A Survey on Privacy Preservation in Cloud Computing

Gurudyal Singh Bhandari¹, Abhishek Chauhan²
Department of Computer Science, NRIIST, Bhopal, India

Abstract — By integrating multiple private and public cloud services, hybrid clouds can effectively provide dynamic scalability of service and data migration. Security plays a vital during the transmission of data from the sender to the receiver in any environment. The challenge in privacy preserving Back-Propagation Neural Network Learning is avoiding the attack of personal data privacy. Due to the enlargement of distributed computing environment, in such distributed scenarios, privacy concerns often become a big concern. Secure computation provides a solution to this problem. With the invention of new technologies, Computing, it has been more convenient than ever for users across the Internet, who may not even know each other, whether it is data mining, in databases or in any networks, resolving privacy problems has become very important.

Index Terms— Privacy Preservation, Back Propagation Neural Network, cloud computing.

I. INTRODUCTION

Learning algorithms are widely used in intrusion detection, medical diagnosis, homeland security bioinformatics, and other fields. The familiar things of these applications are that all of them need to extract patterns and predict trends from a large number of composite data. In these applications, protection and the privacy of sensitive data and personal information from disclosure is an significant issue. Presently the vast preponderance of existing learning algorithms did not consider how to protect the data privacy in the process of learning.

Despite the potential benefits, one crucial issue pertaining to the Internet-wide collaborative neural network learning is the protection of data privacy for each applicant. Especially, the participants from dissimilar trust domains may not want to disclose their private data sets that may include privacy or proprietary information, to anybody else. In order to embrace the Internet wide collaborative learning, it is very important to afford a solution, which allows the applicants, who require mutual trust to perform neural network learning jointly without disclosing their respective private data sets. If possible the solution shall be proficient and scalable enough to support an arbitrary number of participants with every possessing randomly partitioned data set [1].

Neural Networks have been an active research area for decades. Trained neural networks can predict efficient outputs which might be difficult to obtain in the real world.

The expansion of internet and World Wide Web has made it easier to gather data from many sources [23]. Training neural network from the distributed data is common: for example, making use of data from many hospitals to train the neural network to predict a certain disease, collecting datasets of purchased items from different grocery stores and training neural network from those to predict a certain pattern of purchased items. When training neural network from distributed data, privacy is a major concern.

With the invention of new technologies, whether it is data mining, in databases or in any networks, resolving privacy problems has become very important. Because all sorts of data is collected from many sources, the field of machine learning is equally growing and so are the concerns regarding the privacy. Data providers for machine learning are not willing to train the neural network with their data at the expense of privacy and even if they do participate in the training they might either remove some information from their data or can provide false information.

In recent years, Secure Multiparty Computation (SMC) and privacy preservation have attracted much attention in incorporating security into data mining and machine learning algorithms [17], [21], [13], [22], [20], [18]. A natural question is why the researchers would want to build a learning model (e.g., neural networks) without first collecting all the training data on one computer. If there is a learner trusted by all the data holders, then the authenticated learner can assemble data first and build a learning model. Though in many real-world situations, it is rather difficult to find such a trusted learner, since some data holders will always have concerns like “What will you do to my data?” And “Will you discover private information beyond the possibility of research?” alternatively, given the scattered and networked computing environments nowadays, collaborations will greatly benefit the scientific advances. In this they focus on one of the most popular techniques in machine learning, multilayer neural networks in which the privacy preservation problem is far from being practically solved. In [16], a Privacy preserving light weight two party protocol is proposed and implemented. However, it is not generalized for multiparty computation.

The classification quality of neural network depends, amongst others, on the number of training data. As a result, the classification presentation can be increased by using images from more than one health care institution.
However, in this case, privacy issues come up because medical images are considered as sensitive data. As a consequence, conventional methods for knowledge discovery from data are not appropriate; in this case, their use is even restricted by privacy defense laws like the Health Insurance Portability and Accountability Act (HIPAA). Privacy preserving data mining is methods over the chance to build models and extract patterns without disclosing private data. Privacy preservation methods for knowledge discovery can be categorized into two groups, data perturbation methods and cryptographic methods. Methods of the first group use data distortion, such as adding uniform noise, with the purpose of "hiding" private data, or, more formally, of guaranteeing k-anonymity. Methods of the second group are used for collaborative model learning: P >=2 parties contribute their data for the learning of a shared model according to protocols that prevent the disclosure of the contributed data.

In today’s environment, making such assumptions can be difficult and infeasible. Moreover, in certain situations, even though we could trust that the other parties will not abuse our private information, they cannot guarantee that their computer systems and network are secure enough to prevent our information from being stolen. Alternatively, from the trusted party’s point of view, in order to conduct such a cooperative computation, they have to carry the extra burden of securing other party’s data. A security breach that compromises the data may result in serious ramifications. Therefore, it is desirable if nobody knows the other parties’ secret information. Techniques that can support this type of joint computation while protecting the participant’s privacy are of growing importance. There are many multivariate data analysis techniques, such as regression, classification, factor analysis, T2 test, etc. In this they focus on two techniques: multivariate linear regression and classification. Multivariate linear regression concerns about determining a linear function that best fits a set of data observations. Multivariate classification concerns about building a classification model for predicting membership of objects from their measurements on one or more predictor variables [17].

In order to provide practical solutions for privacy preserving back-propagation neural (BPN) network learning, there are three major challenges require being gathered simultaneously [1]:

1) To protect each participant’s private dataset and intermediate results generated during the BPN network learning procedure, it have need of protected computation of diverse operations, e.g. addition, scalar product and the nonlinear sigmoid function that are desired by the BPN network algorithm;

2) To ensure the practicality of the suggested solution, the computation or communication cost commenced to each participant shall be affordable. In order to accommodate a large range of collaborative learning, this solution shall reflect on system scalability. Specifically, it shall be competent to support an arbitrary number of participants without introducing tremendous computation/communication costs to each participant.

3) On behalf of collaborative training, the training data sets may be owned by different parties and partitioned in arbitrary ways rather than a single way of partition.

DATA perturbation, a widely employed and accepted Privacy Preserving Data Mining (PPDM) scheme, tacitly supposes single-level trust on data miners. Diversity utilization across differently disturbed copies, the data miner may be intelligent to assemble a more accurate reconstruction of the original data than what is allowed by the data owner. It includes the colluding attack scenario where adversaries combine their copies to mount an attack; it also includes the scenario where an adversary utilizes public information to perform the attack on its own. Protecting multiplicity attacks is the key challenge in solving the MLT-PPDM problem [22].

Several trends are opening up the era of Cloud Computing that is an Internet-based enlargement and use of computer technology. The constantly cheaper and extra influential processors, together with the “software as a service” (SaaS) computing design, are changing data centers into pools of computing service on a massive range. Temporarily, the rising network bandwidth and reliable yet flexible network connections make it even possible that clients can now subscribe high quality services from data and software that reside solely on remote data centers. Although visualized as a talented service platform for the Internet, this new data storage paradigm in “Cloud” brings about many challenging design issues which have profound influence on the security and performance of the complete system. One of the largest apprehensions with cloud data storage is that of data integrity verification at untrusted servers [11].

Although schemes with private verifiability can achieve higher method effectiveness, public verifiability consents to anyone, not just the client, to face the cloud server for appropriateness of data storage while keeping no private information. Then, clients are capable to entrust the estimation of the service performance to an independent third party auditor (TPA), without devotion of their calculation resources. According to the cloud, the clients themselves are unreliable or cannot afford the overhead of performing frequent integrity checks. Therefore, for practical use, it looks like more coherent to equip the verification protocol with public verifiability that is estimated to play a more imperative role in achieving economies of scale for Cloud Computing.
In order to address this problem, they believe a hybrid cloud storage service involving three different entities, as illustrated in Figure 1: the cloud client who stores or uses data in the cloud; the cloud service provider (CSP) which has significant storage space and computation resources to manage and provide storage services; and the trusted third party (TTP) who stores the clients’ audit data and offers the query services for their data [9].

II. BACKGROUND THEORY

Cloud computing is Internet based computing, whereby information, software and shared resources are provided to computers and other devices on-demand, like a public utility. Infrastructure as a Service is a single tenant cloud layer where the Cloud computing vendor’s dedicated resources are only shared with contracted clients at a pay-per-use fee. This mainly reduces the need for great initial investment in computing hardware such as processing power, networking devices and servers. Software as a Service also operates on the virtualized and pay-per-use costing model whereby software applications are leased out to contracted organizations by specialized SaaS vendors. This is traditionally accessed remotely using a web browser via the Internet. Platform as a service cloud layer works like IaaS but it provides an additional level of ‘rented’ working nature. Clients using PaaS services transmit even more costs from capital investment to operational expenses but must acknowledge the additional constraints and possibly some degree of lock-in posed by the additional functionality layers.

By a privacy-preserving version of inference (for example), we informally mean a protocol in which each party learns their conditional probability of exposure to the disease and absolutely nothing else. More precisely, anything a party can efficiently compute after having participated in the protocol, they could have efficiently computed alone given only the value of their conditional probability— thus, the protocol leaked no additional information beyond its desired outputs.

Secure Multiparty Function Computation

Let \( f(x_1, \ldots, x_k) \) be any function on \( k \) inputs. Imagine a scenario in which there are \( k \) distinct parties, each in possession of exactly one of these inputs (i.e. party \( i \) initially knows \( x_i \)) and the \( k \) parties would like to jointly compute the value of \( f(x_1, \ldots, x_k) \). Perhaps the simplest protocol would have all parties share their private inputs and then individually compute the value of \( f \). However, in many natural settings, we would like the parties to be able to perform this joint computation in a privacy-preserving fashion, with each party revealing as little as possible about their private input.
Simple examples include voting — we would all like to learn the results of the election without having to broadcast our private votes—and the so-called “Millionaire’s Problem” in which two individuals would like to learn who is wealthier, without revealing their precise wealth to each other. If a trusted “third party” is available, one solution would be to provide the private inputs to them, and have them perform the computation in secrecy, only announcing the final result. The purpose of the theory of secure multiparty function computation [16] is to show that under extremely general circumstances, a third party is surprisingly unnecessary.

Privacy-Preserving Belief Propagation

In this section we briefly review the standard algorithm for belief propagation on trees and outline how to run this algorithm in a privacy-preserving manner such that each variable learns only its final marginal’s and no additional new information that is not implied by these marginal’s. The privacy-preserving protocol can be extended easily to handle loopy belief propagation, belief propagation on junction trees, and other message passing algorithms on graphs. In standard belief propagation, we are given an undirected graphical model with vertex set X for which the underlying network is a tree. We denote by V (Xi) the set of possible values of Xi ∈ X, and by N(Xi) the set of Xi’s neighbors. For each Xi ∈ X, we are given a non-negative potential function ψi over possible values xi ∈ V (Xi). Similarly, for each pair of adjacent vertices Xi and Xj, we are given a non-negative potential function ψi,j over joint assignments to Xi and Xj.

The main inductive phase of the belief propagation algorithm is the message-passing phase. At each step, a node Xi computes a message μi→j to send to some Xj ∈ N(Xi). This message is indexed by all possible assignments xj ∈ V(Xj), and is defined inductively by

\[ \mu_{i \rightarrow j}(X_j) = \sum_{x_i \in V(X_i)} \psi_i(X_i)\psi_i,j(X_i,X_j) \prod_{k \in N(X_i) \setminus \{i\}} \mu_k \rightarrow i(X_i) \]

Belief propagation follows the so-called message-passing protocol, in which each vertex of degree d that has received the incoming messages from any d−1 of its neighbors can perform the computation above in order to send an outgoing message to its remaining neighbor. Eventually, the vertex will receive a message back from this last neighbor, at which point it will be able to calculate messages to send to its remaining d−1 neighbors. The protocol begins at the leaves of the tree, and it follows from standard arguments that until all nodes have received incoming messages from all of their neighbors, there must be some vertex that is ready to compute and send a new message.

The message-passing phase ends when all vertices have received messages from all of their neighbors. Once vertex X_i has received all of its incoming messages, the marginal distribution P is proportional to their product. That is, if x_i is any setting to X_i, then

\[ P[X_i = x_i] \propto \prod_{j \in N(X_i)} \mu_{i \rightarrow j}(x_i) \]

**Mask Propagation and the Privacy-Preserving Protocol**

It is assume that at the beginning of the privacy-preserving protocol [19], each node X_i knows its own individual potential function ψ_1, as well as the joint potential functions ψ_{i,j} for all X_j ∈ N(X_i). Recall that our fundamental privacy goal is to allow each vertex Xi to compute its own marginal distribution P[X_i = x_i] (or P[X_i = x_i | E = e] if there is evidence), but absolutely nothing else. In particular, Xi should not be able to compute the values of any of the incoming messages from its neighbors. Knowledge of μ_{i→j} (x_i), for example, along with μ_{i→j} and ψ_{i,j}, may give X_i information about the marginal’s over X_j, a clear privacy violation. They thus must somehow prevent Xi from being able to “read” any of its incoming messages—nor even its own outgoing messages — yet still allows each variable to learn its own marginal’s at the end. To accomplish this we combine tools from secure multiparty function computation with a method we call “mask propagation”, in which messages remain “masked” (that is, provably unreadable) to the vertices at all times. The keys required to unmask the messages are generated locally as the computation propagates through the tree, thus preserving the original communication pattern of the standard (non-private) algorithm.

Before diving into the secure protocol [19], first must fix conventions regarding the encoding of numerical values. We will assume throughout that all potential function values, all message values and all the required products computed by the algorithm can be represented as n-bit natural numbers and thus fall in Z_N = {0, ..., N−1} where N = 2^n. As expressed by Equation, marginal probabilities are obtained by taking products of such n-bit numbers and then normalizing to obtain finite-precision real-valued numbers in the range [0, 1]. It will be convenient to think of values in Z_N as elements of the cyclic group of order N with addition and subtraction modulo N. In particular, we will make frequent use of the following simple fact: for any fixed x ∈ Z_N, if r ∈ Z_N is chosen randomly among all n-bit numbers, then x + r mod N is also distributed randomly among all n-bit numbers. We can think of the random value r as “masking” or hiding the value of x to a party that does not know r, while leaving it readable to a party.
Privacy-Preserving Gibbs Sampling

We now move on to the problem of secure Gibbs sampling on an undirected graphical model G. The local potential functions accompanying G can be preprocessed to obtain conditional distributions for each variable given a setting of all its neighbors (Markov blanket). Thus we henceforth assume that each variable has access to its local conditional distribution, which it will be convenient to represent in a particular tabular form. To simplify presentation, we will assume each variable is binary, taking on values in \{0, 1\}, but this assumption is easy to relax.

If a node \(X_i\) has degree \(d\), the conditional distribution of \(X_i\) given a particular assignment to its neighbors will be represented by a table \(T_i\) with \(2d\) rows and \(d + 1\) columns. The first \(d\) columns range over all \(2d\) possible assignments \(-x\) to \(N(X_i)\), while the final column contains the numerical value \(P[X_i = 1|N(X_i) = -x]\). We will use \(T_i(-x)\) to denote the value \(P[X_i = 1|N(X_i) = -x]\) stored in the \(d + 1\)st column in the row corresponding to the assignment \(-x\). With this notation, the standard (non-private) Gibbs sampling algorithm can be easily described. After choosing an initial assignment to all of the variables in \(G\) (for instance, uniformly at random), the algorithm repeatedly re-samples values for individual variables conditioned on the current values of their neighbors. More precisely, at each step, a variable \(X_i\) is chosen for re-sampling.

Its current value is replaced by randomly drawing value 1 with probability \(T_i(-x)\) and value 0 with probability \(1 - T_i(-x)\) where \(-x\) is the current set of assignments to \(N(X_i)\).

To implement a privacy-preserving variant of Gibbs sampling, we must solve the following cryptographic problem: how can a set of vertices communicate with their neighbors in order to repeatedly that is implied by these values? Again, we would like to accomplish this with limited communication so that no vertex is required to communicate with a vertex more than two hops away. Resample their values from their conditional distributions given their neighbors’ current assignments, without learning any information except their own final values at the end of the process and anything that is implied by these values? Again, we would like to accomplish this with limited communication so that no vertex is required to communicate with a vertex more than two hops away.

III. LITERATURE SURVEY

Yuan, Jiawei, and Shucheng Yu Proposed Privacy Preserving Back-Propagation Neural Network Learning Made Practical with Cloud Computing.

The main idea of this scheme can be summarized as follows: each participant first encrypts her/his private data with the system public key and then uploads the cipher texts to the cloud; cloud servers then execute most of the operations pertaining 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. For the duration of this process, cloud servers discover no privacy data of a participant even if they collude with all the rest participants. During off-loading the computation tasks to the resource-abundant cloud, this scheme makes the computation and communication complexity on each participant independent to the number of participants and is thus extremely scalable. For privacy preservation they crumble most of the sub-algorithms of BPN network into simple operations such as multiplication, addition, and scalar product. To sustain these functions over cipher texts, they adopt the BGN (Boneh, Goh and Nissim) ‘doubly homomorphic’ encryption algorithm [15] and tailor it to split the decryption capability among multiple participants for collusion-resistance decryption [1].

To protect the intermediate data during the learning process, they commence a novel arbitrary sharing algorithm to randomly split the data without decrypting the actual value. Comprehensive security analysis demonstrates that this scheme is secure. They were enabling multiple parties to jointly conduct BPN network learning without revealing their private data. The sets of input data owned by the parties can be randomly partitioned. The communicational and computational costs on each party shall be practically efficient and the system shall be scalable. Thorough analysis investigating privacy and efficiency guarantees of proposed scheme is presented; real experiments on Amazon Cloud further show this scheme’s several magnitudes lower computation/communicational costs than the existing ones [1].

In this approach, the parties encrypt their arbitrarily partitioned data and upload the cipher texts to the cloud. The cloud can implement a large amount operations pertaining to the BPN network learning algorithm without knowing any private information. The cost of each party in this method is independent to the number of parties. This tailors the BGN homomorphic encryption algorithm to support the multi-party scenario, which can be used as an independent solution for other related applications. Complexity and security examination shows that this scheme is secure, efficient and scalable. One interesting future work is to enable multiparty collaborative learning without the help of TA [1].
International Journal of Emerging Technology and Advanced Engineering

W. Du et al [17] defines two S2-MSA problems: Secure 2-party Multivariate Linear Regression (S2-MLR) problem, and Secure 2-party Multivariate Classification (S2-MC) problem. Because MLR and MC problems are built upon matrix computations (multiplication, inverse, etc.), they have developed a set of basic protocols for secure 2-party matrix computations; then we develop our solutions to the S2-MLR and S2-MC problems based on these basic protocols. It should be noted that the building blocks and the methodologies proposed in this paper can be used to solve other privacy-preserving problems beyond the Multivariate Classification and the Linear Regression problems [17].

Wenliang Du and Zhan Zhijun proposed Building decision tree classifier on private data Classification is an important problem in data mining. Although categorization has been considered expansively in the past, the different techniques offered for classification do not work for situations where the data are vertically partitioned: one piece is known by one party, the other piece by another party, and neither party wants to disclose their private pieces to the other party. They presented a solution to this problem using a semi-trusted commodity server. They also discussed the security of our solution and the possible security problem inherent in the decision tree classification method [21].

Q. Wang et al [11] presented a framework and an efficient construction for seamless integration of these two components in our protocol design. Their contribution can be summarized as follows:

1) They suggested a general formal PoR model with public verifiability for cloud data storage, in which blockless verification is achieved;
2) They provide the offered PoR construction with the function of supporting for fully dynamic data operations, particularly to sustain block insertion that is absent in most offered schemes;
3) They also demonstrate the security of our proposed construction and justify the performance of our scheme through concrete implementation and comparisons with the state-of-the-art [11].

Three dissimilar network entities can be recognized as follows:

Client: an entity, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and calculation, can be either personage consumers or organizations;

Cloud Storage Server (CSS): an entity that is supervised by Cloud Service Provider (CSP) has significant storage space and computation resource to maintain clients’ data;

Third Party Auditor (TPA): a TPA, which has expertise and capabilities that clients do not have, is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request.

In the cloud paradigm, by putting the large data files on the distant servers, the clients can be reassured of the burden of storage and calculation. As clients no longer acquire their data locally, it is of vital significance for the clients to ensure that their data are being correctly stored and preserved. That is, clients should be prepared with certain security means so that they can periodically verify the correctness of the remote data even without the existence of local copies. Within situation those clients do not essentially have the feasibility time, or resources to supervise their data; they can entrust the supervising task to a trusted TPA. In this paper, they only consider verification schemes with public verifiability: any TPA in possession of the public key can act as a verifier. They assume that TPA is unbiased while the server is untrusted. Note that they don’t address the issue of data privacy in this work as the subject matter of data privacy in Cloud Computing is orthogonal to the problem they studied here. For application purposes, the clients may interact with the cloud servers via CSP to access or retrieve their pre-stored data. More prominently, in convenient scenarios the client may frequently perform block-level operations on the data files [9].

To ensure cloud data storage defense, it is serious to facilitate a third party auditor (TPA) to evaluate the service quality from an objective and independent perspective. Public verifiability also allows clients to delegate the integrity verification tasks to TPA while they themselves can be unreliable or not be able to commit necessary computation resources performing continuous verifications. An additional major apprehension is how to build verification protocols that can accommodate dynamic data files. In this they explored the problem of providing simultaneous public verifiability and data dynamics for remote data integrity check in Cloud Computing. This construction is deliberately designed to meet these two important goals while efficiency being kept closely in mind [9].

Zhu, Yan et al [9] suggested efficient provable data possession for hybrid clouds. They focused on the construction of PDP scheme for hybrid clouds, supporting privacy protection and dynamic scalability. They first provide an effective construction of Cooperative Provable Data Possession (CPDP) using Homomorphic Verifiable Responses (HVR) and Hash Index Hierarchy (HIH). This construction uses homomorphic property, such that the responses of the client’s challenge computed from multiple CSPs can be combined into a single response as the final result of hybrid clouds. By using this mechanism, the clients can be convinced of data possession without knowing what machines or in which geographical locations their files reside. More importantly, a new hash index hierarchy is proposed for the clients to seamlessly store and manage the resources in hybrid clouds. Their experimental results also validate the effectiveness of their construction.
In CPDP scheme, the client’s communication overhead is not changed in contrast to common PDP scheme, and the interaction among CSPs needs $c \times 1$ times constant-size communication overheads, where $c$ is the number of CSPs in hybrid clouds. Therefore, the total of communication overheads is not significantly increased. Next, they evaluated the performance of CPDP scheme in terms of computational overhead. For the sake of comparison, their experiments were executed in the following scenario: a fixed-size file is used to generate the tags and prove data possession under the different number of sectors $s$. Moreover, there exists an optimal value of $s$ from 15 to 25. The results indicate that the overheads are reduced when the values of $s$ are increased. Hence, it is necessary to select the optimal number of sectors in each block to minimize the computation costs of clients and storage service providers [9].

Li, Yaping et al [4] offered Enabling Multi-level Trust in Privacy Preserving Data Mining. In particular, we focus on the additive perturbation approach where random Gaussian noise is added to the original data with arbitrary distribution, and make available a systematic solution. During a one-to-one mapping, this solution allows a data owner to generate distinctly perturbed copies of its data according to different trust levels. They expand the scope of perturbation based PPDM to multi-level trust, by relaxing the implicit assumption of single-level trust in existing work. MLT-PPDM introduces another dimension of flexibility which allows data owners to generate differently perturbed copies of its data for different trust levels. They identify a key challenge in enabling MLTPPDM services. In MLT-PPDM, data miners may have access to multiple disconcerted copies. By combining numerous perturbed copies, data miners may be capable to achieve assortment attacks to reconstruct the original data more accurately than what is allowed by the data owner [4].

They address this challenge by properly correlating perturbation across copies at different trust levels. They also demonstrate that their solution is robust against diversity attacks. They recommend several algorithms for different targeting scenarios. They also demonstrate the effectiveness of solution through experiments on real data. This solution allows data owners to generate perturbed copies of their data at arbitrary trust levels on-demand. This belonging offers data owner’s highest flexibility. Under the multi-level trust setting, data miners at privileged trust levels can entrance less perturbed copies. Such fewer perturbed copies are not reachable by data miners at lower trust levels. Data miners at unusual trust levels might also colluded to share the perturbed copies between them. As such, it is regular that data miners can have access to more than one perturbed copies. It is true that the data owner may consider releasing only the mean and covariance of the original data.

They remark that simply releasing the mean and covariance does not provide the same utility as the perturbed data. For many real applications, expressive only the mean and covariance may not be sufficient to apply data mining methods, for instance clustering, principal component analysis, and classification. By using random perturbation to release the dataset, the data owner allows the data miner to exploit more statistical information without releasing the exact values of sensitive attributes [4].

In this they [4] expand the scope of additive perturbation based PPDM to multi-level trust (MLT), by relaxing an implicit assumption of single-level trust in exiting work. MLT-PPDM permits data owners to produce differently perturbed copies of its data for different trust levels. The key forefront dishonesty in preventing the data miners from combining copies at different trust levels to jointly reconstruct the original data more accurate than what is allowed by the data owner. They addressed this challenge by properly correlating noise across copies at different trust levels. They also prove that if design the noise covariance matrix to have corner-wave property, then data miners will have no variety gain in their joint reconstruction of the original data. They verify their claim and demonstrate the effectiveness of this solution through numerical evaluation. Many interesting and important directions are significance exploring. Such as, it is not clear how to enlarge the scope of other approaches in the area of partial information hiding, like retention replacement, arbitrary rotation based data perturbation and k-anonymity to multi-level trust. It is also of great interest to extend this approach to handle evolving data streams [4].

Cong Wang et al [2] propose a privacy-preserving public auditing system for data storage security in cloud computing. They utilize the homomorphic linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process that not only removes the burden of cloud user from the tedious and possibly expensive auditing task, although also assuages the users’ fear of their outsourced data escape. Taking into consideration TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, they further extend our privacy-preserving public auditing protocol into a multiuser situation, where the TPA can execute numerous auditing tasks in a batch manner for better efficiency [2].

To accomplish privacy-preserving public auditing, they suggest to uniquely integrating the homomorphic linear authenticator with random masking method. In this protocol, the linear combination of sampled blocks in the server’s reaction is masked with randomness generated by the server.
With random masking, the TPA no longer has all the essential information to construct a accurate group of linear equations and for that reason cannot derive the user’s data content, no issue how many linear combinations of the identical set of file blocks can be composed. Alternatively, the rightness corroboration of the block-authenticator pairs can still be accepted in a new way even with the occurrence of the randomness. Their design makes employ of a public key-based HLA, to provide the auditing protocol with public auditability [2].

By integrating the HLA with random masking, this protocol guarantees that the TPA could not learn any knowledge about the data content stored in the cloud server (CS) during the efficient auditing procedure. The algebraic and aggregation properties of the authenticator further benefit of this design for the batch auditing. This public auditing system of data storage security in cloud computing and provide a privacy-preserving auditing protocol. This scheme enables an external auditor to audit user’s cloud data without learning the data content. This also supports scalable and efficient privacy-preserving public storage auditing in cloud. Specifically, this scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA in a privacy-preserving manner [2].

They may also vigorously work together with the CS to access and update their stored data for various application intentions. As users no longer possess their data in the neighborhood, it is of critical importance for users to ensure that their data are being correctly stored and maintained. To save the computation resource as well as the online burden potentially brought by the periodic storage correctness verification, cloud users may resort to TPA for ensuring the storage integrity of their outsourced data, whereas hoping to keep their data confidential from TPA [2].

In this [2] here they propose a secure cloud storage system supporting privacy-preserving public auditing. Here they extra expand to enable the TPA to perform audits for multiple users simultaneously and efficiently. They utilize the homomorphic linear authenticator and random masking to assurance that the TPA would not gain any knowledge of about the data substance stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the dull and probably costly auditing task, but also improves the users’ fright of their outsourced data outflow. Specifically, there method accomplishes batch auditing where numerous assigned auditing tasks from different users can be performed concurrently by the TPA. This confirms the security and give explanation for the performance of their proposed methods through concrete experiments and comparisons with the demonstrable secure and highly efficient [2].

Srinivas, D. proposed a privacy-preserving public auditing system for data storage security in Cloud Computing. He tries to utilize the homomorphic non-linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process that not only reduces the burden of cloud user from the tedious and possibly pricey auditing task, but also alleviates the users’ terror of their outsourced data security. Taking into account TPA may concurrently handle multiple audit sessions from dissimilar users for their outsourced data files, he further extend this privacy-preserving public auditing protocol into a multi-user scenario, where the TPA can perform multiple auditing tasks in a batch manner for better effectiveness. Extensive examination shows that these schemes are almost certainly secure and highly efficient [8].

The Cloud security responsibilities can be taken on by the customer, if he is managing the cloud, but in the case of a public cloud, such responsibilities are more on the cloud provider and the customer can just try to assess if the cloud provider is able to provide security. Cloud data storage service is involving three different entities.
The cloud user (U), who has large amount of data files to be stored in the cloud; the cloud server (CS) that is managed by cloud service provider (CSP) to provide data storage service and has significant storage space and computation resources; the third party auditor (TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service security on behalf of the user upon request. Cloud users dynamically interact with the CS to access and update their stored data for various application purposes. The traditional cryptographic technologies for data integrity and availability, cannot work on the outsourced data without a local copy of data. It is not a practical solution for data validation by downloading them due to the expensive communications, especially for large size files [8].

The ability to audit the correctness of the data in a cloud environment can be formidable and expensive for the cloud users. Therefore, it is crucial to realize public auditability for CSS, so that data owners may resort to a third party auditor (TPA). This TPA has expertise and capabilities that a common user does not have, for periodically auditing the outsourced data. This audit service is significantly important for digital forensics and credibility in clouds. The users may resort to TPA for ensuring the storage security of their outsourced data, while hoping to keep their data private from TPA. Namely, in most of time it behaves properly and does not deviate from the prescribed protocol execution. However, during providing the cloud data storage based services, for their own benefits the CS might neglect to keep or deliberately delete rarely accessed data files which belong to ordinary cloud users. Moreover, the CS may decide to hide the data corruptions caused by server hacks or Byzantine failures to maintain reputation [8].

They [8] assume the TPA, who is in the business of auditing, is reliable and independent, and thus has no incentive to collude with either the CS or the users during the auditing process. TPA should be able to efficiently audit the cloud data storage without local copy of data and without bringing in additional on-line burden to cloud users. However, any possible leakage of user’s outsourced data towards TPA through the auditing protocol should be prohibited. The audit delegation and authorize CS to respond to TPA’s audits, the user can sign a certificate granting audit rights to the TPA’s public key, and all audits from the TPA are authenticated against such a certificate. The privacy-preserving public auditing, they suggest to uniquely integrate the homomorphic non-linear authenticator with random masking technique. In this protocol, the non-linear blocks in the server’s response is masked with randomness generated the server.

With random masking, the TPA no longer has all the necessary information to build up a correct group of non-linear equations and therefore cannot derive the user’s data content, no matter how many linear combinations of the same set of file blocks can be collected. On the other hand, the correctness validation of the block authenticator pairs can still be carried out in a new way which will be shown shortly, even with the presence of the randomness. This design makes use of a public key based HLA, to equip the auditing protocol with public auditability [8].

In year 2010, Wang, Cong et al [10] offered Privacy-preserving public auditing for data storage security in cloud computing. This work is among the first few ones to sustain privacy-preserving public auditing in Cloud Computing, with a spotlight on data storage. Above and beyond, with the occurrence of Cloud Computing, a predictable increase of auditing tasks from diverse users may be delegated to TPA. As the entity auditing of these growing tasks can be tedious and unwieldy, a natural demand is then how to enable TPA to efficiently perform the multiple auditing tasks in a batch manner, i.e., simultaneously. Allowing for TPA may concurrently handle multiple audit sessions from different users for their outsourced data files [10].

In 2008 by Stephen S. Yau, et. al [12] gives a concept about warehouse for integrating data from various data sharing services without central authorities is existing with our warehouse, data sharing services can update and control the access and limit the usage of their shared data, as a substitute of submitting data to establishment, and our repository will support data sharing and addition. The main differences between their storehouse and existing central authorities are: 1) repository collects data from data sharing services based on users’ integration requirements rather than all the data from the data sharing services as existing central establishment. 2) While existing central establishment have full control of the collected data, the capability of warehouse is controlled to computing the integration results required by users and cannot get other information about the data or use it for other work. 3) The data composed by warehouse cannot be used to generate other results except that of the specified data addition request, and, hence, the cooperation of warehouse can only reveal the results of the specified data integration demand, while the compromise of central establishment will reveal all data and presented a privacy preserving repository to integrate data from various data distribution services. In contrast to existing data allocation techniques, warehouse only collects the least amount of information [12].
Miao Zhou et al. [6] considered the privacy of users in the cloud environment and proposed a flexible method of access control. Each cloud user is linked with certain attributes, which determines their access rights. The paper propounded a two-tier encryption model in which the base phase and surface phase builds up the two tiers of the model respectively. At the first phase, the data owner performs local attribute-based encryption on the data that has to be outsourced. The surface phase on the other hand is performed by the cloud servers, after the initialization done by the cloud data owner. This phase implements the Server re-encryption mechanism (SRM). The SRM dynamically re-encrypts the encrypted data in the cloud, when the owner of that data requests. The request for SRM arises either when a new user has to be created or an existing user has to be repealed. Though the re-encryption takes place in cloud server, the privacy of users data is not compromised as the access policies remains hidden to the cloud servers [6].

Juels et al. [14] describe a proof of retrievability (PoR) model, where spot-checking and error-correcting codes are used to ensure both “possession” and “retrievability” of data files on remote archive service systems. On the other hand, the number of audit confronts a user can perform is predetermined a priori, and public auditability is not maintained in their main method. Even though they explain a straightforward Merkle-tree construction for public PoRs, this method only works with encrypted data. Their scheme utilizes the RSA based homomorphic non-linear authenticators for auditing outsourced data and suggests randomly sampling a few blocks of the file. Nevertheless, the public auditability in their scheme demands the linear combination of sampled blocks exposed to exterior auditor. When used straight, their procedure is not provably privacy preserving, and consequently may escape user data information to the auditor [8].

Taeho Jung et al [21] recommended Privacy Preserving Cloud Data Access with Multi-Authorities. This scheme is able to protect user’s privacy against each single authority. The offered scheme is tolerant against authority compromise, and compromising of up to (N −2) authorities does not bring the whole system down. They provide detailed analysis on security and performance to show feasibility of this scheme. They first implement the real toolkit of multi-authority based encryption scheme. In this scheme, several trees are required in every data file to verify users’ identity and to grant him a privilege accordingly. In this system, there are four types of entities: N Attribute Authorities (denoted as A), Cloud Server, Data Owners and Data Consumers (fig. 3) [21].

A user can be a Data Owner and a Data Consumer simultaneously. Authorities are assumed to have powerful computation abilities that are supervised by government offices since keys act as IDs and partially contain users’ PII (Personally Identifiable Information). The whole attribute set is divided into N disjoint sets and controlled by each authority. One convenient method to separate the attributes set is to divide them by category. The authorities jointly compute a system-wide public key, and independently calculate their master keys at the initialization segment. The public key is employed for all operations inside the system, and the master keys are employed by each attribute authority when he generates private keys for Data Consumers. A Data Owner achieves public key from any one of the establishment, and he utilizes the public key to encrypt the data file before outsourcing it to the Cloud Servers. The Cloud Server, who is assumed to have adequate storage capacity, does nothing but store them. Newly joined Data Consumers request private keys from all of the authorities, and they do not be acquainted with which attributes are controlled by the authorities. Conversely, authorities do not identify which Data Consumers are interacting with them because each of them knows only a part of Data Consumers’ attributes. When the Data Consumers demand their confidential keys from the authorities, establishment jointly generate corresponding private key and send it to them [21].

The proposed method [21] is competent to protect user’s privacy in opposition to each one single authority. This paper presents an anonymous attribute-based privilege control scheme Anony Control to address only the data privacy problem in cloud storage, but also the user uniqueness privacy concerns in existing access control methods.
The proposed scheme understands adjacent to authority give and take, and compromising of up to (N−2) multiple authorities does not bring the entire system downward. By using multiple authorities in cloud computing system, their method attains anonymous cloud data access and fine grained privilege control. They also make available specified analysis on security and performance to demonstrate likelihood of our method demonstrates that Anony Control is both secure and efficient for cloud computing situation. They first implement the genuine toolkit of multi-authority based encryption method. As well, their method stand for the give and take attack towards attributes authorities, which is not faced in many existing efforts [21].

Ranjita Mishra and Sanjit Kumar Dash suggested a Privacy Preserving Repository for Securing Data across the Cloud. In this a privacy preserving repository is being presented for acceptance of integration requirements from clients to share data in the cloud and maintain their privacy, collect and integrate the appropriate data from data distribution services, and return the combination results to users. Their key points are: The data sharing services in the cloud possess the ability to update and control the access and usages of their shared data. That is, data can be updated when required and it can be inferred who is using the data and in what way. The sharing of data in the cloud is done based on the need-to-share principle, which states that the dispatched information of the data is adequate to support client’s integration requirements, but carries no extra information of the data. The repository is limited to gathering data from data sharing services and combining the data to satisfy users’ requirements. The repository will contain no other information apart from that required to deliver the results to the user and it cannot use this data for other purposes [7].

In continuing data integration systems, the concept of a central and trusted authority collecting all data from data sharing services and computing the integration results for users based on the data collected is usually not valid for data sharing services across various organizations. In this system, as in the Fig. 4 given below, our repository will collect only that data required for generating users’ requests. It is assumed that our repository will correctly construct the query plans for users’ integration requirements, decompose the query plans, discover and fetch data from distributed data sharing services, assimilate all data together, and, finally, return the final results to users. Further, they assume that this repository is granted the access to the shared data by all data sharing services, and all shared data is well protected. Because the data sharing services use our context-aware data sharing concept, our repository cannot learn any extra information from the inferential relations of the information it obtains during the process.

This repository consists of two components: the query plan wrapper and the query plan executor. The query plan wrapper is responsible for scrutinizing integration requirements and constructing query plans for the query plan executor [7].

This paper [7] aims at concurrently achieving data confidentiality while still maintaining the harmonizing relations integral in the cloud. Their proposed method enables the data holder to assign most of calculating demanding tasks to cloud servers without revealing data contents or user access privilege information. It also supports other cloud intentions such as lower costs for hardware, maintenance, tuning and support. It delivers high-availability in support of Service Level Agreements (SLAs). As with every tectonic shift in technology, there is a Darwinian ripple effect as we realize which technologies support these changes and which are transferred to legacy systems. Because of their compatibility, cloud computing will used in an ascendance of the shared-disk database. The privacy preserving repository distributes the serious capabilities necessitated for a robust, cost-effective, and secure cloud security implementation.

Ulrich Greveler et al [5] suggested A Privacy Preserving System for Cloud Computing. They aim to build a secure system that can fend off both external and internal attackers. Many previous work deals with issues related to external attackers. Data availability has a very high priority in any company operations. In this system, all data are stored encrypted.
The backup of the database is performed regularly by the cloud service; in addition we require a backup of the Encryption Proxies with the corresponding decryption keys for the system integrity. They automate the backup procedure for Encryption Proxies by establishing system integrity first, then exchanging the decryption keys over a secure channel. All session keys are TPM sealed. By comparing specific PCR values, they are able to attest the integrity between identical hardware. And only if both Encryption Proxies have the same state, the exchange of their key material can take place [5].

The productive database is stored in the cloud, and their architecture aims to protect the content of the database. The system consists of a Data Management, Encryption Proxy and a User Interface. The cloud provides database services and plays the role of data management. The size of the cloud is not restricted, for example that this system would work on a private cloud or single database server. The content of the productive database is encrypted. The user privacy and confidentiality is achieved because the clouds have no access to the original database content. The clients are not bound to a single Encryption Proxy, and it is also possible to use a load balancer to enhance the performance of the system. The productive database includes a Meta data table where Meta information of each user’s transactions is stored. For example, the table could contain a counter which holds the number of service request by a client. This information can be used in the case, when the system is designed to limit the number of service request by a single or a group of clients [5].

A user is not bound to a single encryption proxy, and a load balancer can be used to distribute work load over several proxies. The meta-table has to be signed for protection against replay attacks from cloud administrators. The signing operation can be achieved by the TPM Quote functionalities. When signing, the TPM quote contains information about the hardware state and the status of request tables. The encryption proxy is the key part of the system. It provides user access to the (unencrypted) data. The encryption proxy acts as an intermediary between cloud and users (figure 5). An encryption proxy also serves secure data storage. The secure storage is achieved by the use of a full disk encryption with a TPM protected key-file. The storage is dedicated to the use of data management. All other information are encrypted and stored in external data storage in a cloud [5].

This paper proposed [5] a cloud database storage architecture that avoids the local administrator in addition to the cloud administrator to study about the outsourced database content. Furthermore, machine readable rights appearances are used in order to limit users of the database that require to recognize source. These limitations are not unpredictable by administrators after the database related application is launched, in view of the fact that a new responsibility of rights editors is defined once an application is get underway. Cloud data must be protected not only against external attackers, but also corrupt insiders. This system follows the information-centric approach which aims to make cloud data self-intelligent. In this approach, cloud data are encrypted and packaged with a usage policy. The data when accessed will consult its policy, create a virtualization environment, and attempt to assess the trustworthiness of the data environment via Trusted Computing.

IV. CHALLENGING PROBLEMS

Cloud Computing provide enormous computation power and storage capacity which make possible customer to arrange computation and data demanding applications lacking any investment. Along the processing of such application, a large volume of in-between dataset will be produced and repeatedly to save re-computing cost. On the other hand, preserving the privacy of transitional datasets developed into a challenging problem by examined multiple transitional datasets.

- **Pattern Classification**

The task of pattern classification is to assign an input pattern (like a speech waveform or handwritten symbol) represented by a feature vector to one of many pre specified classes.

Well-known applications include character recognition, speech recognition, EEG waveform classification, blood cell classification, and printed circuit board inspection.
• Clustering & Categorization

In clustering, also known as unsupervised pattern classification, there are no training data with known class labels. A clustering algorithm explores the similarity between the patterns and places similar patterns in a cluster. Well-known clustering applications include data mining, data compression, and exploratory data analysis.

• Prediction & Forecasting

Given a set of n samples \{y(t1), y(t2) y(tn)\} in a time sequence, t, t2, tn, the task is to predict the sample y(t1) at some future time Prediction/forecasting has a significant impact on decision-making in business, science, and engineering. Stock market prediction and weather forecasting are typical applications of prediction/forecasting techniques.

V. OPTIMIZATION

A wide variety of problems in mathematics, statistics, engineering, science, medicine, and economics can be posed as optimization problems. The goal of an optimization algorithm is to find a solution satisfying a set of constraints such that an objective function is maximized or minimized. The Travelling Salesman Problem (TSP), an NP-complete problem, is a classic example.

VI. CONCLUSION

Using Privacy preserving method applied for Back Propagation Neural network on distributed datasets. We know how to preserve privacy of dataset owners and no one can get additional private data then besides our neural network is gets trained for distributed datasets. Various learning problems now have distributed input data, suitable to the improvement of distributed computing background. The cloud computing has rapidly grown in recent years due to the advantages of greater flexibility and availability of computing resources at lower cost. Security and privacy, however, are a concern for agencies and organizations considering migrating applications to public cloud computing environments. The Cloud security responsibilities can be taken on by the customer, if he is managing the cloud, but in the case of a public cloud, such responsibilities are more on the cloud provider and the customer can just try to assess if the cloud provider is able to provide security. In this paper, some of the privacy preservation and security are addressed and the techniques to overcome them are surveyed. While some approaches utilized traditional cryptographic methods to achieve privacy, some other approaches kept them away and focused on alternate methodologies in achieving privacy.

REFERENCES


[4] Li, Yaping, Minghua Chen, Qiwei Li, and Wei Zhang "Enabling multilevel trust in privacy preserving data mining", IEEE Transactions on Knowledge and Data Engineering, Vol. 24, no. 9 pp. 1598-1612, 2012.


