IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 30, 2019, accepted April 9, 2019, date of publication April 12, 2019, date of current version April 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2910885

Enhance PATE on Complex Tasks With Knowledge Transferred From Non-Private Data

LULU WANG[1], JUNXIANG ZHENG[1], YONGZHI CAO[1,2], (Senior Member, IEEE), AND HANPIN WANG[1,3]

[1]Key Laboratory of High Confidence Software Technologies, Ministry of Education, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
[2]School of Mathematics and Statistics, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China
[3]School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China

Corresponding author: Yongzhi Cao (caoyz@pku.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572003, Grant 61772035, and Grant 61751210, and in part by the National Key R&D Program of China under Grant 2018YFC130024 and Grant 2018YFB1003904.

ABSTRACT Privacy protection is considered as an important problem in learning-based systems. Recently, various works based on differential privacy have been proposed to protect an individual's privacy in the machine learning and deep learning contexts. One of the state-of-the-art approaches is Private Aggregation of Teacher Ensembles (PATE), a generic framework which can be successfully applied to many different learning algorithms. In PATE, we need to split the private dataset into many disjoint subsets and train an ensemble of teachers on these subsets. Then, we transfer noisy predictions from the ensemble of teachers to a student model. In this paper, we show that for complex datasets and tasks, such as nature image classification, the training set allocated for one teacher may be too small with respect to the corresponding task to achieve an ideal performance. To alleviate this problem, we propose the TrPATE framework which extends PATE with transfer learning. Based on PATE, we transfer and share the knowledge extracted from a publicly available non-private dataset to the teachers. The extensive experiments are conducted on various datasets, and the empirical results demonstrate the effectiveness of our method.

INDEX TERMS Data privacy, knowledge transfer, machine learning.

I. INTRODUCTION

Recent advances in machine learning and deep learning have achieved great success in many fields such as computer vision, nature language processing, and speech processing. Based on these techniques, learning based systems are becoming ubiquitous in our society. Infrastructure like recommender systems [1], face recognition systems [2], [3], disease prediction systems based on healthcare data [4], etc are emerging. Learning based systems are well known to be data-hungry. They need huge amount of data to be generalized well to unseen input. Many of these applications need to train their model on a dataset that may contain users' private information, such as clinical records, user profiles, or historical behaviors. Releasing models trained on private datasets has been showed to be unsafe [5]–[9], because attackers can recover private information from model parameters. Releasing sensitive results of statistical analysis or machine learning models while protecting privacy has been studied in the past few decades, during which lot of definitions have been proposed such as k-anonymity [6], l-diversity [7], and t-closeness [8]. However, these methods cannot prevent background attacks in which attackers have already known some information about the dataset. One promising approach to this problem is differential privacy [9]–[12] which provides a strong guarantee for privacy protection by injecting noise to the statistical results computed from the private dataset.

Differential privacy has been widely adopted by the machine learning research community as the standard for privacy protection [13]–[23]. Some of the studies focus on a certain machine learning algorithm and develop a differentially private version of it, such as differentially private logistic regression [13], differentially private principal components analysis (PCA) [14], and differentially private matrix

The associate editor coordinating the review of this manuscript and approving it for publication was Bilal Alatas.

factorization [15], and some of them apply differential privacy to release data in a privacy preserving manner [16]–[18]. While some others focus on a general framework that can be applied to many algorithms in a uniform way, such as PrivGene [19], a novel framework for differentially private model fitting based on genetic algorithms, differential private SGD [20], which can be applied to algorithms optimized with stochastic gradient descent, and PATE [21], which outperforms existing approaches on MNIST and SVHN with an improved privacy analysis and a novel teacher-student framework. A refined aggregation process proposed by [24] improves the performance of PATE with Gaussian instead of Laplace noise and a better privacy analysis. However, it also has the same limitations as PATE which we will demonstrate and try to handle in this paper.

PATE uses an ensemble of teachers trained on private data to ''teach'' a student model, and the student model which does not access to the private data directly is released finally. Due to the privacy concern, certain scale of noise should be added to the output of the teacher ensembles. In PATE, the ensemble of teachers is designed to be trained in disjoint private training set in order to restrict its sensitivity which determines the scale of noise.

It has been shown experimentally that generally we need hundreds of teachers to make the aggregated votes robust to the noise injected [21]. The demand for large number of teachers and the requirement of disjoint training set for each teacher are the main reasons why PATE's performance will decrease on complex datasets and tasks. For a complex dataset like CIFAR-10, its intra-class variance is much larger than that of a simpler dataset. Thus we usually need a larger model to ''explain'' this variance. Accordingly, due to a great number of parameters required in a larger model, training one model needs considerable amount of samples. Since training hundreds of teachers needs to split the training set into hundreds of disjoint parts, there is a dilemma. If the number of teachers is large enough to be robust to the noise injected, a single teacher's performance will be very poor due to the limited training data. On the other hand, if we want each teacher to have an ideal accuracy, the number of teachers will be very small, thus the output of the ensemble is vulnerable to the injected noise. For example, CIFAR-10 contains 50000 training instances which is enough to train a single model. However, in PATE if we want to train 200 teachers, there will be only 250 instances for training one teacher.

Fig. 1 shows the classification accuracy on MNIST and CIFAR-10 with respect to the training set size. We can observe the difference between a simple dataset and a complex one: the accuracy on MNIST does not reduce much as the training set size decreases, but for CIFAR-10, the accuracy decreases sharply. We can also find from Fig. 1 that given only 500 samples, the model trained on MNIST can still achieve a 91.7% accuracy but for CIFAR-10 the accuracy decreases to only 42.8%. So for a simple dataset, the limited number of training instances may not be a big concern. In PATE, the model ensemble can compensate for the slightly loss of

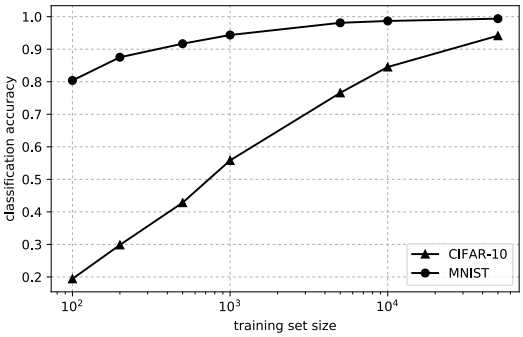


FIGURE 1. Influence of training set size on the accuracy of MNIST model and CIFAR-10 model. The architecture for MNIST model consists of two convolutional layers and each convolutional layer is followed by a pooling layer. It achieves 99.4% accuracy on full training set. The model for CIFAR-10 is VGG13 which achieves 94.13% on full training set.

single teacher's accuracy. But for a complex dataset, the performance degrades significantly because the training set is too small with respect to its complexity and the corresponding task.

Since a small training set cannot provide enough information for a large model to capture the large variance of the distribution of a complex dataset. One way to handle this problem is just collecting more data to satisfy the demand for training set size of each teacher. However, this will be too expensive due to the large number of teachers.

In this paper, we extend the original PATE framework and propose TrPATE which uses transfer learning to alleviate the performance degradation problem. We leverage transfer learning [25] to transfer useful prior from a publicly available non-private dataset to the teacher ensembles. In TrPATE, the transferred knowledge is shared among all of the teachers and each teacher is trained on both the private data and the transferred knowledge.

Note that the idea of leveraging another dataset to promote the performance of a model trained on few labeled data has been exploited in various scenarios. For example, in computer vision, to promote the performance of image classifier, [26] propose to transfer knowledge from the abundant text data available on the Web with the help of some annotated images which serve as a bridge to connect the two domains. In speech recognition, the external data from multiple languages is sometimes used to train a multilingual bottleneck feature extractor [27]–[30] which is used to overcome the limited quantity of data in the new languages. Similarly, to improve the performance of machine translation on low resource language pair, a model is trained on high-resource language pair, and then transfer some of the learned parameters to the model trained on low-resource pair [31], [32].

This paper try to use transfer learning to alleviate the proposed limited data problem of PATE caused by splitting the training set for each teacher. The main contributions of this paper can be summarized as follows:

1) We demonstrate that the disjoint training sets for teacher ensembles of PATE will cause the performance

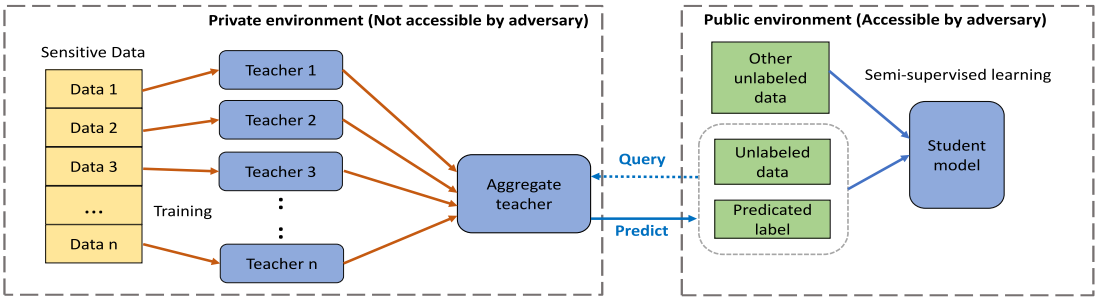


FIGURE 2. Overview of PATE: (1) the ensemble of teachers is trained on disjoint subsets of the sensitive dataset, (2) the ensemble of teachers predicates non-sensitive labels for queries, (3) a student model is trained on public dataset and a few non-sensitive labels with semi-supervised learning.

degradation problem on complex datasets and tasks due to the limited training set for each teacher.

2) To alleviate this problem, we propose TrPATE (an extension of PATE) which use transfer learning to transfer and share the knowledge to every teachers of the ensemble from a related non-private dataset.
3) We apply TrPATE to deep neural networks and Bayesian models, and conduct extensive experiments on different datasets with these models. The experimental results demonstrate the effectiveness of our proposed method.

The rest of this paper is organized as follows. Section II reviews background of differential privacy, PATE and transfer learning. Section III demonstrates the details of our approach. Section IV shows the application of TrPATE to two widely used models. The experimental results are reported in Section V. The last section concludes this paper.

II. PRELIMINARY

In this section, we firstly give an overview of differential privacy, after that we review the details of the PATE framework, and then we give the basic concepts of transfer learning.

A. DIFFERENTIAL PRIVACY

Differential privacy provides a strong guarantee of privacy protection for the released information of a private dataset. Intuitively, differential privacy requires the information we want to release about a dataset should be robust to any changes of one sample. To formalize this intuition, one of the definitions was given by [33],

Definition 1: A randomized mechanism $\mathcal{M}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ϵ, δ)-differential privacy if for any two adjacent datasets (differing at most one record) $d, d' \in \mathcal{D}$ and for any subset of outputs $\mathcal{S} \subseteq \mathcal{R}$, it holds that

$$\Pr(\mathcal{M}(d) \in \mathcal{S}) \leq e^{\epsilon} \Pr(\mathcal{M}(d') \in \mathcal{S}) + \delta.$$

The (ϵ, δ)-differential privacy is a relaxed form of the original definition: ϵ-differential privacy [9]. We can informally interpret (ϵ, δ)-differential privacy as ϵ-differential privacy that may fail with probability δ [34]. Accordingly, when $\delta = 0$, the two definitions are equivalent.

We denote a deterministic function by $f : \mathcal{D} \to \mathcal{R}$ which computes the released information $r \in \mathcal{R}$ from a sensitive dataset $d \in \mathcal{D}$. A random mechanism adds some noise to the result of f, so that the released information becomes a random variable. The random mechanism can be defined as follows

$$\mathcal{M}(d) = f(d) + \eta(\epsilon, s(f)),$$

where $\eta(\epsilon, s(f))$ denotes the injected noise. Different mechanisms have different forms of η. For example, in Laplacian mechanism η follows the Laplacian distribution, and in Gaussian mechanism η follows the Gaussian distribution. The value $s(f)$ means the sensitivity of f. Note that the scale of the noise is controlled by both the sensitivity $s(f)$ and the privacy parameter ϵ. Specifically, for Laplacian mechanism we have

$$\eta(\epsilon, s(f)) = \text{Lap}(\frac{s(f)}{\epsilon}).$$

Sensitivity bounds the possible change in computing the output of f over any two adjacent datasets. For Laplacian mechanism, the sensitivity is defined as follows.

Definition 2: The sensitivity s of $f : \mathcal{D} \to \mathcal{R}$ is defined as

$$s(f) = \max_{d, d' \in \mathcal{D}} \|f(d) - f(d')\|_1$$

where d and d' are any two adjacent datasets which differ at most one record.

B. PRIVATE AGGREGATION OF TEACHER ENSEMBLES

In this subsection, we give a brief introduction of the PATE framework. PATE assumes that we have access to additional public unlabeled data, which is non-sensitive. This is a reasonable assumption, because in reality, a non-overlapping, unlabeled set of data often exists. The overview of PATE is illustrated in Fig.2. PATE consist of three main parts:

1) Teacher ensembles: PATE trains many teachers and use these teachers to vote for the labels of given public unlabeled data. Before selecting the most voted label, some noise needs to be added to the votes. In order to reduce the noise scale injected to the votes, the sensitivity of votes should be small, so we need to split the private dataset into some disjoint sets, and train every teacher separately on different sets, such that the

arbitrary changes of one sample in the private dataset can only influence one teacher at most. Therefore the difference of the votes of any input x on adjacent datasets d and d' is at most 1 in at most 2 locations, therefore the sensitivity of the output votes is 2.

2) Noisy aggregation mechanism: After training an ensemble and getting the votes of the teachers for a given sample, we need to add some noise to this votes. Let m be the number of classes in our task, $f_j(x)$ be the j-th teacher's prediction for input x, and $n_i(x)$ be the number of votes for class i given input data x. We have $n_i(x) = |\{j : f_j(x) = i, j \in [1, n]\}|$, where $i \in [1, m]$ and $\sum_{i=1}^{m} n_i = n$. For convenience, we denote $i \in \{1, 2, \ldots, n\}$ by $i \in [1, n]$ in the rest of this paper. The aggregation mechanism selects the class label for input x in the following way:

$$\mathcal{A}(x) = \arg\max_i \{n_i(x) + \mathrm{Lap}(\frac{1}{\gamma})\},$$

where γ is privacy parameter and it controls the noise scale we add to the votes. Because the sensitivity of votes is 2, according to the Laplacian mechanism, we have that $\epsilon = 2\gamma$. So it is $(2\gamma, 0)$-differential privacy for querying one time of the teacher ensembles with aggregation mechanism.

3) Student model: The student is trained in a semi-supervised way on the public unlabeled data and a few labeled data which are obtained by labeling some unlabeled data using the aggregation mechanism. The student model does not access the private data directly. Thus, the privacy of users who contributed to the original training dataset is preserved even if the student model's architecture and parameters are released to the public.

We have just reviewed the framework of PATE, and now we give a brief introduce of its privacy analysis. To train a student model, we need to query the teacher ensembles many times to get enough labeled data, and we need to apply the mechanism once per query. In order to measure the total privacy loss of these queries, a lot of analytic results have been proposed and proved, such as strong composition theorem [12]. The moments accountant introduced recently by [20] provides a much tighter estimate of the privacy loss. Based on moments accountant, the following privacy guarantee of PATE's aggregation mechanism is provided in [21].

Theorem 1: (Privacy Guarantee of PATE). For any selected γ and δ, queuing T times with aggregation mechanism satisfies (ϵ, δ)-differential privacy, where

$$\epsilon = 4T\gamma^2 + 2\gamma\sqrt{2T\ln\frac{1}{\delta}}$$

Note that, a data dependent method to calculate the privacy guarantee of PATE based on moments accountant is also proposed in [21]. It can track the actual privacy loss more accurate by bounding specific moments for each query based on its votes. Because it is data dependent, the privacy parameters cannot be released directly. Thus it needs further process: bounding the smooth sensitivity [35] of the moments and adding noise to the moments themselves. Since computing a more accurate privacy guarantee is not the focus of this paper, and due to the limited space, we will not give the details of this method.

C. TRANSFER LEARNING

Transfer learning [36]–[41], [51] is proposed to improve the performance of a model trained on a target domain by transferring information from a related domain.

Formally, a domain $\mathcal{D}$ is defined by two components $\mathcal{D} = \{\mathcal{X}, \mathrm{P(X)}\}$, where $\mathcal{X}$ is the feature space, and P(X) is the marginal probability distribution of sample X in space $\mathcal{X}$. For a given domain $\mathcal{D}$, a task $\mathcal{T}$ is defined by $\mathcal{T} = \{\mathcal{Y}, \mathrm{P(Y|X)}\}$, where $\mathcal{Y}$ is the label space, and P(Y|X) is a predictive function which predicts the conditional probability of $\mathrm{Y} \in \mathcal{Y}$ given $\mathrm{X} \in \mathcal{X}$.

Given a source domain $\mathcal{D}_S$ with its corresponding source task $\mathcal{T}_S$, and a target domain $\mathcal{D}_T$ with its corresponding target task $\mathcal{T}_T$, transfer learning aims to improve the performance of predictive function in $\mathcal{T}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$.

Given the above definitions, the condition $\mathcal{D}_S \neq \mathcal{D}_T$ means that $\mathcal{X}_S \neq \mathcal{X}_T$ or $\mathrm{P}(\mathrm{X_S}) \neq \mathrm{P}(\mathrm{X_T})$. This condition implies that the target domain and the source domain have different feature spaces or different marginal distributions of the input features. Similarly, another possible condition $\mathcal{T}_S \neq \mathcal{T}_T$ means that $\mathcal{Y}_S \neq \mathcal{Y}_T$ or $\mathrm{P}(\mathrm{Y_S}|\mathrm{X_S}) \neq \mathrm{P}(\mathrm{Y_T}|\mathrm{X_T})$. This condition indicates that the source task and the target task have different label spaces or different conditional distributions of the labels.

Even the source and the target are not required to be exactly same in transfer learning, they need to be sufficiently related. If they are poorly related, the attempt to transfer knowledge from the source may have a negative impact on the target learner. This situation is often referred to as negative transfer [42]. Therefore, whether transfer learning will improve the performance of the target learner depends on the relationship between the source and the target. Unfortunately, it is inherently difficult to find a true measurement of this relationship, and how to avoid negative transfer is still an open problem [36], [37].

III. METHOD

In this section will present TrPATE an extension of PATE to handle the performance degradation problem we have proposed in Section I. TrPATE transfers knowledge in two directions: (1) from public environment to private environment and (2) from private environment to public environment. To be more specific, we transfer knowledge not only from private teacher ensembles to publicly released student, but also from public dataset to private teacher ensembles. Note that, in TrPATE we need two kinds of public datasets: (1) one unlabeled dataset for querying the teacher ensembles

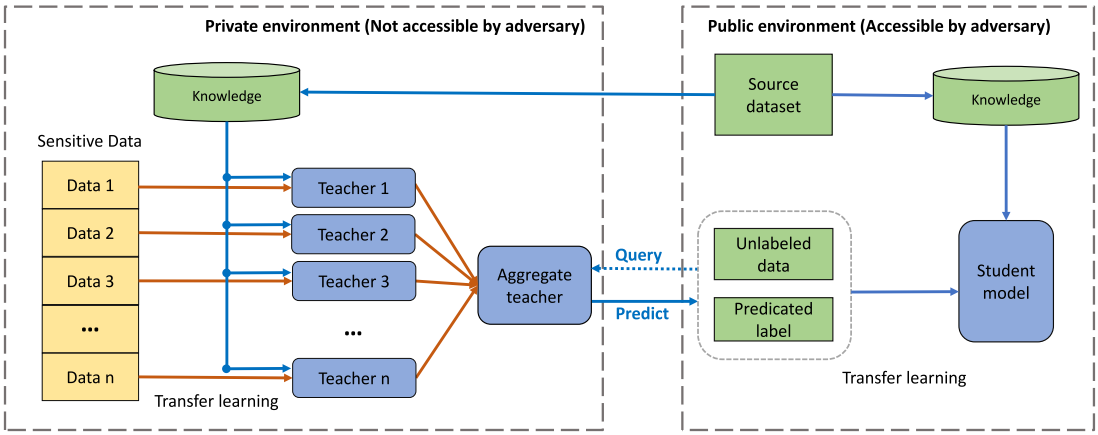


FIGURE 3. Overview of TrPATE: Knowledge is transformed in two directions (1) from public environment to private environment, and (2) from private environment to public environment. Each teacher is trained with both the corresponding private subset and the transferred knowledge from the source dataset. All of the teachers share the same transferred knowledge. (Blue arrow denote the flow of non-sensitive information and orange arrow denote the flow of sensitive information.)

and training student, which are required to have the same distribution with the private dataset; (2) one publicly available labeled dataset used as source dataset for transfer learning, which does not need to have the same distribution with the private dataset. In what follows, we will first introduce the overview TrPATE framework. Then we will talk about the training of student model. After that we give the privacy analysis of TrPATE. At last, we will discuss the considerations we need to give when apply transfer learning to TrPATE.

A. OVERVIEW OF TrPATE

In this subsection, we will reformulate the PATE framework and give the generalized formulation of TrPATE.

We denote the private target dataset by D_T. In PATE, we train n teacher models $m_1, m_2, \ldots, m_n$ on disjoint subsets of $\mathrm{D}_{T_1}, \mathrm{D}_{T_2}, \ldots, \mathrm{D}_{T_n}$, where

$$\mathrm{D}_T = \mathrm{D}_{T_1} \cup \mathrm{D}_{T_2} \cup \mathrm{D}_{T_3} \cdots \cup \mathrm{D}_{T_n}.$$

Suppose that $\mathcal{A}$ is a learning algorithm which returns the model m_i for teacher i. We can formulate the training of teacher models of PATE framework as follows:

$$m_i = \mathcal{A}(\mathrm{D}_{T_i}), \text{ for } i \in [1, n].$$

Usually, a teacher m_i is parameterized by a high dimensional vector, in which each value can be viewed as a statistic estimated from dataset D_{T_i} with the algorithm $\mathcal{A}$. For complex tasks, the dimension of m_i's parameter tends to be extremely large, and the dataset D_{T_i} may be insufficient for training m_i, so the m_i's performance can be very poor.

In TrPATE, we consider another source dataset D_S which is assumed to be public available and is related to the private target dataset D_T. The training of teacher models in TrPATE can be formulated as:

$$m'_i = \mathcal{A}'(\mathrm{D}_{T_i}, \mathcal{B}(\mathrm{D}_S)), \text{ for } i \in [1, n],$$

where $\mathcal{B}$ is another learning algorithm which can extract some prior knowledge from D_S for $\mathcal{A}'$, the $\mathcal{A}'$ is a modified algorithm of $\mathcal{A}$ which can make use of not only the private training data D_{T_i} but the transferred knowledge from $\mathcal{B}$, and m'_i is the model returned by the modified algorithm $\mathcal{A}'$. This new framework allows the training of m'_i with both the original private dataset D_{T_i} and the auxiliary dataset D_S. The overview of TrPATE is demonstrated in Fig.3.

B. TRAINING STUDENT MODEL

The noisy aggregation mechanism labels some of the public unlabeled data in a privacy-preserving fashion, and then these labeled data are used to train the student model. In PATE, the student model is trained with semi-supervised learning on these labeled date and some unlabeled data. In TrPATE, we provide a new way to train the student model. We can train the student and the teachers in a uniform way. That is we train the student on the predicted non-sensitive labeled data and the transferred knowledge. This approach not only simplifies the training of student by reusing the transferred knowledge (e,g: pre-trained lower layers for neural networks or prior for Bayesian models), but also reduces the requirements for the quantity of unlabeled data.

C. PRIVACY ANALYSIS OF TrPATE

Since the source dataset used in TrPATE is assumed to be publicly available, there is no privacy concern for the transferred knowledge of the teacher ensembles. The training process of the teachers and the aggregation mechanism for processing the outputs of the ensemble are the same as PATE. So the privacy guarantee of PATE still holds for TrPATE. We can directly apply Theorem 1 to calculate the privacy loss of TrPATE.

D. DISCUSSION

The logic behind TrPATE is to promote the poor performance (caused by the disjoint split of the private dataset) of each

teacher on complex datasets and tasks. Therefore, whether the performance of TrPATE can be really promoted depends directly on whether each teacher can be enhanced by transfer learning with the chosen source dataset. Previous results show that negative transfer may happen if the domain or task of the target dataset and the source dataset are poorly related. An extreme case is when these datasets are totally unrelated, the performance of the target model may even be harmed. Therefore, the effectiveness of TrPATE is closely related to the choice of source dataset.

Naturally, to exploit the full potential of transfer learning in TrPATE, we should choose a source dataset which is similar enough (closely related) to the target dataset. However, the criteria to measure the similarity between domains or tasks exactly, the definition of the transferability between domains and tasks, and the mechanism of how to avoid negative transfer are still open issues in transfer learning [36], [37].

Despite the lake of theoretical result on how to avoid negative transfer, some proposed methods [43]–[45] are demonstrated empirically to be able to lessen the effects of negative transfer. Such as TrAdaBoost [43] which assign different weights for source instances in every boosting iteration, and its extension [44] which import knowledge from multiple sources to increase the chance of finding one source closely related to the target. various domain adaption methods can also alleviate negative transfer by mapping features of source domain and target domain into a common space, and minimizing some measure of domain variance (such as the Maximum Mean Discrepancy (MMD) [41], [46], [47]), or matching moments of the marginal distributions of the two domains [48]. Besides, some other works exploit the relationship between common tasks such as Taskonomy [49] which discovered a task similarity structure empirically of more than 20 common tasks in computer vision, and [50] which exploiting similarities among several different languages in machine translation task.

In order to reduce the possibility of negative transfer in TrPATE, we should pay close attention to the choice of source dataset. We can take advantage of the existing methods mentioned above to lessen the effect of negative transfer, or we can select certain domain and task of source dataset according to the existing exploited similarity structure (like Taskonomy [49]) to get a more similar source dataset for our target dataset. Heuristically, we can also choose a source dataset that has a large overlap with target dataset in terms of domain spaces and label spaces. Note that, all of the advises mentioned above may be helpful to avoid negative transfer but cannot ensure it.

In TrPATE, given that the source dataset and task are related to our target dataset and task, it is expected that the performance of each teacher can be enhanced. However, it is still unclear whether the enhancement of the ensemble of teachers (due to the improvement of each teacher) is robust to the noise added to its output. It is possible that the noise added to the votes of the teacher ensemble may take away the improvement of each teacher. To verify whether or to what extent TrPATE can enhance the performance of PATE on complex tasks and limited training data, we will apply TrPATE to concrete models in Section IV and conduct various experiments with these models under different condition in Section V.

IV. TrPATE FOR CONCRETE MODELS

The TrPATE we have just demonstrated is an abstract framework. The original algorithm $\mathcal{A}$, the algorithm $\mathcal{B}$ for extracting knowledge, and the modified algorithm $\mathcal{A}'$ are not specified. In this section, we will provide some examples of how to apply TrPATE to concrete models. Besides, to verify whether TrPATE is effective, and to make the experimental results in the next section more convincing. We will design TrPATE for two widely used models deep neural networks and Bayesian models with the common paradigms of knowledge transfer method. Because using specially selected models and heavily designed knowledge transfer methods would harm the generalizability of experimental results.

A. TrPATE FOR DEEP NEURAL NETWORKS

1) KNOWLEDGE TRANSFER FOR DEEP NEURAL NETWORKS

Training a deep neural network with a few labeled data has been well studied in a lot of works. Transfer learning is one of the commonly used method. In this subsection, we show how to use transfer learning to transfer knowledge from a public dataset to private teacher ensembles. Firstly, we use a non-private source dataset to train a network. Then we reuse the learned feature extractor by transferring it to the target network which will be trained on a subset of private dataset. When the target dataset is significantly smaller than the source dataset, transfer learning can be a powerful tool to enable training a large target network without overfitting [51]. Note that transfer learning does not require that the public (source) dataset and the private (target) dataset have exactly domain and task, instead it requires that the domains and tasks of the two datasets are related. Also note that we still need some unlabeled public data that has the same domain with the private dataset to be used as queries to the teacher ensembles. Another benefit of using transfer learning is that we can directly use a neural network that has already been trained before to solve a similar problem instead of building a model from scratch.

Transfer learning works in neural networks because many tasks especially similar tasks share features in their lower layers. So, in practice we usually train a base network on source dataset and then copy its first t layers to the first t layers of a target network, and then randomly initialize the remaining layers of the target network. After that, we can choose to train or freeze the first t layers of the target neural network during training on the new task (on target dataset). When the target dataset is small and the number of parameters is large, training the whole neural network may result in over-fitting. In that case we often choose

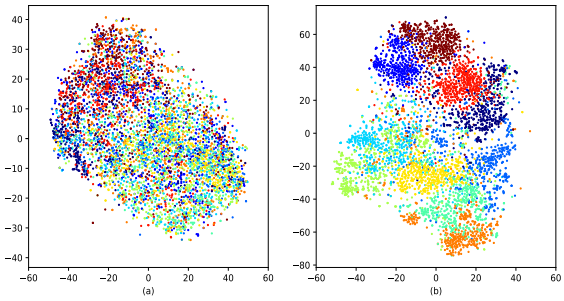


FIGURE 4. Visualization of t-SNE embedding of the high dimensional manifold of 5000 CIFAR-10 instances with respect to its raw features (a) and features transformed with lower layers of VGG13 (b).

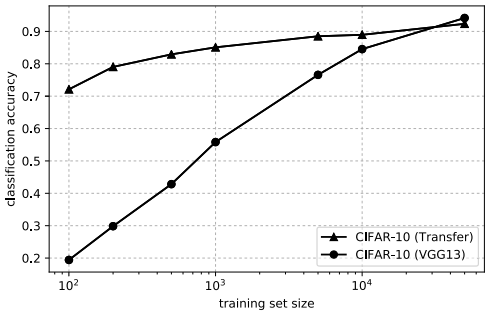


FIGURE 5. Influence of training set size on the accuracy of two models trained on CIFAR-10. The first model is VGG13 and is denoted by CIFAR-10 (VGG13) in this figure. The second model is a two layer neural networks build on top of the lower layers of VGG13 pre-trained with ImageNet and is denoted by CIFAR-10 (Transfer) in this figure.

to freeze the first t layers when training the top layers on the target dataset.

In deep neural networks, the output of each layer can be viewed as a transformation of the input features. The input of the bottom layer is the raw features, and the output of the final layer is the final low-dimensional discriminative features. Datasets of similar domains and tasks usually share the similar raw features and the low level transformation. In Fig.4, we select the first 5000 samples of CIFAR-10 and reduce its dimension to 2 with t-SNE and visualize them in (a). Then we feed these samples to the VGG13 [54] trained on ImageNet [53] dataset (without the top layer), and get a 4096 dimension representation of the samples, and then visualize them in (b) after reducing to 2 dimensions. From this picture we can see that after the transformation, the features of CIFAR-10 become easier to separate because the instances of the same class are roughly clustered into groups. This indicates that the lower layers of the VGG13 trained on ImageNet did extract useful features for CIFAR-10. Thus, comparing with training a classifier from scratch, training on top of a transferred neural network is much easier. Usually, a shallow neural network is capable, and the required instances are also much fewer, since there are fewer parameters to fit.

Fig. 5 shows the comparison of two models trained on CIFAR-10. The first model is VGG13 [54] which is trained from scratch and the other model uses a two layer neural network (256 hidden units) on top of the transferred lower layers

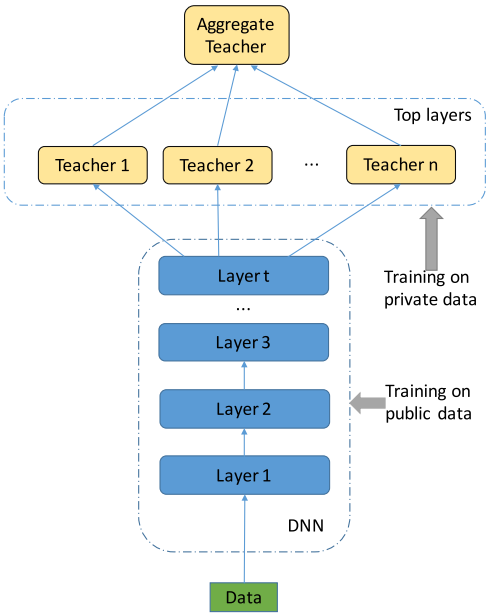


FIGURE 6. The method for training teacher ensembles for deep neural networks. The lower layers are trained on public source dataset and the top layers are trained on private dataset. The lower layers are frozen when training the top layers and all of the teachers share the same lower layers.

from a pre-trained VGG13 on ImageNet. We can see from this figure that when the size of training set decreases, the accuracy of the model with transferred lower layers decreases much slower than that of the model trained from scratch. For example, when there are only 500 instances, the former model can still achieve an accuracy of 83% while the accuracy of the later one is only 42.8%. Note that, in this model the transferred lower layers of VGG13 trained on ImageNet contains 129 million parameters, and the top 2 layers trained on private dataset CIFAR-10 contains 10.5 million parameters. Therefore, only 7.5% of the parameters are trained on private dataset.

We can view transfer learning as a method that can transform a complex dataset to a simpler one. In light of this, we can make it possible for TrPATE to achieve an ideal performance on a complex dataset by training teachers with transfer learning.

2) SHARING LOWER LAYERS BETWEEN TEACHERS

Since the lower layers are trained on public dataset and are frozen when training the top layers on private dataset, the weights of the lower layers do not depend on any private information. Thus there is no privacy concern, and we could reuse the lower layers for every teacher. Fig. 6 shows our approach for training the teacher ensembles. All of the teachers share the same lower layers transferred from public dataset, and are different only in their top layers.

B. TrPATE FOR BAYESIAN MODELS

In this subsection we focus on another family of models: Bayesian models. Bayesian modeling is one of the most

important branches of traditional machine learning. Different from deep neural networks where we transfer knowledge by transferring lower layers trained on source dataset, Bayesian models, such as Bayesian linear regression, Bayesian logistic regression, and probabilistic graphic models, cannot apply the knowledge transfer method proposed for deep learning directly. In Bayesian setting, we transfer knowledge from source dataset to the priors distribution of the target model's parameters. We will give the details of this approach for knowledge transfer.

In Bayesian approach, machine learning problem can be viewed as estimating the posterior distribution of the model's parameters. Assume that the data follow an unknown distribution $P(x)$, and we want to model this distribution with some observed instances: $D = \{x_1, x_2, \ldots, x_n\}$. The model we use to fit this distribution is parameterized by θ, and we want to know the posterior distribution of θ given its prior distribution $P(\theta)$ and the observed instances. According to the Bayes' theorem,

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)},$$

where $P(D|\theta)$ is the likelihood and $P(D)$ is a constant which normalizes $P(\theta|D)$ to be a distribution.

In Bayesian inference, the prediction of a Bayesian model depends not only on the training data, but also the prior. This important property can be leveraged to transfer knowledge from a datasets of similar domains and tasks. In this subsection, we will design TrPATE for Bayesian models by transferring knowledge to the prior distribution and sharing this distribution with all of the teachers.

To make it intuitive, we will provide an example to demonstrate the reason why transfer useful knowledge to prior can mitigate the problem of limited training data. For conciseness and clarity, here we only focus on Bernoulli distribution. We use Beta distribution as the prior for its parameters, so that the posterior can be derived in closed form and has the same form as the prior. Therefore, we can observe clearly how the posterior of model parameters depend on the observed data and its prior. Although we only focus on this simple case in the following analysis, our method can also be applied straightforwardly to other models.

Suppose that the collected data $D = \{x_1, x_2, \ldots, x_n\}$ follows an unknown Bernoulli distribution: Burn(λ). We want to estimate the posterior distribution of its parameter λ. We choose the Beta distribution as the prior distribution of the parameter λ. The probability density function of Beta distribution is

$$p(\lambda) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\lambda^{\alpha-1}(1-\lambda)^{\beta-1},$$

where $\Gamma(x)$ is Gamma function, and it is defined as

$$\Gamma(z) = \int_0^\infty x^{z-1}e^{-x}dx.$$

For short, we abbreviate Beta distribution to

$$p(\lambda) = \text{Beta}(\lambda|\alpha, \beta),$$

where α, β are hyper-parameters, the value of which can be transferred from a model trained on a similarity dataset. Given the conjugate prior distribution, we can drive the posterior distribution of λ with Bayes' theorem

$$\begin{aligned} p(\lambda|D) &= \frac{p(D|\lambda)p(\lambda)}{p(D)} \\ &= \frac{\text{Burn}(D|\lambda)\text{Beta}(\lambda|\alpha,\beta)}{p(D)} \\ &= \text{Beta}(\lambda|\hat{\alpha},\hat{\beta}), \end{aligned}$$

where

$$\begin{aligned} \hat{\alpha} &= \alpha + \sum x_i \\ \hat{\beta} &= \beta + \sum (1 - x_i). \end{aligned}$$

We can observe from this posterior distribution that the parameter λ is determined by not only the training data, but also the prior we choose.

Now we consider the mean of the prior distribution of λ. It can be easily derived as

$$\text{E}(\lambda) = \frac{\alpha}{\alpha+\beta}.$$

In the same way, we can get the mean of the posterior distribution of λ as follows

$$\begin{aligned} \text{E}(\lambda) &= \frac{\hat{\alpha}}{\hat{\alpha}+\hat{\beta}} = \frac{\alpha+\sum x_i}{\alpha+\beta+n} \\ &= \frac{n}{\alpha+\beta+n}\cdot\frac{\sum x_i}{n} + \frac{\alpha+\beta}{\alpha+\beta+n}\cdot\frac{\alpha}{\alpha+\beta}, \end{aligned}$$

where $\frac{\sum x_i}{n}$ is the mean value of the data, $\frac{\alpha}{\alpha+\beta}$ is the expectation of the prior. We can also find that $\frac{n}{\alpha+\beta+n}$ and $\frac{\alpha+\beta}{\alpha+\beta+n}$ sum to one. This equation demonstrates clearly that, for Bernoulli distribution the expectation of the posterior distribution of λ is the weighted sum of the mean value of the observed data and the expectation of its prior. This observation motivates us to transfer knowledge from public data set to the prior of each teacher. Now we give the details of our strategy for knowledge transfer of Bayesian models.

1) KNOWLEDGE TRANSFER FOR BAYESIAN MODELS

From the analysis of Bernoulli distribution, we see that the source of the information of the parameters' posterior comes from both the training data and the prior we choose. So it is natural to choose a more informative prior instead of the usual manner in which we choose the Gaussian distribution or Laplacian distribution with zero mean and fixed variance as the prior. In this paper, we choose the prior that contains the information obtained from the public dataset of a similar task. A carefully selected informative prior can provide a good starting point for the training of the teacher model.

To make it clear, we call the model trained on publicly available source dataset as source model and the teacher

model as target model. Note that the source model and the target model should be in the same form, so that both models have the same number of parameters and it is possible to use the parameters of the trained source model as a starting point for the training of the target model. This requires that the domains of the source dataset and the target dataset should have a same feature space or can be transformed to share a same feature space.

Now we will introduce the details of knowledge transfer method. The first step is to fit the source model on the public dataset to get a posterior. If the source model's parameters have a conjugate prior, then we can derive its posterior distribution in closed form and use it as the prior for the target model's parameters directly. If the posterior distribution of the source model's parameters is intractable to derive in closed form, we can use deterministic approximation method such as Laplace approximation or variational inference to approximate the posterior distribution with a restricted family of distributions. Then this approximated distribution can be used as the prior distribution of the target model's parameters.

In the experiments part of the paper, we will use a model that has conjugate prior so that its posterior is easier to derive in closed from.

2) SHARING PRIOR BETWEEN TEACHERS

In the case of the deep neural networks, we share the transferred lower layers between all teacher models. For Bayesian models, we share the transferred knowledge by sharing the prior constructed from the public dataset between all teacher models. Given the informative prior constructed from the public non-private dataset, we train each teacher with the common prior and different subsets of private training data. After that each teacher will have different posterior distributions over their parameters. When the teacher ensembles are used to vote for the query of the student model, each teacher will predict a conditional distribution of the classes given the input. Then the class with the largest conditional probability is chosen as its vote. The votes accumulated from all teachers will be perturbed by noises sampled from a certain distribution to satisfy differential privacy. Then the most voted class will be chosen as the non-sensitive label of the student's query.

We have provided two concrete examples of TrPATE on deep neural networks and Bayesian models. It does not mean that TrPATE can only be applied to these two kinds of models. Similar to PATE, TrPATE is also a general framework. Existing knowledge transfer methods for different kinds of models have been explored extensively for decades. For example, [55] proposes a method for transferring knowledge in decision trees, [56] studies the knowledge transfer for support vector machines, and [57] proposes an approach for transfer learning in sequential data. Given a certain model and a corresponding knowledge transfer method, we can design TrPATE for it without much effort by transferring and sharing knowledge between teachers based on the existing knowledge transfer method. Due to the limited space, we won't talk about these models.

TABLE 1. Details of the reallocation of CIFAR-10, STL-10 and movie review as well as their source datasets for knowledge transfer.

Dataset	#Private	#Public	#Test	Source Dataset
CIFAR-10	40000	10000	10000	ImageNet
STL-10	10000	1000	2000	ImageNet
Movie Review	8000	1000	1662	SST

V. EXPERIMENTS

In this section, we will verify empirically the effectiveness of TrPATE with the models designed in Section IV and the quantitative improvements compared with the original PATE framework. We conduct various experiments on three datasets: CIFAR-10 [58], STL-10 [59] and Movie Review (MR) [60]. The first dataset in our experiments is CIFAR-10 which consists of 60000 images (50000 training images and 10000 test images) in 10 classes with $32 \times 32 \times 3$ dimensions per image. In our experiment we reallocate the images into three part: private dataset, public dataset and test dataset. The second dataset is STL-10, which is more challenging than CIFAR-10, in that it contains much fewer labeled images and each image is of higher dimensions. STL-10 contains only 13000 labeled images in 10 classes, with $96 \times 96 \times 3$ dimensions per image. There are only 5000 images for training and 8000 images for testing. Similarly, we reallocate the labeled images of STL-10 into three parts: private dataset, public dataset and test dataset. The last dataset is MR, which is a text dataset and contains only 5331 positive and 5331 negative movie reviews. The details of the partition of the datasets and their corresponding source datasets for knowledge transfer in the following experiments are presented in Table 1.

A. EXPERIMENTS ON CIFAR-10 AND STL-10

1) CIFAR-10 DATASET

In this experiment, we use 40000 samples as private dataset, 10000 samples as unlabeled public dataset and 10000 samples as test dataset. The source dataset used for training lower layers of teacher ensembles is ImageNet. In this experiment, the feature spaces of the source domain and the target domain are the same since pictures can be resized into same size, and the marginal distributions are related because both CIFAR-10 and ImageNet are nature images of certain objects. Besides, the tasks of source dataset and the target dataset are related, because both of them are classification tasks and the label space of target dataset is a subset of that of the source dataset.

To make a comparison, our non-private baseline model is made up of two parts. The first part is a ResNet50 model without the top layer, and the second part is a two layers neural networks with a 256 unit hidden layer and a 10 unit output layer. Before feeding the images to this model, we resize the images from $32 \times 32 \times 3$ dimensions to $197 \times 197 \times 3$ dimensions to meet the input size of ResNet50.

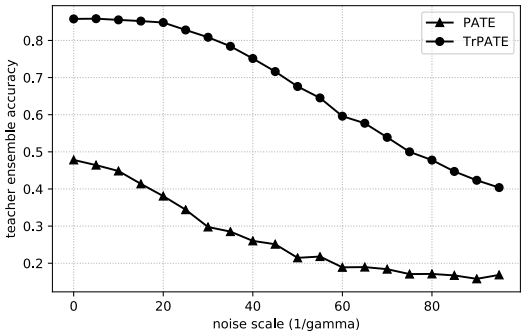


FIGURE 7. Comparison of the accuracy of the noisy labels predicted by the teacher ensembles of PATE and TrPATE on CIFAR-10 with respect to different noise scale.

The baseline model is trained with all of the 50000 training instances in 50 epochs and the batch size is set to 32. The accuracy of this model is 92.34%. Then, we train the teachers and student model with the same architecture as the baseline model, and the lower layers of ResNet50 for this model are transferred from a ResNet50 model pre-trained in source dataset ImageNet. In training the teacher ensembles, we train 200 teachers on 40000 private instances, thus each teacher is trained on 200 instances. We set $\gamma = 0.05$ in the aggregation mechanism. Since the noisy labels predicted by the teacher ensemble will be used as the private ground truth for training the student model, its accuracy is essential for the performance of the student model.

To verify whether the enhancement of each teacher with the transferred knowledge can benefit the teacher ensembles, and to explore how the noise scale will influence the prediction accuracy of the teacher ensembles. We compare the accuracy of the labels predicted by the teacher ensembles of PATE and TrPATE with different scales of noise, and the results are reported in Fig.7. We can see from this figure that with the transferred knowledge from non-private public dataset, the accuracy of the labels predicted by TrPATE outperforms PATE consistently by a large margin under different noise scales. When the noise scale is 0, the performance of TrPATE outperform that of PATE by nearly 40 percent. When the noise scale increased the performance of both PATE and TrPATE decreased, however the superiority of TrPATE still remains. This implies that the enhancement of the teacher ensembles are robust to the injected noise.

In training the student model, we set $\delta = 10^{-5}$ and try different numbers of queries. Then we use Theorem 1 to calculate the privacy loss. Each experiment are executed 20 times and the average results are reported in Table 2. We can observe from Table 2 that, when querying 1000 times, the student could achieve an accuracy of 82.64%. Although this is still lower than the non-private baseline, this result is much better than that of PATE.

In this experiment we calculate the privacy guarantee with Theorem 1 which gives us $\epsilon = 25.2$ when querying 1000 times. This may be a little large for real applications,

TABLE 2. Accuracy and privacy guarantee of student model trained with different number of queries when $\gamma = 0.05$ and $\delta = 10^{-5}$. The target dataset is CIFAR-10 and the source dataset is ImageNet.

Queries	ϵ	Baseline	PATE	TrPATE
1000	25.2	92.34%	47.03%	82.64%
500	15.7	92.34%	46.31%	80.69%
200	8.8	92.34%	44.42%	76.27%
150	7.38	92.34%	44.29%	74.80%
100	5.8	92.34%	42.24%	71.37%

however if we use a more sophisticated data dependent method [21] or a better mechanism [24] we can achieve a much lower ϵ. Since computing a more tight privacy guarantee is orthogonal to this paper and is not the focus of this paper, due to the limited space, we will not expand it further.

2) STL-10 DATASET

STL-10 is a much smaller dataset, which contains only 13000 labeled images. We choose 10000 samples as private dataset, 1000 samples as unlabeled public dataset and 2000 as test dataset. In this experiment, we want to explore experimentally the impacts of the quantity of source dataset to the performance of TrPATE. We will choose ImageNet as the source dataset. To control the quantity of source dataset size, we create two additional datasets ImageNet0.1x and ImageNet0.5X, and we denote the original ImageNet dataset as ImageNet1x. The ImageNet0.1x is created by randomly sampling 10% of the original ImageNet dataset within each class without replacement. Similarly, the ImageNet0.5x is created by randomly sampling 50% of the ImageNet dataset within each class without replacement. In this experiment, the domains of source dataset and target dataset have same feature space and similar marginal distributions. And the tasks are also related since the label space STL-10 is a subset of the label space of ImageNet.

In training TrPATE we train ResNet50 models on ImageNet1x, ImageNet0.5x and ImageNet0.1x specifically and remove their top 1 layers, then we freeze the lower layers of these ResNet50 models and add two layers (one 256 units hidden layer and one 10 units output layer) on each of them. We train a model with the same architecture from scratch with all of the 11000 training data (private + public) and get an accuracy of 71.8% which will be used as the non-private baseline for this experiment.

In training the teacher ensembles, we train 200 teachers with 10000 private instances, therefore each teacher is trained on 50 instances. We set $\gamma = 0.05$ in the aggregation mechanism. The student accuracy and the privacy guarantee are reported in Table 3.

We can observe from Table 3 that with the knowledge transferred from the source dataset TrPATE outperforms PATE in all of the experiments. When the quantity of source dataset increases, the improvements of TrPATE increases consistently. We can also find that a tradeoff can be made

TABLE 3. Comparison of the accuracy and privacy guarantee of student model trained on STL-10 with different number of queries and different source datasets quantities when $\gamma = 0.05$ and $\delta = 10^{-5}$.

Source Dataset	Queries	ϵ	Baseline	PATE	TrPATE
ImageNet1x	1000	25.2	71.8%	53.67%	92.45%
	500	15.7	71.8%	51.79%	91.70%
	200	8.8	71.8%	48.34%	89.46%
	100	5.8	71.8%	44.26%	85.71%
ImageNet0.5x	1000	25.2	71.8%	53.67%	89.49%
	500	15.7	71.8%	51.79%	88.17%
	200	8.8	71.8%	48.34%	86.82%
	100	5.8	71.8%	44.26%	83.30%
ImageNet0.1x	1000	25.2	71.8%	53.67%	80.22%
	500	15.7	71.8%	51.79%	79.84%
	200	8.8	71.8%	48.34%	79.13%
	100	5.8	71.8%	44.26%	77.56%

between the privacy cost ϵ and the quantity of source dataset. For example, TrPATE trained on ImageNet1x achieved a performance of 89.46% with privacy cost $\epsilon = 8.8$, while a similar performance achieved by TrPATE trained on ImageNet0.5x requires privacy cost $\epsilon = 25.2$. Note that when the source dataset is ImageNet and the query is more than 500 times, the accuracy of TrPATE reaches up to 91.7%. To the best of our knowledge, this is the first time for such a complex dataset to achieve this accuracy in a differentially private manner.

B. EXPERIMENTS ON MR

In this subsection, we want to explore whether TrPATE also works on text data, and we want to test whether the designed TrPATE for Bayesian models can also outperform PATE. We use the Movie Reviews (MR) to perform sentiment analysis. MR has 10662 movie reviews with 5331 positive reviews and 5331 negative reviews, and its average sentence length is about 20 words. The task is to classify new reviews into positive class (represents with 1) or negative class (represents with 0). The source dataset that we use to transfer knowledge from is Stanford Sentiment Treebank (SST) [61], which contains 11855 reviews with 8544 training instances, 1101 validation instances and 2210 test instances. It has 5 classes ranging from very negative to very positive, and its average sentence length is about 18 words. In this subsection, we will conduct two experiments on the dataset. In the first experiment we use deep neural networks, and in the second experiment we choose naive Bayes as our model. Note that the source dataset task is a 5 way classification which is correlated to the binary positive or negative classification of the target dataset task. To make the label spaces of the source dataset and the target dataset to be the same, we process source dataset SST in the following way, we assign the label of very negative and negative classes with 0, the label of very positive and positive with 1, and discard all the data in neutral class. After that the label spaces of the two tasks are the same. In both of the experiments, we choose 5000 words with the highest frequency of occurrence in SST as the vocabulary, and treat all other words as out-of-vocabulary (OOV) words.

TABLE 4. Comparison of the accuracy and privacy guarantee of student model trained on MR with different γ and query times. The model used for this experiment is convolutional neural networks.

Privacy Parameters	Queries	ϵ	Baseline	PATE	TrPATE
$\gamma = 0.05$ $\delta = 10^{-5}$	1000	25.2	75.85%	51.60%	79.34%
	500	15.7	75.85%	51.25%	78.99%
	200	8.8	75.85%	50.10%	71.45%
	100	5.8	75.85%	49.73%	64.32%
$\gamma = 0.1$ $\delta = 10^{-5}$	1000	70.3	75.85%	51.04%	78.97%
	500	41.5	75.85%	50.69%	78.69%
	200	21.6	75.85%	50.46%	73.63%
	100	13.6	75.85%	50.23%	67.81%

In this way the features of the source domain and the target domain are embedded into a same space. Note that, there are about 4200 words of MR's vocabulary appear in the top 5000 frequency words of SST. Therefore the domains of the two dataset are highly related.

1) DEEP NEURAL NETWORKS MODEL

In this experiment, we use deep neural networks as our model for the sentiment classification task. The network is constructed with one embedding layer, one 3×3 convolution layer, one global pooling layer and two fully connected layers. In this experiment, we train the embedding layer and the 3×3 convolution layer on the public dataset SST, and train the last two fully connected layers on the private dataset MR. We choose 5000 words with the highest frequency of occurrence in SST as our vocabulary and set the word embedding dimensions to 50. The convolutional layer has 250 filters and the first fully connected layer has 250 dimensions.

Since the size of the dataset is so small, we use only 8000 instances as the training set, 1000 instances as the public unlabeled set and the last 1662 instances as the test set. We train 80 teachers and each teacher is trained on only 100 instances. The comparison is reported in Table 4. We also report the baseline of a non-private model trained on all of the training data with the same architecture. We can observe from the table that MR dataset is too small for PATE to learn a useful model since its performance is just slightly better than random selection. While TrPATE achieve a much higher accuracy even with such limited dataset under the help of the transferred low layers pre-trained on a much larger source dataset SST. We can also find that TrPATE outperforms the baseline again when we query more than 500 times.

2) NAIVE BAYES MODEL

In this experiment, we use naive Bayes as the model to classify reviews' type. In naive Bayes, we assume that the features are independent, and different categories are modeled independently. We use two Categorical distribution individually over 5000 vocabularies to model the distribution of positive and negative reviews, and use the Dirichlet distribution as its conjugate prior. Suppose that for each category the vocabulary size is n and we denote it by $\{w_1, w_2, \ldots, w_n\}$. The probability of observing word w_i is λ_i, where $\lambda_i \in [0, 1]$ and

$\sum_{i=1}^{n} \lambda_i = 1$. We denote by x_i the i-th word of a sentence and $x_i = (a_{i1}, a_{i2}, \ldots, a_{in})$, where $a_{ij} = 1$ if $x_i = w_j$. Otherwise $a_{ij} = 0$. So the probability of observing x_i is denoted by

$$p(x_i) = \text{Cat}(x_i|\lambda_1, \lambda_2, \ldots, \lambda_n) = \prod_{j=1}^{n} \lambda_j^{a_{ij}}.$$

Due to the independence assumption of each word, the probability of observing a sentence $S = x_1\, x_2 \ldots x_m$ is

$$p(S) = \prod_{i=1}^{m} \text{Cat}(x_i|\lambda_1, \lambda_2, \ldots, \lambda_n) = \prod_{j=1}^{n} \lambda_j^{c_j},$$

where c_j is the number of w_j appears in S.

The prior distribution of $\lambda_1, \lambda_2, \ldots, \lambda_n$ is Dirichlet distribution which is parameterized by n positive parameters $\alpha_1, \alpha_2, \ldots, \alpha_n$. We denote the prior distribution as follows

$$p(\lambda_1, \lambda_2, \ldots, \lambda_n) = \frac{1}{\text{B}(\alpha_1, \alpha_2, \ldots, \alpha_n)} \prod_{j=1}^{n} \lambda_j^{\alpha_j - 1},$$

where $\text{B}(\alpha_1, \alpha_2, \ldots, \alpha_n)$ is the normalization constant and it is defined as $\prod_{i=1}^{n} \Gamma(\alpha_i) / \Gamma(\sum_{i=1}^{n} \alpha_i)$. For short, we denote it by

$$p(\lambda_1, \lambda_2, \ldots, \lambda_n) = \text{Dir}(\lambda_1, \lambda_2, \ldots, \lambda_n|\alpha_1, \alpha_2, \ldots, \alpha_n).$$

The mean of a Dirichlet distribution can be derived as follows

$$\text{E}(\lambda_i) = \frac{\alpha_i}{\sum_{j=1}^{n} \alpha_j}, \text{for} i \in [1, n].$$

According to Bayes' theorem, we write the posterior distribution of $\lambda_1, \lambda_2, \ldots, \lambda_n$ after observing a sentence S as

$$\begin{aligned} &p(\lambda_1, \lambda_2, \ldots, \lambda_n|S) \\ &= \frac{\text{Cat}(S|\lambda_1, \lambda_2, \ldots, \lambda_n)\text{Dir}(\lambda_1, \lambda_2, \ldots, \lambda_n)}{p(S)} \\ &= \text{Dir}(\lambda_1, \lambda_2, \ldots, \lambda_n|c_1 + \alpha_1, c_2 + \alpha_2, \ldots, c_n + \alpha_n), \end{aligned}$$

where c_i is the number of w_i appears in S.

In training PATE, we choose the non-informative prior in which $\alpha_i = 1, i \in [1, n]$. This prior is also named as Laplacian smoothing in some contexts. In TrPATE, instead of using the non-informative prior, we use the posterior distribution of $\lambda_1, \lambda_2, \ldots, \lambda_n$ estimated on public dataset SST as the prior distribution for each teacher model. The experiment results are reported in Table 5. We can find that on MR dataset, PATE performs only a slightly better than random guess. This is because the training set for for each teacher is too small. It is too hard to train a naive Bayes classifier of two classes and 5000 features using just about 2000 words from 100 reviews. We can see that TrPATE when transferred with an informative prior extracted from the public dataset SST outperforms PATE with a large margin. This comparison demonstrates the effectiveness of knowledge transfer and sharing in TrPATE when the training set is too small with respect to the corresponding task.

TABLE 5. Comparison of the accuracy and privacy guarantee of student model trained on MR with different γ and query times. The model used for this experiment is naive Bayes.

Privacy Parameters	Queries	ϵ	Baseline	PATE	TrPATE
$\gamma = 0.05$ $\delta = 10^{-5}$	1000	25.2	75.21%	53.28%	73.35%
	500	15.7	75.21%	52.22%	73.81%
	200	8.8	75.21%	52.53%	73.71%
	100	5.8	75.21%	52.70%	73.73%
$\gamma = 0.1$ $\delta = 10^{-5}$	1000	70.4	75.21%	53.57%	73.16%
	500	41.5	75.21%	52.58%	73.76%
	200	21.6	75.21%	52.22%	73.72%
	100	13.6	75.21%	52.25%	73.75%

VI. CONCLUSION

In this paper, we have demonstrated that the disjoint training sets for teacher ensembles of PATE will cause the performance degradation problem on complex datasets and tasks due to the limited training set for each teacher. To alleviate this problem, we proposed TrPATE which uses transfer learning to extract useful knowledge from a related publicly available non-private dataset, and trains all teachers with both the private data and the transferred knowledge. Specifically, we applied TrPATE to deep neural networks and Bayesian models, and conducted extensive experiments on different datasets with these models. The effectiveness of our proposed framework has been demonstrated empirically.

ACKNOWLEDGMENT

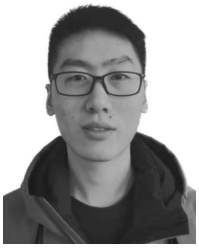
The authors would like to thank the reviewers for their invaluable suggestions and comments.

REFERENCES

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl.-Based Syst.*, vol. 46, pp. 109–132, Jul. 2013.
[2] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.
[3] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1701–1708.
[4] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, "Disease prediction by machine learning over big data from healthcare communities," *IEEE Access*, vol. 5, pp. 8869–8879, 2017.
[5] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1322–1333.
[6] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.
[7] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," in *Proc. 22nd Int. Conf. Data Eng. (ICDE)*, Apr. 2006, p. 24.
[8] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *Proc. 23nd Int. Conf. Data Eng. (ICDE)*, Apr. 2007, pp. 106–115.
[9] C. Dwork, "Differential privacy," in *Proc. 33nd Int. Colloq. Autom. Lang. Program.*, 2006, pp. 1–12.
[10] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proc. 48th Ann. IEEE Symp. Found. Comput. Sci. (FOCS)*, 2007, pp. 94–103.
[11] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to noninteractive database privacy," *J. ACM*, vol. 60, no. 2, pp. 12:1–12:25, Apr. 2013.
[12] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *Proc. 51th Ann. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2010, pp. 51–60.

[13] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2008, pp. 289–296.
[14] K. Chaudhuri, A. Sarwate, and K. Sinha, "Near-optimal differentially private principal components," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 998–1006.
[15] J. Hua, C. Xia, and S. Zhong, "Differentially private matrix factorization," in *Proc. 24nd Int. Joint Conf. Artif. Intell. (IJCAI)*, 2015, pp. 1763–1770.
[16] T. Chanyaswad, C. Liu, and P. Mittal, "RON-Gauss: Enhancing utility in non-interactive private data release," *Proc. Privacy Enhancing Technol.*, vol. 2019, no. 1, pp. 26–46, 2018.
[17] C. Xu, J. Ren, Y. Zhang, Z. Qin, and K. Ren, "DPPro: Differentially private high-dimensional data release via random projection," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 12, pp. 3081–3093, Dec. 2017.
[18] X. Ren *et al.*, "Lopub: High-dimensional crowdsourced data publication with local differential privacy," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 9, pp. 2151–2166, Sep. 2018.
[19] J. Zhang, X. Xiao, Y. Yang, Z. Zhang, and M. Winslet, "PrivGene: Differentially private model fitting using genetic algorithms," in *Proc. ACM SIGMOD Int. Conf. Managem. Data*, 2013, pp. 665–676.
[20] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.
[21] N. Papernot, M. Abadi, U. Erlingsson, "Semi-supervised knowledge transfer for deep learning from private training data," in *Proc. Int. Conf. Learn. Repres. (ICLR)*, 2017.
[22] M. Abadi *et al.* (2017). "On the protection of private information in machine learning systems: Two recent approaches." [Online]. Available: https://arxiv.org/abs/1708.08022
[23] J. Li *et al.*, "Enforcing differential privacy for shared collaborative filtering," *IEEE Access*, vol. 5, pp. 35–49, 2017.
[24] N. Papernot *et al.*, "Scalable private learning with PATE," in *Proc. Int. Conf. Learn. Repres. (ICLR)*, 2018.
[25] S. Thrun, "Is learning the n-th thing any easier than learning the first?" in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 1995, pp. 640–646.
[26] Y. Zhu *et al.*, "Heterogeneous transfer learning for image classification," in *Proc. 25th AAAI Conf. Artif. Intell. (AAAI)*, 2011, pp. 1304–1309.
[27] N. T. Vu, F. Metze, and T. Schultz, "Multilingual bottle-neck features and its application for under-resourced languages," in *Proc. 3th Workshop Spok. Lang. Technol. Under-Resourced Lang. (SLTU)*, 2012, pp. 90–93.
[28] F. Grèzl, M. Karafiát, and K. Veselý, "Adaptation of multilingual stacked bottle-neck neural network structure for new language," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 7654–7658.
[29] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey," *Speech Commun.*, vol. 56, pp. 85–100, Jan. 2014.
[30] H. Chen, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Multilingual bottle-neck feature learning from untranscribed speech," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2017, pp. 727–733.
[31] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2016, pp. 1568–1575.
[32] T. Q. Nguyen and D. Chiang, "Transfer learning across low-resource, related languages for neural machine translation," in *Proc. 8th Int. Joint Conf. Natural Lang. Process. (IJCNLP)*, vol. 2, 2017, pp. 296–301.
[33] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Proc. 25th Ann. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2006, pp. 486–503.
[34] I. Mironov, "Rényi differential privacy," in *Proc. IEEE 30th Comput. Secur. Found. Symp. (CSF)*, Aug. 2017, pp. 263–275.
[35] K. Nissim, S. Raskhodnikova, and A. D. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proc. 39th Ann. ACM Symp. Theory Comput. (STOC)*, 2007, pp. 75–84.
[36] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[37] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," in *J. Big Data*, vol. 3, p. 9, May 2016.
[38] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis, Pattern Recognit. (CVPR)*, Jun. 2014, pp. 806–813.
[39] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, 2016.
[40] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis, Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7167–7176.
[41] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 97–105.
[42] M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich, "To transfer or not to transfer," in *Proc. NIPS Workshop Transf. Learn.*, vol. 898, 2005, pp. 1–4.
[43] W. Dai, Q. Yang, G. R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 193–200.
[44] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1855–1862.
[45] B. Cao, S. J. Pan, Y. Zhang, D.-Y. Yeung, and Q. Yang, "Adaptive transfer learning," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 407–412.
[46] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *CoRR*, vol. abs/1412.3474, Dec. 2014. [Online]. Available: http://arxiv.org/abs/1412.3474
[47] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 2208–2217.
[48] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 443–450.
[49] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 3712–3722.
[50] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," *CoRR*, vol. abs/1309.4168, Sep. 2013. [Online]. Available: http://arxiv.org/abs/1309.4168
[51] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 3320–3328.
[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[53] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[54] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.
[55] J. W. Lee and C. G. Giraud-Carrier, "Transfer learning in decision trees," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2007, pp. 726–731.
[56] V. H. Ablavsky, C. J. Becker, and P. Fua, "Transfer learning by sharing support vectors," School Comput. Commun. Sci., Swiss Federal Inst. Technol., EPFL, Lausanne, Switzerland, Tech. Rep. 181360, 2012.
[57] S. Sun, H. Liu, J. Meng, C. L. P. Chen, and Y. Yang, "Substructural regularization with data-sensitive granularity for sequence transfer learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2545–2557, Jun. 2018.
[58] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.
[59] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Int. Stat. (AISTATS)*, 2011, pp. 215–223.
[60] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proc. 56th Ann. Meeting Assoc. Comput. Linguistics (ACL)*, 2005, pp. 115–124.
[61] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2013, pp. 1631–1642.

LULU WANG received the B.S. degree in computer science from Jilin University, Changchun, China, in 2016. He is currently pursuing the M.S. degree with the Department of Computer Science and Technology, Peking University, China. His research interests include machine learning, big data analytics, and data privacy.



JUNXIANG ZHENG received the B.S. degree from EECS, Peking University, Beijing, China, in 2017, where he is currently pursuing the M.S. degree. His research interest includes security and privacy in machine learning.

YONGZHI CAO (SM'15) received the B.S. and M.S. degrees from Central China Normal University, Wuhan, China, in 1997 and 2000, respectively, and the Ph.D. degree from Beijing Normal University, Beijing, China, in 2003, all in mathematics.

From 2003 to 2007, he was a Postdoctoral Researcher with Tsinghua University, Beijing, and from 2007 to 2015, he was an Associate Professor of computer science with Peking University, Beijing, where he is currently a Professor of computer science. He has published some papers in academic journals, such as the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, *Theoretical Computer Science*, *Information and Computation*, and the *Journal of Computer and System Sciences*. His current research interests include formal methods, reasoning about uncertainty in artificial intelligence, security, and privacy.

HANPIN WANG received the B.Sc. degree from Anhui Normal University, in 1985, and the Ph.D. degree from Beijing Normal University, in 1993. He is currently a Professor of computer science with the School of EECS, Peking University. He has been the Vice Director of the Institute of Software and the Head of the Laboratory of Theoretical Computer Science, since 2003. His research interests include formal semantics and verification of computer systems, algorithms, and computational complexity.