Tensor Recurrent Neural Network with Differential Privacy

Jun Feng, Laurence T. Yang, *Fellow, IEEE*, Bocheng Ren, Deqing Zou, Mianxiong Dong, and Shunli Zhang

Abstract—Recurrent neural network (RNN), a branch of deep learning, is a powerful model for sequential data that has outstanding performance on a wide range of important Internet of Things (IoT) tasks. This unprecedented growth of RNN model has however encountered both heterogeneous IoT data and privacy issues. Existing RNN model can not deal with heterogeneous sequential data; often the larger datasets used in training of RNN model contain sensitive information. To tackle these challenges and for the first time, this research proposes a novel differentially private tensor-based RNN (DPTRNN) that can be applied in many challenging deep learning sequence tasks for IoT systems. Specifically, to process heterogeneous sequential data, we propose a tensor-based RNN model. To guarantee privacy, we develop a tensor-based back-propagation through time algorithm with perturbation to avoid exposing the sensitive information for training the tensor-based RNN model within the framework of differential privacy. Thorough security analysis shows that the differential private tensor-based RNN efficiently protects the confidentiality of sensitive user information for IoT. Our results from extensive experiments on two challenging large video datasets suggest that our proposed scheme is practical with guarantee of data privacy preservation and acceptable accuracy loss.

Index Terms—Recurrent neural networks, differential privacy, tensor model, deep learning, privacy protection, Internet-of-Things, artificial intelligence.

1 Introduction

The Internet of Things (IoT) integrates many things with computations and communications that can be powered by Artificial neural network (ANN) [1]. ANN has gained much attention in both industry and academia and has witnessed a flux in its usage in different IoT application domains [2]. Recurrent neural networks (RNN) are ANN with recurrent connections (feedback loops) [3]. Currently, machine learning schemes based on RNN are showing remarkable results in many challenging sequence modeling tasks in IoT systems [4, 5]. RNN model is increasingly gaining popularity as a result of the improved performance in results obtained in different challenging sequence problems. RNN is undoubtedly the tool of choice that has been used in a wide range of natural language processing (NLP) applications, e.g., language translation, speech recognition, input decoding etc. Although RNN was initially designed for usage in speech recognition [6, 7], it is increasingly being applied to other application domains like social media, text summarization etc. Specific use cases where RNN has been successful include language translation where the input of the RNN is the source language (e.g. Chinese) and the output is the target language (e.g. English), text summarization where given a very large text as input, the RNN gives as output a condensed summary of the whole text that explains the whole original text [8].

It is very meaningful and challenging to process high-order big data in IoT systems. The existing RNN models are hard to process high-order big data [9]. Tensor is a promising representation model for high-order data processing [10, 11]. Tensor relations occur naturally in our everyday life due to the complex nature of data. These relations takes different forms depending on the number of entities involved. Relations can be modeled as one-way tensors (vectors), two-way tensors (two-dimensional array or matrix), and three or more way tensors generally referred to as high order tensors. The tensor-based schemes have shown significant effectiveness in many artificial intelligence applications, such as deep computation model, deep convolutional computation model [10, 12]. Therefore, it is possible to propose a tensor-based RNN (TRNN) model for processing high-order big data in IoT systems. However, neural network for high-order data is a challenging task and the limited number of researches in this area is an attestation of this fact.

This increased usage of RNN in different application domains has however encountered obvious trust issues, the topnotch concerns in IoT systems [13]. Trust-oriented IoT systems should involve ingredients contributing to data privacy protection. Only a few scientific researches on how to protect the privacy of data on RNN however exist. Data privacy issues are the biggest hurdle parts of IoT systems,

- *J. Feng, and D. Zou are with the School of Cyber Science and Engineering, Huazhong University of Science and Technology, and Hubei Engineering Research Center on Big Data Security, Wuhan* 430074*, China. E-mails: junfeng989@gmail.com, deqingzou@hust.edu.cn.*
- *L. T. Yang is with the School of Cyber Science and Engineering, Huazhong University of Science and Technology, and Hubei Engineering Research Center on Big Data Security, Wuhan* 430074*, China, and the Department of Computer Science, St. Francis Xavier University, Antigonish, Canada. E-mail: ltyang@ieee.org.*
- *Z. Ren, and S. Zhang are with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan* 430074 *China.*
- *M. Dong is with the Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. E-mail: mxdong@mmm.muroran-it.ac.jp.*

and have prevented the RNN-based applications in many IoT tasks that involve personal or sensitive information such as photo, shopping history, private financial data, and clinical records of patients [14, 15].

The last few decades has witnessed a growing concern on how best to release sensitive information from statistical analysis, data mining, and artificial intelligence while still protecting its privacy in IoT systems. One state-of-the-art mechanism that has been deployed widely in IoT systems is differential privacy [16]. This mechanism guarantees stronger privacy protection of a specific record such that an adversary having all the other records in the dataset except the record under consideration can not be able to infer any information about this record. The privacy guarantee is controlled by a privacy budget, a measure that controls the amount by which distributions of output induced by two adjacent datasets differs. Privacy budget is inversely proportional to the privacy guarantee i.e. a smaller privacy budget gives stronger privacy guarantees. A lot of study has been done in privacy-preservation using differential privacy both in theory and practice [17]. There are some main ways of achieving differential privacy: Injecting Laplace noise [16], Gaussian noise mechanism, exponential mechanism [18] and functional perturbation mechanism [19]. Neural network models can memorize sensitive, or private data records used in their training. By using membership inference attacks and model inversion attacks, adversaries can infer the training data. The publishing of trained recurrent neural network models may disclose the training data in practice. Therefore, it is of great importance to propose differentially-private recurrent neural network schemes to protect sensitive, or private training data against the adversaries who have full information of neural network model parameters.

Considering the strengths of RNN, tensor model and differential privacy, combining the technologies is expected to yield better results both in performance and privacy protection. To the best of our knowledge, this is the first work to combine RNN, tensor and differential privacy. It is therefore deemed paramount to develop a data privacy preservation and heterogeneous RNN scheme that satisfies two goals. First of all, it should be able to be applied across a number of heterogeneous recurrent neural network application domains. Secondly, it should not reveal private information in the datasets used during the training of heterogeneous RNN models. A scheme that satisfies these two conditions definitely can dramatically enhance the usage of RNN while still ensuring data privacy protection.

Based on the above goals, we proposed a novel differentially private tensor-based recurrent neural network (DPTRNN) scheme, which combines the current most advanced artificial intelligent algorithm with the current most advanced privacy protection techniques, for heterogeneous data processing and privacy protection in IoT systems. The main contributions of this paper include the following three aspects from the deep learning and the differential privacy perspectives.

- A tensor-based RNN model is innovatively designed to deal with high order data by extending RNN model to tensor space. The tensor-based RNN model includes input tensor layer, hidden tensor layer and output tensor layer.
- A new tensor-based back-propagation through time (TBPTT) algorithm with perturbation is proposed to achieve privacy protection for tensor-based RNN. The main idea of the proposed scheme is that we use functional mechanism to enforce ϵ-differential privacy by perturbing the loss function of the tensor-based RNN model rather than its results. People can make use of the proposal to build intelligent video surveillance systems in a privacy-preserving way. As far as we know, this is the first attempt to apply differential privacy to the task of tensor-based deep computation. The proposed scheme can prevent adversaries who have strong background knowledge to achieve privacy protection for tensor RNN model.
- Extensive empirical evaluations were performed on two challenging well-known datasets: The KTH dataset and the UCF50 dataset. Experimental results demonstrate that our scheme is competitive even with a non-privacy-enforcing tensor-based RNN model and introduces acceptable side effects on the precision.

The reminder of the paper is organized as follows: We discuss the preliminaries in Section 2, followed by detailed discussion of the construction of DPTRNN scheme in Section 3. Section 4 describes the experiments conducted including performance evaluations. Section 5 presents related works and finally we have conclusion and future work in Section 6.

2 PRELIMINARIES

In this section, we describe various important technologies used in our work, e.g, RNN, differential privacy, and Laplace mechanism. Some key notations used in this paper are introduced in Table 1.

TABLE 1
Table of symbols.

Symbol	Definition
$\mathcal{X}$	input tensor
$t_{k_1 k_2 \cdots k_N}$	one element of tensor $T \in R^{K_1 \times K_2 \times \cdots \times K_N}$
$\mathcal{H}$	hidden tensor
$\mathcal{O}$	output tensor
$\mathcal{U}$	input-to-hidden weight tensor
$\mathcal{W}$	hidden-to-hidden weight tensor
$\mathcal{V}$	hidden-to-output weight tensor
η, ξ, λ	learning rate
$\odot$	tensor multi-dot product
ϵ	privacy budget

2.1 Recurrent Neural Network

A recurrent neural network (RNN) is a deep learning model that has achieved far greater success in modeling time series [20]. The input layer can take T inputs in form of a variable-length vector sequences through time [21]. Suppose

$x = (x_1, x_2, \cdots, x_T)$ is an input vector, the RNN updates its hidden state h_t by

$$h_t = \begin{cases} 0, & t = 0 \\ (h_{t-1}, x_t), & \text{otherwise} \end{cases}, \quad (1)$$

where ϕ is a non-linear activation function (which can be as simple as an element-wise sigmoid function or as complex as an LSTM) and h_t is an element in the hidden vector sequence $h = (h_1, h_2, \cdots, h_T)$. The RNN can optionally have an output vector sequence $y = (y_1, y_2, \cdots, y_T)$ of variable length. The RNN updates both its hidden and output layers by using equations 2 and 3 below in each iteration stating from $t = 1$ to T,

$$h_t = (W_{xh}x_t + W_{hh}h_{t-1} + b_h), \quad (2)$$

$$y_t = W_{hy}h_t + b_y. \quad (3)$$

A RNN can be linear network or non-linear network. Non-linear networks are very powerful compared to their linear counterparts. However, due to the non-linearity, it is difficult to determine the output given an input. This is one of the major reason why we must learn the correlations between a given input and output. There are a number of activation functions used in RNN, the three major ones being sigmoid, tanh and ReLU. Loss function is a function that assesses the performance of a model by comparing the output of the model y_t to a corresponding target z_t. The loss function is thus defined as

$$L(y, z) = \sum_{t=1}^{T} L_t(y_t, z_t), \quad (4)$$

which indicates summation of all the losses in every timestep. The choice of the loss function depends on the issue to be solved. Popular loss functions used include the Euclidean distance for real-values forecasting and cross-entropy for classification problems over probability distributions of outputs.

2.2 Differential Privacy

In this subsection, we introduce the concept of differential privacy, functional mechanism and finally take a deep look at how to achieve the functional mechanism [22, 23].

Supposing D is a database that contains m records $r_1, r_2, \cdots, r_m$ and $p + 1$ attributes $R_1, R_2, \cdots, R_p, Y$ then each record r_i can be expressed as $r_i = r_{i1}, r_{i2}, \cdots, r_{ip}, r_y$. Without loss of generality, we also assume that $\sqrt{\sum_{j=1}^{p} r_{ij}^2} \leq 1$ where $r_{ij} \geq 0$. We can enforce this this assumption by changing each r_{ij} by $\frac{r_{ij} - \alpha_j}{(\beta_j - \alpha_j) \cdot \sqrt{p}}$ where β_j and α_j are the maximum and minimum values of the domain R_j.

For purposes of formulating a formal definition of ϵ-differential privacy, we let : $\mathcal{D}^n \rightarrow \mathcal{Y}$ be a randomized algorithm and $\mathcal{D}_1, \mathcal{D}_2 \in \mathcal{D}^n$ be two neighbouring databases that differ with only one record (row).

Definition 1. *Let $\epsilon > 0$. X is ϵ-differentially private if for all neighbouring databases and for all subsets of $Y \in \mathcal{Y}$, we have*

$$\frac{\Pr[(\mathcal{D}_1) \in \mathcal{Y}]}{\Pr[(\mathcal{D}_2) \in \mathcal{Y}]} \leq \exp(\epsilon), \quad (5)$$

where the probability space is over the coin flips of the mechanism ϕ.

Numeric queries and functions of the form $f : D \rightarrow R^k$ by far represents the most common queries in a majority of the databases. The function f maps databases to k real numbers. The l_1-sensivity of these queries is a very important parameter as it determines how accurately we can answer these queries.

Definition 2. *(l_1-sensivity). The l_1-sensitivity of a function $f : D \rightarrow R^k$ is:*

$$\Delta f = \max_{\substack{x, y \in D \\ \|x - y\|_1 = 1}} \|f(x) - f(y)\|_1. \quad (6)$$

This l_1-sensivity of any given function f acquires the magnitude by which one individual data could vary f in the worst case.

The Laplace distribution (centered at 0) with scale b is the distribution with the following probability density function:

$$Lap(x|b) = \frac{1}{2b} \exp(-\frac{|x|}{b}). \quad (7)$$

The Laplace distribution has a variance of $\sigma^2 = 2b^2$. We often denote this distribution by $Lap(b)$ where b is a scale of the Laplace distribution. This denotation basically represents a random variable (i.e. $X \sim Lap(b)$). This distribution is just like the exponential distribution except for the fact that it is symmetrical.

The Laplace mechanism, as its name suggest, simply computes f, and perturbs every coordinate with Laplace distribution noise. The scale of the noise will be calibrated to the sensitivity of f (divided by ϵ).

Definition 3. *(The Laplace Method). Given any function $f : D \rightarrow R^k$, the Laplace method is defined as:*

$$L(x, f(\cdot), \epsilon) = f(x) + (Y_1, \cdots, Y_k), \quad (8)$$

where $Y_i \sim Lap(\nabla f/\epsilon)$ iid (random variables taken from the distribution $Lap(\nabla f/\epsilon)$). The Laplace mechanism preserves ϵ-differential privacy.

Differential privacy exhibits a contrasting behaviour between the minimization of privacy loss and maximization of utility. Privacy loss is measured by parameter ϵ (the privacy budget) while utility can be measured by either L_1 or L_2 losses among other mechanisms [22]. The selection of parameter ϵ depends on several factors including how differentially private algorithms behave under compositions and the way the privacy loss accumulates over time during multiple analyses.

Functional mechanism is a general framework used in some machine learning tasks that preserves ϵ-differential privacy. By using this framework, perturbation (adding noise) is done on the objective function (optimization goal) of the machine learning task rather than the outputs. The exact noise you use can come from a number of distributions including Laplace distribution, Gaussian distribution etc. In our case, we make use of Laplace distribution for perturbing our objective function.

3 The Construction of Tensor-Based Recurrent Neural Network with Differential Privacy

This section details the main components of the proposed scheme toward differentially private training of tensor-based recurrent neural network: Tensor-based RNN, and tensor-based back-propagation through time algorithm with perturbation.

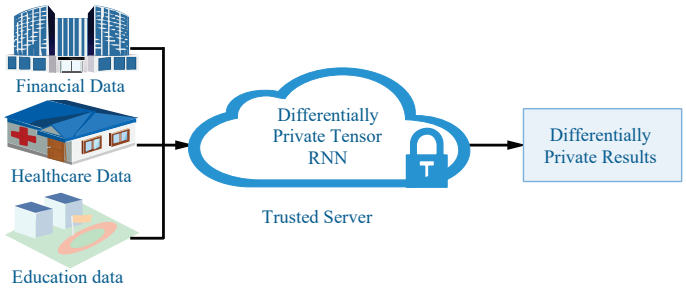


Fig. 1. Overview for Differentially Private Tensor RNN.

Our main goal is to design differentially private tensor-based RNN model. Our system involves three main entities: Data owner, trusted server, and data user. The following system model (See Fig. 1) is considered in this paper: A data owner has a mass of data involving business information, medical information, and education information. Suppose the data owner would like to publish the tensor-based RNN model based on the data for collaboration with a data user. The data owner sends the data to a trusted server. Because of data privacy concerns, the differentially private tensor-based RNN is computed and the differentially private results are released to the data user by the trusted server. Specifically, the following requirements should be met in the work.

- Privacy: The core concern in the proposed scheme is to protect data privacy. The released results (trained tensor-based RNN model) should satisfy differential privacy to prevent the membership inference attack of the data user.
- Utility: The tensor-based RNN should be correctly trained. The released results should retain as much accurate information as possible for subsequent analysis.

The type of model required by data analyzer must be known beforehand by the data owner. For instance, if the requirement of the data analyzer implies a model that uses patient status to estimate possible diseases, the matching model must be built by the data owner. Even though tensor-based RNN model presents a minimal risk in the sense of leakage of personal information, there exists a potential to publish two tensor-based RNN models trained from similar databases. Personal attribute values included only in one database for our case, maybe estimated through comparison of the published tensor-based RNN models. By using differential privacy techniques, we can guarantee that the risk of private information leakage caused by the tensor-based RNN model is lower than a predefined threshold.

3.1 Tensor-Based RNN Model

A tensor-based RNN model is proposed to deal with high order data by using tensor computations [10, 24]. Tensors (multidimensional arrays) are generalized forms of one dimensional arrays (vectors) and two dimensional arrays (matrices). Tensor order is defined as the number of dimensions of a tensor. Other names for tensor order are ways or modes. The tensor-based RNN is the tensor space extension of RNN. The tensor-based RNN models comprise many-to-many tensor-based RNN model, many-to-one tensor-based RNN model, and one-to-many tensor-based RNN model. Fig. 2 shows the structure of one RNN model.

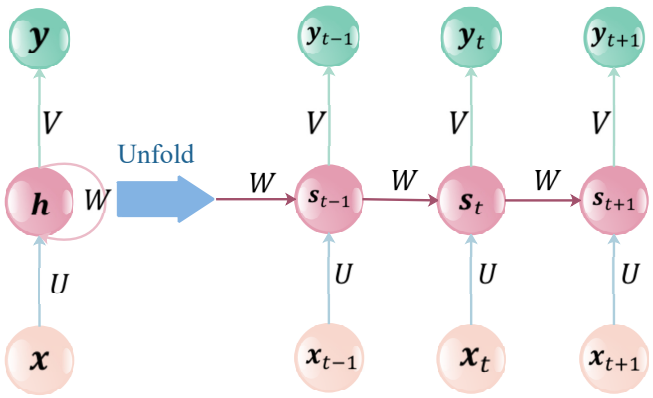


Fig. 2. Structure of One Many-to-many RNN Model.

In this work, we concentrate on one many-to-one tensor-based RNN model. The structure of the many-to-one tensor-based RNN model is shown in Fig. 3. A simple many-to-one tensor-based RNN model has three layers: Input tensor layer, hidden tensor layer, and output tensor layer. Specically, the related definitions of many-to-one tensor-based RNN model are given as follows. The input tensor is $\mathcal{X} \in R^{I_1 \times I_2 \times \cdots \times I_N \times 1}$. The variable of hidden tensor layer is $\mathcal{H} \in R^{J_1 \times J_2 \times \cdots \times J_H \times 1}$. The activation function of the hidden tensor layer is $f_h(\cdot)$. The hidden-to-hidden fully-connected weight tensor is $\mathcal{W} \in R^{\alpha \times J_1 \times J_2 \times \cdots \times J_H}$. The input-to-hidden weight tensor is $\mathcal{U} \in R^{\alpha \times I_1 \times I_2 \times \cdots \times I_N}$, where $\alpha = J_1 \times J_2 \times \cdots \times J_H$. The variable of output tensor layer is $\mathcal{O} \in R^{L_1 \times L_2 \times \cdots \times L_M \times 1}$. The activation function of the output tensor layer is $f_o(\cdot)$. The hidden-to-output weight tensor is $\mathcal{V} \in R^{\beta \times J_1 \times J_2 \times \cdots \times J_H}$, where $\beta = L_1 \times L_2 \times \cdots \times L_M$.

Definition 4. *Tensor Multi-Dot Product* ($\odot$)*: Assume* $Y \in R^{\varphi \times K_1 \times K_2 \times \cdots \times K_M}$ *represents* $(M+1)$*-order tensor having* φ *sub-tensors, every sub-tensor is expressed as* $Y_\delta \in R^{K_1 \times K_2 \times \cdots \times K_M}$*, and* B *represents a tensor of size* $K_1 \times K_2 \times \cdots \times K_M$*. The multi-dot product* ($\odot$) *of the tensors* Y*,* B *produces a new* M*-order tensor* $T \in R^{I_1 \times I_2 \times \cdots \times I_M} (I_1 \times I_2 \times \cdots \times I_M = \varphi)$*.* T *is delineated as* $T = Y \odot B, \forall t_{i_1 i_2 \cdots i_M} \in T, t_{i_1 i_2 \cdots i_M} = Y_\delta \cdot B, (\delta = i_M + \sum_{k=1}^{M-1} (i_k - 1) \prod_{j=k+1}^{M} I_j)$.

The many-to-one tensor-based RNN model includes forward calculation and backward propagation.

The forward calculation for the tensor-based RNN model is given as follows. As illustrated in Fig. 3, the forward

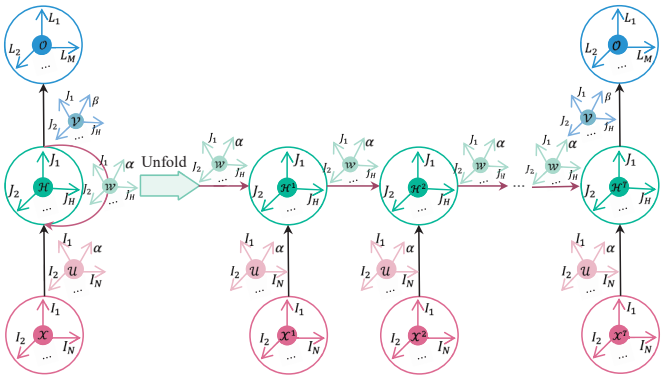


Fig. 3. Structure of Many-to-one Tensor RNN Model.

calculation at time t ($t < T$) can be formulated as:

$$\begin{aligned} \mathcal{H}^{(t)*} &= \mathcal{U} \odot \mathcal{X}^t + \mathcal{W} \odot \mathcal{H}^{t-1}, \\ \mathcal{H}^t &= f_h\left(\mathcal{H}^{(t)*}\right), \\ \mathcal{H}^0 &= 0. \end{aligned} \tag{9}$$

And the forward calculation at time T can be formulated as follows:

$$\mathcal{O} = f_o\left(\mathcal{V} \odot \mathcal{H}^T\right). \tag{10}$$

In the tensor-based RNN model, backward propagation is utilized to compute gradients to achieve a better parameter update. The backward propagation for the model is given as follows. Let M be the number of samples. The loss function is defined to be:

$$L = \frac{1}{M}\sum_{m}^{M}\sum_{t=1}^{T} L^t. \tag{11}$$

Using gradient chain rule, the partial derivatives of L with respect to each tensor and element can be calculated by the following equations:

$$\frac{\partial L^t}{\partial o^{(t)*}_{l_1 l_2 \cdots l_M}} = \frac{\partial L^t}{\partial o^{(t)}_{l_1 l_2 \cdots l_M}} f'_o(o^{(t)*}_{l_1 l_2 \cdots l_M}), \tag{12}$$

$$\frac{\partial L}{\partial V_\mu} = \sum_{t=1}^{T}\frac{\partial L^t}{\partial V_\mu}, \frac{\partial L^t}{\partial V_\mu} = \frac{\partial L^t}{\partial o^{(t)*}_{l_1 l_2 \cdots l_M}} \cdot H^t, \tag{13}$$

$$\frac{\partial L^t}{\partial h^{(t)*}_{j_1 j_2 \cdots j_H}} = \frac{\partial L^t}{\partial o^{(t)*}_{l_1 l_2 \cdots l_M}} \cdot v_{\mu j_1 j_2 \cdots j_H} \cdot f'_h(h^{(t)*}_{j_1 j_2 \cdots j_H}), \tag{14}$$

$$\frac{\partial L}{\partial U_\gamma} = \sum_{t=1}^{T}\frac{\partial L^t}{\partial U_\gamma}, \frac{\partial L^t}{\partial U_\gamma} = \sum_{k=1}^{t-1}\frac{\partial L^t}{\partial h^{(k)*}_{j_1 j_2 \cdots j_H}} \cdot X^k, \tag{15}$$

$$\frac{\partial L}{\partial W_\gamma} = \sum_{t=1}^{T}\frac{\partial L^t}{\partial W_\gamma}, \frac{\partial L^t}{\partial W_\gamma} = \sum_{k=1}^{t-1}\frac{\partial L^t}{\partial h^{(k)*}_{j_1 j_2 \cdots j_H}} \cdot H^{k-1}. \tag{16}$$

In the tensor-based RNN model, parameters are updated by utilizing stochastic gradient descent. The weight tensors are updated by:

$$\begin{aligned} U_\gamma &= U_\gamma - \eta\frac{\partial L}{\partial U_\gamma}, \\ W_\gamma &= W_\gamma - \xi\frac{\partial L}{\partial W_\gamma}, \\ V_\mu &= V_\mu - \lambda\frac{\partial L}{\partial V_\mu}, \end{aligned} \tag{17}$$

where η, ξ, λ indicate learning rate parameters.

3.2 Tensor-Based RNN under Differential Privacy

Differential privacy offers a strong standard for privacy protection for training tensor-based RNN model on aggregate big data. A differentially private tensor-based RNN model is proposed in this subsection. The popular privacy mechanisms, Laplace mechanism and functional mechanism, are applied as concrete construction.

Algorithm 1 Tensor-Based Back-propagation through Time algorithm with Perturbation

Input: Examples $D = \{\mathcal{Z}_1, \mathcal{Z}_2, \cdots, \mathcal{Z}_m, \cdots, \mathcal{Z}_M\}$. Parameters: maximum number of iterations $iterMax$, privacy budget cb, total privacy budget tb, learning rate η, ξ, λ, threshold thr.

Output: Model parameter $\theta = \{\mathcal{U}, \mathcal{W}, \mathcal{V}\}$.

Initialize $\theta, \mathcal{H}^0, \Delta$;
for $iter = 1, 2, \cdots, iterMax$ **do**
  for $m = 1, 2, \cdots, M$ **do**
    for $t = 1, 2, \cdots, T$ **do**
      Compute H^t according to equation (9);
      for $j_1 = 1, 2, \cdots, J_1$ **do**
        $\cdots$
        for $t = 1, 2, \cdots, T$ **do**
          for $l_1 = 1, 2, \cdots, L_1$ **do**
            $\cdots$
            for $l_N = 1, 2, \cdots, l_M$ **do**
              Compute $o^{(t)}_{l_1 l_2 \cdots l_M}$ according to equation (10);
              Compute $\omega^{(0)}_m, \omega^{(1)}_m, \omega^{(2)}_m$ according to equation (23);
              Compute the perturbed loss function $\overline{L}$ according to equation (24);
              Compute the derivative $\frac{\partial \overline{L}}{\partial o^{(t)*}_{l_1 l_2 \cdots l_M}}$ according to equation (12);
              Compute the derivative $\frac{\partial \overline{L}}{\partial h^{(t)*}_{j_1 j_2 \cdots j_H}}$ according to equation (14);
            end for
            $\cdots$
          end for
        end for
        $\cdots$
      end for
    end for
    Generate the gradients $\frac{\partial \overline{L}}{\partial V_\mu}, \frac{\partial \overline{L}}{\partial U_\gamma}, \frac{\partial \overline{L}}{\partial W_\gamma}$ according to equations (13, 15, 16);
  end for
  Update cb;
  if $L > thr$ and $cb < tb$ **then**
    Update the weight tensors $U_\gamma, W_\gamma, V_\mu$ according to equation (17).
  else
    RETURN 0.
  end if
end for
RETURN 0.

A Straightforward Scheme: We may propose a differentially private scheme, which directly injects random noise only into the final results gotten from the users' sensitive

data, to protect the training data privacy in tensor-based RNN model. However, this straightforward scheme may compromise the availability of the learned tensor-based RNN model. An Advanced Scheme: We prefer an alternative advanced differentially private tensor-based RNN scheme in this paper. More specifically, a sophisticated tensor-based back-propagation through time (TBPTT) algorithm with perturbation is designed to achieve privacy guarantees for the tensor-based RNN models by relying on the ϵ-differential privacy.

The softmax function, a monotonically increasing activation function, is widely used in multi-class classification problems. We denote the result of the softmax function on tensor O as $softmax(O)$. The tensor cross-entropy of two tensors $O, Y \in R^{I_1 \times I_2 \times \cdots \times I_N}$ is defined as:

$$CrossEntropy(O, Y) = -Y \cdot \ln(softmax(O)). \tag{18}$$

Let the model parameter $\theta = \{U, W, V\}$. The loss function, which denotes the penalty for mismatching the training samples, is defined to be:

$$L(\theta) = -\sum_{m=1}^{M}\sum_{p=1}^{\beta} y_p^m \cdot \ln(\frac{e^{o_p^m}}{e^{o_p^m} + \sum_{q=1,q\neq p}^{\beta} e^{o_q^m}}), \tag{19}$$

where $o_p^m = V_p \cdot H^m, p = l_m + \sum_{j=1}^{M-1}(l_j - 1)\prod_{t=j+1}^{t=M} L_t, \beta = L_1 \times L_2 \times \cdots \times L_M$. In order to use differential privacy, o_p^m is seen as independent variable in equation (19). Then we get the following form:

$$L(o_1^m, o_m^m, \cdots, o_\beta^m) = -\sum_{p=1}^{\beta} y_p^m \cdot \ln(\frac{e^{o_p^m}}{e^{o_p^m} + \sum_{q=1,q\neq p}^{\beta} e^{o_q^m}}). \tag{20}$$

Applying the Taylor expansion on equation (20) and truncating the Taylor series, we obtain the following low-order polynomial approximation:

$$\begin{aligned}\widetilde{L}(o_1^m, o_m^m, \cdots, o_\beta^m) &= L(0, 0, \cdots, 0) \\ &+ \sum_{p=1}^{\beta} {L_{o_p^m}}^{(1)}(0, 0, \cdots, 0) \cdot o_p^m \\ &+ \tfrac{1}{2}\sum_{p=1}^{\beta}\sum_{q=1}^{\beta} {L_{o_p^m o_q^m}}^{(2)}(0, 0, \cdots, 0) \cdot o_p^m o_q^m,\end{aligned} \tag{21}$$

where the polynomial coefficients can be computed as follows:

$$\begin{aligned}&L(0, 0, \cdots, 0) = \ln\beta, \\ &{L_{o_p^m}}^{(1)}(0, 0, \cdots, 0) = \tfrac{1}{\beta} - y_p^m, \\ &{L_{o_p^m o_q^m}}^{(2)}(0, 0, \cdots, 0) = \begin{cases} \frac{\beta-1}{\beta^2}, p = q \\ -\frac{1}{\beta^2}, p \neq q \end{cases}.\end{aligned} \tag{22}$$

Let Δ be the global sensitivity. In order to achieve ϵ-differential privacy, Laplace noises are injected into the polynomial coefficients:

$$\begin{aligned}\omega_m^{(0)} &= L(0, 0, \cdots, 0) + Lap_m^0\left(\tfrac{\Delta}{\varepsilon}\right), \\ \omega_m^{(1)} &= L_{o_p^m}^{(1)}(0, 0, \cdots, 0) + Lap_m^1\left(\tfrac{\Delta}{\varepsilon}\right), \\ \omega_m^{(2)} &= \frac{L_{o_p^m o_q^m}^{(2)}(0,0,\cdots,0)}{2} + Lap_m^0\left(\tfrac{\Delta}{\varepsilon}\right),\end{aligned} \tag{23}$$

where $Lap\left(\frac{\Delta}{\varepsilon}\right)$ follows the Laplace distribution with mean 0 and scale parameter $\frac{\Delta}{\varepsilon}$.

Using equation (23), we can construct the following perturbed loss function, which is employed in the differentially private tensor RNN:

$$\begin{aligned}\overline{L}(o_1^m, o_m^m, \cdots, o_\beta^m) &= \omega_m^{(0)} + \sum_{p=1}^{\beta} \omega_m^{(1)} \cdot o_p^m \\ &+ \tfrac{1}{2}\sum_{p=1}^{\beta}\sum_{q=1}^{\beta} \omega_m^{(2)} \cdot o_p^m o_q^m.\end{aligned} \tag{24}$$

After the perturbed loss function is obtained, the model parameter θ can be derived by minimizing the perturbed loss function.

The main steps of the proposed TBPTT algorithm with perturbation are highlighted in Algorithm 1. The input and parameters of our algorithm are examples $D = \{\mathcal{Z}_1, \mathcal{Z}_2, \cdots, \mathcal{Z}_m, \cdots, \mathcal{Z}_M\}$, $Z^m = (X^m, Y^m)$, maximum number of iterations $iterMax$, privacy budget ϵ, learning rate η, ξ, λ, and threshold thr. The output of our algorithm is the model parameter $\theta = \{\mathcal{U}, \mathcal{W}, \mathcal{V}\}$. By employing functional mechanism, Laplace distribution noises are added in order to achieve differential privacy in Line 13 in Algorithm 1. The gradients $\frac{\partial L}{\partial \mathcal{V}}$, $\frac{\partial L}{\partial \mathcal{U}}$, $\frac{\partial L}{\partial \mathcal{W}}$ are computed by exploiting the chain rule in Line 24 in Algorithm 1. We take a step in the opposite direction of each gradient to update the parameters in Line 28 in Algorithm 1. If the privacy cost is more than the total privacy budget or $L < thr$, the algorithm will be stopped.

The correctness of the proposed differentially private tensor-based RNN is based on Theorem 1. Before Theorem 1 is stated and proven, the definition of adjacent tensor sets is given firstly.

Definition 5. *Adjacent Tensor Sets: For any two sets D, D', where every element in the sets is one tensor of the same order and the same dimension, if the two sets D, D' contain at most one different tensor, then the two sets D, D' are adjacent tensor sets, which is written as $D \approx D'$.*

Theorem 1. *The proposed tensor-based back-propagation through time (TBPTT) algorithm with perturbation satisfies ϵ-differential privacy.*

Proof. Assume tensor sets D, D' are adjacent tensor sets (i.e. $D \approx D'$). To prove that the proposed algorithm satisfies ϵ-differential privacy, the global sensitivity Δ on the adjacent tensor sets D, D' is computed firstly. Without loss of generality, assume that D and D' differ only in their last tensor element, $X^{T_D} \neq X^{T_{D'}}, |D| = |D'| = T, \beta = L_1 \times L_2 \times \cdots \times L_M$. Then, we have

$$\begin{aligned}\Delta &= \max_{X^{t_1}\in D, X^{t_2}\in D'} \sum_{R=0}^{2} \left\|\omega_{t_1}^{(R)} - \omega_{t_2}^{(R)}\right\| \\ &= \sum_{R=0}^{2} \left\|\omega_{T_D}^{(R)} - \omega_{T_{D'}}^{(R)}\right\| \\ &\leq \sum_{R=0}^{2} (\left\|\omega_{T_D}^{(R)}\right\| + \left\|\omega_{T_{D'}}^{(R)}\right\|) \\ &\leq 2 \max_{X^t\in D, D'} \sum_{R=0}^{2} \left\|\omega_t^{(R)}\right\|.\end{aligned} \tag{25}$$

The outputs of hidden layer are normalized. According to equation (22), we have $\left\|\omega_{T_D}^{(0)}\right\| = \left\|\omega_{T_{D'}}^{(0)}\right\|$. Therefore, we

have the following inequality:

$$\begin{aligned}\Delta &\leq 2 \max_{X^t \in D, D'} \sum_{R=0}^{2} \left\| \omega_t^{(R)} \right\| \\ &= 2 \max_{X^t \in D, D'} \sum_{R=1}^{2} \left\| \omega_t^{(R)} \right\| \\ &= 2 \max_{X^t \in D, D'} \left\| \sum_{p=1}^{\beta} \left(\frac{1}{\beta} - y_p^t \right) \right\| + \left\| \sum_{p=1}^{\beta} \sum_{q=1, q \neq p}^{\beta} -\frac{1}{\beta^2} \right\| \\ &+ \left\| \sum_{p=q=1}^{\beta} \frac{\beta-1}{\beta^2} \right\| \\ &\leq 2 \left(\beta + \frac{1}{\beta^2} \cdot \beta \cdot (\beta - 1) + \frac{\beta-1}{\beta^2} \cdot \beta \right) \\ &= \frac{2\beta^2 + 4\beta - 4}{\beta}.\end{aligned} \tag{26}$$

The sensitivity of one function provides an upper bound on how much we should disturb the output of the function to preserve privacy. According to equation (26), the sensitivity Δ has upper bound. For multi-class problem, β is the number of classes. The number of classes is usually limited, therefore we can get appropriate sensitivity Δ.

Let $\omega_t^{(R)}$ be the real output, $\omega_{T_D}^{(R)}$ be the output of the algorithm with perturbation on the tensor set D, and $\omega_{T_{D'}}^{(R)}$ be the output of the algorithm with perturbation on the tensor set D'.

We can derive that:

$$\begin{aligned}\frac{\Pr(\overline{L}(\theta)|D)}{\Pr(\overline{L}(\theta)|D')} &= \frac{\prod_{t=1}^{T} \prod_{R=0}^{2} e^{\frac{\varepsilon \left\| \omega_{T_D}^{(R)} - \omega_t^{(R)} \right\|}{\Delta}}}{\prod_{t=1}^{T} \prod_{R=0}^{2} e^{\frac{\varepsilon \left\| \omega_{T_{D'}}^{(R)} - \omega_t^{(R)} \right\|}{\Delta}}} \\ &= \frac{\prod_{R=0}^{2} e^{\frac{\varepsilon \left\| \omega_{T_D}^{(R)} - \omega_t^{(R)} \right\|}{\Delta}}}{\prod_{R=0}^{2} e^{\frac{\varepsilon \left\| \omega_{T_{D'}}^{(R)} - \omega_t^{(R)} \right\|}{\Delta}}} \\ &= e^{\frac{\varepsilon}{\Delta} \sum_{R=0}^{2} \left(\left\| \omega_{T_D}^{(R)} - \omega_t^{(R)} \right\| - \left\| \omega_{T_{D'}}^{(R)} - \omega_t^{(R)} \right\| \right)}.\end{aligned} \tag{27}$$

By the triangle inequality, we have the following inequality $\left\| \omega_{T_D}^{(R)} - \omega_t^{(R)} \right\| - \left\| \omega_{T_{D'}}^{(R)} - \omega_t^{(R)} \right\| \leq \left\| \omega_{T_D}^{(R)} - \omega_{T_{D'}}^{(R)} \right\|$. Therefore, we have that:

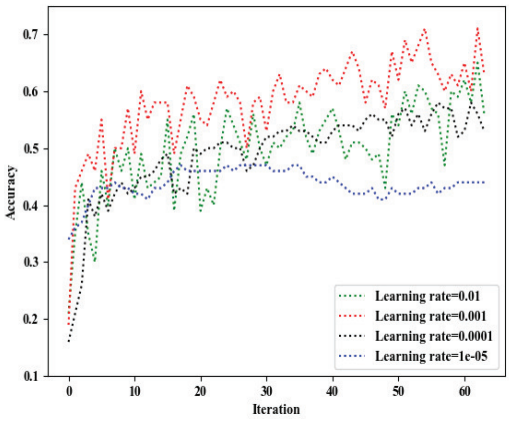
$$\begin{aligned}\frac{\Pr(\overline{L}(\theta)|D)}{\Pr(\overline{L}(\theta)|D')} &\leq e^{\frac{\varepsilon}{\Delta} \sum_{R=0}^{2} \left(\left\| \omega_{T_D}^{(R)} - \omega_{T_{D'}}^{(R)} \right\| \right)} \\ &= e^{\frac{\varepsilon}{\Delta} \cdot \Delta} \\ &= e^{\varepsilon}.\end{aligned} \tag{28}$$

$\frac{\Pr\{\mathcal{U}=\mathcal{U}(D)\}}{\Pr\{\mathcal{U}=\mathcal{U}(D')\}}$ is less than or equal to e^{ϵ}. Consequently, we obtain the conclusion: The resulting model satisfies ϵ-differential privacy. Therefore, the proposed scheme can be used to train a tensor RNN model while guaranteeing the differential privacy preservation of the user's data. The scheme can protect against a strong adversary who has the complete knowledge of training the tensor-based RNN model.

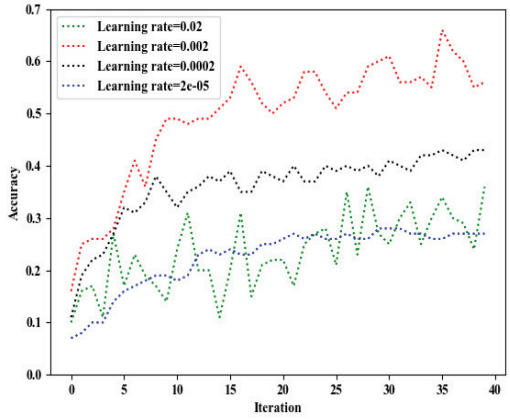
□

4 Experimental Results

In this section, we report the experimental results of the differentially private tensor-based RNN model. To validate the proposed model, rigorous experimental evaluations were conducted on two well-known real-world action recognition datasets: KTH [25] and UCF11 [26], which are used extensively in IoT-enabled smart city monitoring systems research.



(a) KTH dataset

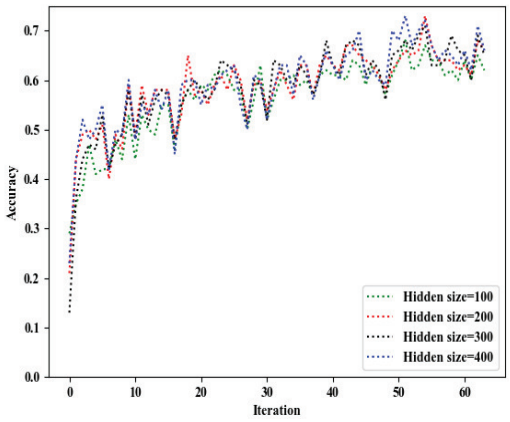


(b) UCF11 dataset

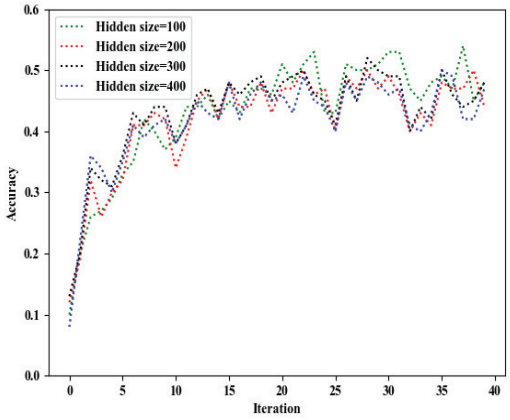
Fig. 4. Accuracies of the tensor-based RNN models for varying learning rate.

The real-world KTH dataset [25] comprises six categories of human actions i.e jogging, boxing, hand clapping, walking, running and hand waving each of which is performed severally by 25 subjects under 4 different conditions: Outdoors c_1, outdoors with varying scale c_2, outdoors using different clothes c_3 and finally indoors c_4. The current database contains 2391 sequences taken on similar backgrounds using a static camera having a frame rate of 25 fps. The composition of the UCF11 dataset includes 1600 video clips divided into 11 action groups. They include horse riding, diving, jumping and others. Every group consists of 25 video categories, whereby each category contains 4 or more clips. For a detailed explanation about UCF11 dataset, please refer to [26].

We programmed all of the algorithms in tensor-based RNN and differentially private tensor-based RNN in Python. Our code is executed on a machine with a 3.6 GHz Quad-Core i7 Intel processor. For each dataset, we varied iteration counts, learning rate, hidden size, and privacy budget ϵ in a series of experimental evaluations. The tensor-based RNN model does not enforce ϵ-differential privacy while the differentially private tensor-based RNN model

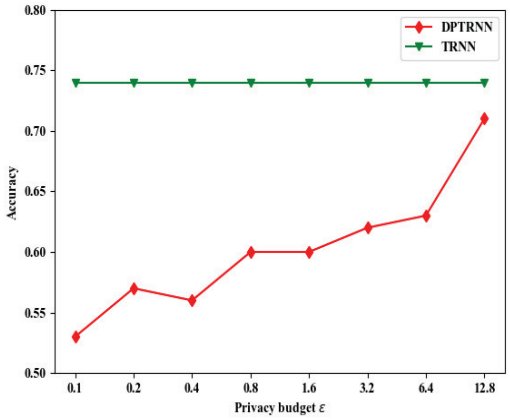


(a) KTH dataset

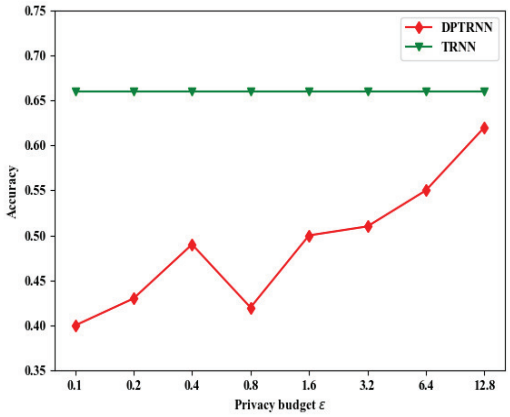


(b) UCF11 dataset

Fig. 5. Accuracies of the differentially private tensor-based RNN models for varying size of hidden units.



(a) KTH dataset



(b) UCF11 dataset

Fig. 6. Results on accuracies of the tensor-based RNN models and the differentially private tensor-based RNN models for varying privacy budget ϵ.

enforces ϵ-differential privacy. It is obvious that our differentially private tensor-based RNN mainly incurs the following side-effect on tensor-based RNN: It incurs inaccuracies due to the noise injection in training tensor-based RNN model. Therefore, accuracy as the key metric is employed in our performance evaluations.

To determine how the accuracy of the tensor-based RNN model was affected by the learning rate, we conducted experiments that varied the learning rate. Figure 4(a) and Figure 4(b) show the algorithm's results on classification accuracy, as a function of the iteration number, for varying learning rate on the KTH dataset and the UCF11 dataset, respectively. We observe that when the learning rate is 0.001, the algorithm has higher accuracies than when the learning rates are 0.01, 0.0001, 0.00001 on the KTH dataset. We observe that when the learning rate is 0.002, the algorithm has higher accuracies than when the learning rates are 0.02, 0.0002, 0.00002 on the UCF11 dataset. We also observe that the learning rates have significant influence on accuracies of algorithm on the KTH dataset and the UCF11 dataset.

To determine how the accuracy of the differentially private tensor-based RNN model was affected by the size of hidden units, we conducted experiments that varied the size of hidden units in the differentially private tensor-based RNN model from 100 to 400 on the KTH dataset and the UCF11 dataset. Figure 5(a) and Figure 5(b) show the algorithm's classification accuracy, as a function of the iteration number, for varying size of hidden units on the KTH dataset and the UCF11 dataset, respectively. The results from Figure 5(a) and Figure 5(b) indicate that increasing the size of hidden units in the differentially private tensor-based RNN model does not decrease the classification accuracy, and that the effects of the size of hidden units on the differentially private tensor-based RNN models are insignificant on the KTH dataset and the UCF11 dataset.

To determine how the accuracies of the tensor-based RNN models and the differentially private tensor-based RNN models were affected by the privacy budget ϵ, we conducted experiments that varied the privacy budget ϵ. Figure 6(a) shows each algorithm's classification accuracy as a function of the privacy budget ϵ on the KTH dataset. As can be observed from the figure, the tensor-based RNN scheme's accuracies remain unchanged with the varying privacy budget ϵ. This is simply because tensor-based RNN scheme does not enforce ϵ-differential privacy. Since the size of ϵ is inversely proportional to the amount of noise (i.e. small values of privacy budget ϵ requires more noise to be injected), the differentially private tensor-based RNN model that enforces ϵ-differential privacy suffers from inaccuracies with decreasing ϵ. It is worth noting that the differentially private tensor RNN model is competitive with the non-privacy preservation model. Figure 6(b) shows each algo-

rithm's classification accuracy as a function of the privacy budget ϵ on the UCF11 dataset. The experimental results on the UCF11 dataset are consistent with those on the KTH dataset. Therefore, the proposed differentially private tensor RNN model is feasible.

5 Related Works

In this section, we present some of the most recent related works done in our line of research.

There have been a number of efforts to develop and use RNN in IoT [27]. The main distinguishing feature of RNN from standard neural networks is the persistence of information about previous states used to determine the current state i.e. hidden layers are interconnected with each connection associated with a time delay [28]. By utilizing these connections, the model is able to remember past states which enables it to discover temporal correlations among events that are far away from each other in the dataset. The RNN is thus an extension of a feedforward neural network that is capable of handling a variable-length input sequence by utilizing a recurrent hidden state whose activation at any given time depends on that of previous time [27]. Recently, Cao et al. [29] proposed a RNN based short-term forecasting method for land use change. Lai et al. [21] proposed a method combining LSTM and edge computing to analyze data features for industrial IoT. Alazab et al. [27] presented a novel multidirectional long short-term memory model for the stability prediction of smart grid network, and this model has shown to be very superior to prior approaches. Gong et al. [1] proposed a novel hybrid deep neural network model for friend recommendation, and the resulting model has shown to be effective. The existing RNN models are almost designed based on matrix operations, therefore they are not suitable for heterogeneous IoT data.

There have been some efforts to protect data privacy using differential privacy. With regard to existing literature, Dwork et al. [16] proposed a ϵ-differential private scheme and showed that the Laplace method (supporting queries having real numbers as outputs) can be used to enforce the privacy. Although this method has been used in a wide range of existing work, it is limited to aggregate queries (such as sum, histograms) or queries that are reducible into simple aggregates. As a way of complementing the Laplace method, the exponential method (supporting queries with discrete output spaces) was proposed by McSherry and Talwar [19]. This made it possible for implementation of various difficult differential privacy problems whose outputs are not real numbers. Despite the two methods having achieved substantial success in implementing differential privacy solutions, it is still challenging to use them in neural network tasks as a result of complex correlations between the inputs and outputs through the objective functions. There are several related efforts where differential privacy is used with neural networks. Abadi et al. in the literature [30] employed the gradient approximation method based on differential privacy, which is first to add Gaussian noise to perturb the gradients at each iteration of neural network training to provide provable privacy guarantees. Furthermore, they also proposed moments accountant method to tightly estimate the cumulative privacy loss. Li et al. [31] proposed an efficient secure computation scheme for differentially private data publishing in a cloud environment. Phan et al. [22, 23] presented differentially private deep neural networks based on functional mechanism. Recently, Chen et al. [3] proposed a differentially private recurrent neural network model for the privacy protection of dynamic trajectory. Unlike the existing schemes, our proposed scheme focuses on high order data in IoT. The existing schemes are different from the scheme proposed in our paper in the target model.

6 Conclusions

Processing heterogeneous IoT data and ensuring IoT data security and privacy become more fundamental concerns in practical IoT systems. In this paper, we have presented a tensor-based RNN model and a mechanism of preserving privacy of information in training of tensor-based RNN, differentially private tensor-based RNN (DPTRNN), for IoT systems for the first time. Our proposed scheme is able to process heterogeneous IoT data and achieve stronger privacy guarantees by enforcing ϵ-differential privacy, the most promising privacy metric used in privacy-preserving machine learning tasks, during the training process of the tensor-based RNN model. By using the scheme, membership inference attack exposure can be reduced. Extensive experiments were conducted by using the KTH dataset and the UCF11 dataset. The results of our experiments demonstrated the accuracy and effectiveness of the proposed tensor-based RNN with differential privacy.

As future work, we would like to set our sights on the schemes under differential privacy that can improve the efficiency of the tensor-based RNN models while still preserving privacy of IoT user data. We would like to present several optimization strategies (such as adaptive learning rate to improve the training procedure) to enhance the accuracy of the differentially private tensor-based RNN model. We also plan to adapt our proposal to other commonly used tensor-based deep computation algorithms for IoT systems.

References

[1] J. Gong, Y. Zhao, S. Chen, H. Wang, L. Du, S. Wang, M. Z. A. Bhuiyan, H. Peng, and B. Du, "Hybrid deep neural networks for friend recommendations in edge computing environment," *IEEE Access*, vol. 8, pp. 10 693–10 706, 2019.

[2] M. Z. A. Bhuiyan, J. Wu, G. M. Weiss, T. Hayajneh, T. Wang, and G. Wang, "Event detection through differential pattern mining in cyber-physical systems," *IEEE Trans. Big Data*, vol. 6, no. 4, pp. 652–665, 2020.

[3] S. Chen, A. Fu, J. Shen, S. Yu, H. Wang, and H. Sun, "RNN-DP: A new differential privacy scheme base on recurrent neural network for dynamic trajectory privacy protection," *Journal of Network and Computer Applications*, vol. 168, p. 102736, 2020.

[4] S. Huang, K. Ota, M. Dong, and F. Li, "MultiSpectralNet: Spectral clustering using deep neural network for multi-view data," *IEEE Trans. Computational Social Systems*, 2019, DOI: 10.1109/TCSS.2019.2926450.

[5] A. Azmoodeh, A. Dehghantanha, and K.-K. R. Choo, "Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning," *IEEE Trans. Sustainable Computing*, vol. 4, no. 1, pp. 88–95, 2018.

[6] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. 38th IEEE Int'l Conf. Acoustics, Speech and Signal Processing (ICASSP'13)*, 2013, pp. 6645–6649.

[7] Z. M. Fadlullah, B. Mao, F. Tang, and N. Kato, "Value iteration architecture based deep learning for intelligent routing exploiting heterogeneous computing platforms," *IEEE Trans. Computers*, vol. 68, no. 6, pp. 939–950, 2019.

[8] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang *et al.*, "Abstractive text summarization using sequence-to-sequence rnns and beyond," *arXiv preprint arXiv:1602.06023*, 2016.

[9] K.-C. Li, H. Jiang, L. T. Yang, and A. Cuzzocrea, *Big data: Algorithms, analytics, and applications*. CRC Press, 2015.

[10] J. Feng, L. T. Yang, R. Zhang, W. Qiang, and J. Chen, "Privacy preserving high-order bi-lanczos in cloud-fog computing for industrial applications," *IEEE Trans. Industrial Informatics*, 2020, DOI: 10.1109/TII.2020.2998086.

[11] N. Park, S. Oh, and U. Kang, "Fast and scalable method for distributed boolean tensor factorization," *The VLDB Journal*, pp. 1–26, 2019.

[12] Q. Zhang, L. T. Yang, and Z. Chen, "Deep computation model for unsupervised feature learning on big data," *IEEE Trans. Services Computing*, vol. 9, no. 1, pp. 161–171, 2016.

[13] W. Feng, Z. Yan, H. Zhang, K. Zeng, Y. Xiao, and Y. T. Hou, "A survey on security, privacy, and trust in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2971–2992, 2017.

[14] D. Liu, Z. Yan, W. Ding, and M. Atiquzzaman, "A survey on secure data analytics in edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4946–4967, 2019.

[15] W. Ding, X. Jing, Z. Yan, and L. T. Yang, "A survey on data fusion in internet of things: Towards secure and privacy-preserving fusion," *Information Fusion*, vol. 51, pp. 129–144, 2019.

[16] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[17] F. Zhang, V. E. Lee, and K.-K. R. Choo, "Jo-DPMF: Differentially private matrix factorization learning through joint optimization," *Information Sciences*, vol. 467, pp. 271–281, 2018.

[18] X. Yang, T. Wang, X. Ren, and W. Yu, "Survey on improving data utility in differentially private sequential data publishing," *IEEE Trans. Big Data*, 2017, 10.1109/TBDATA.2017.2715334.

[19] M. Li, L. Zhu, Z. Zhang, and R. Xu, "Achieving differential privacy of trajectory data publishing in participatory sensing," *Information Sciences*, vol. 400, pp. 1–13, 2017.

[20] Q. Feng, D. He, Z. Liu, H. Wang, and K. R. Choo, "SecureNLP: A system for multi-party privacy-preserving natural language processing," *IEEE Trans. Information Forensics and Security*, vol. 15, pp. 3709–3721, 2020.

[21] C.-F. Lai, W.-C. Chien, L. T. Yang, and W. Qiang, "LSTM and edge computing for big data feature recognition of industrial electrical equipment," *IEEE Trans. Industrial Informatics*, vol. 15, no. 4, pp. 2469–2477, 2019.

[22] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential privacy preservation for deep auto-encoders: an application of human behavior prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[23] N. Phan, X. Wu, H. Hu, and D. Dou, "Adaptive laplace mechanism: Differential privacy preservation in deep learning," in *Proc. IEEE Int'l Conf. Data Mining (ICDM'17)*, 2017, pp. 385–394.

[24] S. Oh, N. Park, J.-G. Jang, L. Sael, and U. Kang, "High-performance tucker factorization on heterogeneous platforms," *IEEE Trans. Parallel and Distributed Systems*, 2019, DOI: 10.1109/TPDS.2019.2908639.

[25] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th IEEE Int'l Conf. Pattern Recognition (ICPR'04)*, 2004.

[26] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'09)*, 2009.

[27] M. Alazab, S. Khan, S. S. R. Krishnan, Q. Pham, M. P. K. Reddy, and T. R. Gadekallu, "A multidirectional LSTM model for predicting the stability of a smart grid," *IEEE Access*, vol. 8, pp. 85 454–85 463, 2020.

[28] Z. Wang, M. Song, S. Zheng, Z. Zhang, Y. Song, and Q. Wang, "Invisible adversarial attack against deep neural networks: An adaptive penalization approach," *IEEE Trans. Dependable and Secure Computing*, 2019, DOI: 10.1109/TDSC.2019.2929047.

[29] C. Cao, S. Dragićević, and S. Li, "Short-term forecasting of land use change using recurrent neural network models," *Sustainability*, vol. 11, no. 19, p. 5376, 2019.

[30] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. 23rd ACM SIGSAC Conf. Computer and Communications Security (CCS'16)*, 2016, pp. 308–318.

[31] J. Li, H. Ye, T. Li, W. Wang, W. Lou, T. Hou, J. Liu, and R. Lu, "Efficient and secure outsourcing of differentially private data publishing with multiple evaluators," *IEEE Trans. Dependable and Secure Computing*, 2020, DOI: 10.1109/TDSC.2020.3015886.

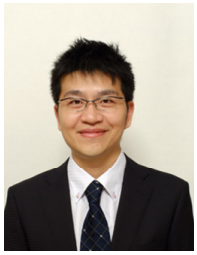
Jun Feng is a postdoctoral research fellow with the School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan, China. He received the Ph.D degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. He is particularly interested in privacy-preserving machine learning, cyber-physical-social systems, differential privacy, deep learning, blockchain, and big data.

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2023.3236868

Laurence T. Yang received the B.E. degree in Computer Science and Technology from Tsinghua University, Beijing, China, and the Ph.D. degree in Computer Science from the University of Victoria, Victoria, BC, Canada. He is a professor with the School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan, China, and with the Department of Computer Science, St. Francis Xavier University, Antigonish, NS, Canada. His research focuses on cyber-physical-social systems, big data, parallel and distributed computing, and embedded and ubiquitous/pervasive computing. His research has been supported by the National Sciences and Engineering Research Council, Canada, and the Canada Foundation for Innovation.

Bocheng Ren received the master's degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2020. He is pursuing the Ph.D degree in the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. He is particularly interested in deep learning, differential privacy, big data trust, security, and privacy.

Deqing Zou is currently a professor in the School of Cyber Science and Engineering at Huazhong University of Science and Technology, Wuhan, China. He received his PH.D at Huazhong University of Science and Technology in 2004. His main research interests include cloud security, virtualization, system security, and trusted computing. He has been the leader of one 863 project of China and three National Natural Science Foundation of China projects, and core member of several important national projects, such as National 973 Basic Research Program of China. He has applied almost 80 patents, published four books and more than 100 papers, including papers published by NDSS, ASE, ASPLOS, TDSC, TPDS, and TOSEM.



Mianxiong Dong received B.S., M.S. and Ph.D. in Computer Science and Engineering from The University of Aizu, Japan. He is currently a Professor in the Department of Sciences and Informatics at Muroran Institute of Technology, Japan. Dr. Dong was selected as a Foreigner Research Fellow (a total of 3 recipients all over Japan) by NEC C&C Foundation in 2011. He is the recipient of IEEE TCSC Early Career Award 2016, IEEE SCSTC Outstanding Young Researcher Award 2017, The 12th IEEE ComSoc Asia-Pacific Young Researcher Award 2017, Funai Research Award 2018 and NISTEP Researcher 2018 (one of only 11 people in Japan) in recognition of significant contributions in science and technology from MEXT. He is 2019 Highly Cited Researcher (Web of Science).

Shunli Zhang is a PhD student in School of Computer Science and Technology at Huazhong University of Science and Technology, Wuhan, China. His research focuses on artificial intelligence, cloud computing security and privacy, and applied cryptography.