluded with some subsets of malicious users. In the absence of strong privacy-preserving technology applied, honest users will face a higher risk of privacy leakage.

III. COMMON PRIVACY-PRESERVING TECHNOLOGY

There have been a large number of papers devoted to the study of improving the learning model without revealing sensitive data, and it mainly includes technologies such as secure multi-party computing, homomorphic encryption, and differential privacy.

A. Secure multi-party computing

Secure multi-party computing originated from the millionaire problem proposed by AC Yao [9] in 1982. Its purpose is to solve the problem of collaborative computing that protects privacy among a group of non-trusted users. SMC needs to ensure the independence of input and calculations. Accuracy, without leaking the input values to other members participating in the calculation. The SMC ensure the independence of the input and the correctness of the calculation, while not leaking the input values to other members participating in the calculation. When training a neural network using a secure multi-party computing technology based on Yao's garbled circuits, it is not suitable for deep learning techniques because the cost of calculating nonlinear activation functions such as sigmoid or softmax is large during training. Moreover, Yao's garbled circuits is suitable for 2 or 3 sided security calculations, and it is not easy to expand to a collaborative environment with more users. Reference [10] proposes a two-server model for privacy-preserving training. Users split their data into two separate copies and send them to two different servers. Two servers use secure two-party computation (2PC) to train neural network and other machine learning models. Both servers cannot see the user's entire data during the training process.

B. Homomorphic encryption

The Homomorphic Encryption (HE) scheme was originally proposed by [11] as a way to encrypt data such that certain operations can be performed on it without decrypting it first, which preserves some of the original message space structure in computations. Reference [12] proposes a fully homomorphic encryption (FHE) based on [11] to provide a function for processing encrypting data which allows arbitrarily many operations to be performed on the encrypted data. At the same time, the user who owns the key decrypts the processed data and gets the result of the processing. When applying homomorphic encryption in collaborative deep learning, each user first encrypts his local data with the system public key and uploads the ciphertext to the server. The server performs most of the operations related to the learning process in ciphertext and returns the encrypted result to the user. In this process, the server knows nothing about the user's data, and the user is also ignorant of the server's model.

C. Differential privacy

Differential privacy was first proposed by C. Dwork [13,14,15]. This method injects random noise into the statistical results calculated from the original sensitive data. When a single record is replaced or deleted in the original data set, it will not affect the output of the algorithm. Differential privacy enables insights to be obtained from the original sensitive data, but it is mathematically proven that a single record cannot be recognized. Reference [16] applies differentially private principal projection at the deep neural networks input layer and then train deep neural networks with non-convex objectives under a modest privacy budget.

IV. PRIVACY-PRESERVING TECHNOLOGY IN COLLABORATIVE DEEP LEARNING

A. Privacy-preserving during training phase

1) Direct collaborative deep learning: In direct collaborative deep learning, each user needs to upload local data to the server. In this process, privacy-preserving is mainly to solve the user's data cannot be recognized by the server.

Reference [17] is the first work on this problem, they provides privacy-preserving for multiparty collaborative back-propagation network learning over arbitrarily partitioned data, using the BGN homomorphic encryption that supports one multiplication and unlimited number of additions [18]. Because users must re-encrypt after each multiplication operation, this protocol is not very effective for deep neural networks. To solve this problem, It proposed the all-homomorphic encryption algorithm based on the BGV scheme [21], which allows unlimited number of multiplications and additions on ciphertext to avoid excessive multiplication depth. However, the updated weights need to be sent to the parties for decryption and re-encryption after each iteration, so the communication complexity of this solution is very high. Reference [22] uses homomorphic encryption to protect the gradient information passed between users and honest but curious servers. Each user communicates the homomorphically encrypted ciphertext with the server using different secure channels. All gradient information is encrypted and stored in the server, which protects user privacy without reducing the accuracy of deep learning. Unlike [23] focus on the mobile environment, this solution applies to a large and stable collaborative environment where the user's local data set is large, without considering the exit mechanism.

When a malicious server collaborates with some malicious users, randomized data are vulnerable to statistical data recovery attacks [24,25]. Reference [26] proposes a two-stage perturbation mechanism for privacy-preserving collaborative learning called RG+RP. The first stage is designed to perturb each user's data through a nonlinear function called repeated Gompertz (RG). The second stage use a row-orthogonal random projection (RP) matrix to maintain accuracy and reduce transmission energy. This solution ensures that when user publishes data, it is guaranteed that the original sensitive information cannot be re-identified while retaining the analysis attributes of the data and can defend against maximum a posteriori (MAP) estimation attacks [51] and independent component analysis (ICA) attacks [52].

Reference [46] trains large recurrent language models with user-level differential privacy guarantees with only a negligible cost in predictive accuracy and introduce a noised version of the federated averaging algorithm [30], which satisfies user-adjacent differential privacy via use of the moments accountant [16]. The federated averaging approach groups multiple Stochastic Gradient Descent (SGD) updates together, enabling large-step model updates.

Reference [49] proposes an alternative privacy-preserving multi-party machine learning system based on trusted SGX processors [53] and data-obliviousness algorithms. The server creates processor-protected memory region (called an enclave) that contains code and data, establishes connections by using different secure channels to authenticate identities for different users. The user uses a separate, locally generated key to encrypt his own input data set and uses its secure channel to share the key with the enclave. After the server communicates with the users and obtains the keys for all data set, the enclave code runs the target algorithm over the entire data set and outputs a machine learning model that is encrypted and protected with the integrity of the new symmetric key.

With the increasing demand of users for privacy, many companies use privacy-preserving technologies to make their products more competitive in the market. For example, Apple applies differential privacy in macOS and iOS. When users use iOS to search for queries and geo-location applications, the data sent to Apple is summarized and anonymized. What Apple sees is a general pattern that does not identify individuals.

2) Indirect Collaborative Deep Learning: Compared to a centralized architecture that directly collaborates with deep learning, indirect collaborative deep learning retains the user's data locally, eliminating the need to pay for large data centers and retaining the convenience of a centralized architecture.

Google's federal learning enables android users to predict the next word when they compose a text message. It greatly increases the typing efficiency of a phone's on-screen keyboard [48]. Each user securely maintains their private database of text messages on their own mobile device, and trains the shared global model under the coordination of a highly processed central server.

Reference [30] proposes a practical method for the federated learning of deep networks that enables mobile devices to learn together to share global models while retaining all training data locally. This method allows high-quality models to be trained in relatively few rounds of communication by optimizing non-convex loss functions and using parameter averaging over updates from multiple users. Compared with [31], this method proves robust to the unbalanced and non-IID (independently and identically distributed) data distributions that naturally arise.

Reference [23] puts forward a secure aggregation protocol based on [30]. When facing mobile environments where communication is extremely expensive and dropouts are common. It operates on high-dimensional vectors and provides the strongest possible security under the constraints of a server-mediated, unauthenticated network model. It is highly communication efficient and robust to users dropping out. In order to prevent the server from cheating real users by simulating other unreal existing virtual users, model uses public-key infrastructure (PKI) to ensure that users receive messages from other users (and not the server). This preserves the privacy of the rest of the honest users even if a malicious server collaborates with a malicious user.

References [32,33,34] apply differential privacy to indirect collaborative machine learning. Reference [35] proposes a stochastic gradient descent algorithm that satisfies differential privacy. The server can access the data in plain text and ensure that the published model cannot be used to infer the data used during training. Reference [31] proposes a distributed selective stochastic gradient descent algorithm that enables multiple users to jointly learn an accurate model for a given objective without sharing their input data set. Users train independently on their own data set and selectively share small subsets of their models' key parameters during training. Reference [36] proposes a collaborative privacy-preserving supervised deep learning system in mobile environment based on [31]. The mobile device locally trains data and uploads the trained parameters to a XMPP (global server) [54] through round robin and asynchronous parameter exchange protocol. Reference [14] proposes deep private auto-encoder (dPA), which enforce ε-differential privacy by perturbing the objective functions of the traditional deep auto-encoder. They apply the dPA to human behavior prediction in a health social network. In order to solve the problem of significant utility loss in the global model caused by overly conservative injected randomization in collaborative deep learning, [39] proposes a multi-party deep learning framework (αMDL) based on asynchronous optimization, lightweight homomorphic encryption, and threshold secret sharing. This solution provides strong privacy assurance and desirable model utility simultaneously through integrates the local differential privacy (LDP) mechanism with a coordination mechanism.

References [27,28,29] propose a hybrid framework to segment traditional deep learning algorithms. Reference [27] splits a deep learning processing sequence of the Caffe framework [54] by defining new layers and performs distributed processing between the user side and the server side in a pipeline manner. This approach reduces communication costs between the user and the server and protects privacy by sending feature values instead of sensitive data. Reference [28] proposes a hybrid framework that consists of a feature

extractor and classifier. They use the Siamese architecture to protect the data privacy against unauthorized tasks. Reference [29] divides the model into a user and a server. The user trains sensitive data and uploads the trained non-sensitive data to the server to continue training. The server centralizes the aggregation model. Unlike [28,29], the model does not require a global view of the user data, only a part of it is willing to provide a small part of the real data to guide the model.

Unlike existing work focusing on problems where users seek to agree on a global model, Reference [4] studies the case where each user learns a personalized model according to its own learning objective in a fully decentralized peer-to-peer network. Reference [40] proposes a decentralized and asynchronous block coordinate descent algorithm for collaborative learning without any master node to perform aggregation or coordinate the protocol. The local data distribution is different for each user and every user learns a personal model instead of a single global model. Compared with [4], this algorithm accommodates general loss functions and uses a differential privacy scheme based on randomly perturbing to guarantees that even if a malicious adversary knows the result posted by the user, he cannot infer the user's sensitive data. Besides, it is more difficult for a malicious adversary to systematically collect all the information transmitted over the network. Since there is no communication bottleneck for the master node, these decentralized architectures can scale to large sets of users. However, for current mobile environments, users cannot usually establish direct communication channels with other users (relying on a server to mediate such communication). Therefore, this algorithm can only be applied to a more demanding network environment.

B. Privacy-preserving during Using phase

References [42,43] propose a CryptoNets system based on fully homomorphic encryption. First, the user encrypts the sensitive data and sends it to the server. The server operates the data without decryption and sends the encrypted result back to the user. Finally, the user decrypts and gets the result. However, CryptoNets transformations result in high performance overhead and can only be applied when the number of non-linear layers is small. Reference [44] proposes a solution which combines the polynomial approximation of the ReLU activation function with a batch normalization layer [55]. Compared with CryptoNets, this approach can be applied to neural networks with a large number of non-linear layers while maintaining high accuracy.

Reference [45] proposes the first approach named MiniONN for transforming an existing neural network to an oblivious neural network supporting privacy-preserving predictions with reasonable efficiency. This model uses secret sharing and garbled circuits in using phase. Except online phase, it introduces an offline precomputation phase to perform request-independent operations using additively homomorphic encryption together with the single instruction multiple data (SIMD) batch processing technique. Compared with CryptoNets, MiniONN does not require changes to how neural networks are trained and increase efficiency in the prediction phase.

Reference [47] proposes a strengthened strategy named Private Aggregation of Teacher Ensembles (PATE) based on [34]. It consists of an ensemble of teacher models and a student model. The former is trained on disjoint subsets of the sensitive data and the student learns to predict a noise vote among a set of teachers. This strategy uses the state-of-the-art moments accountant technique [16] which restricts student training to a limited number of teacher votes and reveals only the topmost vote after carefully adding random noise. After the training is completed, an ensemble of teacher models will not be made public, and the user invokes the student model when using it. Since the student model does not depend on any single sensitive data, the privacy of the training data is protected even if a malicious adversary can observe the student's internal model parameters.

V. CONCLUSION

In this paper, we first introduce the architecture of collaborative deep learning and the issue of privacy leakage. Secondly, we analyze the application of the commonly used privacy-preserving technology in the two phases of collaborative deep learning and its advantages and disadvantages. Although secure multi-party computing and homomorphic encryption can achieve a high level of privacy and accuracy, the cost is high computational and communication overhead for the users. A more practical and efficient approach is to use differential privacy, where the users insert random noise into their data before sending them to the server. However, it will reduce the accuracy of the model. Compared with traditional machine learning, many factors need to be taken into account such as the hardware performance of user equipment, transmission costs and time constraints when using privacy-preserving technology in collaborative deep learning. When organizations such as hospitals or banks that have large amounts of sensitive data act as users, homomorphic encryption technology is required to ensure the security of the model. When a large number of individuals with weak computing power act as users, differential privacy technology is required to ensure the efficiency of the model. Each privacy-preserving technology has its own characteristics, and more and more studies are currently focusing on providing a reasonable trade-off between data privacy and utility through a combination of secure multi-party computing, homomorphic encryption, and differential privacy.

With the improvement of mobile devices performance, more and more applications use collaborative deep learning to provide users with useful personalized services. Researching the privacy-preserving technology in a complex mobile environment is the next step in our work.

REFERENCES

[1] L Deng, D Yu, "Deep learning: methods and applications," Foundations and Trends® in Signal Processing, 2014, 7(3–4), pp. 197-387.

[2] B Perozzi, R Al-Rfou, S Skiena, "Deepwalk: Online learning of social representations," Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2014, pp. 701-710.

[3] HY Xiong, B Alipanahi, LJ Lee, et al, "The human splicing code reveals new insights into the genetic determinants of disease," Science, 2015, 347(6218): 1254806.

[4] P Vanhaesebrouck, A Bellet, M Tommasi, "Decentralized collaborative learning of personalized models over networks," International Conference on Artificial Intelligence and Statistics (AISTATS). 2017.

[5] A Hern, "Fitness tracking app Strava gives away location of secret US army bases", https://www.theguardian.com/world/2018/jan/28/fitness-tracking-app-gives-away-location-of-secret-us-army-bases

[6] B Hitaj, G Ateniese, F Pérez-Cruz, "Deep models under the GAN: information leakage from collaborative deep learning," Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017, pp. 603-618.

[7] M Fredrikson, S Jha, T Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 2015, pp. 1322-1333.

[8] R Shokri, M Stronati, C Song, V Shmatikov, "Membership inference attacks against machine learning models," Security and Privacy (SP), 2017 IEEE Symposium on. IEEE, 2017, pp. 3-18.

[9] AC Yao, "Protocols for secure computations," Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on. IEEE, 1982, pp. 160-164.

[10] P Mohassel, Y Zhang, "Secureml: A system for scalable privacy-preserving machine learning," Security and Privacy (SP), 2017 IEEE Symposium on. IEEE, 2017, pp. 19-38.

[11] RL Rivest, L Adleman, ML Dertouzos, "On data banks and privacy homomorphisms," Foundations of secure computation, 1978, 4(11), pp. 169-180.

[12] C Gentry, "A fully homomorphic encryption scheme," Stanford University, 2009.

[13] C Dwork, "Differential privacy: A survey of results," International Conference on Theory and Applications of Models of Computation. Springer, Berlin, Heidelberg, 2008, pp. 1-19.

[14] C Dwork, "A firm foundation for private data analysis," Communications of the ACM, 2011, 54(1), pp. 86-95.

[15] C Dwork, A Roth, "The algorithmic foundations of differential privacy," Foundations and Trends® in Theoretical Computer Science, 2014, 9(3–4), pp. 211-407.

[16] M Abadi, A Chu, I Goodfellow, et al, "Deep learning with differential privacy," Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016, pp. 308-318.

[17] J Yuan, S Yu, "Privacy preserving back-propagation learning made practical with cloud computing," International Conference on Security and Privacy in Communication Systems. Springer, Berlin, Heidelberg, 2012, pp. 292-309.

[18] D Boneh, EJ Goh, K Nissim, "Evaluating 2-DNF formulas on ciphertexts," Theory of Cryptography Conference. Springer, Berlin, Heidelberg, 2005, pp. 325-341.

[19] F Bu, Y Ma, Z Chen, H Xu, "Privacy Preserving Back-Propagation Based on BGV on Cloud," High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conferen on Embedded Software and Systems (ICESS), 2015 IEEE 17th International Conference on. IEEE, 2015, pp. 1791-1795.

[20] Q Zhang, LT Yang, Z Chen, "Privacy preserving deep computation model on cloud for big data feature learning," IEEE Transactions on Computers, 2016, 65(5), pp. 1351-1362.

[21] Z Brakerski, C Gentry, V Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," ACM Transactions on Computation Theory (TOCT), 2014, 6(3), p. 13.

[22] LT Phong, Y Aono, T Hayashi, L Wang, S Moriai, "Privacy-Preserving Deep Learning via Additively Homomorphic Encryption," IEEE Transactions on Information Forensics and Security, 2018, 13(5), pp. 1333-1345.

[23] K Bonawitz, V Ivanov,B Kreuter, et al, "Practical Secure Aggregation for Privacy-Preserving Machine Learning," Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017, pp. 1175-1191.

[24] Y Sang, H Shen, H Tian, "Effective reconstruction of data perturbed by random projections," IEEE Transactions on Computers, 2012, 61(1), pp. 101-117.

[25] CR Giannella, K Liu, H Kargupta, "Breaching Euclidean distance-preserving data perturbation using few known inputs," Data & Knowledge Engineering, 2013, 83, pp. 93-110.

[26] L Lyu, X He, YW Law, M Palaniswami, "Privacy-Preserving Collaborative Deep Learning with Application to Human Activity Recognition," Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. ACM, 2017, pp. 1219-1228.

[27] A Ichinose, M Oguchi, A Takefusa, H Nakada, "Evaluation of distributed processing of caffe framework using poor performance device," Big Data (Big Data), 2016 IEEE International Conference on. IEEE, 2016, pp. 3980-3982.

[28] SA Ossia, AS Shamsabadi, A Taheri, et al, "A hybrid deep learning architecture for privacy-preserving mobile analytics," arXiv preprint arXiv:1703.02952, 2017.

[29] P Veličković, ND Lane, S Bhattacharya, et al, "Scaling health analytics to millions without compromising privacy using deep distributed behavior models," Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare. ACM, 2017, pp. 92-100.

[30] HB McMahan, E Moore, D Ramage, BA y Arcas, "Federated learning of deep networks using model averaging," 2016.

[31] R Shokri, V Shmatikov, "Privacy-preserving deep learning," Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. ACM, 2015, pp. 1310-1321.

[32] M Pathak, S Rane, B Raj, "Multiparty differential privacy via aggregation of locally trained classifiers," Advances in Neural Information Processing Systems. 2010, pp. 1876-1884.

[33] A Rajkumar, S Agarwal, "A differentially private stochastic gradient descent algorithm for multiparty classification," Artificial Intelligence and Statistics. 2012, pp. 933-941.

[34] J Hamm, Y Cao, M Belkin, "Learning privately from multiparty data," International Conference on Machine Learning. 2016, pp. 555-563.

[35] S Song, K Chaudhuri, AD Sarwate, "Stochastic gradient descent with differentially private updates," Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE. IEEE, 2013, pp. 245-248.

[36] M Liu, H Jiang, J Chen, et al, "A collaborative privacy-preserving deep learning system in distributed mobile environment," Computational Science and Computational Intelligence (CSCI), 2016 International Conference on. IEEE, 2016, pp. 192-197.

[37] NH Phan, Y Wang, X Wu, D Dou, "Differential Privacy Preservation for Deep Auto-Encoders: an Application of Human Behavior Prediction," AAAI. vol. 16, pp. 1309-1316, February 2016.

[38] K Roose, "How Facebook's Data Sharing Went From Feature to Bug", https://www.nytimes.com/2018/03/19/technology/facebook-data-sharing.html

[39] X Zhang, S Ji, H Wang, T Wang, "Private, Yet Practical, Multiparty Deep Learning," Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on. IEEE, 2017, pp. 1442-1452.

[40] A Bellet, R Guerraoui, M Taziki, M Tommasi, "Fast and Differentially Private Algorithms for Decentralized Collaborative Machine Learning," arXiv preprint arXiv:1705.08435, 2017.

[41] X Lian, C Zhang, H Zhang, et al, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," Advances in Neural Information Processing Systems. 2017, pp. 5336-5346.

[42] P Xie, M Bilenko, T Finley, et al, "Crypto-nets: Neural networks over encrypted data," arXiv preprint arXiv:1412.6181, 2014.

[43] R Gilad-Bachrach, N Dowlin, K Laine, et al, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," International Conference on Machine Learning. 2016, pp. 201-210.

[44] H Chabanne, A de Wargny, J Milgram, C Morel, E Prouff, "Privacy-preserving classification on deep neural network," IACR Cryptology ePrint Archive, 2017, 2017, 35.

[45] J Liu, M Juuti, Y Lu, N Asokan, "Oblivious neural network predictions via minionn transformations," Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017, pp. 619-631.

[46] HB McMahan, D Ramage, K Talwar, L Zhang, "Learning differentially private language models without losing accuracy," arXiv preprint arXiv:1710.06963, 2017.

[47] N Papernot, M Abadi, U Erlingsson, I Goodfellow, K Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," arXiv preprint arXiv:1610.05755, 2016.

[48] HB McMahan, E Moore , D Ramage, S Hampson, BA y Arcas, "Communication-efficient learning of deep networks from decentralized data," arXiv preprint arXiv:1602.05629, 2016.

[49] O Ohrimenko, F Schuster, C Fournet, et al, "Oblivious Multi-Party Machine Learning on Trusted Processors," USENIX Security Symposium. 2016, pp. 619-636.

[50] Y Gao, N Zhang, H Wang, et al, "iHear food: Eating detection using commodity bluetooth headsets," Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2016 IEEE First International Conference on. IEEE, 2016, pp. 163-172.

[51] Y Sang, H Shen, H Tian, "Effective reconstruction of data perturbed by random projections," IEEE Transactions on Computers, 2012, 61(1), pp. 101-117.

[52] K Liu, H Kargupta, J Ryan, "Random projection-based multiplicative data perturbation for privacy preserving distributed data mining," IEEE Transactions on knowledge and Data Engineering, 2006, 18(1), pp. 92-106.

[53] F McKeen, I Alexandrovich, A Berenzon, et al, "Innovative instructions and software model for isolated execution," HASP@ ISCA, 2013, 10.

[54] Xmpp library for python. http://sleekxmpp.com/index.html.

[55] S Ioffe, C Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.