2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

A Lightweight, Privacy-Preserving Tensor Completion Framework for Internet of Things

Shuyu Wu[†], Linghe Kong[†*], Qiao Xiang[§], Zhenzhe Zheng[†], Luoyi Fu[†], Guihai Chen[†]
[†]Shanghai Jiao Tong University, [§]Xiamen University
[*]Corresponding author: linghe.kong@sjtu.edu.cn

***Abstract*—Machine learning enabled Internet of Things (IoT) applications (*e.g.*, object recognition and autonomous driving) require collecting large amounts of multi-dimensional data, called *tensors*, at IoT devices. Due to the lower-power and distributed nature of these devices, however, many data are missing during collection (*i.e.*, incomplete tensors), impairing the performance of IoT applications. The best practice to cope with this issue is to collect the private raw data from all devices to a centralized server to execute tensor completion algorithms, causing severe privacy concerns. This paper systematically investigates the new problem of privacy-preserving tensor completion, and designs *TwilightTensor*, a lightweight, privacy-preserving tensor completion framework. In *TwilightTensor*, each IoT device works in parallel to retrieve public shared data from a logically centralized server, uses a novel, lightweight obfuscation mechanism to individually disguise its private raw data, and sends the obfuscated raw data to the server. The server then executes tensor completion algorithms with the obfuscated tensors from all devices as input, and distributes the corresponding results back to devices. Each device then independently de-obfuscates the received results to get its own reconstructed complete tensors. We rigorously analyze the performance of *TwilightTensor*, implement a prototype of *TwilightTensor* and conduct extensive experiments using real-world datasets for different IoT applications. Results show that *TwilightTensor* achieves comparable recovery accuracy of state-of-the-art tensor completion algorithms, while preserving the data privacy of IoT devices.**

I. Introduction

Many IoT scenarios benefit substantially from machine learning [1], [2]. These IoT applications collected large amounts of data from IoT devices such as sensors, wearable devices, and smartwatches to train models that are used in various scenarios, including computer vision [3], traffic prediction [4], weather forecasting [5], health care alerting [6] and recommendation [7]. The collected data are of multiple dimensions and the most widely used approach to encoding such data is to organize them in a multi-dimension array, which is also called *tensors*.

The completeness of tensors, representing to what extent data elements are collected in tensors, plays a fundamental role in deciding the efficiency of IoT machine learning applications (*e.g.*, accuracy). However, due to the lower-power and distributed nature of IoT devices, data missing and corruption are common in IoT devices [8][9]. To cope with this, many *tensor completion* algorithms [10], [3], [11] are developed, with the aim of reconstructing the missing data in tensors.

Although much progress has been made in developing tensor completion algorithms, the current best practice to deploy these algorithms in IoT machine learning applications consists of collecting the raw data from all IoT devices to a centralized server to execute the tensor completion algorithms. This is due to the limited computation capability of IoT devices [12].

However, collecting the raw data of IoT devices to a third party server raises severe privacy concerns [13], [14] (*e.g.*, Fig 1). First, adversaries participating in the data collection and transmission process may conduct sniffing and spoofing attacks [15]. Second, with the prevalence of cloud services, the centralized server becomes the single point of failure, increasing the chance of service hijacking [16][17]. Recent work [18], [19] proposed solutions to preserve the privacy of clients by having clients encrypt the private data before sending it to the server. However, these methods only focus on the protection of private data. Using the encrypted data from these methods as input, the tensor completion algorithms cannot yield accurate recovery results [8].

In this paper, we systematically investigate the privacy-preserving tensor completion problem: *how can the missing data in tensors be accurately estimated while the private data at devices are not exposed?* This is a *non-trivial* task due to the conflicting requirements of data privacy preservation and tensor completion. In particular, to achieve the accurate estimation of missing data in tensors, the data (*i.e.*, the input of tensor completion algorithms) collected from distributed IoT devices should maintain their authenticity. In contrast, to avoid the leakage of clients' private data on IoT devices, these private data should be distorted (*e.g.*, encrypted or obfuscated) before the transmission, and the distorted data are largely different from the original one. However, using distorted data as input may result in the severe inaccuracy of tensor completion. Moreover, the solution to this problem must not introduce heavy computation or communication overhead to IoT devices, given the lower-power nature of these devices.

Our solution to this challenging problem is *TwilightTensor*, a novel, lightweight, privacy-preserving tensor completion framework. The basic idea of *TwilightTensor* is to equip each IoT device with a lightweight obfuscation mechanism, so that each device can work in parallel to disguise its private raw data with public shared data retrieved from the server, and only send the obfuscated data to the server (Fig 1). As such, adversaries cannot access devices' private data by tampering the device-server communication or hacking the server. In the meantime, the obfuscation mechanism is carefully designed to maintain the homomorphic property of data. As such, upon receiving the obfuscated data, the server can execute

978-1-6654-3574-1/21/$31.00 ©2021 IEEE
DOI 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00094

a tensor completion algorithm with the obfuscated data as input and distributes the obfuscated results back to the devices individually. Each IoT device can then independently de-obfuscate the data to get the reconstructed complete tensors, and use them to train local learning models (*e.g.*, federated learning [20] and distributed learning [21]). With this novel design, *TwilightTensor* simultaneously protects the private data of IoT devices from leaking, and maintains the accuracy of tensor completion. In addition, the design of *TwilightTensor* is also modular such that different tensor completion algorithms can be plugged in at the server.

The **main** contributions of this paper are as follows:

- We systematically study the new problem of privacy-preserving tensor completion. This problem sheds light on many practical tensor completion use cases, *e.g.*, traffic prediction, health care alerting, and recommendation. To the best of our knowledge, we are the first to systematically investigate and address this problem.
- We design *TwilightTensor*, a novel, lightweight, privacy-preserving tensor completion framework, which allows efficient, accurate tensor completion while protecting the private data of clients from being exposed.
- We fully implement a prototype of *TwilightTensor* and evaluate its performance extensively using real-world datasets for different applications. Results show that *TwilightTensor* can achieve similar recovery accuracy compared with tensor completion algorithms that use un-obfuscated raw data as input, while preserving the private data of clients from being exposed.

II. Motivation

Tensor completion is used to tackle many realistic problems in IoT. For example, many computer vision applications can be formulated as tensor completion problems (*e.g.*, image in-painting [22], video completion [11], and compressed sensing [23]). In addition, applications in intelligent transportation systems [7], social network [24], and mobile medical and health monitoring [25] also rely on tensor completion to reconstruct corrupted data. As such, many tensor completion algorithms [3], [26], [10] are developed.

As shown in **Fig. 1**, a common design paradigm of all these applications is to collect data from different sources to a powerful centralized server, where tensor completion algorithms are executed to reconstruct the missing data. Without any encoding strategies, however, these data can be easily acquired during the data transmission stage, which will lead to privacy leakage, especially for sensitive private data, *e.g.*, GPS location data, medical data, and personal images or videos. This potential drawback will decrease the users' willingness to share their data for tensor completion. Considering the rapid growth of tensor completion and the increasing popularity of IoT services we believe it is crucial to take the privacy problem into consideration and address the privacy issues. Current tensor privacy systems protect the data while do not address the tensor completion problem. These privacy concerns prompt us to study the novel problem of accurate, privacy-preserving tensor completion problem. We design *TwilightTensor*, the first system to simultaneously provide accurate tensor completion and protect tensor privacy in IoT applications.

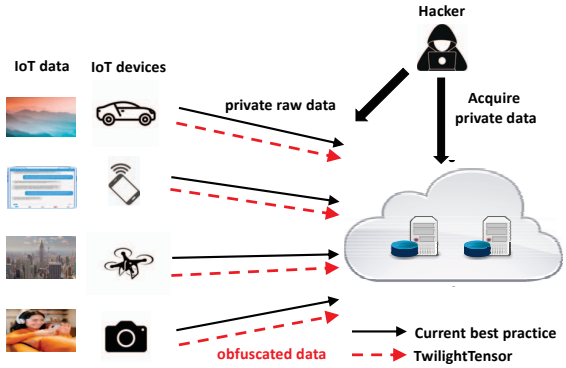


Fig. 1: Current best practice vs. *TwilightTensor*: collecting private, raw data from IoT devices may cause severe privacy issues. *TwilightTensor* addresses them by sending obfuscated data from IoT devices to the server.

III. Problem Formulation

Next, we present the system settings, and then formally define the privacy-preserving tensor completion problem.

A. System Settings

Client/Server system model. We consider a typical client/server model for data collection, process, and analysis. IoT devices are clients that collect both public shared data, *e.g.*, environment temperature, and private data, *e.g.*, the trajectory data. After data are collected, clients need to send the data to a centralized server for processing and learning, due to their limited local computing capability. For private data, the problem of data preservation should be addressed.

Matrix and tensor representation. The data collected by clients can be represented as tensors. We use upper case letters for matrices, *e.g.*, X, and lower case letters for entries, *e.g.*, $x_{i_1 i_2}$, which represents the element in the i_1-th row and the i_2-th column. The Frobenius norm of matrix X is defined as $||X||_F := (\sum_{i_1,i_2} |x_{i_1 i_2}|^2)^{\frac{1}{2}}$. An n-D tensor is defined as $\mathscr{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$. And the elements are denoted as $x_{i_1 i_2 \ldots i_n}$, where $1 \le i_k \le I_k$, $1 \le k \le n$. The unfolding of a tensor $\mathscr{X}$ along the k-mode is defined as $\text{unfold}_k(\mathscr{X}) := \mathscr{X}_k \in \mathbb{R}^{I_k \times (I_1 \times \ldots I_{k-1} \times I_{k+1} \cdots \times I_n)}$. The folding operation is defined as $\text{fold}_k(\mathscr{X}_k) := \mathscr{X}$. The Frobenius norm of tensor $\mathscr{X}$ is defined as $||\mathscr{X}||_F := (\sum_{i_1,i_2,\ldots,i_n} |x_{i_1 i_2 \ldots i_n}|^2)^{\frac{1}{2}}$, or written as $||\mathscr{X}||_*$. We use Ω to indicate the boolean index set of the same size of $\mathscr{X}$. We use $\mathscr{X}_\Omega$ to denote the tensor copying the entries in set Ω from $\mathscr{X}$, while the remaining entries are set to 0. And a **Slice** of an n-D tensor is a 2-D section defined by fixing all but two indices. A **Fiber** of an n-D tensor is a 1-D section defined by fixing all but one. For instance, a third-order tensor of size $n_1 \times n_2 \times n_3$ is denoted as $\mathscr{X}$, and its $(i,j,k)th$ entry is represented as $\mathscr{X}_{ijk}$. Moreover, a tensor tube of size $1 \times 1 \times n_3$ is represented as $\boldsymbol{x}$. And we use the Matlab notion $\mathscr{X}(k,:,:)$, $\mathscr{X}(:,k,:)$ and $\mathscr{X}(:,:,k)$ to denote the k_{th} horizontal, lateral and

frontal slices, and $\mathscr{X}(:,i,j)$, $\mathscr{X}(i,:,j)$ and $\mathscr{X}(i,j,:)$ to denote the $(i,j)_{th}$ mode-1, mode-2 and mode-3 fiber. In particular, we use $\mathscr{X}^{(k)}$ to denote $\mathscr{X}(:,:,k)$. We can view $\mathscr{X}$ as an $n_1 \times n_2$ matrix of tubes.

And the commutative operation $*$ between the tubes $\boldsymbol{a},\boldsymbol{b} \in \mathbb{R}^{1\times 1\times n_3}$ via $\boldsymbol{a}*\boldsymbol{b} = \boldsymbol{a}\circ\boldsymbol{b}$, where $\circ$ denotes the circular convolution between two vectors. And the t-product $\mathscr{T} = \mathscr{A} * \mathscr{B}$ of $\mathscr{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$ and $\mathscr{B} \in \mathbb{R}^{n_2\times n_4\times n_3}$ is a tensor of size $n_1 \times n_4 \times n_3$ where the $(i,j)_{th}$ tube denoted by $\mathscr{T}(i,j,:)$ for $i = 1,2,\ldots,n_1$ and $j = 1,2,\ldots,n_4$ of the tensor $\mathscr{T}$ is given by $\sum_{k=1}^{n_2} \mathscr{A}(i,k,:) * \mathscr{B}(k,j,:)$.

Then we define the tensor transpose. Let $\mathscr{X}$ be a tensor of size $n_1 \times n_2 \times n_3$, and $\mathscr{X}^T$ is the $n_2 \times n_1 \times n_3$ tensor obtained by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through n_3. And we use $\hat{\mathscr{X}}$ to denote the tensor obtained by taking the Fourier Transform (FFT) of all the tubes along the third dimension of $\mathscr{X}$. For details of the computation, please refer to [27]. A tensor is called f-diagonal if each frontal slice of the tensor is a diagonal matrix.

Then, we define the tensor Singular Value Decomposition (t-SVD). For$\mathscr{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$, the t-SVD of $\mathscr{X}$ is given by

$$\mathscr{X} = \mathscr{U} * \mathscr{S} * \mathscr{V}^T, \tag{1}$$

where $\mathscr{U}$ and $\mathscr{V}$ are orthogonal tensors of size $n_1 \times n_1 \times n_3$ and $n_2 \times n_2 \times n_3$. $\mathscr{S}$ is a rectangular f-diagonal tensor of size $n_1 \times n_2 \times n_3$, and $*$ represents the t-product. This decomposition can be computed by SVD in the Fourier domain.

Now we define a measure of tensor complexity based on t-SVD: tensor multi-rank. The multi-rank of $\mathscr{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is a vector $p \in \mathbb{R}^{n_3\times 1}$ with the i_{th} element equal to the rank of the i_{th} frontal slice of $\hat{\mathscr{X}}$ obtained by taking the Fourier transform along the third dimension of the tensor. And the tensor-nuclear-norm (TNN) is denoted as $||\mathscr{X}||_{TNN}$ and defined as the sum of the singular values of all the frontal slices of $\hat{\mathscr{X}}$. And it is the tightest convex relaxation to l_1 norm of the tensor multi-rank. We need to note that $||\mathscr{X}_{TNN}|| = ||blkdiag(\hat{\mathscr{X}})||_*$, and $blkdiag(\hat{\mathscr{X}})$ is a block diagonal matrix.

Low rank tensor. In practical applications, low rank or approximately low rank tensors are broadly observed. For example, in signal processing, [28] evaluates the scheme based on low rank simulated exponential signals. The brain MRI data used in [3] is also approximately low rank. As such, we focus on low rank tensors in this paper.

Tensor completion. Suppose there is a n-D low rank tensor $\mathscr{T}$ with the entries in the set Ω of $\mathscr{T}$ given while the remaining are set to 0. For different tensor completion algorithms, this problem can be formulated in a different way. Here we discuss two methods of formulating the tensor completion problem for Tensor Nuclear Norm (TNN) penalized algorithm and high accuracy low rank tensor completion algorithm.

According to [11], We can solve the following problem to complete this tensor with missing values and produce the output $\mathscr{X}$:

$$\begin{aligned} &\min : ||\mathscr{X}||_{TNN} \\ &s.t. : P_\Omega(\mathscr{X}) = P_\Omega(\mathscr{T}), \end{aligned} \tag{2}$$

where P_Ω is the orthogonal projector onto the span of tensors vanishing outside of Ω. Let $\mathscr{Y}$ be the available data: $\mathscr{Y} = P_\Omega\mathscr{T}$. Then, we define $\mathscr{G} = \zeta_3 P_\Omega \zeta_3^{-1}$ where ζ_3 and ζ_3^{-1} are the operators representing the Fourier and inverse Fourier transform along the third dimension of tensors. Then, we have $\hat{\mathscr{Y}} = \mathscr{G}(\hat{\mathscr{T}})$ where $\hat{\mathscr{Y}}$ and $\hat{\mathscr{T}}$ are the Fourier transforms of $\mathscr{Y}$ and $\mathscr{T}$ along the third mode. Thus, Eq. (2) is equivalent to

$$\begin{aligned} &\min : ||blkdiag(\hat{\mathscr{X}})||_* \\ &s.t. : \hat{\mathscr{Y}} = \mathscr{G}(\hat{\mathscr{X}}), \end{aligned} \tag{3}$$

where $\hat{\mathscr{X}}$ is the Fourier transform of $\mathscr{X}$ along the third mode.

Another way to formulate the problem is to utilize the trace norm. [3] defines the trace norm for tensors

$$||\mathscr{X}||_* := \sum_{i=1}^{n} \alpha_i ||\mathscr{X}_{(i)}||_*, \tag{4}$$

where α_i are constants satisfying $\alpha_i \geq 0,\ 1 \leq i \leq n$ and $\sum_{i=1}^{n} \alpha_i = 1$ for n-D tensor $\mathscr{X}$. The trace norm for tensor is a convex combination of the matrix trace norm of tensor unfolded along each mode. Then [3] proposes the tensor completion optimization as

$$\begin{aligned} &\min_{\mathscr{X}} : \sum_{i=1}^{n} \alpha_i ||\mathscr{X}_{(i)}||_* \\ &s.t. : \mathscr{X}_\Omega = \mathscr{T}_\Omega, \end{aligned} \tag{5}$$

where $\mathscr{X}$ and $\mathscr{T}$ are n-D tensors with identical size, with the entries in the set Ω of $\mathscr{T}$ given while the remaining are 0.

Adversary models. In this paper, we focus on two representative types of adversaries, whose goal is to acquire the complete recovered private data of IoT devices.

- Eavesdroppers and hackers: An eavesdropper is an adversary who intercepts the data transmission process to capture the data traffic between clients and servers and a hacker is an adversary who directly hacks into the servers to obtain the data stored in the server. The data they can acquire is in the same form. Thus, in the following context, they are called by a joint name: hackers.
- Stalkers: A stalker is a special adversary model in trajectory related scenarios. A stalker can tail after the client and acquire k records of the actual data. We assume the number of k is a small number compared with the original data the client owns.

B. Problem Formulation

Performance metrics. We define two metrics to jointly evaluate the performance of our *TwilightTensor* system: recovery error and distortion.

We use root mean squared error $\varepsilon_{(i)}$ to evaluate the recovery accuracy, which is defined as

$$\varepsilon_{(i)} = ||\overline{\mathscr{X}}_{(i)} - \mathbb{X}_{(i)}||_F, \tag{6}$$

where $\overline{\mathscr{X}}_{(i)}$ is the original complete tensor and $\mathbb{X}_{(i)}$ is the recovered tensor of our system for client i. To evaluate the overall recovery performance of our system, we use ε, which

is defined as

$$\varepsilon = \sqrt{\frac{||\overline{\mathscr{X}} - \mathbb{X}||_F^2}{n}}, \tag{7}$$

where $\overline{\mathscr{X}}$ is the original complete tensor, $\mathbb{X}$ is the recovered tensor of our system and n is the number of private users.

We use distortion $\beta_{(i)}$ to describe the variance between the obfuscated data and the original data of client i, which is defined as

$$\beta_{(i)} = ||\overline{\mathscr{X}}_{(i)} - \boldsymbol{\mathscr{X}}_{(i)}||_F, \tag{8}$$

where $\overline{\mathscr{X}}_{(i)}$ is the original complete tensor and $\boldsymbol{\mathscr{X}}_{(i)}$ is the recovered obfuscated tensor for client i. The measurement of distortion is root mean squared error. To evaluate the overall distortion performance of our system, we use β, which is defined as

$$\beta = \sqrt{\frac{||\overline{\mathscr{X}} - \boldsymbol{\mathscr{X}}||_F^2}{n}}, \tag{9}$$

where $\overline{\mathscr{X}}$ is the original complete tensor, $\boldsymbol{\mathscr{X}}$ is the recovered obfuscated tensor and n is the number of private users.

Problem formulation. After defining the performance metrics, we then define The privacy-preserving tensor completion problem (PPTC problem)s as follows:

Definition 1 (PPTC Problem): Design a tensor data collection, completion and encryption mechanism which minimizes $\varepsilon_{(i)}$ while maximizing $\beta_{(i)}$ to accurately recover the missing values in tensors while preserving the privacy for clients.

IV. *TwilightTensor* System

In this section, we present *TwilightTensor*, a novel system that tackles the PPTC problem. We first give an overview of *TwilightTensor*, and then present the details of its components.

A. TwilightTensor Overview

The overview of *TwilightTensor* is shown in **Fig. 2**. Specifically, *TwilightTensor* consists of three key components:

1) **Data obfuscation:** Clients obfuscate their private data with public shared data based on a *K-Tensor Obfuscation* (KTO) mechanism. Then, clients upload their obfuscated data to the server.
2) **Tensor completion:** The server assembles the obfuscated data from clients into a multi-dimensional tensor and applies the tensor completion algorithm to the tensor to estimate the missing values in the tensor. We need to note that the tensor completion module is replaceable. Users of *TwilightTensor* can plug the most suitable tensor completion algorithm for the private data into the tensor completion module.
3) **Data de-obfuscation:** Clients download their corresponding recovered data and de-obfuscate it into the recovered complete one.

B. Data Obfuscation

To tackle the privacy issues, the obfuscation operation of private data is applied at the client side.

We use f_{ob} to denote the obfuscation operation. The goal of f_{ob} is to disguise the original data with public data, so even the adversaries acquire the obfuscated data, they cannot distinguish the original data from the intercepted data. We use $\mathscr{X}_{(i)}$ to denote the original data with missing values and $\mathscr{X}_{(i)}$ for the obfuscated data after obfuscation operation of client i and $\mathscr{X}_{(i)}$ can be obtained by

$$\mathscr{X}_{(i)} = f_{ob}(\mathscr{X}_{(i)}). \tag{10}$$

Then, we explain the details of *K-Tensor Obfuscation* (KTO) mechanism, which serves the obfuscation function in our system. In our system, we assume that the public data are complete without missing values. Client i acquires K public tensors $\mathscr{I}_{(i_1)}, \mathscr{I}_{(i_2)}, \ldots, \mathscr{I}_{(i_K)}$ from server or other public clients. The sources of public tensors are uncertain, which partially strengthens the privacy preservation feature. Client i generates a length of $K+1$ randomly generated vector $< \theta_{i_0}, \theta_{i_1}, \theta_{i_2}, \ldots, \theta_{i_K} >$, where $\theta_{i_j} \in (0,1)$ and $\sum_{j=0}^{K} \theta_{i_j} = 1$, as the private key. The main obfuscation operation of f_{ob} function over $\mathscr{X}_{(i)}$ can be presented by

$$\mathscr{X}_{(i)} = (\theta_{i_0}\mathscr{X}_{(i)} + \theta_{i_1}\mathscr{I}_{(i_1)} + \cdots + \theta_{i_K}\mathscr{I}_{(i_K)}) \circ \Omega, \tag{11}$$

where $\mathscr{X}_{(i)}$ is the original data with missing values, $\mathscr{X}_{(i)}$ is the obfuscated data, Ω is the boolean index set, $\theta_{i_j} \in (0,1)$ and $\sum_{j=0}^{K} \theta_{i_j} = 1$.

The impact of θ_{i_0} is essential in our system. Its value determines the portion of original data in the obfuscated data. The value of θ_{i_0} cannot be too small, otherwise, it will lead to poor recovery accuracy. Meanwhile, the value of θ_{i_0} cannot be too large, otherwise, the obfuscated data will be highly identical to the original one, which makes the obfuscation operation less effective. Empirically, we set $\theta_{i_0} \in [0.2, 0.8]$ to achieve high recovery accuracy and privacy preservation.

C. Tensor Completion

After collecting m obfuscated data from clients, the server forms them into a multi-dimensional tensor with missing values $\mathscr{X} \in \mathbb{R}^{m \times I_1 \times I_2 \times \cdots \times I_{n-1}}$. Then, *TwilightTensor* operates the tensor completion algorithm on the obfuscated tensor. Most existing low rank tensor completion algorithms can be used in the tensor completion module of *TwilightTensor*. The tensor completion algorithm is denoted as f_{TC}. We use $\boldsymbol{\mathscr{X}}$ to denote the recovered tensor after the tensor completion operation

$$\boldsymbol{\mathscr{X}} = f_{TC}(\mathscr{X}), \tag{12}$$

where $\mathscr{X}$ is the obfuscated data with missing values.

The design of *TwilightTensor* is modular so that different tensor completion algorithms can be plugged in. As a proof of concept, in this paper, we choose a High accuracy Low Rank Tensor Completion (HaLRTC) algorithm [3], which leverage the alternating direction method of multipliers (ADMM) framework to solve the tensor completion problem.

HaLRTC Algorithm. In HaLRTC algorithm, additional tensors $\mathscr{M}_1, \ldots, \mathscr{M}_n$ are introduced to obtain the formulation of

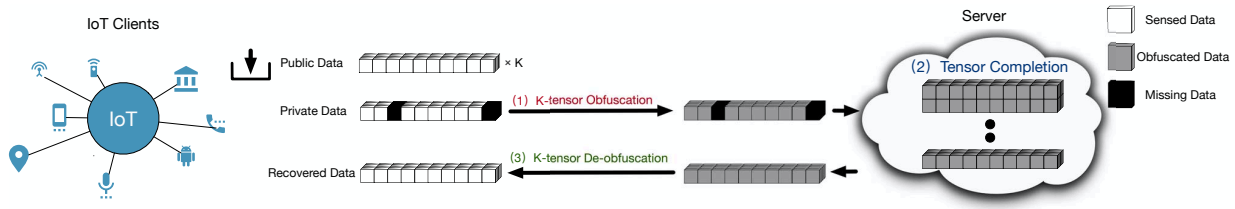


Fig. 2: Illustration of *TwilightTensor* system.

this problem for n-mode tensor:

$$\begin{aligned} &\min_{\hat{\mathcal{X}},\mathcal{M}_1,\ldots,\mathcal{M}_n} : \sum_{i=1}^{n} \alpha_i ||\mathcal{M}_{i(i)}||_* \\ &s.t. : \hat{\mathcal{X}}_\Omega = \mathcal{X}_\Omega \\ &\hat{\mathcal{X}}_{(i)} = \mathcal{M}_i, \text{ for } i = 1,\ldots,n, \end{aligned} \tag{13}$$

where α_i are constants satisfying $\alpha_i \geq 0$, $1 \leq i \leq n$, $\sum_{i=1}^{n} \alpha_i = 1$ and Ω is the binary index tensor. Then, we define the following Lagrangian function according to ADMM formulation:

$$\begin{aligned} &L_\rho(\hat{\mathcal{X}},\mathcal{M}_1,\ldots,\mathcal{M}_n,\mathcal{Y}_1,\ldots,\mathcal{Y}_n) = \\ &\sum_{i=1}^{n} \alpha_i ||\mathcal{M}_{i(i)}||_* + (\hat{\mathcal{X}} - \mathcal{M}_i) \circ \mathcal{Y}_i + \frac{\rho}{2} ||\mathcal{M}_i - \hat{\mathcal{X}}||_F^2, \end{aligned} \tag{14}$$

where ρ is the input of the algorithm. According to the architecture of ADMM, we can iteratively update $\mathcal{M}_i$s, $\hat{\mathcal{X}}$ and $\mathcal{Y}$s as follows:

$$\begin{aligned} &\{\mathcal{M}_1^{k+1},\ldots,\mathcal{M}_n^{k+1}\} = \arg\min_{\mathcal{M}_1,\ldots,\mathcal{M}_n} : L_\rho \\ &= (\hat{\mathcal{X}}^k,\mathcal{M}_1,\ldots,\mathcal{M}_n,\mathcal{Y}_1^{k+1},\ldots,\mathcal{Y}_n^{k+1}), \\ &\hat{\mathcal{X}}^{k+1} = \arg\min_{\hat{\mathcal{X}} \in \mathbb{Q}} : L_\rho \\ &= (\hat{\mathcal{X}},\mathcal{M}_1^{k+1},\ldots,\mathcal{M}_n^{k+1},\mathcal{Y}_1^{k},\ldots,\mathcal{Y}_n^{k}), \\ &\mathcal{Y}_i^{k+1} = \mathcal{Y}_i^k - \rho(\mathcal{M}_i^{k+1} - \hat{\mathcal{X}}^{k+1}). \end{aligned} \tag{15}$$

HaLRTC can be accelerated by changing the value of ρ iteratively. According to [29], we set $\rho^0 = \rho$ and $\rho^{k+1} = t\rho^k$.

D. Data De-obfuscation

After the obfuscated tensor is recovered, clients can download their corresponding data and de-obfuscate it into a complete one and f_{de} to denote the de-obfuscation operation. We use $\mathbb{X}_{(i)}$ to represent the recovered data after de-obfuscation for client i. The de-obfuscation operation over the obfuscated recovered data $\hat{\mathcal{X}}_{(i)}$ to produce $\mathbb{X}_{(i)}$ can be defined as

$$\mathbb{X}_{(i)} = f_{de}(\hat{\mathcal{X}}_{(i)}). \tag{16}$$

The de-obfuscation operation is the inverse of obfuscation operation. Based on the private key and public data, we formulate the de-obfuscation operation as

$$\mathbb{X}_{(i)} = (\hat{\mathcal{X}}_{(i)} - (\theta_{i_1}\mathcal{I}_{(i_1)} + \cdots + \theta_{i_K}\mathcal{I}_{(i_K)}))/\theta_{i_0}. \tag{17}$$

Due to the fact that the private key and the set of public data are only known to clients themselves, their privacy concerns are addressed against the attack from adversaries or latent danger of information leakage.

V. *TwilightTensor* System Analysis

In this section, we analyze the performance of recovery accuracy, privacy preservation and time and space complexity.

A. Accuracy Analysis

In *TwilightTensor*, we adopt obfuscation operation to disguise the private data collected from clients and de-obfuscation operation to decrypt the recovered obfuscated data to produce the final output. From Eq. (11) and Eq. (17), we find that these two operations are linear transformation, which will not change the rank of the tensors. Thus, the rank of the tensors before and after the obfuscation and de-obfuscation operations are identical [30]. Because the essence of the tensor completion is to minimize the tensor-nuclear-norm which is the tightest convex relaxation to l_1 norm of the tensor multi-rank, the obfuscation and de-obfuscation operations do not degrade the recovery accuracy of tensor completion algorithm used in the server due to the consistent tensor rank.

B. Privacy Preservation

From the definition of distortion in Eq. (8), we can observe that the degree of distortion is related to the distance δ between the pair of points in the recovered obfuscated tensor and original complete tensor and a large δ indicates stronger privacy preservation against hackers. The encrypted tensor is obtained by randomly combine K public records with randomly generated vector $< \theta_{i_0}, \theta_{i_1}, \theta_{i_2}, \ldots, \theta_{i_K} >$. Thus, we can use the random distance distribution to approximate the distortion of a pair of nodes.

We consider the random distance between two random points in a $a \times b \times c$. The density function $k(v) = 2vh(v^2)$, where v is the distance and $h(v^2)$ is the density of the probability that $(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 \leq v^2$. The details of $h(v^2)$ can be found in [31]. With this distribution, we simulate in a cube with $a = 4$, $b = 5$, and $c = 6$. The statistic distribution of the density function $k(v)$ is shown in **Fig. 3**. For more generic cases and types of adversary models, we will discuss in Section VI through real-world datasets.

C. Complexity Analysis

Computational Complexity. In *K-Tensor Obfuscation* mechanism, $K+1$ records of data of size T are needed to output an encrypted tensor. The obfuscation and de-obfuscation are linear operations of the whole size of data used. Thus, their time complexities are $O((K+1)T$.

Communication Overhead. The obfuscation and de-obfuscation method operate locally at the user side. To achieve the goal of privacy preservation, users need to download K records of public data of size T from the server and upload the encrypted data in obfuscation operation. Thus, the commu-

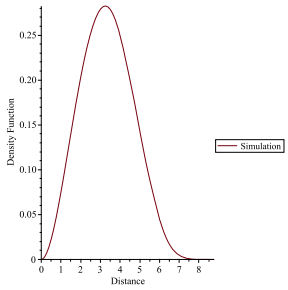


Fig. 3: Density function of the distance.

nication overhead for obfuscation is $O((K+1)T)$. Similarly, for de-obfuscation operation, its communication overhead for de-obfuscation is $O(T)$.

VI. Performance Evaluation

We implement a prototype of *TwilightTensor* and conduct extensive experiments to evaluate its performance by using 2 real-world datasets: visual data, and climate data.

A. Experiment Settings

To simulate the missing values in real datasets, we randomly generate the Ω boolean index set of the same size of our datasets to indicate whether the entry is missing or not. The total number of missing values in tensor is constrained by data loss ratio α. Due to the constraints of limited client number and the aim to simplify the system, we set the top 10 records of data as public data. In this case, the K in *K-tensor obfuscation* is set to 10 by default.

B. Experiment on Visual datasets

Dataset Description. There is a massive amount of applications of IoT services in the area of computer vision. We use the BioID Face Database [32] for analysis and evaluation. The dataset consists of images with a resolution of 384×286 pixels. Each one shows the frontal view of the face of one out of 23 different test persons. We randomly select 100 images to form a tensor with a size of $286 \times 384 \times 100$ to simulate the settings that users will upload pictures to the server for recovery. In **Fig. 4**, we present the cumulative distribution function (CDF) of the top i singular values of the tensor unfolding along the first, the second, and the third dimension. The sum of the singular values can be replaced by several leading singular values. If we remove small singular values of the matrices unfolding along each mode, which is less than 1% of the tensor's Frobenius norm, the rank of the BioID dataset can decrease to $46 \times 57 \times 46$ to satisfy the requirements for low rank tensor completion algorithms.

Performance of Recovery Accuracy. We compare the results of directly applying high accuracy low rank tensor completion (HaLRTC) algorithm [3] and adopting this in the tensor completion module of the *TwilightTensor*. The recovery error CDF of two options is shown in **Fig 5 (a)** with the loss ratio set to 50%. The overall recovery error is 1155.8 and 1239.0, respectively. The overall recovery performance with loss ratio increasing from 10% to 60% by every 10% is shown in **Fig. 5 (b)**. Overall the recovery error in *TwilightTensor* is tolerable for the feature of privacy preservation it can achieve.

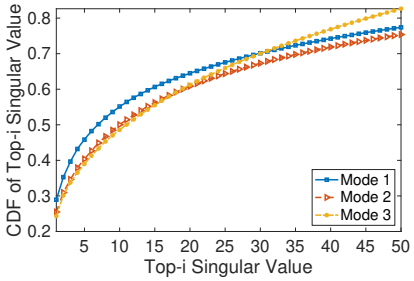


Fig. 4: Low rank property of visual data.

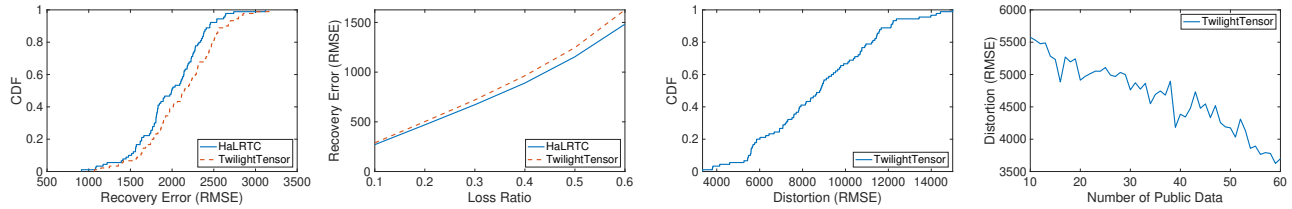
Performance of Privacy Preservation. The CDF of the distortion for each user with the loss ratio set to 50% is shown in **Fig. 5 (c)**. The distortion of over 50% images is greater than 8793.0 and the overall distortion is 5218.6. Then, we adjust the number of public records of data from 10 to 60 and record the overall distortion performance in **Fig. 5 (d)**. We can infer that there is no clear pattern between the number of public records and overall distortion.

Visual Results. We randomly sample a portrait image to give a visual demonstration of *TwilightTensor*, as shown in **Figure 6**. **Fig. 6 (a)** is the ground truth of the original image. **Fig. 6 (b)** is the sampled image with the loss ratio set to 50%. **Fig. 6 (c)** is the output of directly applying the HaLRTC algorithm. **Fig. 6 (d)** is the output of our *TwilightTensor* and **Fig. 6 (e)** is the distorted image after recovery. The results of directly applying the tensor completion algorithm and output of our system are comparable. The obfuscation mechanism also has an effect on privacy preservation.

C. Experiment on Climate dataset

Dataset Description. We also use climate data to validate the efficiency of our system. From the climate data, one can infer the location-based data which will expose the uploaders' private information. Thus, there is still a privacy concern. The climate dataset we use is the U.S. Historical Climatology Network Monthly (USHCN) dataset which consists of monthly climatological data of 108 stations with 17 variables spanning from the year 1915 to 2000. It can form a 3-D tensor with a size of $125 \times 156 \times 17$. The first mode represents 125 locations, the second represents 156 time series, and the third is for 17 variables. In **Fig. 7**, we present the CDF of the top i singular values of tensors unfolding along the first and the second dimension in the USHCN dataset. If we consider the top 75% singular value along each mode, the rank of the tensor can be decreased to $46 \times 64 \times 17$, which meets the requirement of the low rank tensor for our system.

Performance of Recovery Accuracy. For climate data, we only observe the measurements for a subset of locations and time series matrix. For missing data, the 17 variables at one certain location and time frame are blank. We consider the user should upload the climate data of 17 variables along different time frames at one fixed location. When the loss ratio is set to 50%, the recovery error CDF of the HaLRTC algorithm and *TwilightTensor* adopting this algorithm in the tensor completion module on the climate dataset are shown in



(a) Recovery error CDF of visual dataset (b) Recovery error (RMSE) - loss ratio (c) Distortion CDF of visual dataset (d) Distortion (RMSE) - number of public records

Fig. 5: Experimental result of visual dataset.

(a) Ground truth (b) Sampled result (c) Directly recovered (d) *TwilightTensor* result (e) Distorted result

Fig. 6: A demonstration showing that *TwilightTensor* provides accurate tensor completion using obfuscated incomplete tensors.

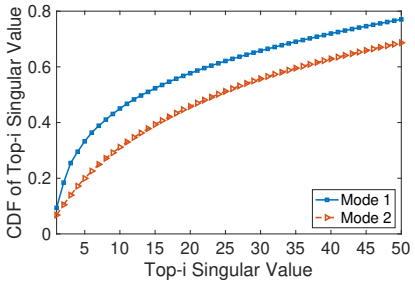


Fig. 7: Low rank property of climate data.

Fig. 8 (a). The overall recovery performance of directly applying HaLRTC and *TwilightTensor* is 14.2 and 14.6 separately. With the loss ratio increasing from 10% to 60% by every 10%, the overall recovery errors of directly applying HaLRTC and *TwilightTensor* are shown in **Fig. 8 (b)**, which are comparable in all the explored cases.

Performance of Privacy Preservation. For the climate dataset, we mainly focus on privacy preservation against hackers. The distortion CDF for each user is shown in **Fig. 8 (c)**. The distortion of *TwilightTensor* of over 80% locations is greater than 23.6 and the overall distortion is 35.0. Considering the overall recovery error of *TwilightTensor* is 14.6, this kind of distortion can protect the information from leakage. Moreover, we apply the *TwilightTensor* system on the climate dataset with the number of public records K varying from 10 to 60. The overall distortion results are shown in **Fig. 8 (d)**. The conclusion is the same as it in the visual dataset and GPS dataset that there is no clear pattern between the number of public records and overall distortion.

VII. Related Work

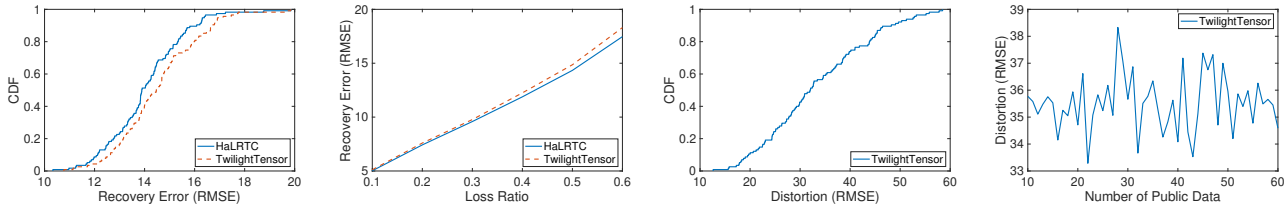
We classify the related work into two categories: *tensor completion* and *tensor privacy preservation*.

Tensor completion. The goal of tensor completion is to accurately estimate the missing data in tensors. Many algorithms for tensor completion have been developed [10], [33], [3], [34]. And [3] applied the alternating direction method of multipliers (ADMM) to develop several tensor completion algorithms built on tensor trace norm. In addition, another approach involved in applying the singular value decomposition on the unfolding matrices along each mode of tensors [33]. To further improve the accuracy and efficiency of tensor completion, algorithms based on tensor-singular value decomposition are recently proposed [26], [11]. [34] gives a comprehensive survey on tensor completion algorithm. Although much progress has been made on improving the accuracy and efficiency of tensor completion, directly applying them to IoT applications [13], [14], [7], [25] would cause privacy concerns.

Tensor privacy preservation. There have been many studies investigating the privacy preservation of tensor data. The common design of these studies preserve the privacy of tensor owners is data encryption. Qiu et al. [8], [18] defined two attack models for cloud servers: direct access model and data mining attack model, and proposed a scheme for tensor-based encryption to balance the privacy and functionality. [35] presented a privacy-preserving High-Order Probabilistic C-Means algorithm (HOPCM), which integrates the Brakerski-Gentry-Vaikuntanathan (BGV) encryption into HOPCM. However, current works on tensor privacy prevent the adversary from getting the original tensor but have no guarantee on tensor completion accuracy [36]. These designs have not investigated the problem of estimating missing data in tensors. On the contrary, we systematically study the privacy-preserving tensor completion problem and design *TwilightTensor*, a novel, modular and efficient system that solves this problem.

VIII. Conclusion

In this paper, we study the novel problem of privacy-preserving tensor completion for IoT applications, and design *TwilightTensor*, a lightweight, privacy-preserving framework



(a) Recovery error CDF of climate dataset (b) Recovery error (RMSE) - loss ratio (c) Distortion CDF of climate dataset (d) Distortion (RMSE) - number of public records

Fig. 8: Experimental results of climate dataset.

that realizes privacy preservation and accurate tensor completion. We implement a prototype of *TwilightTensor* and perform extensive evaluations to demonstrate its efficiency and efficacy for supporting two representative IoT applications.

ACKNOWLEDGEMENT

This work was supported in part NSFC grant 61972253, U1908212, 72061127001, the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning. Linghe Kong is the corresponding author.

REFERENCES

[1] K. Gai and M. Qiu, "Reinforcement learning-based content-centric services in mobile sensing," *IEEE Network*, vol. 32, no. 4, pp. 34–39, 2018.

[2] K. Gai, M. Qiu, H. Zhao, and X. Sun, "Resource management in sustainable cyber-physical systems using heterogeneous cloud computing," *IEEE Transactions on Sustainable Computing*, vol. 3, no. 2, pp. 60–72, 2017.

[3] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, 2013.

[4] M. Zhu, X. Liu, F. Tang, M. Qiu, R. Shen, W. Shu, and M. Wu, "Public vehicles for future urban transportation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3344–3353, 2016.

[5] M. T. Bahadori, Q. R. Yu, and Y. Liu, "Fast multivariate spatio-temporal analysis via low rank tensor learning," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 3491–3499.

[6] J. A. Bazerque, G. Mateos, and G. B. Giannakis, "Rank regularization and bayesian inference for tensor completion and extrapolation," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5689–5703, 2013.

[7] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang, "Collaborative filtering meets mobile recommendation: A user-centered approach," in *AAAI 2010*.

[8] K. Gai and M. Qiu, "Blend arithmetic operations on tensor-based fully homomorphic encryption over real numbers," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3590–3958, 2017.

[9] H. Qiu, M. Qiu, and Z. Lu, "Selective encryption on ECG data in body sensor network based on supervised machine learning," *Information Fusion*, vol. 55, pp. 59–67, 2020.

[10] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, pp. 279–311, 1966.

[11] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. E. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-svd," in *CVPR 2014*.

[12] D. Kotani, "An architecture of a network controller for qos management in home networks with lots of iot devices and services," in *CCNC 2019*, 2019, pp. 1–4.

[13] I. Singh, M. Butkiewicz, H. V. Madhyastha, S. V. Krishnamurthy, and S. Addepalli, "Twitsper: Tweeting privately," *IEEE Security & Privacy*, vol. 11, no. 3, pp. 46–50, 2013.

[14] M. Xia, L. Gong, Y. Lyu, Z. Qi, and X. Liu, "Effective real-time android application auditing," in *IEEE Symposium on Security and Privacy, SP*, 2015, pp. 899–914.

[15] Y. Wang and J. Chen, "Hijacking spoofing attack and defense strategy based on internet tcp sessions," in *IMSNA*, 2013, pp. 507–509.

[16] Y. Li, Y. Song, L. Jia, S. Gao, Q. Li, and M. Qiu, "Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2831–2842, 2021.

[17] H. Qiu, T. Dong, T. Zhang, J. Lu, G. Memmi, and M. Qiu, "Adversarial attacks against network intrusion detection in iot systems," *IEEE Internet of Things Journal*, pp. 1–9, 2021.

[18] K. Gai, Y. Wu, L. Zhu, and M. Qiu, "Privacy-preserving data synchronization using tensor-based fully homomorphic encryption," in *IEEE TrustCom*, 2018, pp. 1149–1156.

[19] J. Feng, L. T. Yang, X. Liu, and R. Zhang, "Privacy-preserving tensor analysis and processing models for wireless internet of things," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 98–103, 2018.

[20] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[21] J. B. Predd, S. B. Kulkarni, and H. V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 56–69, 2006.

[22] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *SIGGRAPH 2000*.

[23] M. F. Duarte and R. G. Baraniuk, "Kronecker compressive sensing," *IEEE Trans. Image Processing*, vol. 21, no. 2, pp. 494–504, 2012.

[24] S. Rendle and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *WSDM 2010*.

[25] J. Dauwels, L. Garg, A. Earnest, and L. K. Pang, "Handling missing data in medical questionnaires using tensor decompositions," in *8th Int'l Conf. on Infor. Comm. Signal Proc.*, 2011, pp. 1–5.

[26] Z. Zhang and S. Aeron, "Exact tensor completion using t-svd," *IEEE Trans. Signal Processing*, vol. 65, no. 6, pp. 1511–1526, 2017.

[27] M. E. Kilmer, K. S. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM J. Matrix Analysis Applications*, vol. 34, no. 1, pp. 148–172, 2013.

[28] J. Ying, H. Lu, Q. Wei, J. Cai, D. Guo, J. Wu, Z. Chen, and X. Qu, "Hankel matrix nuclear norm regularized tensor completion for n-dimensional exponential signals," *IEEE Trans. Signal Processing*, 2017.

[29] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *CoRR*, vol. abs/1009.5055, 2010.

[30] V. De Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1084–1127, 2008.

[31] J. PHILIP, "The probability distribution of the distance between two random points in a box," 03 2020.

[32] "Bioid face database," https://www.bioid.com/facedb/, 2010.

[33] J. Salmi, A. Richter, and V. Koivunen, "Sequential unfolding SVD for tensors with applications in array signal processing," *IEEE Trans. Signal Processing*, vol. 57, no. 12, pp. 4719–4733, 2009.

[34] Q. Song, H. Ge, J. Caverlee, and X. Hu, "Tensor completion algorithms in big data analytics," *CoRR*, vol. abs/1711.10105, 2017.

[35] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "Pphopcm: Privacy-preserving high-order possibilistic c-means algorithm for big data clustering with cloud computing," *IEEE Transactions on Big Data*, pp. 1–1, 2018.

[36] K. Gai, Y. Wu, L. Zhu, Z. Zhang, and M. Qiu, "Differential privacy-based blockchain for industrial internet-of-things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4156–4165, 2019.