RESEARCH ARTICLE OPEN ACCESS

Privacy-Preserving Back Propagation Neural Network Learning In Signature Scheme

Mrs.S.Blessy,
Assistant Professor, AIHT, Indhumathi.S, Jayashree.R.

Abstract-
Cloud computing allows their clients to share data. Multiple parties may join through conducting joint Back propagation neural network learning on the union of their respective data sets. During this process no party wants to disclose her/his private data to others. Existing schemes supporting this kind of collaborative learning are either limited in the way of data partition or just consider two parties. There lacks a solution that allows two or more parties, each with an arbitrarily partitioned data set, to join conduct the learning. This paper solves this open problem by utilizing the power of cloud computing. In our proposed scheme, each party encrypts his/her private data locally and uploads the cipher texts into the cloud. The cloud then executes most of the operations over cipher texts without knowing the original private data. Each participants have their own login id to see their report in cloud. To connect with Cloud user must give their username and password then only they can able to connect the server. Here the users are able to view their own details efficiently.

Index Terms-Privacy preserving learning, Neural Network, Signature scheme, Back Propagation, Cloud computing, Computing Outsource

I. INTRODUCTION

Back propagation is an effective method for learning neural networks and has been widely used in various applications. The accuracy is highly affected by the volume of high quality data. The participating parties carry out learning not only on their own data sets, but also on others' data sets. With the recent remarkable growth of new computing infrastructures such as Cloud Computing, it has been more convenient than ever for users across the Internet, who may not even know each other to conduct joint collaborative learning through the shared infrastructure.

Despite the potential benefits, one crucial issue pertaining, to the Internet-wide collaborative neural network learning is the protection of data privacy for each participant. In particular, the participants from different trust domains may not want to disclose their private data sets, which may contain privacy or proprietary information, to anybody else. In applications such as healthcare, disclosure of sensitive data, e.g. protected issue health information, is not only a privacy issue but of legal concerns according to the privacy rules

II. RELATED WORK

Several parties participating BPN network learning schemes have been proposed recently. Introducing network learning a privacy preserving BPN network learning without disclosing their respective data sets. But the solution is proposed \ only for horizontal partitioned data. This scheme cannot protect the intermediate results, which may also contain sensitive data, during the learning process. A privacy preserving BPN network learning algorithm for two-party scenarios. We enable user login to view his details. This scheme provides strong protection for datasets including the intermediate results. However it supports vertically partitioned data according to [4]. To overcome this limitation, we enhanced and proposed a solution for arbitrarily partitioned data. Already it was proposed for a two party scenario [6]. Directly we extend them to the multi-party setting will introduce a computation and communication complexity for the number of participants. Such a complexity will produce a tremendous cost on each party scenario through itsupports arbitrarily partitioned dataset. Two party scenarios lacks in efficient and scalable solution that supports collaborative BPN network learning [1]. But in the multi-party scenario it efficiently allows arbitrarily partitioned data sets.

In this work, we address this open problem by incorporating the computing power of the cloud. The main idea of our scheme can be summarized as follows: each participant first encrypts her or his private data with the system public key and then uploads the cipher texts and return the encrypted results to the participants; the participants to the learning process over the cipher texts and return the encrypted results to the participants; the participants jointly decrypt the results with which they update their respective weights for the BPN network. During this process, cloud servers learn no privacy data of a participant even if they conclude with all the participants. Through off-loading the computation tasks. We adopt the BGN (Boneh, Goh, Nissim) doubly homomorphicencryption algorithm. Existing schemes supporting this kind of collaborative learning are either limited in the way of data partition or just consider two parties. There lacks a solution that allows two or more parties, each with an arbitrarily partitioned data set, to collaboratively conduct the learning.

Our contribution can be summarized as follows:

To our best knowledge, this paper is the first that provides privacy preserving for multi-party collaborative BPN network learning over arbitrarily partitioned data. Thorough analysis investigating privacy and efficiency guarantees of proposed scheme is presented; real experiments [4]. To support multi-party secure scalar product and introduce designs that allows decryption of arbitrary large messages.

We aim at enabling multiple parties to jointly conduct BPN network learning without revealing their private data. The input data sets owned by the parties can be arbitrarily partitioned. The computational and communicational costs on each party shall be practically efficient and the system shall be scalable

III. OUR PROPOSED SCHEME PRIVACY PRESERVING MULTI-PARTYWITH USER INTERFACE LOGIN

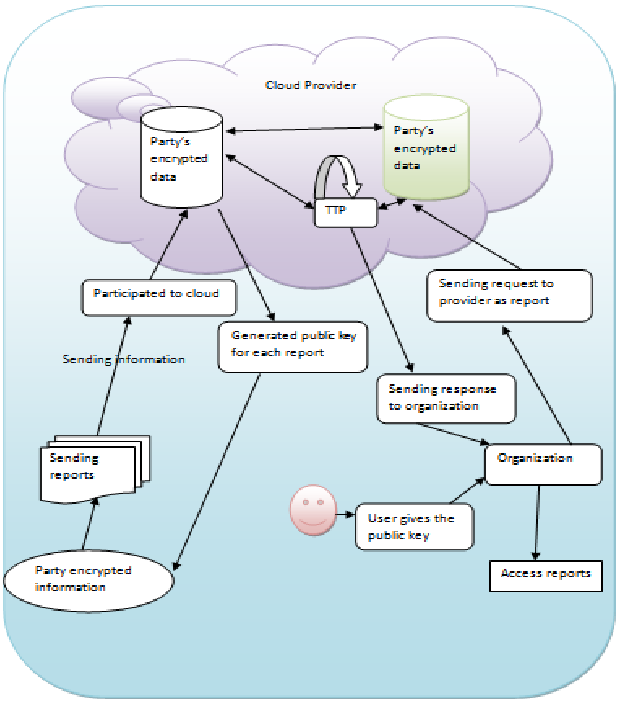
Here, we introduce our cloud based privacy preserving multi-party network learning algorithm over arbitrarily partitioned data.as we describes in the privacy preserving back propagation neural network learning. All the parties generate and assign random weights to each Ps and make agreement on the max number of learning iteration. The learning rate n, error threshold and target value ti, of each output layer node at the beginning of learning [17]. In the feed forward stage, all the parties agree on the terms of approximation for the sigmoid function according to their accuracy

We provide the user login purpose for the users to views their details personally and it is more secure .using the key we provide the security to the user accounts. The users profile consists of their confidential details, and which can logged using their key, they key is the randomly generated key, and it can be according to the user's wish. When the user wants to switch to any other organization or hospital, the user profile will be helpful, the user details will be notified only with the knowledge of user. Each party encrypts her or his own data with the system public key and uploads the cipher texts to the cloud [18]. The cloud servers compute the sum of original messages are vectors, the cloud computes the scalar product is returned to all the parties in cipher texts. Decrypting the results needs the participation of the trusted authority.

The systems architect establishes the basic structure of the system, defining the essential core design features and elements that provide the framework. The systems architect provides the architects view of the users' vision. Above diagram user first login to the account then he can download a file which are available in Cloud. Securely offloading the expensive operations to the cloud, we keep the computation and communication costs on each party minimal and independent to the number of participants. Our scheme can be implemented with easy communication protocol design since each party only needs to communicate with the cloud server. It can efficiently handle therelative large dataset for learning, which is compared to the linearly increasing computation/ communication cost of existing works.

IV. SYSTEM ARCHITECTURE

Each party's information are encrypted by the trusted authority using the encryption techniques. The key will be generated for that report by the trusted authority. The generated key will be issued to the particular user. Then these reports are send and that is stored in the cloud. Each parties have their partition in the cloud. The user information are stored in the cloud according to their partition. The partition in the cloud are inter-connected. If one party needs the other party user information. It directly downloads the report from the cloud that reports will be in the encrypted format. The party has to get the key from the user and decrypts it for the future use.Neural Networks have been an active research area for decades.Secure Multi-party computation can be used to solve problems of this kind. But the extremely high computation and communication complexity of SMC, due to the circuit size, usually makes it far from practical even in the two-party case.


Cloud Provider
Party's encrypted data
Party's encrypted data
TTP
Participated to cloud
Sending information
Generated public key for each report
Sending request to provider as report
Sending response to organization
Organization
Sending reports
User gives the public key
Party encrypted information
Access reports

However, privacy bothers many when the training dataset for the neural networks is distributed between two parties, which is quite common nowadays. Existing cryptographic approaches such as secure scalar product protocol provide a secure way for neural network learning when the training dataset is vertically partitioned. In this paper we present a privacy preserving algorithm for the neural network learning when the dataset is arbitrarily partitioned between the two parties. We show that our algorithm is very secure and leaks no knowledge (except the final weights learned by both parties) about other party's data. We demonstrate the efficiency of our algorithm by experiments on real world data.

V. BACK PROPAGATION NEURAL NETWORK LEARNING

Back Propagation neural network learning algorithm is mainly composed of two stages: Feed forward and error back propagation. We use vectors {x1, x2, and x3} to denote the values of input nodes and vectors {h1, h2, h3} and vectors {a1,a2,a3} to denote the hidden layer nodes and output layer nodes respectively.The goal of any supervised learning algorithm is to find a function that best maps a set of inputs to its correct output. An example would be a simple classification task, where the input is an image of an animal, and the correct output would be the name of the animal. Some input and output patterns can be easily learned by single-layer neural networks. There is no way for it to learn any abstract features of the input since it is limited to having only one layer. A multi-layered network overcomes this limitation as it can create internal representations and learn different features in each layer.[1] The first layer may be responsible for learning the orientations of lines using the inputs from the individual pixels in the image. The second layer may combine the features learned in the first layer and learn to identify simple shapes such as circles. Each higher layer learns more and more abstract features such as those mentioned above that can be used to classify the image. Each layer finds patterns in the layer below it and it is this ability to create internal representations that are independent of outside input that gives multi-layered networks its power. The goal and motivation for developing the back propagation algorithm is to find a way to train multi-layered neural networks such that it can learn the appropriate internal representations to allow it to learn any arbitrary mapping of input to output. During the BPN network learning.

Input: N input sample vectors V, 1<i<N with a dimensions iteration tales place at the rate of n, target value ti, sigmoid function f(x) =1/1+e (-x).

Output: Network with final weights. wjk.

```
Randomly initialize the wjk values
For iteration=1,2….iteration do
for sample=1,2…..N.
//Feed forward stage:
for j=1,2…b do
hj=f$\sum_{k=1} a$(x(k)*wjk)
For i=1,2….c do
oi=f($\sum_{j=1} a$(hj*wjk)
if Error=$\frac{1}{2}\sum_{i=1} c$(ti-0i)*hj
else// it works with Back propagation stage,;
$\boldsymbol{\sigma}$wij=-(ti-oi)*hj
$\sigma wjk = -hj(1-hj)xk\sum_{i=1} c$[(ti-oi)*wij)]
wij=wij-□wjk
wjk=whj-□wjk
else
//learning finish
break.
```

$$\sigma wij = -(ti - oi)hj$$

$$\sigma wjk = -hj(1 - hj)xk \sum_{i=1} c\,[(ti - oi) * wjk)]$$

VI. TECHNIQUE PRELIMINARIES

A.USER INTERFACE DESIGN

To connect with Cloud .user must give their username and password then only they can able to connect the server. If the user already exits directly can login into the server else user must register their details such as username, password, Email id, City and Country into the server. Server will create the account for the entire user to maintain upload and download rate. Name will be set as user id. . Logging in is usually used to enter a specific page, which trespassers cannot see. Once the user is logged in, the login token may be used to track what actions the user has taken while connected to the site.

B. KEY GENERATION

We have generated key for user security. This key have gave to individual participating parties. This key has used to enter the cloud server. This key value did not know another user. Keys were sending secretly to individual user. The key can changed according the user's wish for their confidential purpose. Modern cryptographic systems include symmetric-key algorithms (such as AES) and public-key algorithms Symmetric-key algorithms use a single shared key; keeping data secret requires keeping this key secret. Public-key algorithms use a public key and a private key. The public key is made available to our users. The party encrypts data with the public key; only the holder of the private key can decrypt this data. Since public-key algorithms tend to be much slower than symmetric-key algorithms, modern systems such as: one party receives the other's public key, and encrypts a small piece of data (either a symmetric key or some data used to generate it). The remainder of the conversation uses a (typically faster) symmetric-key algorithm for encryption. Here we use the public key doubly homomorphic encryption technique to decrypt the data,

C. BGN HOMOMORPHIC ENCRYPTION

Homomorphic encryption enables operations on plaintexts to be performed on their respective cipher texts without disclosing the plaintexts. Most existing homomorphic encryption schemes only support single operations –either addition or multiplication. The public key 'doubly homomorphic' encryption scheme is used [21], which supports one multiplication and number of addition operations. Therefore cipher texts c(m1),c(m2),c(m3)….c(mi) as [5][10][7]and c(m1),c(m2),c(m3)….c(mi) gives the result c(m1m1+m2m2+m3m3…mimi) without knowing the plaintext, where c() is the cipher text of message mi. Encrypted by the system's public key. We refer [5] for the reference of the BGN scheme.

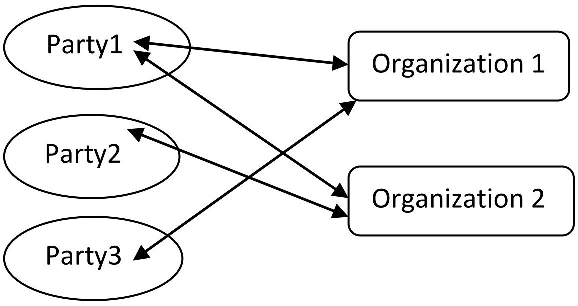
D. SECURE SCALAR AND SHARING DATA WITH CLOUD

We have proposed multiple parties to perform secure scalar product and homomorphic addition operations on cipher texts using cloud computing. We have introduced designs that allow decryption of arbitrary large messages. The product and addition computation used to help of cloud [9][11].To the support consecutive multiplications, the parties need to decrypt the intermediate results first. Decryption of the intermediate values will reveal these values to the parties, which may have privacy implications and shall be avoided [6]. To protect these intermediate results (scalar products or sum), which enables each participating party to get a random value of the intermediate result without knowing its actual value.

E. ANALYSIS

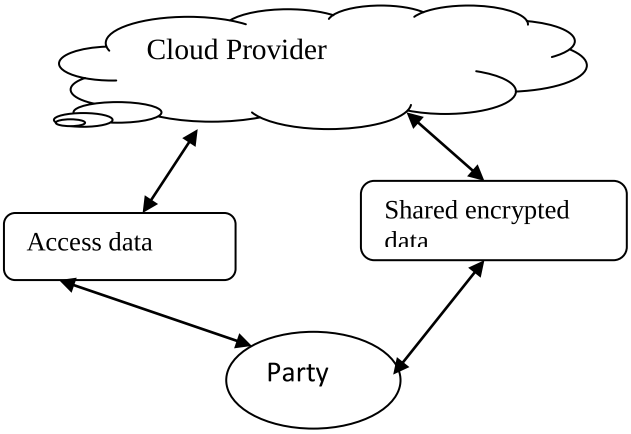
I. SECURITY ANALYSIS

Propose an algorithm that allows multiple parties to perform secure scalar product and homomorphic addition operations on cipher texts using cloud computing. Party encrypts her/his data with the system public key and uploads the cipher texts to the cloud [15]. The cloud servers compute the sum of original messages based on their cipher texts.



II. NUMERIC ANALYSIS

In practice, however, it is hard to guarantee that the final results (numbers) are always small enough for the Pollard's lambda method to efficiently decrypt. This is either because the numbers contained in the vectors are too large, or the vectors are too long (of high dimension). To overcome this limitation, we propose to let the data holders divide the numbers, if they are large, into several numbers, and the cloud then decrypt the smaller "chunks" with which the final result can be recovered[12]. The decryption process can be parallelized for efficiency.



VII. CONCLUSION

In this we proposed the secure and practical multi-party BPN network learning scheme over arbitrarily partitioned data. In our proposed scheme, each parties have their own login to enter into cloud server then they encrypt their arbitrarily partitioned data and upload the chipper text to the cloud. The cloud can execute most operations pertaining to the BPN network learning algorithm without knowing any private information. The cost is independent to the number of parties. Complexity and security analysis shows that our proposed scheme is scalable, efficient and secure. One interesting future work is to enable multi-party collaborative learning without the help of TA.

REFERENCE

[1] The health insurance portability and accountability act of privacy and security rules. url: http://www.hhs.gov/ocr/privacy.

[2] A. Bansal, T. Chen, and S. Zhong. Privacy preserving back-propagation neural network learning over arbitrarily partitioned data. *Neural Comput. Appl.*, 20(1):143–150, Feb. 2011.

[3] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In *Proceedings of the Second international conference onTheory of Cryptography*, TCC'05, pages 325–341, Berlin, Heidelberg,2005.

[4] T. Chen and S. Zhong. Privacy-preserving backpropagation neural network learning. *Trans. Neur. Netw.*, 20(10):1554–1564, Oct. 2009.

[5] L. Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W.Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in Neural InformationProcessing Systems*, pages 396–404. Morgan Kaufmann, 1990.

[6] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Over-encryption: management of access control evo-lution on outsourced data. In *Proceedings of the 33rd internationalconference on Very large data bases*, VLDB '07, pages 123–134. VLDBEndowment, 2007.

[9] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of CRYPTO 84 onAdvances in cryptology*, pages 10–18, New York, NY, USA, 1985.

[10] S. E. Fahlman. *Faster-learning variations on Back-propagation: Anempirical study*, pages 38–51. Morgan Kaufmann, 1988.

[11] K. Flouri, B. Beferull-lozano, and P. Tsakalides. Training a svm-based classifier in distributed sensor networks. In *Proceedings of14nd European Signal Processing Conference*, pages 1–5, 2006.

[12] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[13] R. Grossman and Y. Gu. Data mining using high performance data clouds: experimental studies using sector and sphere. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 920–927, NewYork, NY, USA, 2008.

[14] R. L. Grossman. The case for cloud computing. *IT Professional*, 11(2):23–27, Mar. 2009.

[15] A. Inc. *Amazon Elastic Compute Cloud (Amazon EC2)*. Amazon Inc., http://aws.amazon.com/ec2/#pricing, 2008.

[16] R. Law. Back-propagation learning in improving the accuracy of neural network-based tourism demand forecasting. *TourismManagement*, 21(4):331–340, 2000.

[17] A. J. Menezes, P. C. V. Oorschot, S. A. Vanstone, and R. L. Rivest. Handbook of applied cryptography, 1997.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Parallel distributed processing: explorations in the microstructure of cog-nition, vol. 1. chapter Learning internal representations by error propagation, pages 318–362. MIT Press, Cambridge, MA, USA, 1986.

[19] N. Schlitter. A protocol for privacy preserving neural network learning on horizontal partitioned data. In *Proceedings of thePrivacy Statistics in Databases (PSD)*, Sep. 2008.

[20] S. Stolfo, A. L. P. S. Tselepis, A. L. Prodromidis, S. Tselepis, W. Lee, D. W. Fan, and P. K. Chan. Jam: Java agents for meta-learning over distributed databases. In *In Proc. 3rd Intl. Conf. KnowledgeDiscovery and Data Mining*, pages 74–81. AAAI Press, 1997.

[21] B. Yang, Y.-d. Wang, and X.-h. Su. Research and design of distributed neural networks with chip training algorithm. In *Proceedings of the First international conference on Advances in Nat-ural Computation - Volume Part I*, ICNC'05, pages 213–216, Berlin,Heidelberg, 2005.