NANYANG TECHNOLOGICAL UNIVERSITY

TENSOR COMPUTING FOR BIG DATA ANALYTIC

ONG JENN BING

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2020

TENSOR COMPUTING FOR BIG DATA ANALYTIC

by

ONG JENN BING

Dissertation Supervisor: Assoc. Prof. Ng Wee Keong

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

21 January 2020

. . . . . .
Date

. . . . . .
Ong Jenn Bing

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

23 January 2020

.....
Date

.....
Ng Wee Keong

Authorship Attribution Statement

This thesis contains material from 2 paper(s) published in the following peer-reviewed journal(s) / from papers accepted at conferences / online repository of electronic preprints in which I am listed as an author.

Chapter 2 is published as Jenn-Bing Ong, and Wee-Keong Ng. "Hybrid Subspace Mixture Models for Prediction and Anomaly Detection in High Dimensions." *International Conference on Advanced Data Mining and Applications*. Springer, Cham, 2017.

The contributions of the co-authors are as follows:

- A/Prof Wee-Keong Ng provided the initial project direction, reviewed the manuscript draft, and provided some feedbacks for improvements.
- I prepared the manuscript drafts and carry out the experiments.

1 July 2021

Date

Ong Jenn Bing

Abstract

Tensors have been around since the end of nineteenth century with the development of differential calculus. Tensor research and applications have spanned many areas ranging from mathematics and physics in the early days to psychometrics and linguistics later in around 1960s, and more recently to signal processing and machine learning in the 1990s. Tensor is multi-dimensional extension of matrix, it emerges as important mathematical tool because measurements from experimental studies usually come from more than two dimensions (e.g., space, time, sensors data, and locations), treating these arrays as matrices might lose important structural information such as multi-dimensional correlation structure during data analysis. Tensor decomposition is an established technique for multi-way data analysis such as blind source separation, feature extraction, higher-order statistical analyses, etc. In general, matrix and tensor decomposition have different properties (e.g., uniqueness and interpretability) despite their similarity in concept. Tensor network decomposes higher-order tensors into sparsely-interconnected low-order core tensors, which captures the complex correlation structure in parsimonious manner. Tensor network has been used mostly for data imputation and compressed computation due to the lack of physical interpretability, however, it is promising for big data processing and high-dimensional numerical computing due to their natural support for parallel distributed and compressed computation.

The rationale behind this thesis is to explore new applications of tensor network computing in the big data deep learning era. The contributions of this thesis are summarized as follows: 1) protect big data privacy using randomized tensor network decomposition and dispersed computation, 2) provide theoretical and empirical analysis of adversarial perturbations in deep learning using tensor analysis, and 3) perform blind source separation for human movements sensing using streaming WiFi channel state information.

For big data privacy applications, our primary intuition comes from observing the ability of tensor network to compute wide-range of multi-linear operations using the low-order core tensors without the need to reconstruct the original tensor, which overcomes the curse-of-dimensionality by modeling large parameter space in parsi-

monious manner using the tensor network representations. The tensor distributed computing is implemented in the multi-party computation setting, this enhances the privacy of big data computation by randomized tensor information dispersal. Compared to existing encryptions and data splitting techniques, randomized tensor network computation does not require centralized servers for management and hence completely removes the single point of failure in big data privacy protection.

Adversarial perturbation on deep learning models deviates the inferred output to desired state by the adversary by slightly modifying the input data. Adversarial perturbations have disastrous impact to mission-critical and safety-critical applications such as autonomous vehicles. Many research studies have shown that adversarially-perturbed road signs and adversarial patch can easily cause many deep learning models to misclassification, e.g., detects a speed sign instead of the actual stop sign. Higher-order tensor network decomposition is utilized to provide theoretical analysis of the sensitivity of deep learning models subject to complex correlation structure in the input data, empirical evidence is provided to support the theoretical analysis and adaptive algorithm based on tensor network is proposed to detect strong and static adversarial perturbations.

Human movements are modulated in the surrounding WiFi signals, which can be extracted using data processing and analysis pipeline. Vital sign such as respiration and heartbeat are important predictors of human health status. In this study, we are interested to extract respiration signals from WiFi channel state information (CSI) for occupancy detection and healthcare monitoring. However, the motion dynamics of CSI subcarriers centered at different frequencies cannot be written as linear mixtures of the respiration signals. Thanks to the complementary CSI amplitude and phase information for respiration detection, we propose a complex system made up of the complementary CSI amplitude and phase as the real and imaginary parts, respectively, and model the complex CSI time series of different subcarriers as linear superposition of respiration signals. Stochastic and deterministic separation techniques are then used to extract the respiration source signals from the noisy, multi-modal CSI streams for stationary-person detection and monitoring in quasi-static environments.

Acknowledgements

First and foremost, I would like to thank my supervisor, Assoc. Prof. Ng Wee Keong, for his kindness, trust, and patience throughout the years of my PhD study. The research guidance given by Prof. Ng has important influence to the success of our new discovery. The research environment in NTU SCSE provides us the research freedom to carry out meaningful explorations and experimentations with new inventions or innovations. The regular seminars held by the school and university inspires us with new ideas and expand our core research to stay up-to-date and relevant. I would like to thank Assoc. Prof. Adams Kong Wai Kin, Assoc. Prof. Chen Lihui, and Prof. Cong Gao for being my thesis advisory committee, their valuable feedback and advice have important impact to build up the confidence of our research work from 1st year to the 4th year of my PhD study. I would like to thank the thesis examiners: Prof. Dusit Niyato, Assoc. Prof. Hui Siu Cheung, Asst. Prof. Zhang Tianwei for the insightful comments, questions, and suggestions, which make the PhD thesis examination and oral defence challenging but very exciting.

I am grateful for my family and friends for being there all the time, taking good care of my health and emotions, providing me the joy and happiness in everyday life, and most importantly adding new dimensions to my researcher's life. Special thanks to my wife Ms. Ma Lan for her support and encouragement to my research and entrepreneurial endeavor. Ma Lan's positive attitude and her spirit of never give up have been influential in helping me to expand beyond the research horizon and persist on my startup endeavor.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Tensor computing recently emerges as a promising mathematical technique for big data processing and analytics [1, 6]. As multidimensional generalization of two-dimensional matrices, tensors possess many similar characteristics to techniques based on matrix representation, e.g., dimensionality reduction and constrained optimization, but at the same time provides more flexible and versatile methods in modeling and analyzing disparate data structures such as tabular data and complex networks [7]. Modern areas of data science such as bioinformatics [8, 9] or computational neuroscience [10, 11] generate massive amounts of data collected in various forms of large-scale, sparse tabulars, graphs or networks with multiple aspects and high dimensionality. Tensors, which are multi-dimensional generalizations of matrices, provide often a useful representation for such data. The "flattened view" provided by 2-way component analysis and matrix factorizations may be inappropriate for large classes of real-world data which exhibit multiple couplings and cross-correlations. Higher-order tensor networks provide the opportunity to develop more sophisticated models for capturing multiple interactions and couplings, instead of the standard pairwise interactions provided by matrix representation. Multiway data analysis allows data scientist to take account of the intrinsic multi-dimensional distributed patterns present in the data.

Tensor network (TN) decomposition naturally supports distributed storage and computation on the subtensors; substantial reduction on storage and computational cost can be achieved with low-rank assumption imposed on the subtensors. Tensor

network computing is a well-established technique among the numerical community; the technique provides unprecedented large-scale scientific computing with performance comparable to competing techniques such as sparse-grid methods [12, 13]. Tensor network represents functions in a sparsely-interconnected low-order core tensors and factor matrices and the operators by distributed tensor network operations. It was first discovered in quantum physics in the 1990s, physicists made the first attempt to capture and model the multi-scale interactions among the entangled quantum particles in a parsimonious manner using tensor network and simulate how they evolve over time using a set of dynamical equations [14]. Tensor network was then independently rediscovered in the 2000s by the numerical community and has found wide applications ranging from scientific computing to electronic design automation [15]. Tensor decomposition, as a multidimensional generalization of matrix decomposition, is a decades-old mathematical technique in multiway analysis since the 1960s, see [16] and references therein; tensor techniques are widely applied for signal processing such as blind source separation and multimodal data fusion to machine learning such as model compression and learning latent variable models [17, 18].

The ability of tensor representations and tensor decomposition to work on multimodal data (e.g. linked component analysis or coupled matrix / tensor decomposition for fusion of tabular, graphical, discrete, or continuous data) [19, 20, 21], to deal with different data quality / veracity or incomplete / noisy / inconsistent data (e.g. tensor completion to fill missing data) [22], to provide incremental analysis [23, 24] or real-time analytics such as streaming analytics, and to capture the complex correlation structure in data with large volume and generate valuable insights for many big data distributed applications make it well-suited for big data analytics which is characterized by 4Vs, i.e., Variety, Veracity, Velocity, and Volume. Tensor network computing emerges as a promising solution for big data processing due to its ability to provide large-scale dimensionality reduction and perform parallel, distributed, and compressed computation [1, 6].

1.1 Technical Background

Tensor decomposition has found many applications in signal processing and machine learning. Many review papers have been published throughout the years, more recent and relevant to machine learning and big data applications include [1, 6, 25, 26, 27, 28, 29]. The basic tensor formats and properties are summarized here. Moreover, tensor network computing with distributed tensor operations are described here for big data processing and applications.

Matrix factorization decomposes a data matrix $X_2(i_1, i_2)$ into sum of the unknown source signals in columns of S (latent factors or variables) multiplied by the mixing vectors (factor loadings) in columns of A. Generally, matrix decomposition is defined as follows:

$$X_2(i_1, i_2) \cong \sum_{r=1}^{R} \lambda_r \, A(i_1, r) \, S(i_2, r) \tag{1.1}$$

However, the intrinsic indeterminancies of this model come from the arbitrary scaling of components and permutation of the rank-1 terms. Another indeterminancy comes from the physical meaning of the latent factors, i.e., there are infinitely many combinations of A and S if the model is unconstrained. Standard matrix factorizations such as QR factorizaiton, Eigenvalue Decomposition (EVD), and Singular Value Decomposition (SVD) owe their uniqueness to hard and restrictive constraints such as triangularity and orthogonality. Certain properties of the factors in Equation 1.1 can be represented by appropriate constraints to make unique estimation or extraction of such factors possible. These constraints include statistical independence, sparsity, smoothness, nonnegativity, and uncorrelatedness, which form the foundations for Independent Component Analysis, Sparse Component Analysis, and Nonnegative Matrix Factorization.

Canonical Polyadic (CP) decomposition is one of the most popular tensor technique due to the ease of interpretation. CP is expressed as the sum of rank-1 components or latent factors, CP is defined as follows:

$$X(i_1, \cdots, i_d) \cong \sum_{r=1}^{R} A_1(i_1, r) \, A_2(i_2, r) \cdots A_d(i_d, r), \tag{1.2}$$

where X is a d-dimensional tensor, r is the canonical rank and A_j is the latent factor in scalar representation. Each rank-one component of the decomposition serves as a latent concept or cluster in the data. The latent factors can be interpreted as soft membership to the r-th latent cluster. CP is unique up to scaling and permutation of the r components under very mild conditions, i.e., the components should be "sufficiently different" and their number not unreasonably large. Although CP format bypasses the curse of dimensionality, CP approximation may involve numerical instabilities for very high-order tensors because the problem is generally ill-posed due to intrinsic uncloseness.

Tucker decomposition (TD) captures the interactions between the latent factors A_i using a core tensor G that reflects the main subspace variation in each mode assuming a multilinear structure, TD is defined as follows:

$$X(i_1, \cdots, i_d) \cong \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_d=1}^{R_d} G(r_1, r_2, \cdots, r_d)\, A_1(i_1, r_1)\, A_2(i_2, r_2) \cdots A_d(i_d, r_d) \tag{1.3}$$

TD is non-unique because the latent factors can be rotated without affecting the reconstruction error. TD yields a good low-rank approximation of a tensor, since the core tensor G is the best compression of the original tensor with respect to squared error. The intermediate data explosion problem during Tucker decomposition is well-known and solved in a number of research studies by exploiting sparsity or parallel distributed computing [26]. However, Tucker format is not practical for tensor order $d > 5$ because the number of entries of the core tensor scales exponentially with d, as shown in Table 1.1, therefore storage and computing with Tucker format is not scalable due to the large core tensor when dealing with higher-order tensors [1].

Hierarchical Tucker (HT) decomposition [30, 31] approximates well high-order tensors ($d >> 3$) without suffering from the curse of dimensionality. HT requires a priori knowledge of a binary tree of matricizations of the tensor, HT is defined as

$$\begin{aligned} X(i_1, \cdots, i_d) &\cong \sum_{r_{u_0}=1}^{R_{u_0}} \sum_{r_{v_0}=1}^{R_{v_0}} B_{(12\cdots d)}(r_{u_0}, r_{v_0}) f_{u_0}(i_{u_0}, r_{u_0}) f_{v_0}(i_{v_0}, r_{v_0}) \\ f_t(i_t, r_t) &\cong \sum_{r_u=1}^{R_u} \sum_{r_v=1}^{R_v} B_t(r_u, r_v, r_t) f_u(i_u, r_u) f_v(i_v, r_v) \end{aligned} \tag{1.4}$$

where B_t are "transfer" core tensors (internal nodes), f_u and f_v are the corresponding left and right child nodes respectively. The leaf nodes contain the latent factors. HT is particularly useful when the application provides an intuitive and natural hierarchy over the physical modes of the tensor.

Tensor-Train (TT) decomposition [32, 33] decomposes a given tensor into a matrix, followed by a series of three-mode "transfer" core tensors, and finally ended by a matrix. Each one of the core tensors is "connected" with its neighboring core tensors through a common reduced mode or so-called TT-rank r_k with $r_0 = r_d = 1$. TT is given by Equation 1.5, a variant of TT is called tensor ring with $r_0 = r_d > 1$.

$$X(i_1, \cdots, i_d) \cong \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_{d-1}=1}^{R_{d-1}} G_1(r_0, i_1, r_1) G_2(r_1, i_2, r_2) \cdots G_d(r_{d-1}, i_d, r_d) \quad (1.5)$$

CP and TD are globally-additive models, which mean the tensors are represented by a (global) sum over few separable (rank-1) elements; whereas TT format is locally-multiplicative type, variables only interact directly with few local neighbors (slightly-entangled systems) through the contracted product representations. Table 1.1 tabulates the storage complexity and storage bound of different TN. TT format exhibits both very good numerical properties and allows control of the approximation error within the decomposition algorithm. Mathematical operations in TT format increase the TT-ranks, TT-rounding algorithm, which is mathematically similar to TT-SVD but in TT format, can efficiently reduce the TT-ranks to optimal. Interestingly, any specific TN such as CP and TD can be converted to TT format by reshaping and rearranging the variables.

Decomposition Algorithms [17]: Alternating Least Square (ALS) algorithms update each tensor block successively and iteratively such that the tensor approximation converges to optimal, ALS is a special case of block coordinate descent method. ALS converges at most at local linear rate and therefore ALS is slower than most derivative-based algorithms, it may not converge to stationary point as well. Gradient descent is a well-known optimization strategy for various tensor models, e.g., Newton descent uses a local quadratic approximation of the cost function to obtain a new updating step. As computation of the Hessian matrix may be prohibitively expensive, several approximation schemes lead to a number of Quasi-Newton and

Nonlinear Least Squares optimization techniques. Exact line search seeks to select the optimal step-size after finding the update ("search") direction, e.g., the negative gradient, sometimes through exploiting the multi-linearity of the cost function. Stochastic gradient descent deals with missing data naturally / effortlessly by computing gradient estimates from the observed values only. Another approach is to use expectation-maximization to impute the missing values together with estimation of the model parameters, but imputation is very inefficient in terms of memory for very big and sparse data and is thus avoided. Imposing constraints in tensor network decomposition improves estimation accuracy, ensuring interpretability and identifiability of the problem. Non-parametric constraints such as non-negativity, orthogonality, smoothness, probability simplex, and linear constraints can be formulated as parametric constraints, which can be conveniently handled by derivative-based optimizations such as Quasi-Newton and Non-linear Least Squares.

Scalable algorithms for CP and TD decomposition are well established and can be categorized into compression, exploiting sparsity, sampling, and parallel / distributed computation [26]. Randomized mapping or projection utilize a projection matrix like Gaussian, Rademacher, or random orthonormal matrices to project the data tensor to smaller size prior to the tensor network decomposition process [34]. Randomized sampling techniques such as fiber subset selection or tensor cross approximation choose a smaller subset of tensor fibers that approximate the entire data tensor well for decomposition [34]. Existing randomized mapping / projection and sampling algorithms may be utilized for big data reduction to fit into memory the data tensor for tensor network decomposition, the tensor blocks can be compressed with lossy tensor compression [34]. Tensor sketching (or randomized mapping) using Tucker model is a promising technique to analyze big data efficiently [35, 36]. HT and TT algorithms involve sequential matricization and singular value decomposition (SVD), for scalability, CUR decomposition / cross-approximation / pseudo-skeleton can be performed on big matrices by sampling rows and columns with statistically large influence on the best low-rank fit of the data [37, 38, 39, 40].

TN	**Storage Complexity**	**Storage Bound**
CP	$\sum_{k=1}^{d} I_k R$	$O(dIR)$
TD	$\sum_{k=1}^{d} I_k R_k + \prod_{k=1}^{d} R_k$	$O(dIR + R^d)$
HT	$\sum_{k=1}^{d} I_k R_k + \sum_{u,v,t} R_u R_v R_t$	$O(dIR + dR^3)$
TT	$\sum_{k=1}^{d} I_k R_{k-1} R_k$	$O(dIR^2)$

Table 1.1: Storage complexity of TN [1]. d is the tensor order, I_k and R_k are the size and rank of mode k respectively. The storage bound is calculated by letting $I = \max_k I_k$ and $R = \max_k R_k$ for all possible k in particular TN.

Tensor networks can be represented by a set of shapes or nodes interconnected by the edges. The edges correspond to the contracted modes, whereas lines that do not go from one tensor to another correspond to open (physical) modes, which contribute to the order of the entire tensor network. Fig. 1.1 shows the graphical representations of different tensor networks. Mathematical operations can be expressed using graphical representation of tensors (e.g., tensor contractions and reshaping) in a simple and intuitive way without the explicit use of complex mathematical expressions. For some very high-order data tensors it has been observed that the ranks of the cores increase rapidly with the order of the tensor for any choice of tensor network. For such cases, the Projected Entangled-Pair State (PEPS) [41] or the Multi-scale Entanglement Renormalization Ansatz (MERA) [42] tensor networks with cycles and hierarchical structures can be used. The main advantage of tensor networks with loops is that the size of each core tensor in the internal tensor network structure is usually much smaller than the cores in TT/HT decompositions, additionally they allow us to model more complex functions and interactions between variables, especially MERA. However, it should be noted that the contraction of the resulting tensor network is more computationally expensive and involves approximations due to the fact that the PEPS and MERA tensor networks contain loops and therefore do not have closed-form solution [1, 6].

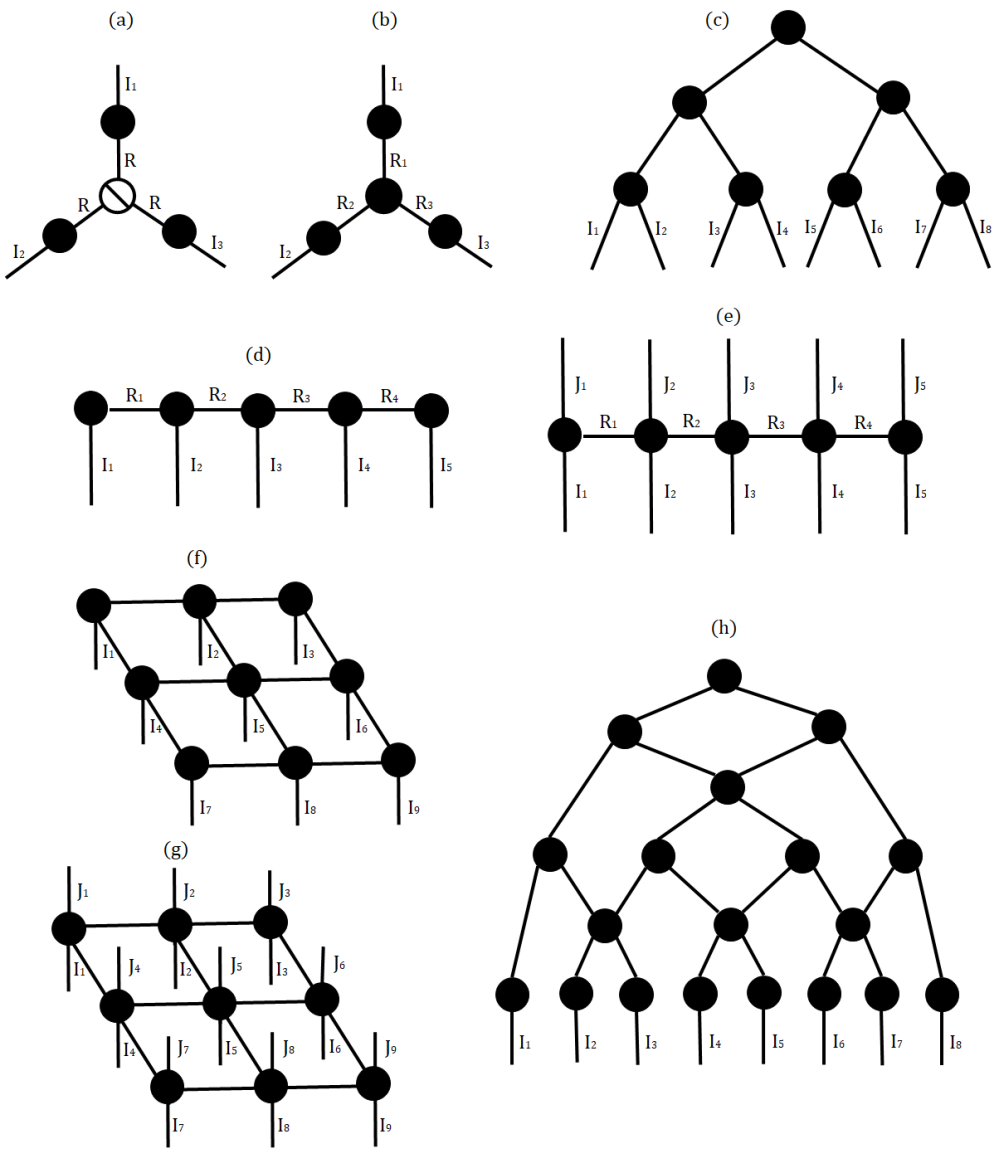

Figure 1.1: Graphical representations of different tensor decompositions. The number of edges connected to a node indicates the order of the tensor, and the mode size is labeled on each edge. (a) Canonical Polyadic (CP) decomposition with a superdiagonal core, (b) Tucker decomposition (TD) that captures all the interactions between the latent factors, (c) Hierarchical Tucker (HT), (d) vector Tensor-Train (TT), and (e) matrix TT formats. HT and TT formats break the curse of dimensionality and suitable for big data processing. More sophisticated tensor networks with loops include (f/g) Projected Entangled-Pair State/Operator (PEPS/PEPO) and (h) Multiscale Entanglement Renormalization Ansatz (MERA), which may have smaller cores compared to other tensor networks but higher cost for contractions.

1.1.1 Basic Distributed Tensor Operations

Tensor network naturally supports distributed / dispersed computation using the smaller tensor blocks after big data decomposition [34, 43, 4]. Figure 1.2 summarizes the basic notations for tensors [4]; Figures 1.3 and 1.4 show some of the basic tensor operations [4], these notations and operations allow us to present the distributed tensor operations in concise manner. For example, CP and Tucker decompositions in Equations 1.2 and 1.3 can be conveniently expressed in the form of multilinear product as

$$\underline{X} = \underline{\Lambda} \times_1 A^{(1)} \times_2 A^{(2)} \ldots \times_N A^{(N)} = [\![\underline{\Lambda};\ A^{(1)},\ A^{(2)}, \ldots,\ A^{(N)}]\!] \tag{1.6}$$

$$\underline{X} = \underline{G} \times_1 A^{(1)} \times_2 A^{(2)} \ldots \times_N A^{(N)} = [\![\underline{G};\ A^{(1)},\ A^{(2)}, \ldots,\ A^{(N)}]\!] \tag{1.7}$$

where $\underline{\Lambda} \in \mathbb{R}^{R \times \cdots \times R}$ is the superdiagonal tensor with diagonals $\lambda_1, \ldots, \lambda_R$, $\underline{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots R_N}$ is the core tensor, and $A^{(n)} \in \mathbb{R}^{I_n \times R_n}$ are the factor matrices. Tensor multilinear operations can be performed in distributed / dispersed manner in tensor network formats. For example, some basic operations in Tucker format are shown as follows. Let $\underline{A} = [\![\underline{G}_A; \underline{A}^{(1)}, \underline{A}^{(2)}, \ldots, \underline{A}^{(N)}]\!]$ and $\underline{B} = [\![\underline{G}_B; \underline{B}^{(1)}, \underline{B}^{(2)}, \ldots, \underline{B}^{(N)}]\!]$,

$$\begin{aligned}
&(a)\ \underline{A} + \underline{B} = [\![\underline{G}_A \oplus \underline{G}_B\ ; \underline{A}^{(1)} \boxplus \underline{B}^{(1)}, \ldots,\ \underline{A}^{(N)} \boxplus \underline{B}^{(N)}]\!] \\
&(b)\ \underline{A} \oplus \underline{B} = [\![\underline{G}_A \oplus \underline{G}_B\ ; \underline{A}^{(1)} \oplus \underline{B}^{(1)}, \ldots,\ \underline{A}^{(N)} \oplus \underline{B}^{(N)}]\!] \\
&(c)\ \underline{A} \circledast \underline{B} = [\![\underline{G}_A \otimes \underline{G}_B\ ; \underline{A}^{(1)} \boxtimes \underline{B}^{(1)}, \ldots,\ \underline{A}^{(N)} \boxtimes \underline{B}^{(N)}]\!] \\
&(d)\ \underline{A} \otimes \underline{B} = [\![\underline{G}_A \otimes \underline{G}_B\ ; \underline{A}^{(1)} \otimes \underline{B}^{(1)}, \ldots,\ \underline{A}^{(N)} \otimes \underline{B}^{(N)}]\!]
\end{aligned} \tag{1.8}$$

Figure 1.5 shows the multilinear operations that can be performed for data tensors represented in TT format [4]. As shown in Figure 1.6, tensor network operations such as TT can be performed naturally in distributed (and compressed) manner, making it well-suited for big data processing and scientific computing. Indeed, there is already intensive research within the numerical computing community to develop new tensor network computing algorithms to overcome the curse-of-dimensionality in solving linear systems such as a set of partial differential equations with huge parameter space to explore and find the optimal solutions [15]. For big data processing, several huge-scale optimization problems have been considered such as tensor com-

Notation	Description
$\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$	Nth-order tensor of size $I_1 \times I_2 \times \cdots \times I_N$
$x, \mathbf{x}, \mathbf{X}$	Scalar, vector, and matrix
$x_{i_1,i_2,\ldots,i_N}, \underline{\mathbf{X}}(i_1, i_2, \ldots, i_N)$	$(i_1, i_2, \ldots, i_N)$th entry of $\underline{\mathbf{X}}$
$\mathbf{x}_{:,i_2,i_3,\ldots,i_N}, \underline{\mathbf{X}}(:, i_2, i_3, \ldots, i_N)$	Mode-1 fiber of $\underline{\mathbf{X}}$
$\mathbf{X}_{:,:,i_3,i_4,\ldots,i_N}, \underline{\mathbf{X}}(:, :, i_3, i_4, \ldots, i_N)$	Frontal slice of $\underline{\mathbf{X}}$
$\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N}$	Mode-n unfolding of $\underline{\mathbf{X}}$
$\mathbf{X}_{<n>} \in \mathbb{R}^{I_1 I_2 \cdots I_n \times I_{n+1} \cdots I_N}$	Mode-$(1, 2, \ldots, n)$ unfolding of $\underline{\mathbf{X}}$
$\underline{\mathbf{G}}, \underline{\mathbf{G}}^{(n)}, \underline{\mathbf{X}}^{(n)}, \underline{\mathbf{A}}^{(n)}$	Core tensors and factor matrices/tensors
R, R_n	Ranks
$\overline{i_1 i_2 \cdots i_N}$	Multi-index, $i_N + (i_{N-1} - 1) I_N + \cdots + (i_1 - 1) I_2 I_3 \cdots I_N$

Figure 1.2: Notations for tensors [3].

pletion, Riemannian optimization, linear and quadratic programming, eigenvalue or singular value decomposition, and sparse principal component analysis [34, 43].

1.2 Tensor Network Software Libraries

A number of tensor network software libraries have been developed over the years, a comprehensive list can be found on the open-source tensor network website (`http://tensornetwork.org/software/`). Most of the software libraries are developed for academic research, only a handful are built for production environments. In particular, software libraries like TensorNetwork, TensorLy, T3f, Quimb, Tntorch, and TorchMPS are built with machine learning / differentiable programming backends such as Tensorflow and Pytorch. Tensor network operations performed in these libraries can be easily accelerated using GPUs, multi-core, and parallel distributed computing. The backpropagation of deep learning models compressed in tensor network formats can be calculated using automatic differentiation built into the backends. Other software libraries including Scikit-TT, TeNPy, Mpnum, Ttpy, TT-toolbox, Tensorlab, Tensor Toolbox, N-way Toolbox, SPLATT, TDALab, iTensor, Uni10 are built for tensor network with different network topology, software platforms / environments, and applications such as structured data fusion, complex optimization in numerical computing, and modeling and simulation studies in

Notation	Description
$\underline{\mathbf{C}} = \underline{\mathbf{A}} \otimes \underline{\mathbf{B}}$	Kronecker product of $\underline{\mathbf{A}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and $\underline{\mathbf{B}} \in \mathbb{R}^{J_1 \times \cdots \times J_N}$ yields a tensor $\underline{\mathbf{C}}$ of size $I_1 J_1 \times \cdots \times I_N J_N$ with entries $\underline{\mathbf{C}}(\overline{i_1 j_1}, \ldots, \overline{i_N j_N}) = \underline{\mathbf{A}}(i_1, \ldots, i_N)\underline{\mathbf{B}}(j_1, \ldots, j_N)$
$\underline{\mathbf{C}} = \underline{\mathbf{A}} \circledast \underline{\mathbf{B}}$	Hadamard (elementwise) product of $\underline{\mathbf{A}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and $\underline{\mathbf{B}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ yields a tensor $\underline{\mathbf{C}}$ of size $I_1 \times \cdots \times I_N$ with entries $\underline{\mathbf{C}}(i_1, \ldots, i_N) = \underline{\mathbf{A}}(i_1, \ldots, i_N)\underline{\mathbf{B}}(i_1, \ldots, i_N)$
$\underline{\mathbf{C}} = \underline{\mathbf{A}} \circ \underline{\mathbf{B}}$	Outer product of $\underline{\mathbf{A}} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ and $\underline{\mathbf{B}} \in \mathbb{R}^{J_1 \times \cdots \times J_N}$ yields a tensor $\underline{\mathbf{C}}$ of size $I_1 \times \cdots \times I_M \times J_1 \times \cdots \times J_N$ with entries $\underline{\mathbf{C}}(i_1, \ldots, i_M, j_1, \ldots, j_N) = \underline{\mathbf{A}}(i_1, \ldots, i_M)\underline{\mathbf{B}}(j_1, \ldots, j_N)$
$\underline{\mathbf{C}} = \underline{\mathbf{A}} \oplus \underline{\mathbf{B}}$	Direct sum of $\underline{\mathbf{A}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and $\underline{\mathbf{B}} \in \mathbb{R}^{J_1 \times \cdots \times J_N}$ yields a tensor $\underline{\mathbf{C}}$ of size $(I_1 + J_1) \times \cdots \times (I_N + J_N)$ with entries $\underline{\mathbf{C}}(k_1, \ldots, k_N) = \underline{\mathbf{A}}(k_1, \ldots, k_N)$ if $1 \leq k_n \leq I_n$ $\forall n$, $\underline{\mathbf{C}}(k_1, \ldots, k_N) = \underline{\mathbf{B}}(k_1 - I_1, \ldots, k_N - I_N)$ if $I_n < k_n \leq I_n + J_n$ $\forall n$, and $\underline{\mathbf{C}}(k_1, \ldots, k_N) = 0$ otherwise
$\underline{\mathbf{C}} = \underline{\mathbf{A}} \times_n \mathbf{B}$	Mode-n product of tensor $\underline{\mathbf{A}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and matrix $\mathbf{B} \in \mathbb{R}^{J \times I_n}$ yields a tensor $\underline{\mathbf{C}}$ of size $I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$ with mode-n fibers $\underline{\mathbf{C}}(i_1, \ldots, i_{n-1}, :, i_{n+1}, \ldots, i_N) = \mathbf{B}\underline{\mathbf{A}}(i_1, \ldots, i_{n-1}, :, i_{n+1}, \ldots, i_N)$
$\underline{\mathbf{C}} = \underline{\mathbf{A}} \bar{\times}_n \mathbf{b}$	Mode-n (vector) product of tensor $\underline{\mathbf{A}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and vector $\mathbf{b} \in \mathbb{R}^{I_n}$ yields a tensor $\underline{\mathbf{C}}$ of size $I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N$ with entries $\underline{\mathbf{C}}(i_1, \ldots, i_{n-1}, i_{n+1}, \ldots, i_N) = \mathbf{b}^\top \underline{\mathbf{A}}(i_1, \ldots, i_{n-1}, :, i_{n+1}, \ldots, i_N)$
$\mathbf{C} = \mathbf{A} \,\vert\otimes\vert\, \mathbf{B}$	Strong Kronecker product of block matrices $\mathbf{A} = [\mathbf{A}_{r_1,r_2}] \in \mathbb{R}^{R_1 I_1 \times R_2 I_2}$ and $\mathbf{B} = [\mathbf{B}_{r_2,r_3}] \in \mathbb{R}^{R_2 J_1 \times R_3 J_2}$ yields a block matrix $\mathbf{C} = [\mathbf{C}_{r_1,r_3}] \in \mathbb{R}^{R_1 I_1 J_1 \times R_3 I_2 J_2}$ with blocks $\mathbf{C}_{r_1,r_3} = \sum_{r_2=1}^{R_2} \mathbf{A}_{r_1,r_2} \otimes \mathbf{B}_{r_2,r_3}$
$\underline{\mathbf{C}} = \underline{\mathbf{A}} \,\vert\otimes\vert\, \underline{\mathbf{B}}$	Strong Kronecker product of block tensors $\underline{\mathbf{A}} = [\underline{\mathbf{A}}_{r_1,r_2}] \in \mathbb{R}^{R_1 I_1 \times R_2 I_2 \times I_3}$ and $\underline{\mathbf{B}} = [\underline{\mathbf{B}}_{r_2,r_3}] \in \mathbb{R}^{R_2 J_1 \times R_3 J_2 \times J_3}$ yields a block tensor $\underline{\mathbf{C}} = [\underline{\mathbf{C}}_{r_1,r_3}] \in \mathbb{R}^{R_1 I_1 J_1 \times R_3 I_2 J_2 \times I_3 J_3}$ with blocks $\underline{\mathbf{C}}_{r_1,r_3} = \sum_{r_2=1}^{R_2} \underline{\mathbf{A}}_{r_1,r_2} \otimes \underline{\mathbf{B}}_{r_2,r_3}$

Figure 1.3: Notations and definitions for tensor operations [3].

Notation	Description
$\underline{\mathbf{C}} = \underline{\mathbf{A}} \boxtimes \underline{\mathbf{B}}$	Partial Kronecker product of factor tensors $\underline{\mathbf{A}} \in \mathbb{R}^{R_1 \times \cdots \times R_L \times I_1 \times \cdots \times I_N \times R_{L+1} \times \cdots \times R_M}$ and $\underline{\mathbf{B}} \in \mathbb{R}^{S_1 \times \cdots \times S_L \times I_1 \times \cdots \times I_N \times S_{L+1} \times \cdots \times S_M}$ yields a tensor $\underline{\mathbf{C}}$ of size $R_1 S_1 \times \cdots \times R_L S_L \times I_1 \times \cdots \times I_N \times R_{L+1} S_{L+1} \times \cdots \times R_M S_M$ with subtensors $\underline{\mathbf{C}}(:, \ldots, :, i_1, \ldots, i_N, :, \ldots, :) = \underline{\mathbf{A}}(:, \ldots, :, i_1, \ldots, i_N, :, \ldots, :) \otimes \underline{\mathbf{B}}(:, \ldots, :, i_1, \ldots, i_N, :, \ldots, :)$
$\underline{\mathbf{C}} = \underline{\mathbf{A}} \boxplus \underline{\mathbf{B}}$	Partial direct sum of factor tensors $\underline{\mathbf{A}} \in \mathbb{R}^{R_1 \times \cdots \times R_L \times I_1 \times \cdots \times I_N \times R_{L+1} \times \cdots \times R_M}$ and $\underline{\mathbf{B}} \in \mathbb{R}^{S_1 \times \cdots \times S_L \times I_1 \times \cdots \times I_N \times S_{L+1} \times \cdots \times S_M}$ yields a tensor $\underline{\mathbf{C}}$ of size $(R_1 + S_1) \times \cdots \times (R_L + S_L) \times I_1 \times \cdots \times I_N \times (R_{L+1} + S_{L+1}) \times \cdots \times (R_M + S_M)$ with subtensors $\underline{\mathbf{C}}(:, \ldots, :, i_1, \ldots, i_N, :, \ldots, :) = \underline{\mathbf{A}}(:, \ldots, :, i_1, \ldots, i_N, :, \ldots, :) \oplus \underline{\mathbf{B}}(:, \ldots, :, i_1, \ldots, i_N, :, \ldots, :)$
$\left[\!\left[\mathbf{G}; \underline{\mathbf{A}}^{(1)}, \ldots, \underline{\mathbf{A}}^{(N)} \right]\!\right]$	Multilinear operator for tensors $\mathbf{G}$ and $\underline{\mathbf{A}}^{(n)}, n = 1, \ldots, N$
$\underline{\mathbf{C}} = \underline{\mathbf{A}} \times^1 \underline{\mathbf{B}}$	Mode-$(M, 1)$ contracted product of tensors $\underline{\mathbf{A}} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ and $\underline{\mathbf{B}} \in \mathbb{R}^{J_1 \times \cdots \times J_N}$ with $I_M = J_1$ yields a tensor $\underline{\mathbf{C}}$ of size $I_1 \times \cdots \times I_{M-1} \times J_2 \times \cdots \times J_N$ with entries $\underline{\mathbf{C}}(i_1, \ldots, i_{M-1}, j_2, \ldots, j_N) = \sum_{i_M=1}^{I_M} \underline{\mathbf{A}}(i_1, \ldots, i_M) \underline{\mathbf{B}}(i_M, j_2, \ldots, j_N)$
$\underline{\mathbf{C}} = \underline{\mathbf{A}} \,\vert\bullet\vert\, \underline{\mathbf{B}}$	Core contracted product (C-product) of block tensors $\underline{\mathbf{A}} = [\underline{\mathbf{A}}_{r_1, r_2}]$ and $\underline{\mathbf{B}} = [\underline{\mathbf{B}}_{s_1, s_2}]$ yields a block tensor $\underline{\mathbf{C}} = [\underline{\mathbf{C}}_{t_1, t_2}]$ with blocks $\underline{\mathbf{C}}_{t_1, t_2} = \underline{\mathbf{A}}_{r_1, r_2} \times^1 \underline{\mathbf{B}}_{s_1, s_2}$ for $t_1 = \overline{r_1 s_1}$, $t_2 = \overline{r_2 s_2}$
$Tr\left(\underline{\mathbf{X}}\right)$	Partial trace operator $Tr : \mathbb{R}^{R \times I_1 \times \cdots \times I_N \times R} \to \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined by $Tr(\underline{\mathbf{X}}) = \sum_{r=1}^{R} \underline{\mathbf{X}}(r, i_1, \ldots, i_N, r)$

Figure 1.4: Notations and definitions for tensor operations (continued) [3].

Operation	TT-cores
TT (global)*	
$\underline{\mathbf{Z}} = \underline{\mathbf{X}} + \underline{\mathbf{Y}} = \left(\underline{\mathbf{X}}^{(1)} \boxplus \underline{\mathbf{Y}}^{(1)}\right) \times^1 \left(\underline{\mathbf{X}}^{(2)} \boxplus \underline{\mathbf{Y}}^{(2)}\right) \times^1 \cdots \times^1 \left(\underline{\mathbf{X}}^{(N)} \boxplus \underline{\mathbf{Y}}^{(N)}\right)$	$(I_n = J_n \, \forall n)$
Ten	$\underline{\mathbf{Z}}^{(n)} = \underline{\mathbf{X}}^{(n)} \boxplus \underline{\mathbf{Y}}^{(n)}$
Mat	$\mathbf{Z}_{i_n}^{(n)} = \mathbf{X}_{i_n}^{(n)} \oplus \mathbf{Y}_{i_n}^{(n)}$
Vec	$\mathbf{z}_{s_{n-1},s_n}^{(n)} = \begin{cases} \mathbf{x}_{s_{n-1},s_n}^{(n)} & \text{for } 1 \le s_n \le R_n^X \\ \mathbf{y}_{s_{n-1}-R_{n-1}^X, s_n - R_n^X}^{(n)} & \text{for } R_n^X < s_n \le R_n^X + R_n^Y \\ \mathbf{0}_{I_n} & \text{otherwise} \end{cases}$
$\underline{\mathbf{Z}} = \underline{\mathbf{X}} \oplus \underline{\mathbf{Y}} = \left(\underline{\mathbf{X}}^{(1)} \oplus \underline{\mathbf{Y}}^{(1)}\right) \times^1 \left(\underline{\mathbf{X}}^{(2)} \oplus \underline{\mathbf{Y}}^{(2)}\right) \times^1 \cdots \times^1 \left(\underline{\mathbf{X}}^{(N)} \oplus \underline{\mathbf{Y}}^{(N)}\right)$	
Ten	$\underline{\mathbf{Z}}^{(n)} = \underline{\mathbf{X}}^{(n)} \oplus \underline{\mathbf{Y}}^{(n)}$
Mat	$\mathbf{Z}_{k_n}^{(n)} = \begin{cases} \mathbf{X}_{k_n}^{(n)} \oplus \mathbf{0}_{R_{n-1}^Y \times R_n^Y} & \text{for } 1 \le k_n \le I_n \\ \mathbf{0}_{R_{n-1}^X \times R_n^X} \oplus \mathbf{Y}_{k_n - I_n}^{(n)} & \text{for } I_n < k_n \le I_n + J_n \end{cases}$
Vec	$\mathbf{z}_{s_{n-1},s_n}^{(n)} = \begin{cases} \mathbf{x}_{s_{n-1},s_n}^{(n)} \oplus \mathbf{0}_{J_n} & \text{for } 1 \le s_n \le R_n^X \\ \mathbf{0}_{I_n} \oplus \mathbf{y}_{s_{n-1}-R_{n-1}^X, s_n - R_n^X}^{(n)} & \text{for } R_n^X < s_n \le R_n^X + R_n^Y \\ \mathbf{0}_{I_n + J_n} & \text{otherwise} \end{cases}$
$\underline{\mathbf{Z}} = \underline{\mathbf{X}} \circledast \underline{\mathbf{Y}} = \left(\underline{\mathbf{X}}^{(1)} \boxtimes \underline{\mathbf{Y}}^{(1)}\right) \times^1 \left(\underline{\mathbf{X}}^{(2)} \boxtimes \underline{\mathbf{Y}}^{(2)}\right) \times^1 \cdots \times^1 \left(\underline{\mathbf{X}}^{(N)} \boxtimes \underline{\mathbf{Y}}^{(N)}\right)$	$(I_n = J_n \, \forall n)$
Ten	$\underline{\mathbf{Z}}^{(n)} = \underline{\mathbf{X}}^{(n)} \boxtimes \underline{\mathbf{Y}}^{(n)}$
Mat	$\mathbf{Z}_{i_n}^{(n)} = \mathbf{X}_{i_n}^{(n)} \otimes \mathbf{Y}_{i_n}^{(n)}$
Vec	$\mathbf{z}_{s_{n-1},s_n}^{(n)} = \mathbf{x}_{r_{n-1}^X, r_n^X}^{(n)} \circledast \mathbf{y}_{r_{n-1}^Y, r_n^Y}^{(n)} \quad (s_n = \overline{r_n^X r_n^Y})$
$\underline{\mathbf{Z}} = \underline{\mathbf{X}} \otimes \underline{\mathbf{Y}} = \left(\underline{\mathbf{X}}^{(1)} \otimes \underline{\mathbf{Y}}^{(1)}\right) \times^1 \left(\underline{\mathbf{X}}^{(2)} \otimes \underline{\mathbf{Y}}^{(2)}\right) \times^1 \cdots \times^1 \left(\underline{\mathbf{X}}^{(N)} \otimes \underline{\mathbf{Y}}^{(N)}\right)$	
Ten	$\underline{\mathbf{Z}}^{(n)} = \underline{\mathbf{X}}^{(n)} \otimes \underline{\mathbf{Y}}^{(n)}$
Mat	$\mathbf{Z}_{k_n}^{(n)} = \mathbf{X}_{i_n}^{(n)} \otimes \mathbf{Y}_{j_n}^{(n)} \quad (k_n = \overline{i_n j_n})$
Vec	$\mathbf{z}_{s_{n-1},s_n}^{(n)} = \mathbf{x}_{r_{n-1}^X, r_n^X}^{(n)} \otimes \mathbf{y}_{r_{n-1}^Y, r_n^Y}^{(n)} \quad (s_n = \overline{r_n^X r_n^Y})$
$\underline{\mathbf{Z}} = \underline{\mathbf{X}} \circ \underline{\mathbf{Y}} = \underline{\mathbf{X}}^{(1)} \times^1 \cdots \times^1 \underline{\mathbf{X}}^{(N)} \times^1 \underline{\mathbf{Y}}^{(1)} \times^1 \cdots \times^1 \underline{\mathbf{Y}}^{(N)}$	
Ten	$\underline{\mathbf{Z}}^{(n)} = \underline{\mathbf{X}}^{(n)} \; (n \le N); \underline{\mathbf{Y}}^{(n-N)} \; (n > N)$
Mat	$\mathbf{Z}_{i_n}^{(n)} = \mathbf{X}_{i_n}^{(n)} \; (n \le N); \mathbf{Y}_{i_n}^{(n-N)} \; (n > N)$
Vec	$\mathbf{z}_{s_{n-1},s_n}^{(n)} = \mathbf{x}_{s_{n-1},s_n}^{(n)} \; (n \le N); \mathbf{y}_{s_{n-1},s_n}^{(n-N)} \; (n > N)$
$\underline{\mathbf{Z}} = \underline{\mathbf{X}} \times_n \mathbf{A} = \underline{\mathbf{X}}^{(1)} \times^1 \cdots \times^1 \underline{\mathbf{X}}^{(n-1)} \times^1 \left(\underline{\mathbf{X}}^{(n)} \times_2 \mathbf{A}\right) \times^1 \underline{\mathbf{X}}^{(n+1)} \times^1 \cdots \times^1 \underline{\mathbf{X}}^{(N)}$	
Ten	$\underline{\mathbf{Z}}^{(m)} = \underline{\mathbf{X}}^{(m)} \; (m \ne n); \underline{\mathbf{X}}^{(m)} \times_2 \mathbf{A} \; (m = n)$
Mat	$\mathbf{Z}_{i_m}^{(m)} = \mathbf{X}_{i_m}^{(m)} \; (m \ne n); \underline{\mathbf{X}}^{(m)} \bar{\times}_2 \mathbf{a}_{i_m,:} \; (m = n)$
Vec	$\mathbf{z}_{s_{m-1},s_m}^{(m)} = \mathbf{x}_{s_{m-1},s_m}^{(m)} \; (m \ne n); \mathbf{A}\mathbf{x}_{s_{m-1},s_m}^{(m)} \; (m = n)$

Figure 1.5: Basic operations on tensors represented by TT format [3].

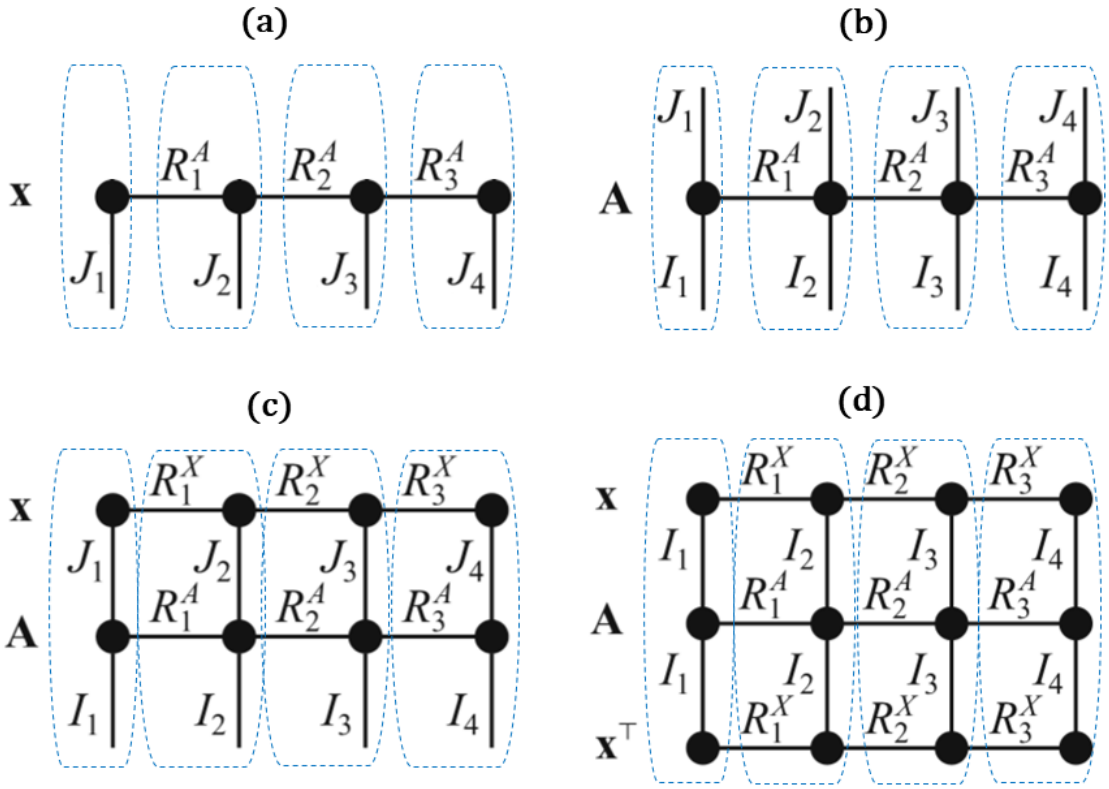

Figure 1.6: Tensor network diagrams of (a) a matrix $A \in \mathbb{R}^{I_1 I_2 I_3 I_4 \times J_1 J_2 J_3 J_4}$ in matrix TT format, (b) matrix-by-vector product $y = Ax$, both A and x are in TT formats, (c) quadratic form $x^T Ax$ when $I_n = J_n$ [3]. The dashed / dotted blue boxes show each of the tensor blocks and operations that can be stored and computed in distributed manner.

quantum physics, quantum chemistry, electronic design automation, or advanced materials modeling.

1.3 Research Areas

Based on the literature survey in Chapter 2, our plan is to further develop tensor methods to be integrated in the production line for different software applications, the research areas of this thesis are given as follows.

- *Big data machine learning or deep learning* are part of the core components to inform artificial intelligence (AI). The success of machine learning depends heavily on the learned representations. To speed up learning process, knowledge learned from one domain can be transferred to closely-related domain. The ultimate quest of machine intelligence is to have an end-to-end pipeline for structural prediction, reasoning, and planning. Major progress in artificial intelligence will come about through comprehensive systems that combine representation learning with complex reasoning, it would be interesting to find more tensor network analysis and applications towards building general AI.

- *Privacy and security* issues of tensor-based computing software is not well studied, only a few research studies are found, e.g., using homomorphic encryption scheme [44, 45] for tensor decomposition by cloud computing resources and applying differential privacy [46] for online tensor decomposition to prevent person identification in tensor analysis. Exploring tensor networks as an alternative data structure for efficient and distributed storage as well as privacy preservation for input data of machine learning models have not be been considered before; our study is particularly relevant in the context of edge and fog computing by exploiting the data redundancy and uses tensor network decomposition to retain the relevant features for image classification.

- *Software engineering for tensor computing.* Tensor-based softwares are commonly used and studied in academic research, however, tensor operations are not optimized and widely tested for production environments. Therefore, it is

imperative to find commercial applications, develop and deploy tensor-based computation software in production environments.

1.4 Thesis Contributions

There are three major contributions made in this thesis:

- We propose a theory based on subspace analysis using Singular Value Decomposition (SVD) for adversarial perturbations in deep learning models, we believe the sensitivity analysis of deep learning models is generally valid for signal processing applications. The complex correlation structure is primarily due to the high dimensionality of Big training Data that results in the sensitivity of deep learning models subject to input perturbations. We provide empirical evidence for different datasets / models and propose tensor network algorithm to mitigate adversarial attacks on convolutional neural networks.

- We propose a arithmetic secret sharing based on randomized tensor network decomposition, we analyze the privacy leakage for different types of data, e.g., image, audio, sensors, graph, and textual data (Variety of Big Data). We further present a randomized incremental tensor updating scheme that caters for High Velocity of Big Data. The computational and communication complexity are compared to classical secret-sharing schemes.

- We propose a complex system made up of the complementary CSI amplitude and phase as the real and imaginary parts, respectively, and model the complex CSI time series as linear mixtures of respiration signals. The blind source separation problem is formulated by tensorization of the CSI signals and solved via tensor decompositions. We design and prototype a system using commodity 5GHz WiFi devices with 40MHz bandwidth and show that reasonable performance can be achieved for respiration detection.

1.5 Organization of Thesis

Chapter 1 starts by providing some preliminary background on tensor network, motivation, and contribution of this thesis. Chapter 2 provides the literature survey. Chapter 3 uses principal component analysis (PCA) to reduce the dimensions of high-dimensional data for hybrid mixture models' analysis, PCA is closely related to the concept of singular value decomposition (SVD), whereas SVD forms the foundation for matrix decomposition and higher-order tensor decomposition thanks to its efficiency. Chapter 4 uses tensor network to compress neural network models and input data, we benchmark the storage and computational efficiency for different datasets. Chapter 3 and 4 serve as the starting points to the research in tensor networks. In chapter 5, a theory for adversarial perturbations of neural networks is proposed using higher-order tensor analysis of data complexity commonly found in training inputs. Chapter 6 further proposes a novel secret-sharing scheme based on randomized tensor network for big data privacy preservation, the commercialization potential of the proposed approach is presented in Appendix A. Blind source separation based on tensor decompositions are utilized in Chapter 7 to extract the respiration signals from WiFi channel state information. Chapter 8 concludes this thesis and provides the future work.

Chapter 2

Literature Survey

There are several reviews on tensor decomposition and applications published throughout the years, they span a number of research fields including quantum physics, quantum information theory, quantum computing, and algebraic statistics. Here, we review the related tensor studies and applications in computer science and engineering that tackle big data processing and analytics.

Signal processing, data mining, and machine learning. Tensor decomposition has been widely used in signal processing and data mining for extracting and explaining their underlying properties or intrinsic relationships [47]. Traditional research on matrix and tensor decomposition focuses on the uniqueness or separabililty of latent components to allow physical interpretability [27, 17]. Tensor usages include blind source separation, feature extraction [48, 49], noise reduction, missing data imputation [50], etc. and has wide applications in many areas [26] due to its ability to capture the correlation structure in multi-aspect / multi-view, heterogeneous data tensors. Tensor decomposition has been used for feature extraction and classification on multidimensional data [48], recent work is extended to high-dimensional data with cutting-edge tensor techniques such as tensor-train decomposition [51, 49]. Tensor-based recommender models improve traditional collaborative filtering techniques by taking into account a multifaceted nature of real environments, which allows to produce more accurate, situational (e.g. context-aware, criteria-driven) recommendations [52]. Coupled tensor decomposition [20, 19] and linked component [21, 53] analysis have been used for multi-modal data fusion to simultaneously extract com-

mon and individual latent components with desired properties and types of diversity. A deep multi-task representation learning framework based on tensor decomposition learns cross-task sharing structure at every layer in a deep network [54], this is in contrast to existing approaches that require a user-defined multi-task sharing strategy. Computationally and statistically efficient parameter estimation of a wide class of latent variable models including Gaussian mixture models, hidden Markov models, and latent Dirichlet allocation can be done by exploiting a certain tensor structure in their low-order observable moments, such as using a robust tensor power method proposed in [28]. Theoretical links have been slowly established between tensor network structure and machine learning techniques, e.g., the expressive power of CP and HT decompositions correspond to shallow and deep networks respectively [55, 56], whereas TT corresponds to recurrent neural network [57] and hidden markov models, PEPS corresponds to markov random field / conditional random field, MERA corresponds to deep belief networks, etc [43]. The theoretical relationships help to characterize the expressive power of different deep learning architectures and shed light on novel model design [43]. Recent research studies further make use of tensor network optimization algorithms for machine learning, e.g., supervised learning using tensor-train [58] and multi-scale tensor networks [59]. Tensor networks naturally support compressed and distributed computation, many large-scale optimization problems have been studied using tensor network computing such as solving system of linear equations, nonlinear functions and system, dynamical and parameteric problems [1, 6]. Tensor networks have also been used to compress and accelerate machine learning and deep learning models due to the high redundancy in model parameters, this is especially true for low-power mobile applications, e.g., fully-connected network [60], convolutional neural network [61, 62, 63, 64], recurrent network [65], sharing residual units [66], multitask learning [54], multimodal learning [67, 68].

Tensor-structured numerical methods in scientific computing. The use of tensor-structured data formats was recognized as the basic concept for breaking the curse of dimensionality in multidimensional numerical simulations and stochastic / parametric partial differential equations [13]. The guiding principle of the tensor methods is

an approximation of multivariate functions and operators relying on a certain separation of variables and keeping the computational process in a low parametric tensor-structured manifold [69]. Novel quantized tensor approximation method (QTT) provides function-operator calculus in higher dimensions in logarithmic complexity rendering super-fast convolution, FFT and wavelet transforms [12]. Algorithms for full tensors cannot be used for very large parameter space because of the high memory and computational requirements, therefore techniques have been developed to decompose incomplete tensor efficiently using compressed sensing [70]. Tensor network computing has been widely applied to many numerical computing problems such as electronic design automation [71], finite element method, and boundary element method modeling [13]. Tensor numerical computing can be broadly categorized into two classes, the first class is based on combining classical iterative algorithm with repeated low-rank truncation, the second class is based on reformulating the numerical problem as an optimization problem, constraining the admissible set to low-rank tensor formats, and applying various optimization techniques [15].

Software engineering, communication, networking, and tensor-based computation. Due to the versatility of tensor representations, tensor techniques have been proposed for big data networking and management [72, 73, 74, 75, 76], e.g., Internet of Things [77, 78]. A unified tensor model has been proposed to represent the unstructured, semi-structured, and structured data in order to extract the smaller core set for software-defined big data networking [76, 78, 73, 74]. Research issues regarding tensor computing for Internet of things have also been discussed in [77]. A big data-as-a-service framework based on tensor network decomposition has been proposed in [79] for cyber-physical-social system, the framework proposes incremental high-order SVD to extract high-quality core tensor in the sensing plane, the distributed tensor representations are then integrated via incremental join and union methods in the cloud plane before mapped into specific applications such as mining, learning, and recommendation. Tensor-based big-data-driven routing recommendation approach has also been proposed for heterogeneous networks [80]. In this framework, a tensor-based, holistic, hierarchical approach is introduced to generate efficient routing paths using tensor decomposition; a tensor matching method includ-

ing the controlling tensor, seed tensor, and orchestration tensor is employed to realize routing recommendation [80]. Tensor network has been used for the identification of high-order discrete-time nonlinear multiple-input multiple-output (MIMO) Volterra systems, the system identification problem is rewritten in terms of a Volterra tensor in tensor network format, which is never explicitly constructed during computation, thus avoiding the curse of dimensionality [81, 82]. Several other studies also employ tensor network to improve symbol/channel estimation in MIMO relay communication systems [83, 84, 85, 86, 87]. The first tensor algebra compiler technique was introduced in [88] to automatically generate kernels for any compound tensor operation on dense and sparse tensors. This technique is implemented in a C++ library called taco, the performance is competitive with best-in-class hand-optimized kernels in popular libraries while support far more tensor operations [88]. Large-scale tensor decomposition usually exploits the sparsity in the data tensor, data sampling, and parallel distributed computing such as Hadoop MapReduce, multi-core CPUs and GPUs hardware acceleration [89, 90, 91, 92, 93, 94].

Chapter 3

Hybrid Subspace Mixture for Prediction and Anomaly Detection in High Dimensions

Robust learning of mixture models in high dimensions remains an open challenge and especially so in current big data era. This chapter investigates twelve variants of hybrid mixture models that combine the G-means clustering, Gaussian, and Student t-distribution mixture models for high-dimensional predictive modeling and anomaly detection. High-dimensional data is first reduced to lower-dimensional subspace using whitened principal component analysis. For real-time data processing in batch mode, a technique based on Gram-Schmidt orthogonalization process is proposed and demonstrated to update the reduced dimensions to remain relevant in fulfilling the task objectives. In addition, a model-adaptation technique is proposed and demonstrated for big data incremental learning by statistically matching the mixture components' mean and variance vectors; the adapted parameters are computed based on weighted average that takes into account the sample size of new and older statistics with a parameter to scale down the influence of older statistics in each iterative computation. The hybrid models' performance are evaluated using

The work in this chapter has been published in the *International Conference on Advanced Data Mining and Applications.* Springer, Cham, 2017.

simulation and empirical studies. Results show that simple hybrid models without the Expectation-Maximization training step can achieve equally high performance in high dimensions that is comparable to the more sophisticated models. For unsupervised anomaly detection, the hybrid models achieve detection rate $\gtrsim 90\%$ with injected anomalies from 1% to 60% using the KDD Cup 1999 network intrusion dataset.

3.1 Introduction

Mixture models have been widely used in many applications such as speaker verification, background subtraction for real-time tracking, and biological applications. Efficient algorithm to learn mixture of Gaussians in high dimensions with small error bound has recently been demonstrated [95]. However, practical algorithms for robust and adaptive learning of high-dimensional mixture models is still an open challenge [96]. This chapter extends the work of a robust subspace mixture model initially developed by [97] for anomaly detection to predictive modeling in high dimensions. The work by [97] provides a way for parameter rating for explainability / interpretability of the model output. Their proposed models pre-process the data using principal component analysis for dimensionality reduction and diffusion-map-based coarse filtering, estimate the robust mixture model parameters using Student-t distribution Mixture Model (SMM) and Expectation Maximization (EM), and finally output the result using Gaussian mixture model (GMM) [97]. Their model parameters estimation is adaptive to handle streaming data [97]. Our motivation is to study, develop, and investigate hybrid subspace mixture models more extensively for robust and adaptive prediction / anomaly detection in high dimensions. High-dimensional data is reduced to lower dimensional subspace using whitened principal component analysis. Diffusion Map (DM)-based coarse-filtering technique is robust to noise perturbation [98] and Student-t distribution Mixture Model (SMM) is robust to outliers [99], both provide a robust statistics of the model developed by [97]. The estimated SMM parameters are then used to form a Gaussian Mixture Model (GMM) statistics for predictive density estimation to ensure robustness and sensi-

tivity to outliers. This chapter aims to further investigate and improve the model performance and computing efficiency for high-dimensional predictive modeling and anomaly detection. The contributions of this chapter are as follows:

- Twelve variants of hybrid mixture models that combine G-means clustering or K-Means clustering using Gaussian algorithm (KM) developed by [100], GMM, and SMM have been compared for predictive modeling and anomaly detection. Results show that simple hybrid models without the Expectation-Maximization (EM) step can achieve equally high prediction accuracy and anomaly detection rates comparable to the sophisticated models.

- For unsupervised anomaly detection, the noise can be removed by a DM-based coarse-filtering technique developed by [97]. However, without the coarse filtering, the hybrid models achieve detection rate $\gtrsim 90\%$ with injected anomalies from 1% to 60% in the KDD Cup 1999 network intrusion dataset. The top-down approach produces results that do not fluctuate with data sampling in contrast to the models using the DM-based coarse-filtering technique that process data in smaller chunks.

- For real-time batch data processing, a technique based on Gram-Schmidt orthogonalization process is proposed and demonstrated to update the reduced dimensions to remain relevant in fulfilling the task objectives. Existing work usually assumes same reduced dimensions for each batch of data or assumes spherical-Gaussian covariance so that the covariance remains conserved after re-projection from one set of dimension vectors to another.

- A model-adaptation technique is proposed and tested for incremental learning of GMM and SMM model parameters. This technique is different from previous [97, 101, 102] in that the adapted model parameters are computed by taking account the data size of new and older statistics, and a parameter is introduced in the technique to scale down the influence of older statistics in each iterative computation.

The organization of this chapter is as follows. Data pre-processing for dimensional-

ity reduction and coarse filtering are provided in Section 3.2. The hybrid mixture models, model adaptation, and the parameter rating techniques are covered in Section 3.3. Section 3.4 evaluates the model performance by simulation / experimental studies and Section 3.5 discusses some of the findings of this work.

3.2 Data Pre-processing

Dimensionality reduction is first applied on high-dimensional data to extract the relevant features for mixture modeling. For anomaly detection, diffusion map-based coarse filtering is used to remove the anomalies for a more robust parameters estimation for the mixture models.

3.2.1 Dimensionality Reduction using whitened PCA

Principal component analysis (PCA) is a linear mapping from a high-dimensional space to a subspace that captures the most variability in the data specified by a set of orthogonal / principal components (PCs). To extract the relevant components from different datasets, different number of PCs with the highest eigenvalues are tested to find the minimum required for better model performance. For batch processing, it is important to ensure that these minimum number of PCs are sufficient for each batch of data to fulfill particular objective; e.g., predictive modeling or anomaly detection. Suppose there exists additional PCs in new batch of data which are not spanned by the older set of PCs, the new dimensions can be appended by using the Gram-Schmidt orthogonalization process to remove the projections on older set of PCs. On the other hand, older PC may be discarded if the absolute value of the Pearson correlation with the set of new PCs is low (< 0.5). The threshold can be determined from empirical experiments to ensure good model performance. The reason this updating technique is proposed because the projection of non-spherical Gaussian covariance from a set of orthonormal vectors to another is not conserved, therefore the PCs can only be appended or discarded, but not re-projected, during the updating process.

3.2.2 Diffusion Map-based Coarse Filtering

Diffusion Map (DM) is a non-linear technique that helps to discover the underlying manifold of high-dimensional data [98]. Given a set of d-dimensional data X, the similarity measure between two data points is defined as

$$\chi(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{\epsilon}\right) \tag{3.1}$$

where $||\bullet||$ is the Euclidean distance of the vectors in the ambient space $\mathbb{R}^d$. The scaling parameter is computed by the average smallest neighbouring distance [103].

$$\epsilon = \frac{1}{N}\sum_{i=1}^{N} \min_{j:x_i \neq x_j} ||x_i - x_j||^2 \tag{3.2}$$

The transition probability between two data points can be computed by normalizing Equation 3.1. The diffusion distance is small if there are many short paths connecting two data points, which implies large transition probability between the two points. DM provides a representation in which the data points are clustered according to their connectivity, which is robust to noise perturbations [98]. In the diffusion space, inliers are expected to be clustered together, whereas outliers might be spread between several small clusters or scattered randomly. The inliers can then be identified by discovering the biggest connected component in the diffusion space using technique proposed in [97].

The transition probability between vectors x_i and x_j is then defined by normalizing the similarity measure in Equation 3.1

$$p(x_i, x_j) = \frac{\chi(x_i, x_j)}{\sum_{h=1}^{K} \chi(x_i, x_h)} \tag{3.3}$$

The transition probability matrix after t time steps is given by

$$P^t_{i,j} = \prod_{t'=1}^{t} p(x_i, x_j) \tag{3.4}$$

and the diffusion distance between vectors x_i and x_j is

$$Q_t(x_i, x_j)^2 = \sum_{\tilde{k}=1}^{\tilde{K}} \left(P^t_{i,\tilde{k}} - P^t_{j,\tilde{k}}\right)^2 \tag{3.5}$$

Q_t is small if there are many short paths connecting x_i and x_j, which implies large transition probability between the two points. Note that the diffusion distance is robust to noise perturbations because it involves all paths of length t connecting x_i and x_j. Applying spectral decomposition, $Pv_i = \lambda_i v_i$ and define a mapping $M_t : \{x_{\tilde{k}}\}_{\tilde{k}=1}^{\tilde{K}} \rightarrow D$,

$$M_t(x_{\tilde{k}}) = |\lambda_1^t v_{1\tilde{k}}, \lambda_2^t v_{2\tilde{k}}, ..., \lambda_\ell^t v_{\ell\tilde{k}}]^T \tag{3.6}$$

where D has dimension $\ell << d$. It can be shown that for $\ell = d - 1$ [Lafon 2004]

$$\begin{aligned} Q_t(x_i, x_j) &= ||M_t(x_i) - M_t(x_j)|| \\ &= \left(\sum_{h=1}^{l} \lambda_h^{2t} (v_{hi} - v_{hj})^2 \right)^{\frac{1}{2}} \end{aligned} \tag{3.7}$$

By applying DM to the data points $X = \{x_{\tilde{k}}\}_{\tilde{k}=1}^{\tilde{K}}$, we obtain the embedding $Y = \{y_{\tilde{k}}\}_{\tilde{k}=1}^{\tilde{K}}$, where $y_{\tilde{k}} = M_t(x_{\tilde{k}})$. Define a graph on Y where the vertices are Y and the adjacency matrix is $A_{ij} = \mathbf{1}_{a(i,j)>\tau}$, where $\mathbf{1}$ is an indicator function with respect to the condition $a(i,j) > \tau$,

$$\begin{aligned} a(i,j) &= \frac{\left|\left| ||y_i - y_j|| - \mu_\zeta \right|\right|}{\sigma_\zeta} \ , \ \delta_{\tilde{k}\zeta} = \sum_{v=1}^{\zeta} ||y_k - y_{Y_{\tilde{k}v}}|| \\ \mu_\zeta &= \tilde{K}^{-1} \sum_{\tilde{k}=1}^{\tilde{K}} \delta_{\tilde{k}\zeta} \ , \ \sigma_\zeta = \sqrt{\tilde{K}^{-1} \sum_{\tilde{k}=1}^{\tilde{K}} (\delta_{\tilde{k}\zeta} - \mu_\zeta)^2} \end{aligned} \tag{3.8}$$

where $Y \in \{1, 2, ..., \tilde{K}\}^{\tilde{K} \times \zeta}$ is the Euclidean nearest neighbour matrix in which the entry $Y_{\tilde{k}v}$ contains the index of the v nearest neighbour of $y_{\tilde{k}}$. ζ and τ are predetermined constants. Then, the breadth-first-search algorithm is applied on Y in order to reveal the biggest connected component in the graph $Y_b \subset Y$ and Y_b is considered as a set of inliers [97]. Notice that $a(i,j)$ is large when the distance between two data points is much smaller or much bigger than the average neighbouring distance μ_ζ. With appropriate τ value, the adjacency matrix A_{ij} captures the connections between inliers from within or different mixture components.

3.3 Hybrid Mixture Models

This section explains the methods to estimate the hybrid models' parameters, some of the algorithms can be found in [97]. Table 3.1 tabulates the sequence of data processing of the hybrid mixture models. The acronym for each hybrid model follows the sequence of data processing. For example, the sequence of KEG model is (1) KM (2) EM (3) GMM. For anomaly detection, the data is first coarse-filtered with a technique based on diffusion map described in Section 3.2.2 to remove anomalies before model training. GMM and SMM parameters are estimated using the EM algorithms described in [104] and [99] respectively. The EM initialization is provided by the K-means clustering using Gaussian algorithm (KM) developed by [100] that repeatedly splits every clusters until each approximates the Gaussian distribution statistically. The statistical test, which is based on the one-dimensional Anderson-Darling statistics, is valid for multidimensional Gaussian distribution. In addition, the mixture model parameters can also be learnt using KM directly and is theoretically shown to require near-optimal sample requirement with well-separated mixture components [105]. The reason this simple model is explored because EM algorithm often converges to local minimum in high dimensions. For computing efficiency, variance vectors are used in the mixture modeling instead of full covariance matrices. For KESG, the estimated SMM parameters are used to form GMM statistics; while for KEGS, the estimated GMM parameters form the SMM statistics assuming the degree of freedom is 1, this is justifiable because real data usually spreads out. Similar applies to other models. In addition, a model-adaptation technique is introduced for incremental learning of big data and the computation of a parameter rating technique developed by [97] is presented here in order to examine the source of anomalies occurrences in the original feature space.

Table 3.1: Sequence of data processing of the proposed hybrid mixture models. The acronym for each hybrid model follows the sequence of data processing. For example, the sequence for DKEGS model is (1) DM (2) KM (3) EM (4) GMM (5) SMM.

Models	DM	KM	EM	GMM	SMM	Remark
KG		1		2		
KS		1			2	
KEG		1	2	3		Prediction and
KES		1	2		3	Anomaly Detection
KESG		1	2	4	3	
KEGS		1	2	3	4	
DKG	1	2		3		
DKS	1	2			3	
DKEG	1	2	3	4		Anomaly Detection
DKES	1	2	3		4	
DKESG	1	2	3	5	4	
DKEGS	1	2	3	4	5	

3.3.1 Model Adaptation

Box's M test is used to statistically compare the sample variance from new and older statistics. After finding a match of a pair of mixture components' variance, Hotelling's T^2 test is used to compare and match the corresponding sample mean. The adapted parameters are estimated using Maximum A-Posteriori (MAP) estimation. Similar model adaptation technique has been developed for GMM by [101]. Our proposed technique differs from previous [97, 101, 102] in that the adapted parameters are computed by taking account the sample size of new and older statistics, and a parameter $f^{\rho}(\mathbf{c})$ is introduced in the technique to scale down the influence of older statistics in the iterative computation. Equation 3.9 summarizes the model adaptation for both GMM and SMM. Although Box's M test and Hotelling's T^2 test

assume the pair of mixture components are multidimensional Gaussian-distributed, short of other alternatives, these statistical tests provide a more stringent criteria for matching SMM mixture components. Additionally, the EM algorithm to estimate the mixture model parameters does not guerantee to find a global optimum since the problem is non-convex and the final solutions depend on the initial parameter values. Therefore, a technique to combine the statistics of a mixture of parametric models for predictive density estimation is proposed in [97]. The technique can be easily parallelized in the expense of computational resources due to model independence [97]. Our model-adaptation technique can also be used to merge the model parameters estimated from multiple trial estimation on a given dataset, this saves the memory space from storing duplicate parameters from different trials.

$$\begin{aligned} Mixing\ coefficient &: \tilde{\omega}_i = \big(\alpha_i^{\omega}\omega_i^{new} + (1-\alpha_i^{\omega})\omega_i\big)\gamma \\ Sample\ mean &: \tilde{\mu}_i = \alpha_i^{\mu}\mu_i^{new} + (1-\alpha_i^{\mu})\mu_i \\ Sample\ variance &: \tilde{\sigma}_i^2 = \alpha_i^{\sigma}\big((\sigma_i^{new})^2 + (\mu_i^{new})^2\big) + (1-\alpha_i^{\sigma})(\sigma_i^2+\mu_i^2) - \tilde{\mu}^2 \\ Degree\ of\ freedom &: \tilde{v}_i = \alpha_i^{v}v_i^{new} + (1-\alpha_i^{v})v_i \end{aligned} \tag{3.9}$$

where γ is a normalization factor which ensures the adapted weights sum to unity, α_i^{ρ} is the data-dependent adaptation coefficient that are computed by $\alpha_i^{\rho} = \frac{n_i^{new}}{n_i^{new}+f^{\rho}(\mathbf{c})n_i}$, where $\rho \in \{\omega, \mu, \sigma, v\}$, $n_i^{new} = N^{new}\omega_i^{new}$ and $n_i = N\omega_i$ is the sample size estimates of the i-th mixture component, $f^{\rho}(\mathbf{c})$ is a function of the context ranges from 0 to 1 that characterizes the decay of the influence of older statistics in the iterative computation. The sample mean and variance vectors have been matched statistically between a pair of mixture components before adaptation, therefore $f^{\rho}(\mathbf{c})$ has a larger impact on the mixing coefficients and degree of freedom than the sample mean and variance. Additionally, the weights of unmatched components may be scaled down appropriately, one way to do this is by applying the normalization factor on the unmatched components but keeping the weights of the matched components unchanged.

3.3.2 Parameter Rating

To understand the source of anomaly occurrences in the original feature space, a technique was developed by [97] for parameter rating from the learnt subspace. An anomaly is detected when its logarithmic probability is extremely low. Let z_a be the observed anomaly in the projected subspace span by the PCs, the associated mixture component of the anomaly is given by

$$i^* \triangleq \arg\max_i \{q_i^{\tilde{K}}(z_a) | 1 \leqslant i \leqslant M\} \tag{3.10}$$

$$q_i^{\tilde{K}}(z_j) = \frac{\omega_i N(z_j; \mu_i, \Sigma_i)}{\sum_{\tilde{k}=1}^{\tilde{K}} \omega_i N(z_{\tilde{k}}; \mu_i, \Sigma_i)} \tag{3.11}$$

where $N(z_{\tilde{k}}; \mu_i, \Sigma_i)$ is the probability density of a Gaussian mixture component, M is the number of mixture components, and $\tilde{K}$ is the number of samples. The explanatory vector, which represents the parameters that account for the anomaly, is computed by

$$\bar{x}_a = \phi_{i^*}(x_a) \triangleq \sigma_{q_{i^*}^{\tilde{K}}}^{-\frac{1}{2}}[S] \left| x_a - E_{q_{i^*}^{\tilde{K}}}[S] \right| \tag{3.12}$$

$$\begin{aligned} E_{q_{i^*}^{\tilde{K}}}[S] &= \sum_{\tilde{k}=1}^{\tilde{K}} q_{i^*}^{\tilde{K}}(z_{\tilde{k}}) x_{\tilde{k}} \\ \sigma_{q_{i^*}^{\tilde{K}}}[S] &= \sum_{\tilde{k}=1}^{\tilde{K}} q_{i^*}^{\tilde{K}}(z_{\tilde{k}}) (x_{\tilde{k}} - E_{q_{i^*}^{\tilde{K}}})^2 \end{aligned} \tag{3.13}$$

$\bar{x}_a$ represents the scaled geometric difference vector between the anomaly and the sample mean associated with the mixture component i^*. The parameters are rated by their responsibility for the anomaly occurrence by sorting the entries in $\bar{x}_a$ in a descending order. In cases that a low confidence in the responsibility of a specific mixture component for an observed anomaly, Equation 3.14 presents a soft parameter rating technique proposed by [97] that takes into account the deviation from all the mixture components.

$$\begin{aligned} q_i^M(z_j) &= \frac{\omega_i N(z_j; \mu_i, \Sigma_i)}{\sum_{m=1}^{M} \omega_m N(z_j; \mu_m, \Sigma_m)} \\ \bar{x}_a &= \mathbf{E}_q[\phi(x_a)] = \sum_{i=1}^{M} q_i^M(z_a)(\phi_i(x_a)) \end{aligned} \tag{3.14}$$

Both soft and hard parameter rating techniques can be applied when the SMM statistics is used, in this case, $N(z_{\tilde{k}}; \mu_i, \Sigma_i)$ should be replaced by SMM distribution. The soft parameter rating technique (Equation 3.14) is more computing-efficient than the hard one (Equation 3.13) because there is no need to search for the associated mixture component for each anomaly. Notice that Equation 3.11 is modified from Equation 3 in [97]; the summation in the denominator is over the sample points instead of the mixture components as in [97], this makes more sense in computing the mean and variance in Equation 3.13.

3.4 Experimental Evaluation

Both simulation studies and empirical studies based on real-life datasets are used to investigate and benchmark the proposed hybrid mixture models' performance.

3.4.1 Simulation Studies

Figure 3.1(a) shows two simulated multidimensional Gaussian-distributed centers with white noise added. The noise constitutes one-third of the sample size. Two hybrid models for anomaly detection (see Table 3.1) are used to remove the white noise, the models differ in sophistication and therefore computing efficiency. Although KG is less sophisticated and hence more computing-efficient compared to DKESG, both models perform equally well in removing the white noise with appropriate logarithmic-probability threshold to identify the outliers. It will be shown later using empirical data that even without the EM step, simple hybrid model like KS shows high performance in anomaly detection.

The model adaptation is demonstrated in Figure 3.1(b) with two datasets sharing a common but slightly deviated multidimensional Gaussian-distributed center between the two datasets; the common distribution centers are $(11, 20)$ and $(11.5, 20.5)$ respectively. Each dataset also includes another non-colocated centers, which are $(1, 3)$ and $(1.5, 10.5)$ respectively. The standard deviation of all the distributions are set to 3. With high confidence level ($p \leqslant 0.0001$) during the matching of model mean and variance vectors using statistical methods, the results show that the pro-

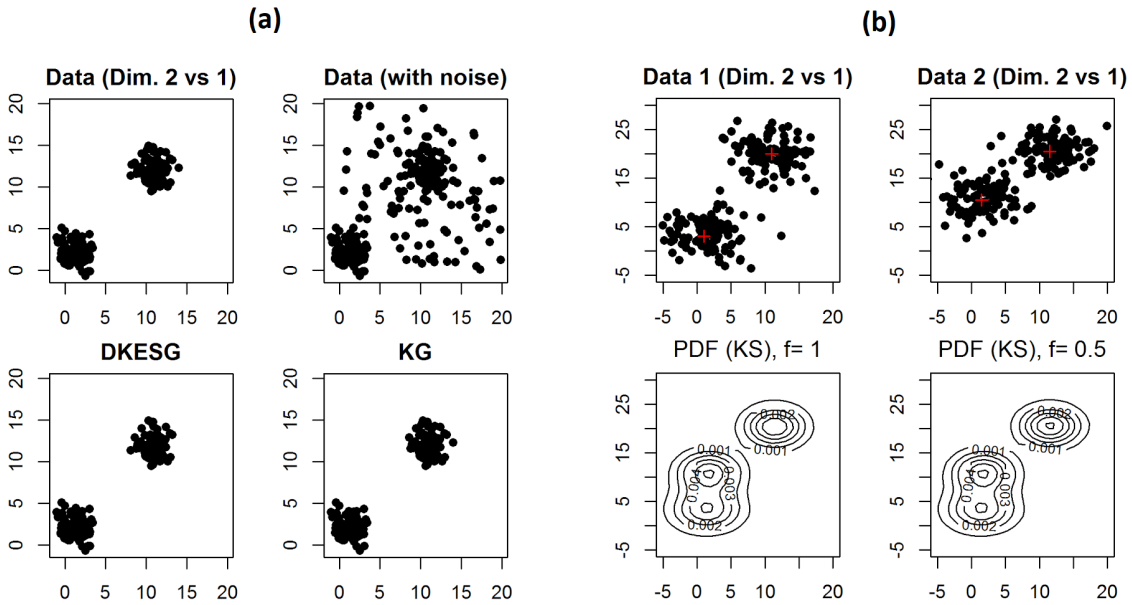

Figure 3.1: **(a)** Top: Simulated multidimensional Gaussian-distributed data with two mixture components (left) and injected white noise (right). The first and second dimensions are plotted here and the distribution centers are $\mu_1 = (1, 2, ..., 10)$ and $\mu_2 = (11, 12, ..., 20)$ respectively with variance $\sigma^2 = (1, 1.5, ..., 5.5)$. The white noise comes from a uniform distribution within the range 0 to 20 in all dimensions. Bottom: Two hybrid models are used to remove the noise, DKESG is more sophisticated and hence less computing-efficient compared to KG but both models perform equally well in removing the noise. **(b)** Top: Two datasets with a common but slightly shifted multidimensional Gaussian-distributed center. The distribution centers are marked as red cross. Bottom: The predictive density of KS model after adaptation of the two datasets with the decay of influence of the older statistics, $f^{\rho}(\mathbf{c})$ set as 1 and 0.5 respectively.

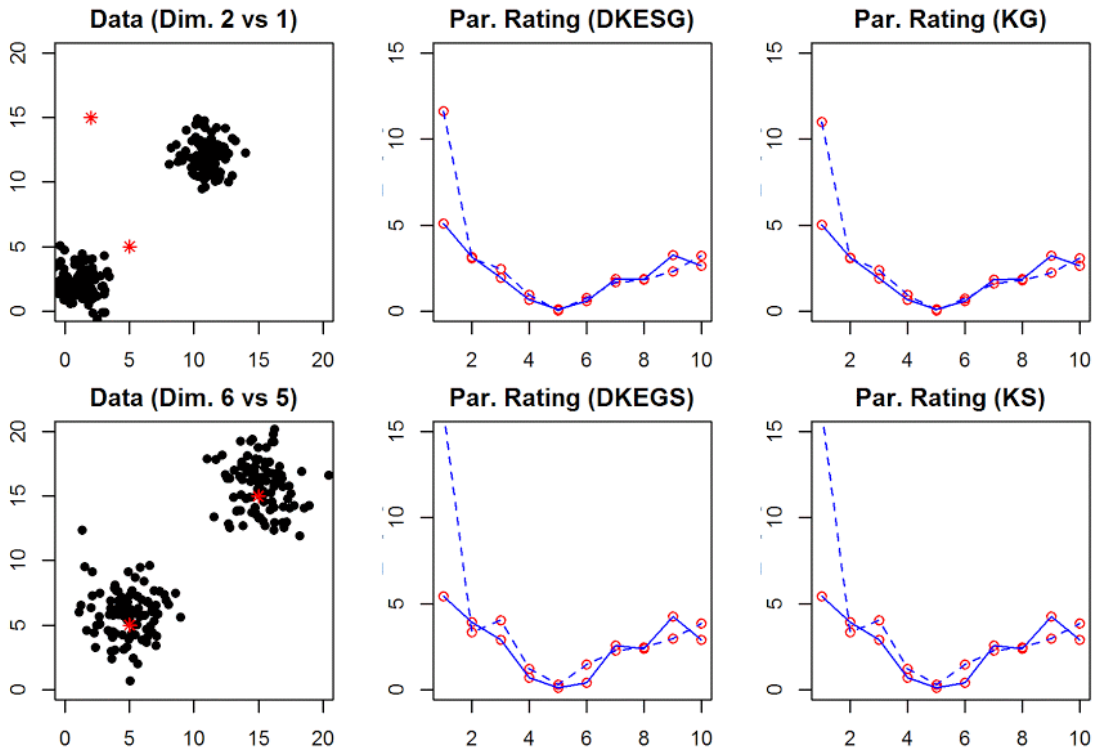

Figure 3.2: First column from left: The first and second dimensions (top) and fifth and sixth dimensions (bottom) of simulated 10 dimensional data with two multidimensional Gaussian-distributed centers and two anomalies marked as red star. The two anomalies are $x_1 = (5, 5, ..., 5)$ and $x_2 = (2, 15, 15, ..., 15)$ respectively, which overlap with the distributions at about the 5th dimension but deviate at other dimensions. Remaining plots are the parameter rating using selected hybrid models. The blue lines correspond to the soft parameter rating using Equation 3.14 and red circles are hard parameter rating using Equation 3.13.

posed model-adaptation technique in Section 3.3.1 provides reasonable predictive density estimation with slight variation near the center of the common distribution between using parameter $f^{\rho}(\mathbf{c}) = 1$ and 0.5 to scale down the older statistics in model adaptation (see Section 3.3.1).

Figure 3.2 simulates the data with the same multidimensional Gaussian distributions as in Figure 3.1 (a) top left plot, but with two anomalies inserted into the dataset to evaluate the parameter rating technique. All the selected hybrid models perform equally well by showing high parameter rating at the dimensions where de-

viation from the Gaussian distributions occur. However, large sample size is required to form a reliable judgement of the parameter ratings because the EM algorithm is sensitive to initial parameter values and better statistics with higher confidence level can be obtained with larger sample size.

Table 3.2: Prediction accuracy of the hybrid mixture models using different datasets obtained from UCI Machine Learning Repository [2].

Dataset	KG	KS	KEG	KES	KEGS	KESG
Iris (Instances: 150, Attributes: 4)	0.97	0.97	0.97	0.97	0.97	0.97
Wine (178, 13)	0.94	0.94	0.94	0.93	0.94	0.93
Adult (48842, 14)	0.81	0.81	0.81	0.80	0.81	0.80
Wisconsin Diagnostic Breast Cancer (569, 32)	0.96	0.96	0.96	0.94	0.96	0.94
KDD Cup 1999 (10% subset: 494020, 41)	0.90	0.90	0.86	N/A	0.86	N/A

3.4.2 Empirical Studies

Dataset. The number of instances and attributes of five popular datasets from UCI Machine Learning Repository [2] is provided in Table 3.2. Iris dataset contains 3 classes of 50 instances each, where each class refers to a type of Iris plant, the attributes consist of petal or sepal width / length. Chemical analysis of Wine dataset contains quantities of 13 constituents found in each of the three types of wines. Adult dataset aims to predict whether a person's income exceeds \$50K/yr based on census data. Breast cancer dataset describes characteristics of the cell nuclei present in a digitized image of a fine needle aspirate of a breast mass. KDD Cup 1999 dataset contains a standard set of network traffic data to be audited, which includes a wide variety of network intrusions simulated in a military network environment. Training and testing on Adult dataset are conducted on two different given datasets (train:

32561, test: 16281), prediction accuracy on other datasets are computed using 10-fold cross validation on a single dataset. The highest prediction accuracy recorded in the repository are Iris (different classification techniques have shown accuracy close to 100%), Adult (Forward Sequential Selection Naive-Bayes: 85.95%), Wine (Regularized Discriminant Analysis: 100%), and Breast Cancer (separating plane: 97.5%) [2]. For KDD Cup 1999 dataset, different prediction accuracy for different network intrusion types were reported using genetic algorithm [106]; i.e., normal (69.5%), probe (71.1%), denial of service (99.4%), user to root attacks (18.9%), and remote to user attacks (5.4%). On average, 88.2% detection rate is achieved in [106].

Prediction Accuracy. Table 3.2 tabulates the prediction accuracy of the proposed hybrid mixture models, all of them perform reasonably well compared to other techniques especially in high-dimensional regime. It is also observed that without the EM step, KG and KS perform equally well compared to the sophisticated models for predictive modeling. To show that the algorithm is scalable, the full KDD Cup 1999 dataset with 41 attributes and close to 5 million instances is used to train and test the hybrid models for large-scale prediction in batch mode of 10^5 instances. The mixture components are adapted using the proposed model-adaptation technique described in Section 3.3.1. However, the variance vectors were not matched here because they are several orders of magnitude smaller than the mean and fluctuate wildly, i.e., only the mean vectors are matched in the adaptation process. The results are shown in Figure 3.3, the highest prediction accuracy is observed when all the PCs are used. The EM algorithm to estimate GMM converges even with reduced dimensions $\gtrsim 40$ and produce higher prediction accuracy compared to the ones without the EM step (compare KEG and KEGS to KG and KS). The prediction accuracy fluctuates with the data sampling, this is likely a characteristic of the dataset which contains both predictable and less-predictable events. New PCs not spanned by the older set of PCs are appended using the Gram-Schmidt orthogonalization process described in Section 3.2.1, the unused PCs are not discarded in the updating process. It is observed that the SMM-based hybrid models (KS, KES, and KESG) do not perform well in reduced dimensions $\gtrsim 30$.

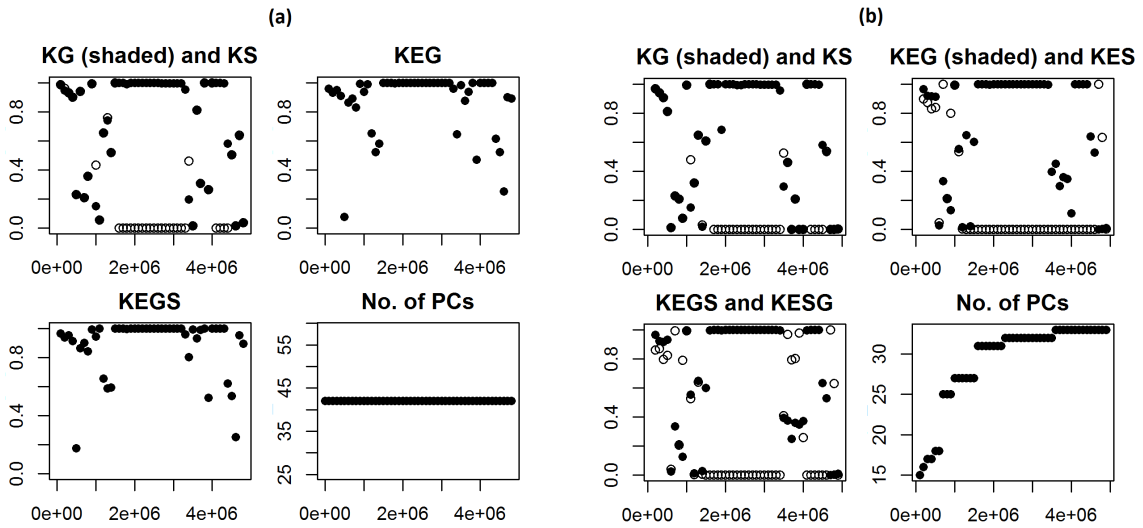

Figure 3.3: Prediction accuracy (y-axis) of the hybrid models for batches of 10^5 instances using KDD Cup 1999 network intrusion dataset **(a)** with same number of PCs and **(b)** changing number of PCs for each batch of data (x-axis represents the time series or incoming batches of data over time). The solid / shaded circles and empty circles represent different models labeled above each plot.

Anomaly Detection. Table 3.3 tabulates the detection rate and false positive rate of the hybrid mixture models using the KDD Cup 1999 dataset (10% subset). Normal-type network data of 95000 instances are extracted from the dataset and injected with different percentages of injected anomalies. The number of PCs required for anomaly detection is lesser compared to predictive modeling, in particular, only seven PCs were used here. The data is coarse-filtered with the technique based on diffusion map described in Section 3.2.2 to remove the anomalies before model training. The logarithmic-probability threshold for anomaly detection is set as the percentile of injected anomalies. It is observed that $\gtrsim$ 80% detection rate is possible with 1% to 60% injected anomalies with coarse-filtering technique based on DM. Higher percentage of injected anomalies biases the training model and lower percentage increases the false positive rate, hence present different challenges to unsupervised anomaly detection. However, the detection rates fluctuate with the data-sampling process because the DM-based coarse-filtering technique processes limited amount of data points at one time due to the need to compute the similar-

ity distance between each pair of data points (see Section 3.2.2). Without coarse filtering, KS detection rate achieves $\gtrsim 90\%$ and because of the top-down approach, the measured detection rates are robust to data sampling process.

Table 3.3: Model performance in anomaly detection using KDD Cup 1999 computer network intrusion dataset (10% subset) with different percentages of injected anomalies. The dataset contains "normal" and "attack" data. The "normal" data is first extracted from the dataset and "attack" data is then artificially injected. The percentages of injected anomalies are calculated based on the ratio of artificially injected "intrusion" data into the extracted "normal" data.

Anomalies		KS	DKG	DKS	DKEG	DKES	DKEGS	DKESG
60%	DR	0.93	**0.94**	0.77	**0.94**	0.47	0.77	0.46
	FP	0.10	0.086	0.35	0.086	0.80	0.35	0.81
50%	DR	0.93	**0.96**	0.95	**0.96**	0.50	0.95	0.51
	FP	0.073	0.045	0.048	0.045	0.50	0.048	0.49
40%	DR	0.93	0.93	**0.94**	0.93	0.84	**0.94**	0.80
	FP	0.047	0.047	0.041	0.047	0.11	0.041	0.13
30%	DR	**0.90**	0.84	0.82	0.85	0.78	0.84	0.72
	FP	0.043	0.067	0.075	0.062	0.094	0.067	0.12
20%	DR	**0.92**	0.85	0.85	0.85	0.32	0.85	0.36
	FP	0.020	0.037	0.037	0.037	0.17	0.037	0.16
10%	DR	**0.93**	**0.93**	**0.93**	**0.93**	0.11	**0.93**	0.90
	FP	0.0079	0.0073	0.0079	0.0073	0.017	0.0079	0.011
5%	DR	**0.91**	0.81	0.80	0.81	0.16	0.80	0.17
	FP	0.0050	0.0099	0.011	0.0099	0.044	0.011	0.014
1%	DR	0.89	**0.92**	**0.92**	**0.92**	0.00	**0.92**	0.00
	FP	0.0012	0.00085	0.00075	0.00085	0.010	0.00075	0.010

Detection rate,

$$DR = \frac{\#detected\ anomalies}{\#injected\ anomalies} \tag{3.15}$$

False positive rate,

$$FP = \frac{\#false\ positive}{\#inliers} \tag{3.16}$$

Parameter Rating. Figure 3.4 shows the soft parameter rating for different network intrusion types using KDD Cup 1999 dataset (10% subset). The model is trained with normal-type network data and the KS statistics is used to compute the parameter rating. This is because KG statistics is too sparse due to the rapidly-decaying GMM tail distribution. For large number of anomalies, the soft parameter rating technique (Equation 3.14) is more computing-efficient than the hard one (Equation 3.13) because there is no need to search for the associated mixture component for each anomaly. Results show that there is overlap between the sources of anomaly occurrences from different network intrusion types; the parameter rating may be used to suggest mitigating actions for each intrusion type.

3.5 Discussion

Twelve variants of hybrid mixture models have been assessed in terms of model performance and computing efficiency. In particular, KG and KS hybrid models are recommended for high-dimensional predictive modeling and anomaly detection respectively. This has implication for big-data applications because the EM algorithm may not be required to estimate mixture model parameters for high-dimensional data, which saves the computing cost. However, it is also found that whitened PCA reduces the dimensions and scales the subspace to a smaller one, which allows the EM algorithm to converge even with reduced dimensions $\gtrsim 10$. For real-time batch data processing, the proposed PC-updating technique based on Gram-Schmidt orthogonalization process is demonstrated; this technique can be used even if new dimensions are added in the original feature space. KS statistics is used to compute the parameter rating because GMM statistics is too sparse due to the rapidly-decaying tail distribution. Soft parameter rating is more computing-efficient than the hard one because there is no need to search for the associated mixture component for each

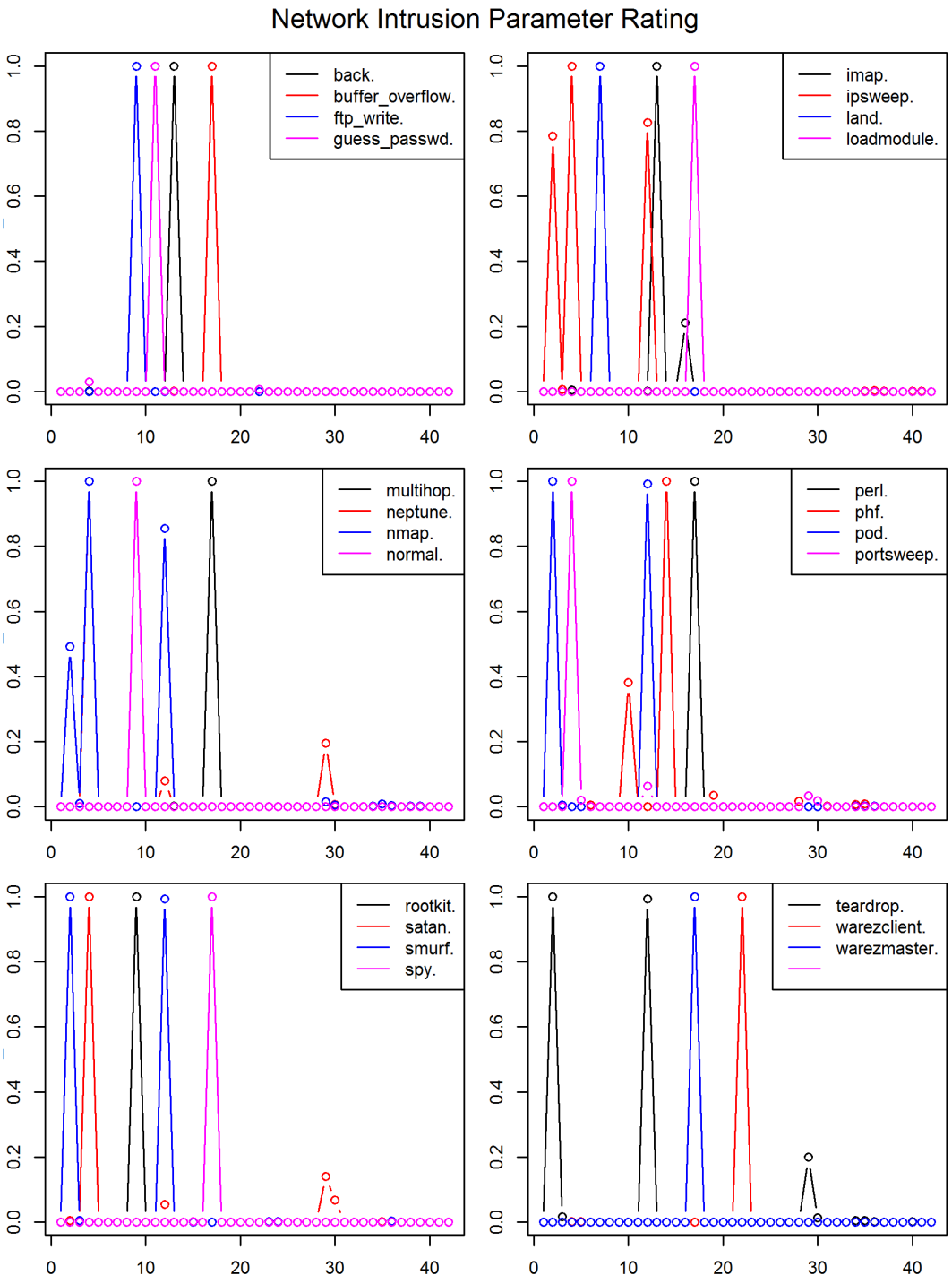

Figure 3.4: Parameter rating of each network intrusion type of KDD Cup 1999 dataset (10% subset). The ratings are normalized to the range $[0, 1]$. Results show that majority of the sources of anomaly occurrences come from dimensions $\lesssim 20$ and may be used to suggest mitigating actions.

anomaly. For anomaly detection, the detection rates measured from a bottom-up approach using DM-based coarse-filtering technique to remove the anomalies tend to fluctuate with the data sampling process. On the other hand, a top-down approach using KS statistics is demonstrated to achieve $\gtrsim 90\%$ detection rate from 1% to 60% of injected anomalies in the KDD Cup 1999 network intrusion dataset and this approach is more robust to the data sampling process. In future work, the proposed hybrid mixture models can be tested for broader range of applications to better understand the models' performance and stability.

Chapter 4

Training Fully-Connected Neural Network using Compressed Input Data Stored and Computed in Low-Rank Tensor-Train Format

Deep learning extract features directly from raw data and construct higher-level features to achieve human-level performance in many designated tasks. Deep learning turns out to be very good at discovering intricate structures in high-dimensional data and the machine learning framework is applicable to many domains of science, business, and government. Therefore, deep learning approach is much more efficient and effective compared to features designed by human engineers which is typically time consuming and task-specific. Currently, deep learning requires millions to billions of structured data and model parameters to learn the underlying features / representations for different tasks. To speed up deep learning for big data, we propose tensor decompositions and tensor networks to reduce the storage and memory during the training and inference using deep learning models. Experiments are conducted by training fully connected network using compressed MNIST grayscale images of handwritten digits and CIFAR-10 color images for objects recognition, both stored and computed in tensor-train formats. The results show that 3.6% (compression ratio = 27) of MNIST and 2.6% (compression ratio = 38) of CIFAR-10 original

data size are enough to train the model with only 1% and 2% drop in accuracy respectively. The model parameters can be reduced by 100 to 1000 times using the same technique (i.e., tensor-train decomposition). The proposed data compression technique is applicable for different data structures and machine learning algorithms because tensors can express multimodal data and the tensor decomposition framework allows arithmetic operations on tensor network formats in distributed manner without compromising the computational efficiency under the low-rank assumption. This work has implications in saving the storage and communication costs without compromising computational efficiency and model performance, which may pave the ways of applying machine learning through fog or edge computing for the Internet of things paradigm.

4.1 Introduction

Deep learning [107] surpasses conventional machine-learning techniques through representation learning on raw data and automatically discover the representations needed for detection or classification. Deep learning constructs multiple levels of representation by composing simple but non-linear modules that each transform the representation into higher / more abstract level. The key aspect of deep learning is that these layers of features are not designed by human engineers: they are learned from data using a general-purpose learning procedure. Deep learning turns out to be very good at discovering intricate structures in high-dimensional data and is therefore applicable to many domains of science, business and government. Current state-of-the-art deep learning models are trained on big data with millions to billions of parameters. Therefore, techniques that enable efficient processing of deep neural network to improving energy-efficiency and throughput without sacrificing accuracy with cost-effective hardware are critical to expanding the model deployments in both existing and new domains, a recent comprehensive survey on accelerated deep learning from both hardware and software perspectives can be found in [108].

Tensor networks have been used independently either to reduce model parameters for large-scale machine learning models or extract physically meaningful latent vari-

ables / complex features from data. However, training a machine-learning model, in particular deep learning model, on compressed data both stored / computed in low-rank Tensor Train (TT) format has not been done, which has the potential to realize the edge / fog computing paradigm for Internet of things by reducing communication bandwidth and provide real-time operations through distributed computing. Specifically, the contributions of this work include:

- Propose and demonstrate the training of a fully connected network using compressed input data stored / computed in low-rank tensor-train format. The storage and computational complexity are analyzed theoretically and empirically. Other factors that have been analyzed empirically include data (relative) approximation error, data shape, minimal and maximal TT-ranks, batch size, number of hidden units and how they affect the storage, computational efficiency, and model performance.

- Benchmark the model performance trained on compressed data using MNIST grayscale images of handwritten digits and CIFAR-10 color images for objects recognition, the results show that 25 to 40 times data compression ratio is possible without significant computational overhead in terms of the data compression speed, network training and inference speed. Empirical experiments suggest that the data should be compressed with mini-batch size to ensure high compression speed and model performance. In addition to data (relative) approximation error, the data shape plays an important role to obtain low TT-ranks for high compression ratio, computational efficiency, and model performance.

- For the fully-connected layer, empirical experiments show that higher number of hidden units improves the model performance but result in slower training and inference speed. The TT-ranks do not affect the model performance a lot but high TT-ranks require orders of magnitude higher storage cost than the number of hidden units and result in higher computational overhead.

The organization of this chapter is as follows. Section 4.2 describes the tensor techniques to tensorize the fully-connected neural network and corresponding operations,

Section 4.3 presents the experimental study, and Section 4.4 discusses implications of the findings.

4.2 Tensorizing the Data Inputs and Weight Matrix of the Fully-Connected Layer

The universal approximation theorem [109, 110] states that a feedforward network with a single hidden layer containing finite number of neurons can approximate any continuous functions under mild assumptions on the activation function. The theorem thus asserts that simple neural networks can represent a wide variety of interesting functions when given appropriate parameters, this result holds even if the function has many inputs and outputs. In a feedforward network, a fully-connected (FC) layer applies a linear transformation to an N-dimensional input vector x:

$$y = Wx + b \tag{4.1}$$

where $W \in \mathbb{R}^{M \times N}$ is the weight matrix and $b \in \mathbb{R}^M$ is the bias vector. Given the nice properties of TT format mentioned before, the weight matrix and input vector of the FC layer are therefore represented in TT formats, the FC layer can be compactly written as

$$\begin{aligned}\mathcal{Y}(i_1, i_2, \cdots, i_d) = &\sum_{j_1, \cdots, j_d} \tilde{W}_1[i_1, j_1] \cdots \tilde{W}_d[i_d, j_d] \cdot \\ &\tilde{X}_1[j_1] \tilde{X}_2[j_2] \cdots \tilde{X}_d[j_d] + \mathcal{B}(i_1, i_2, \cdots, i_d) \\ = &\sum_{j_1, \cdots, j_d} (\tilde{W}_1[i_1, j_1] \ \ |\bullet| \ \ \tilde{X}_1[j_1]) \cdots \\ &(\tilde{W}_d[i_d, j_d] \ \ |\bullet| \ \ \tilde{X}_d[j_d]) + \mathcal{B}(i_1, i_2, \cdots, i_d)\end{aligned} \tag{4.2}$$

where $\tilde{W}_k[i_k, j_k]$ is a $r^W_{k-1} \times r^W_k$ matrix, $\tilde{X}_k$ is a $r^X_{k-1} \times r^X_k$ matrix, $|\bullet|$ is the core contracted product of two block tensors [3], and r^W_k and r^X_k are the TT-ranks of the FC layer's weight matrix and input vector respectively, both stored / computed in TT formats. Figure 4.2 shows graphical representations of the FC weight matrix (W), input vector (x), and the matrix-by-vector multiplication (Wx) in TT formats. Table 4.1 summarizes the computational complexity and memory usage

of the forward and backward pass [60] to train the neural network (see Figure 4.1). The complexity of the matrix-by-vector multiplication in TT formats is $O(dI^2r^4)$ [3], each core contains a copy and therefore the memory usage is $O(I^2r^4)$. Substituting matrix-by-vector multiplication into the backward pass (see Section 5 in [60]) yields $O(d^2I^2r^6)$ complexity and $O(I^2r^6)$ memory usage.

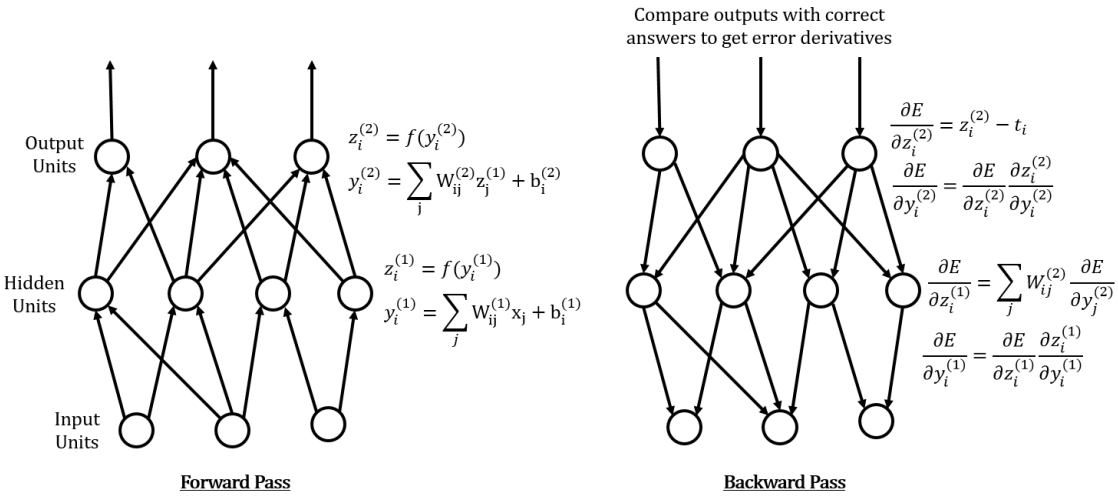

Figure 4.1: A neural network with two fully-connected layers. The hidden units linearly combine its inputs by weighted sum followed by nonlinear activation function such as sigmoid $f(y) = \frac{1}{1+\exp(-y)}$or rectified linear units (ReLU) $f(y) = \max(0, y)$. The network is usually trained by error backpropagation, which consists of the forward and backward pass using equations as shown above. The objective function here is quadratic loss.

4.3 Experimental Study

The computational efficiency and storage cost of FC network with TT input and TT weight matrix are first evaluated and compared to FC layer with full-array input. Different backpropagation algorithms are compared to benchmark against previous work in terms of computational efficiency and classification error rate. This section also investigates various factors affecting the data compression ratio, computational efficiency, and model performance. These factors include data (relative)

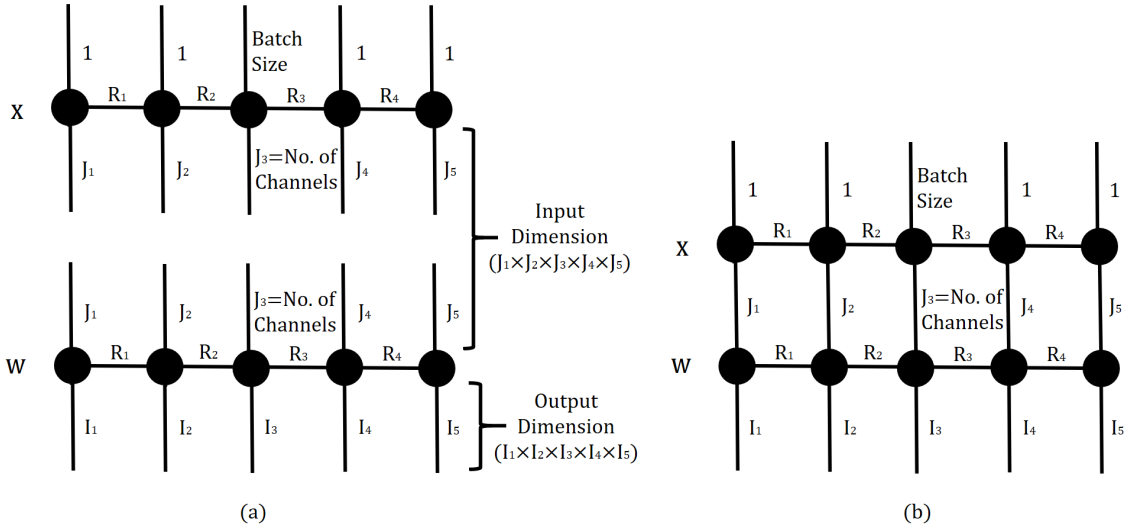

Figure 4.2: Graphical representations of (a) FC network weight matrix (W) and mini-batch size of input data (x) represented in matrix TT formats, (b) matrix-by-vector multiplication (Wx) computed in TT formats.

approximation error, batch size for data compression, and data shape that plays an important role in achieving low TT-ranks for optimal storage and computation. The data (relative) approximation error is defined as the ratio of Frobenius norm of approximation loss and Frobenius norm of original data. The communication cost for parallel distributed computing is worth investigating but beyond the scope of this study.

Data. The model performance is evaluated using MNIST grayscale images of handwritten digits and CIFAR-10 color images for objects recognition. The error rate is defined as the model prediction error. It is worth mentioning the difficulty of the task when comparing different models. The size of the classifier or network (i.e. number of weights) will be larger for more difficult task than the simpler task and thus require more storage and computation. MNIST [111] is a widely used dataset for digit classification that was introduced in 1998. It consists of 28×28 pixel grayscale images of handwritten digits. There are 10 classes (for 10 digits) and $60,000$ training images and $10,000$ test images. CIFAR-10 [112] is a dataset that consists of 32×32 pixel color images of of various objects, which was released in 2009. CIFAR-10 is composed of 10 mutually exclusive classes. There are $50,000$

Table 4.1: Computational complexity and memory usage of the forward and backward pass in training the FC network with input data and weight matrix in full array or TT format. $I = \max_{k=1,\cdots,d}\{i_k, j_k\}$, $r = \max_{k=0,\cdots,d}\{r_k^W, r_k^X\}$, $m = \max_{k=1,\cdots,d}\{i_k\}$

Operation	**Input**	**Computation**	**Memory**
FC forward	Full	$O(MN)$	$O(MN)$
TT forward	Full	$O(dr^2mI^d)$	$O(rI^d)$
TT forward	TT	$O(dI^2r^4)$	$O(I^2r^4)$
FC backward	Full	$O(MN)$	$O(MN)$
TT backward	Full	$O(d^2r^4mI^d)$	$O(r^3I^d)$
TT backward	TT	$O(d^2I^2r^6)$	$O(I^2r^6)$

training images (5000 per class) and $10,000$ test images (1000 per class).

Implementation Details. Table 4.2 tabulates the hardwares and softwares used in this study. MatConvNet is a MATLAB toolbox implementing convolutional neural networks for computer vision applications. AutoNN is a functional wrapper for MatConvNet, implementing automatic differentiation. TT-toolbox is a MATLAB implementation of basic operations with tensors in low-parametric representation of high-dimensional tensors. TensorNet is a Matlab implementation of the tensor-train layer of a neural network. To benchmark the computational efficiency using the proposed compressed data computation in low-rank TT format, Matlab is configured to run on single core to estimate the training and inference speed of the fully connected network.

Error-backpropagation algorithms. Numerical differentiation (ND) approximates gradient using divided difference but introduces rounding error in the discretization process. Automatic differentiation (AD) applies symbolic differentiation on sophisticated functions at elementary operation level by applying chain rule repeatedly to these operations and keep intermediate results. AD is much easier to code, especially for function compositions, and generalize well to higher-order derivatives. Table 4.3 compares the network training and inference speed using ND [60] and AD. AD is

Table 4.2: Implementation details. *Matlab by default runs on single core only. The multicores environment is not fully utilized in order to focus on algorithmic efficiency instead of the communication cost for parallel distributed computing.

Software	**Version**
Matlab	R2016b
MatConvNet	1.0 beta24
Autonn	-
TT-toolbox	2.2.2
TensorNet	-
Hardware	**Specification**
Processor*	Intel(R) Core(TM) i3-4170 CPU @ 3.70GHz 3.70GHz
Installed memory (RAM)	4.00GB
System type	64-bit Operating System, x64-based processor

Table 4.3: FC network training / inference speed and MNIST classification error rate using different backpropagation algorithms with data inputs in full array and TT format (maximal TT-rank= 5). ND and AD refer to numerical differentiation and automatic differentiation respectively. The FC weight matrix in TT-format has 3125 hidden units and maximal TT-rank = 2.

Backpropagation Algorithm	**Training / Inference Speed (Hz, images/s)**	**Error Rate**
ND [60] (Full input array)	820 / 7200	1.9% [60]
ND [60] (TT input format)	750 / 3900	6.5%
AD (Full input array)	3300 / 7150	3.0%
AD (TT input format)	3100 / 7000	3.5-4.0%

much more efficient than ND during the training phase but with about 1% drop in accuracy. Running with input data in TT format slows down the computation by less than 10%, this depends heavily on the TT-ranks as shown by the computational complexity in Table 4.1. The large difference in the ND inference speed between full and TT input by a few kilohertz is likely attributable to the software libraries instead of tensor computation, because AD has about the same inference speed for both full-array and TT input.

4.3.1 Computational Efficiency

Figure 4.3 shows that the storage cost increases with the number of hidden units and maximal TT-rank in linear and quadratic manners respectively, storing the weight matrix in TT format reduces the parameters by 100 to 1000 times. Figure 4.4 shows the training and inference speeds within the range of the number of hidden units and the maximal TT-rank of the FC weight matrix that we use in the experiments, the results show that the higher the number of hidden units, which is critical for good model performance, reduces the training and inference speed dramatically.

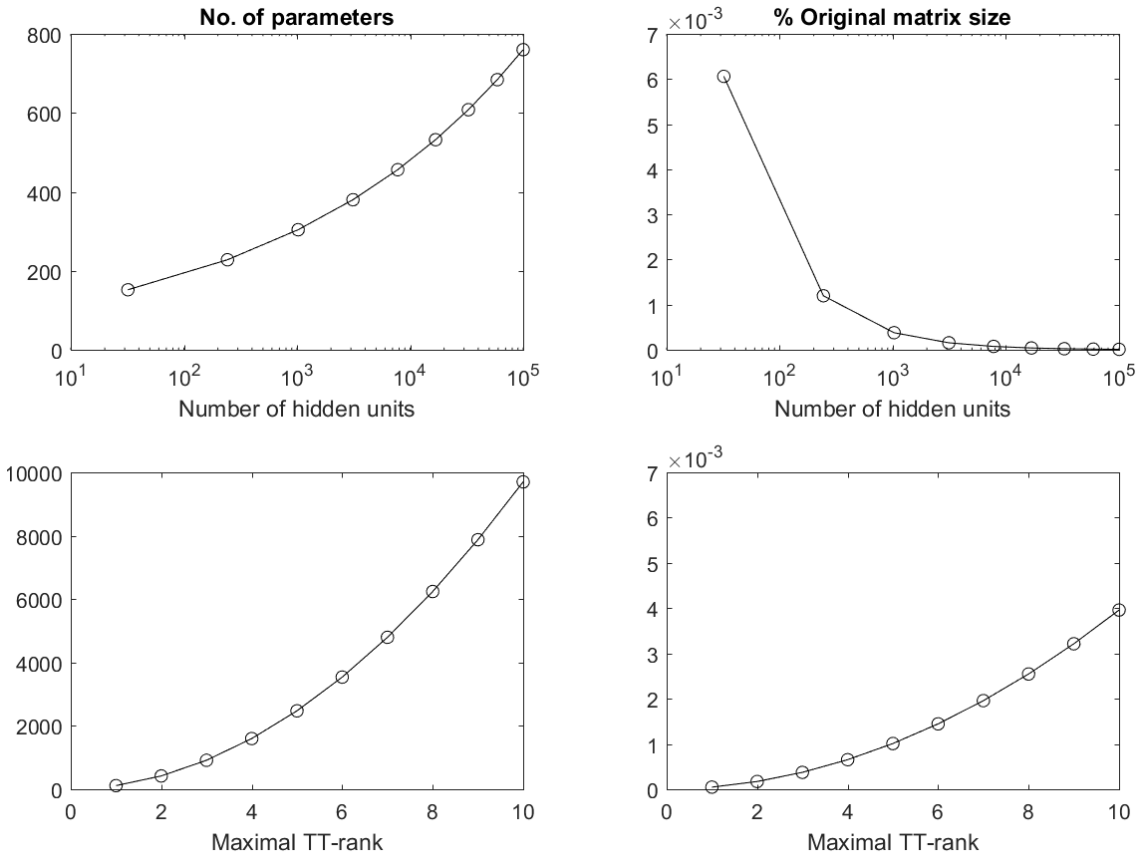

Figure 4.3: Storage cost of FC layer's weight matrix in matrix TT format. Low-rank TT format stores 100 - 1000 times less parameters compared to the FC layer with weight matrix in full array.

4.3.2 MNIST Handwritten Digits Recognition

The fully connected network consists of >3000 hidden units and the nonlinearity is from the rectified linear units. The experimental results are tabulated in Tables 4.4-4.7 and shown in Figures 4.5-4.7. It is observed that even with the same data (relative) approximation error, the data shape plays an important role in controlling the reduced TT-ranks and therefore the compression ratio and computational efficiency. The suboptimal compression observed in Table 4.4 compared to Table 4.5 is due to the high reduced TT-ranks of data inputs. By reshaping the data and impose a maximal TT-rank, large compression ratio can be achieved with less than 1% drop in classification accuracy (see Figure 4.6). Higher maximal TT-rank is required to compress larger batch size of data to maintain the model performance. For the FC

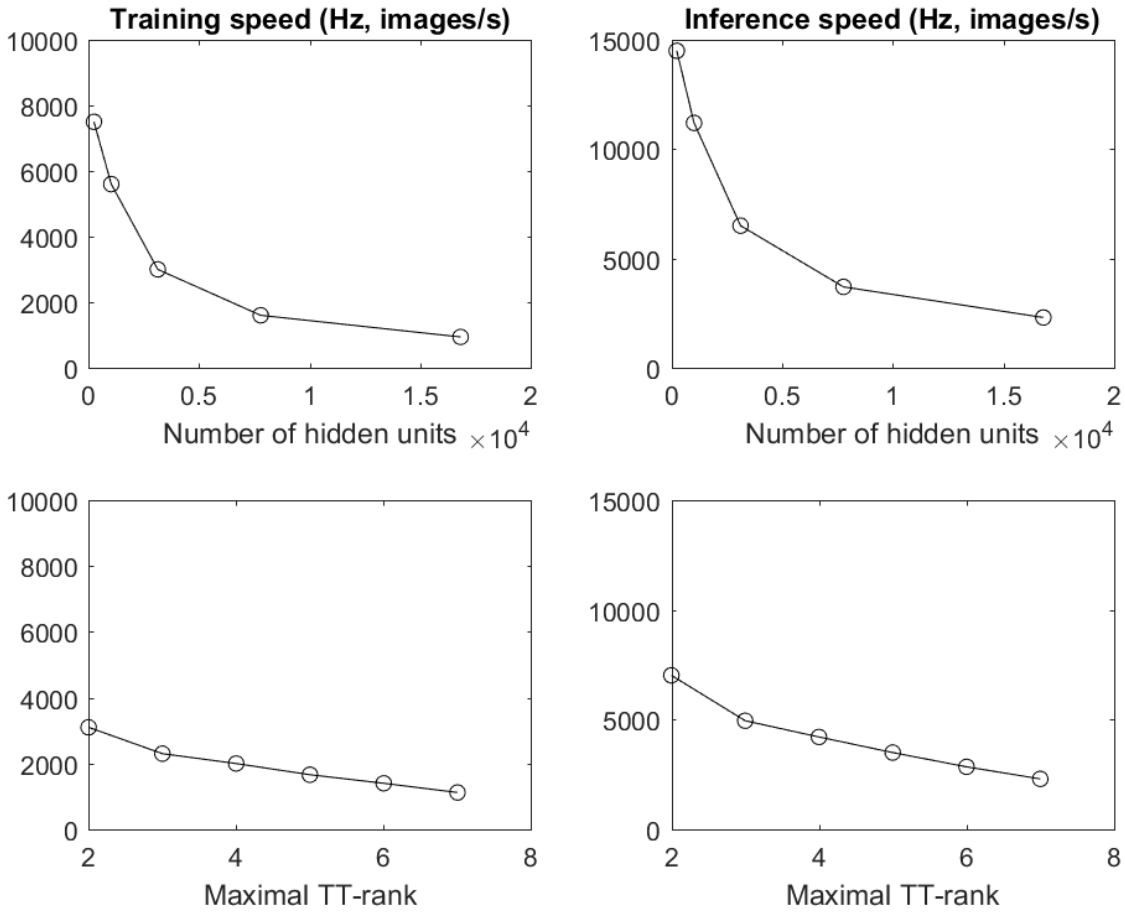

Figure 4.4: Training (left) and inference speed (right) of the FC network with weight matrix of different parameters (i.e. number of hidden units and maximal TT-rank) using MNIST images of handwritten digits stored / computed in TT formats. The training and inference speeds are collected over 10 epoches and averaged in order to reduce fluctuations.

layer, the number of hidden units affects the network training / inference speed and the model performance, as shown in Table 4.7. Low-rank TT approximation blurs the edges but preserve the local features in each grayscale image, see Figure 4.5. The TT-ranks of the FC weight matrix does not affect the classification accuracy too much but the storage scales quadratically with the TT-ranks, therefore all the TT-ranks are set to 2. It is observed that the TT weight matrix or TT data inputs with values too small for each tensor dimension or with too few dimensions perform worse than their more balanced counterparts. Overfitting happens when a model learns the details and noise in the training data to the extent that it negatively

Table 4.4: FC network (TT weight matrix: 3125 hidden units and maximal TT-ranks = 2) trained with MNIST data of input dimension = (4×7×7×4×100 (batch size)) and compressed using low-rank TT decomposition with different data (relative) approximation error. The reduced TT-ranks and % compression vary with different batches of data, therefore these values are averages over the whole dataset.

Reduced TT-ranks of Data Inputs	**Data (Relative) Approximation Error**	**% Original Data Size**	**Error Rate**
(1, 1, 4, 13, 26, 1)	0.9 (low accuracy)	5.55%	6.5%
(1, 2, 6, 21, 37, 1)	0.7	9.93%	4.0%
(1, 2, 8, 35, 52, 1)	0.5	18.57%	3.5%
(1, 3, 13, 66, 72, 1)	0.3	41.45%	3.5%
(1, 4, 23, 129, 94, 1)	0.1 (high accuracy)	101.19%	3.5%

impacts the performance of the model to generalize on new data. In Figure 4.6, the large difference in error rate between training and validation sets indicate that their distributions are different, but no sign of overfitting is observed. Higher fluctuations are observed for input data in TT format. The error rate typically converges within 20-30 epoches, which means the model training is complete. Automatic differentiation decreases the model performance from previous work [60] from 1.9% to 3%, operating on compressed data in TT format decreases another 1% which results in 3.5-4.0% error rate.

Figure 4.5: (Top) original and (bottom) decompressed MNIST images of handwritten digits from low-rank TT format.

Table 4.5: Similar to Table 4.4 but with MNIST data of input dimension = (4×7×100 (batch size)×7×4). By reshaping the data, the optimal compression (compared to Table 4.4) comes with slightly higher classification error rate.

Reduced TT-ranks of Data Inputs	**Data (Relative) Approximation Error**	**% Original Data Size**	**Error Rate**
(1, 1, 4, 3, 2, 1)	0.9 (low accuracy)	4.04%	9-10%
(1, 2, 6, 5, 2, 1)	0.7	7.50%	4.1%
(1, 2, 8, 7, 3, 1)	0.5	13.09%	3.0%
(1, 3, 13, 12, 4, 1)	0.3	29.54%	3.0%
(1, 4, 23, 20, 4, 1)	0.1 (high accuracy)	63.22%	3.2%

Table 4.6: Low-rank TT approximation of the MNIST images with batch size = 10×10 (underlined). The number of hidden units of the trained FC network is > 3000 for all the experiments. It is observed that the compression ratio and model performance is higher when the batch size is not split into multiple dimensions.

Input Dimension	**TT-ranks (full) TT-ranks (low)**	**% Original Data Size**	**Error Rate**
(28×100×28)	(1, 28, 28, 1) (1, 6, 6, 1)	102% 5.02%	- 3.5%
(28×10×10×28)	(1, 28, 280, 28, 1) (1, 28, 30, 28, 1)	202% 20.92%	- 6.0-7.0%
(4×7×100×7×4)	(1, 4, 28, 28, 4, 1) (1, 4, 5, 5, 4, 1)	102% 3.59%	- 3.5-4.0%
(4×7×10×10×7×4)	(1,4,28,280,28,4,1) (1,4,28,30,28,4,1)	202% 20.96%	- 4.0-5.0%

Table 4.7: FC network trained on MNIST data compressed with different batch sizes into low-rank TT-format with maximal TT-rank = 5 and input dimension = (4×7×batch size×7×4). The compression speed is averaged over 10 repeated data-compression experiments to reduce fluctuations.

Batch Size	**Compression Speed (Hz, images/s)**	**% Original Data Size**	**Error Rate**
100	7800	3.59%	3.5 - 4.0%
500	6300	3.27%	4.5%
1000	5400	3.23%	5.2%
5000	3600	3.20%	8%
10000	3400	3.19%	10%

4.3.3 CIFAR-10 Object Recognition In Images

Most of the experiments with MNIST dataset in Section 4.3.2 is repeated with CIFAR-10 dataset and the results are shown in Figures 4.8-4.10 and tabulated in Table 4.8 and Table 4.9. Table 4.8 shows that the input data shape does not affect the compression ratio and model performance as observed in the MNIST experiments, this may be because the intra-class variation in CIFAR-10 color images is high enough such that local interactions always dominate. In Table 4.9, the optimal compression ratio and error rate is observed at maximal rank= 5 for batch size= 100; the increase of error rate with larger batch size suggests that higher maximal TT-rank is required to maintain the model performance. Figure 4.8 shows that the low-rank TT approximation preserves the local features in each color image. Increasing the number of hidden units improve the classification accuracy but no improvement is observed by increasing the TT-ranks of the weight matrix of the FC layer, see Figure 4.10. The data shape does not affect the data compression ratio and model performance in classifying the 10 classes of objects, but results in much higher computational cost due to high TT-ranks as indicated by the theoretical computational complexity in Table 4.1. Similar to MNIST experiment, higher maximal

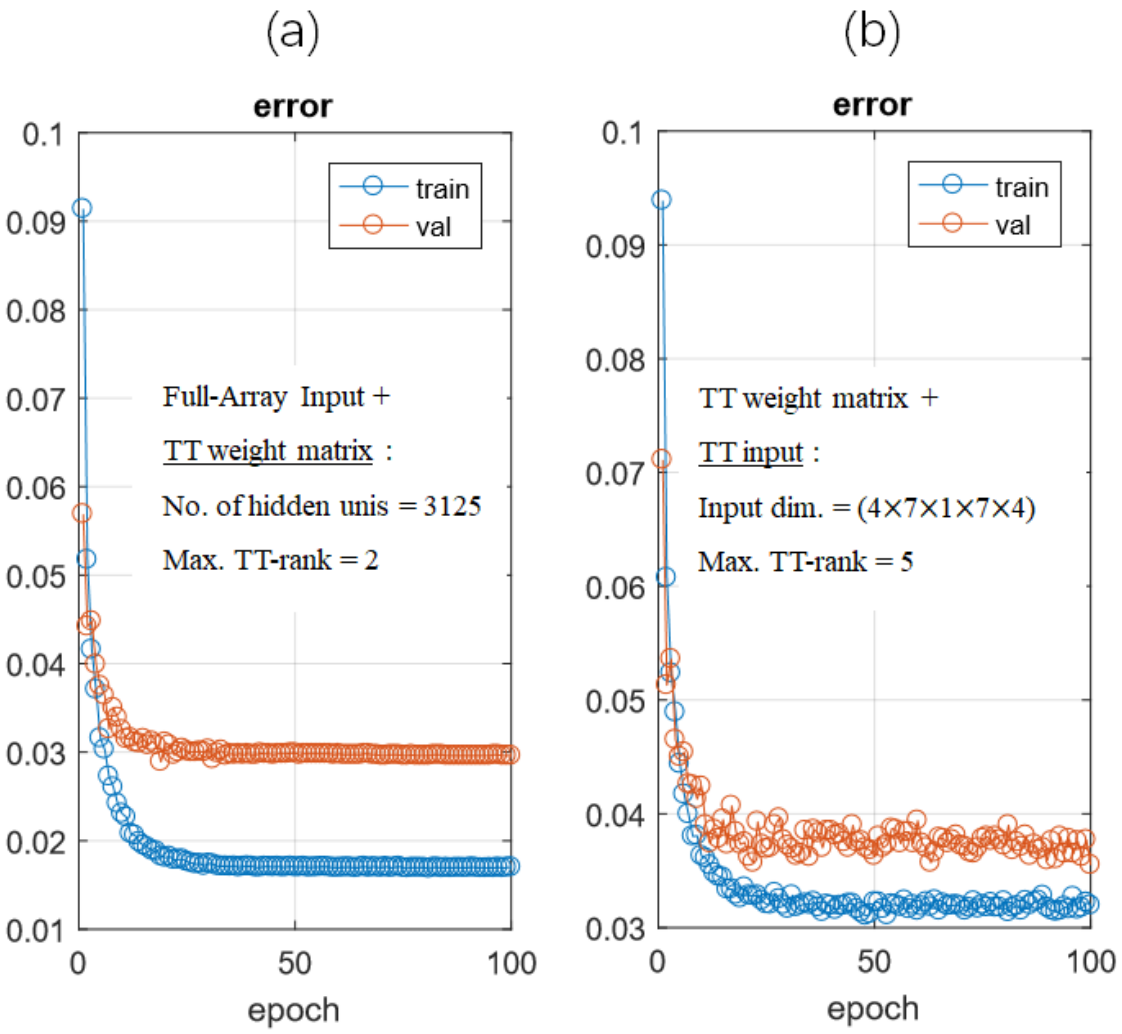

Figure 4.6: Training of a FC network with MNIST data input in (a) full array and (b) TT format.

TT-rank is required to compress data of larger batch size in order to maintain the model performance. As a baseline, the work in [60] first preprocesses CIFAR-10 images by subtracting the mean and performing global contrast normalization and ZCA whitening. Their network consists of convolutional, pooling, and non-linearity layers followed by two fully-connected layers in TT format with 3125 hidden units and all the TT-ranks = 8. Their test error is 23.13% without fine-tuning. In comparison, our fully connected network consists of 16807 hidden units with all the TT-ranks = 2 without data augmentation or fine-tuning, the model performance is at best 40% classification error for both input data in full array or TT format.

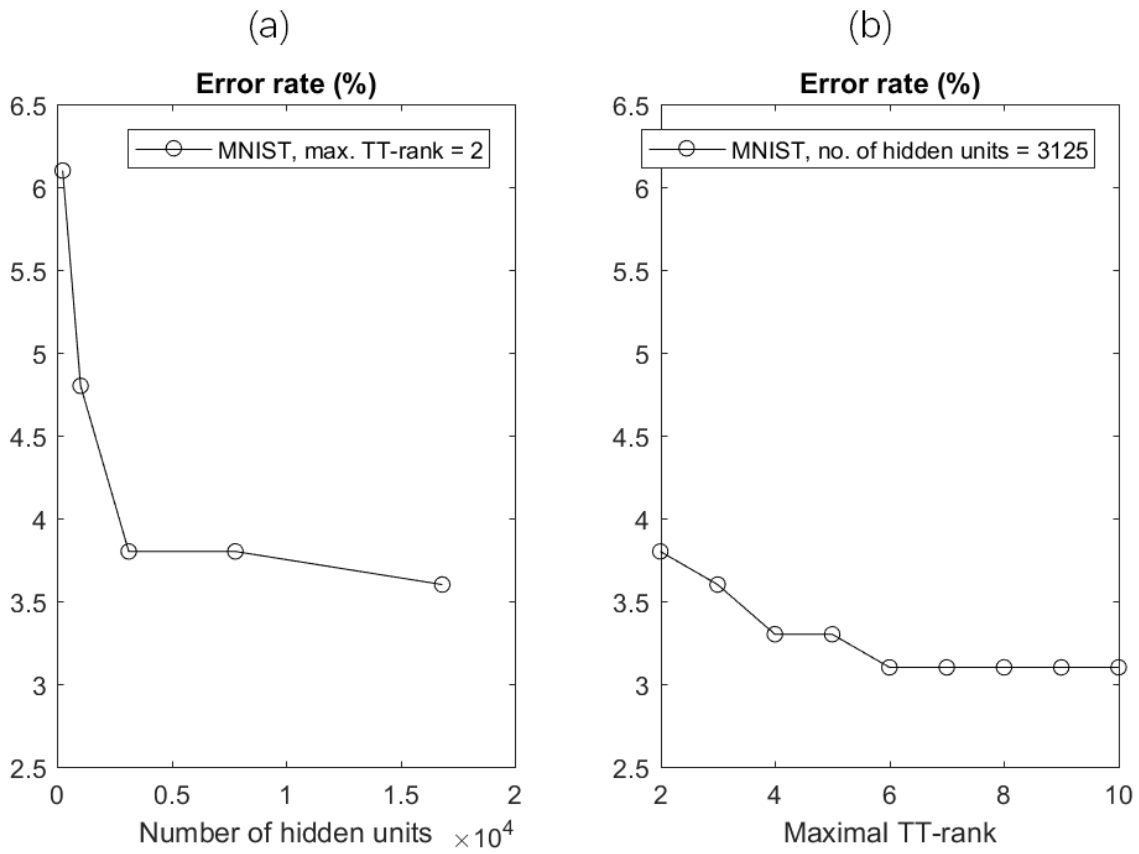

Figure 4.7: The classification error rate of MNIST grayscale images of handwritten digits using different (a) number of hidden units and (b) maximal TT-ranks for the FC weight matrix stored / computed in TT format.

Applying convolutional layer on TT input is beyond the scope of this work.

4.4 Discussion

Low-rank tensor network decomposition has the potential to realize accelerated and multimodal deep learning. Our findings suggest that the data storage cost can be reduced by 25 to 40 times by sacrificing little computational efficiency and model performance. Factors affecting the data compression ratio include data (relative) approximation error, batch size, and crucially data shape to achieve low TT-ranks for massive compression because the storage size scales quadratically with the maximal TT-rank. Empirical results also show that the computational efficiency is not

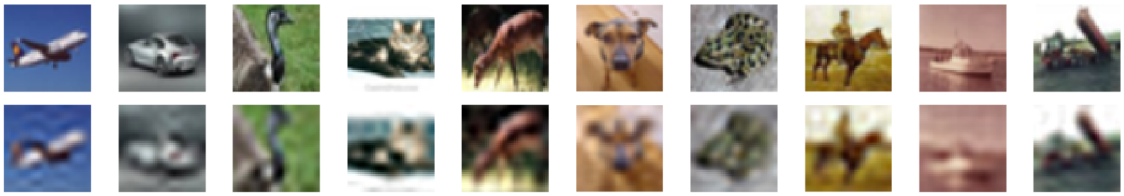

Figure 4.8: (Top) original and (bottom) decompressed CIFAR-10 color images from low-rank TT format.

Table 4.8: Experiments on CIFAR-10 color images for object classification. For the FC layer, the number of hidden units= 16807 and the maximal TT-rank= 2. The Number of channels×Batch Size are underlined.

Input Dimension	**Data (Relative) Approximation Error**	**% Original Data Size**	**Error Rate**
(4, 8, 8, 4, 3×100) (1, 1, 3, 7, 28, 1)	0.7 (low accuracy)	3.05%	42.5%
(4, 8, 8, 4, 3×100) (1, 2, 11, 55, 72, 1)	0.3 (high accuracy)	13.82%	39.5%
(4, 8, 3×100, 8, 4) (1, 1, 4, 7, 4, 1)	0.7	2.82%	43%
(4, 8, 3×100, 8, 4) (1, 2, 14, 16, 4, 1)	0.3	22.12%	38%

affected by the arithmetic operations in TT formats. This is provided that the low-rank assumption in TT format holds, for big data computing, this is almost always true because the data and model complexity is too high that local interactions always dominate, which can be well captured using low-rank TT. To obtain optimal compression and model performance, the data should be compressed in mini-batch size with low TT-ranks, larger batch size requires higher TT-ranks to maintain high model performance, this leads to suboptimal compression. Empirical results also show that increasing the number of hidden units of the weight matrix of a FC layer improve the generalization ability of the model, but is not affected much by

Table 4.9: Experiments on CIFAR-10 color images for object classification. For the FC layer, the number of hidden units= 16807 and the maximal TT-rank= 2. The Number of channels×Batch Size are underlined.

Input Mode : **Full TT-ranks :**	(4, 8, 3×100, 8, 4) (1, 4, 32, 32, 4, 1)	**Batch Size :**	100
Reduced TT-ranks of Data Inputs	**Maximal TT-rank of Data Inputs**	**% Original Data Size**	**Error Rate**
(1, 4, 5, 5, 4, 1)	5	2.56%	40%
(1, 4, 10, 10, 4, 1)	10	9.98%	38%
(1, 4, 15, 15, 4, 1)	15	22.30%	38%
(1, 4, 20, 20, 4, 1)	20	39.49%	38%
Input Dimension (max. TT-rank = 10)	**Compression Speed (Hz, images/s)**	**% Original Data Size**	**Error Rate**
(4, 8, 3×100, 8, 4)	1040	9.98%	38%
(4, 8, 3×500, 8, 4)	870	9.81%	41%
(4, 8, 3×1000, 8, 4)	780	9.79%	42%
(4, 8, 3×5000, 8, 4)	710	9.77%	47%
(4, 8, 3×10000, 8, 4)	660	9.77%	51%

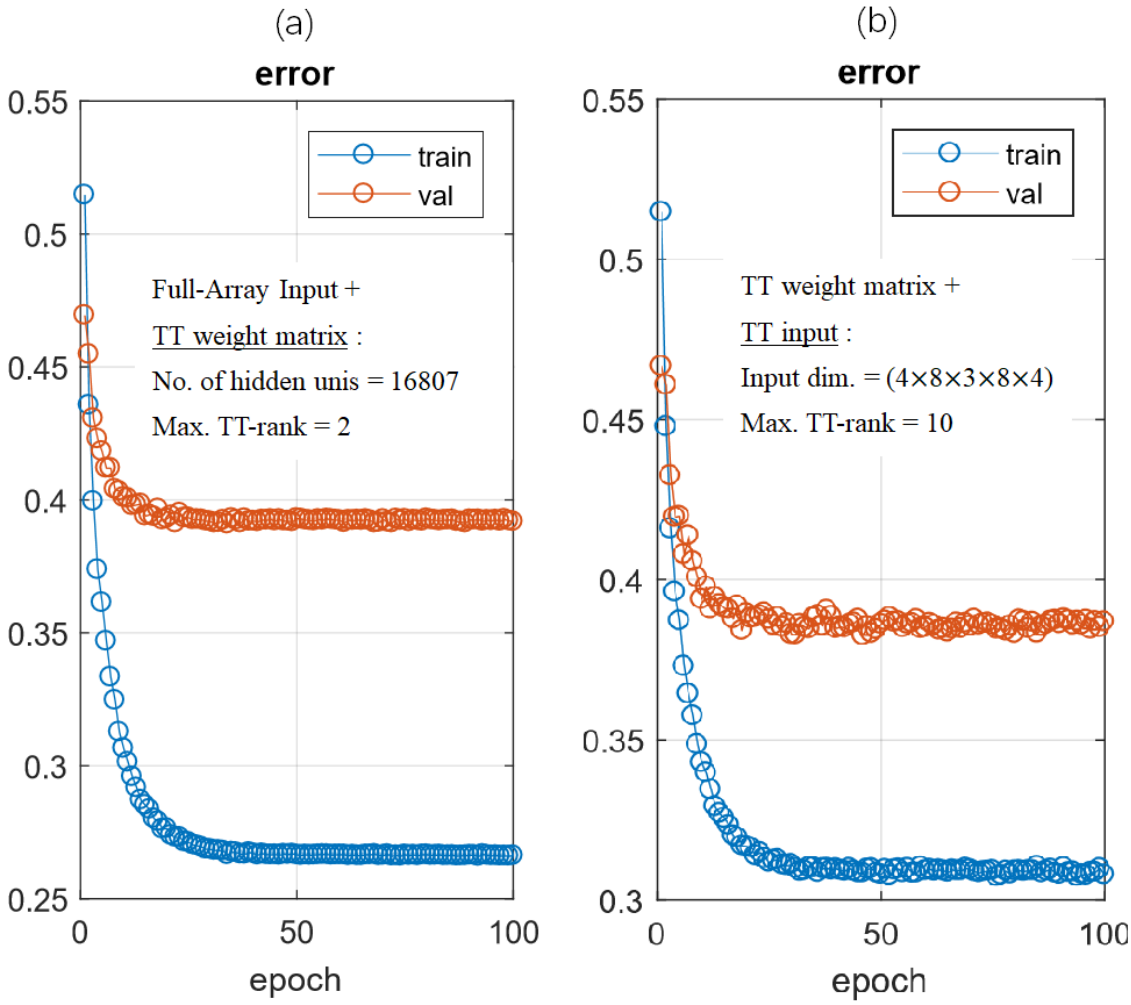

Figure 4.9: Training of FC network with CIFAR-10 input data in (a) full array and (b) TT format. The FC weight matrix is stored / computed in TT format and contains 16807 hidden units with maximal TT-rank = 2.

the TT-ranks. These findings have significant implications in the data storage and memory reduction especially for training deep learning architecture using resource-constrained devices. Our work paves the ways to train machine learning models on compressed input data with both the model and data stored / computed in low-rank tensor-network formats.

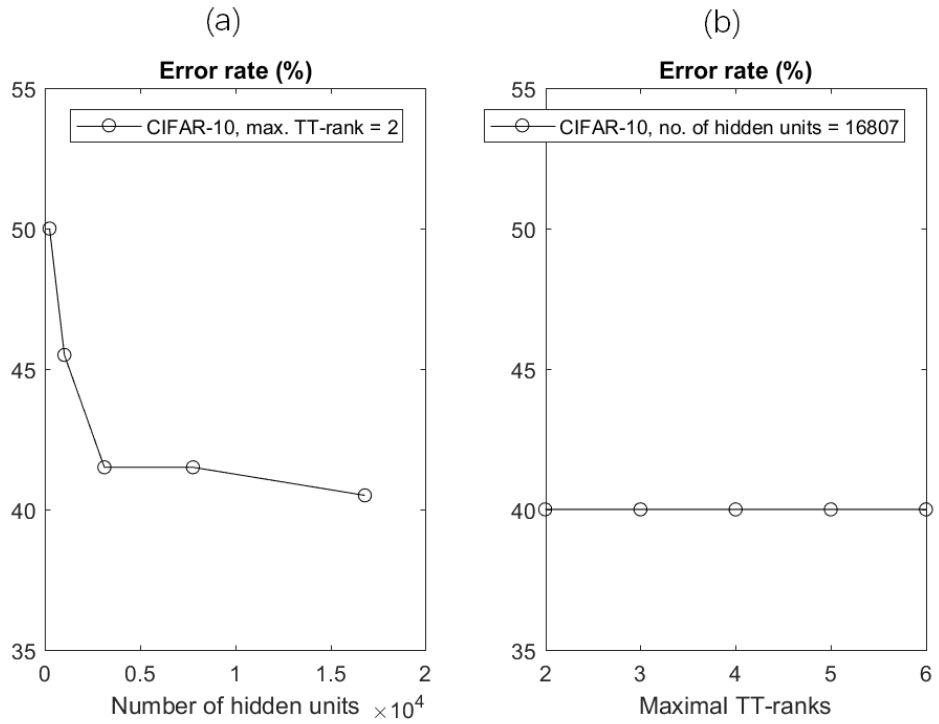

Figure 4.10: The classification error rate of CIFAR-10 object recognition in color images using different (a) number of hidden units and (b) maximal TT-ranks for the FC weight matrix stored / computed in TT format.

Chapter 5

Convolutional Neural Network with Transformed Input based on Tensor Network Decomposition

Many theoretical links have been established between different tensor network formats and machine learning models, therefore it is natural to ask whether we could use tensor analysis to characterize the sensitivity of machine learning models subject to adversarial perturbations. Here, we analyze adversarial examples that mislead convolutional neural networks to misclassification using subspace analysis based on singular value decomposition (SVD). The tensor study is extended to analyze higher-order tensors using tensor-train SVD (TT-SVD); it helps to explain the level of susceptibility of different datasets to adversarial attacks, the structural similarity of different adversarial attacks including global and localized attacks, and the efficacy of different adversarial defenses based on input transformation. An efficient and adaptive algorithm based on robust TT-SVD is then developed to detect strong and static adversarial attacks. We believe that the theory is generally applicable for deep learning models that carry out signal processing with subspace approximation. The intersection between tensor networks and deep learning provides a good prospect for further research and continues to bring new insights that flourish both areas.

The work in this chapter has been posted online in the *arXiv preprint arXiv:1812.02622*, 2018

5.1 Introduction

Deep learning models, despite their impressive performance, are highly susceptible to adversarial attacks that attempt to perturb the inputs in subtle manner (imperceptible or quasi-imperceptible) to achieve the adversary's motives such as targeted or untargeted misclassifications. This has serious implications especially because these models are increasingly being deployed for mission-critical and safety-critical applications such as autonomous vehicles and robotics. Existing theories on adversarial examples such as models' linearity and non-linearity hypotheses, dimensional analyses, etc. do not generalize to different adversarial attacks, these theories typically stem from local empirical observations and do not fully align with each other [113]. In this chapter, we conduct both theoretical and experimental studies using tensor networks as data structure for the input data of convolutional neural networks (CNNs) and analyze their robustness to adversarial attacks. Our contributions include:

- Analyze adversarial examples in CNNs using subspace analysis based on singular value decomposition (SVD).
- We quantify the robustness of different datasets to adversarial attacks, analyze the efficacy of different adversarial defense techniques based on input transformation and the structural similarity of different adversarial attacks.
- The tensor analysis is extended to analyze higher-order tensors with tensor-train SVD (TT-SVD) and an efficient algorithm is proposed to detect strong and static adversarial attacks (see Figure 5.1).

The organization of this chapter is as follows: Our proposed robust TT-SVD algorithm for adversarial detection is presented in Section 5.2. Section 5.3 covers the threat model, adversarial attacks and defenses in CNNs. Experimental studies are conducted and the results are discussed in Section 5.4. Section 5.5 and 5.6 provide the related work and discussion respectively.

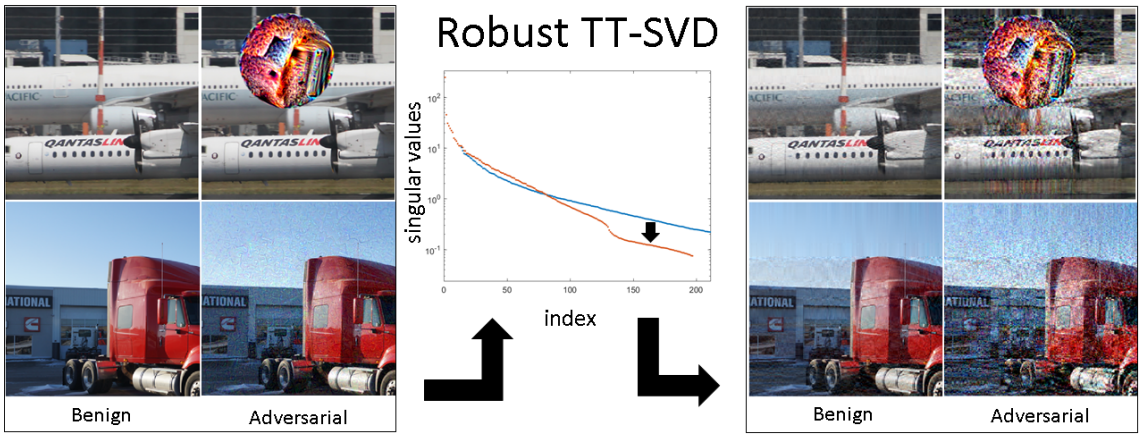

Figure 5.1: Global and localized adversarial examples, as diverse as their form can take, share similar structural properties in increasing the image roughness. This is because the sensitivity of subspace approximation by convolutional neural networks (CNNs) is controlled by the decay rate of singular values of the input image. The larger the decay rate, the smoother is the image input, and the more robust is the approximation. Our proposed robust TT-SVD algorithm linearly combines the singular values and vectors that fall within a (prescribed) bin to examine the robustness.

5.2 Robust TT-SVD Algorithm

Singular value decomposition (SVD) decomposes a matrix A into left and right singular vectors, the basis vectors are ranked by the amount of explained variation in A or the so-called singular values. Mathematically, SVD is given by $A \cong U\Sigma V^T$, where U and V are orthonormal matrices that contain the left and right singular vectors in their respective columns, the diagonal elements of Σ matrix contain the corresponding singular values. As shown in Figure 5.2 and 5.3, the distribution of singular values affect the luminance variation that accounts for textural changes such as smoothness / roughness change; the singular vectors form the basis images that encode the structural information of the original image [114]. Unlike discrete Fourier, cosine, or wavelet transform, the basis images of SVD are not fixed but adaptively-derived, thus allows better representation of the image structure. In the field of computer vision, SVD has been used for image denoising, compression, and forensic

such as steganography and watermarking. Higher-order SVD can help to disentangle the constituents factors or modes of image ensembles, e.g., TensorFaces [115] for facial images with different lighting conditions, viewpoint and poses. Closely related to our work is the use of SVD to extract features for visual quality assessment. Reference images are usually provided for comparison between images before and after processing [114], whereas no reference measure requires assumptions on the patches to be analyzed such as anisotropy [116]; all of the SVD-based image quality metrics work on grayscale images. Adversary does not provide the reference images for comparison; our work extends the SVD properties to higher-order tensors and analyze the robustness of correlation structure extracted from input data for image classification by CNNs.

Perturbation theory of SVD shows that the closeness of a singular value from its neighbors controls the sensitivity of its singular vector to perturbations [117]. It is further shown that when two singular values are close enough, the corresponding singular vectors are not unique, approximation of the change in subspace spanned by the corresponding singular vectors cannot be done with first-order perturbation theory but requires higher-order terms [118]. The singular values of real-life images follow exponential decay distribution, therefore the decay rate provides a scale-independent measure of the closeness of singular values and allows comparison of the robustness of multiscale correlation structure to perturbations between different datasets and image-processing techniques.

Unlike SVD, the transform kernels of convolutional neural networks (CNN) are learned from data and fixed after the training is complete; the CNN filters are not constrained to be orthogonal, but the domain of possible filter choices for both SVD and CNN spans the input space. Once training is complete, the input subspace spanned by CNN's filters is a subset of the whole input space. To make it more precise, suppose we split the input into irrelevant and relevant parts $A + \Delta A$ that activate particular CNN's neuron, both parts share a subset of all the singular vectors, i.e., $A + \Delta A = U(\Sigma + \Delta\Sigma)V^T$. In the case that $||\Sigma|| >> ||\Delta\Sigma||$, the robustness of the subspace spanned by the corresponding singular vectors of $\Delta\Sigma$ is determined by the closeness of the set of singular values and their neighbors.

Figure 5.2: The effect of transferring singular values between images. Top rightmost image shows the original image of a motorbike. Bottom row shows the change of luminance / texture after the transfer of singular value distribution from the top images. Bottom rightmost image shows the transfer of average of all the singular values of the top rightmost image.

There are a few loss terms in CNN: the approximation loss due to limited number of transform kernels, the rectification loss due to nonlinear activations, pooling, and dropout. The theoretical foundations of CNN in subspace approximation are laid down by Kuo [119, 120, 121, 122]. The research on subspace-based signal analysis using SVD is well-established among the signal processing community, in particular the sensitivity of subspace approximation with individual singular vectors when the singular values are close [123, 118]. If the singular values are well-separated, it can be shown that both the singular values and vectors change in the order of noise magnitude [124]. As will be shown in Section 5.4 Experiments, the decay rate of real-life images is small, therefore we hypothesize that the close separation between singular values gives rise to the adversarial examples in deep learning models, i.e., the neuron's activation patterns are not unique and extremely sensitive to input perturbations. We experimentally investigate (1) the decay rate of singular values of different datasets and compare their robustness against adversarial attacks, (2) the change in input's decay rate resulted from different adversarial attacks including global and localized attacks to show their structural similarity, and (3) the change in decay rate of transformed input from different adversarial defenses and how it affects their efficacy. These provide concrete evidence to support our hypothesis.

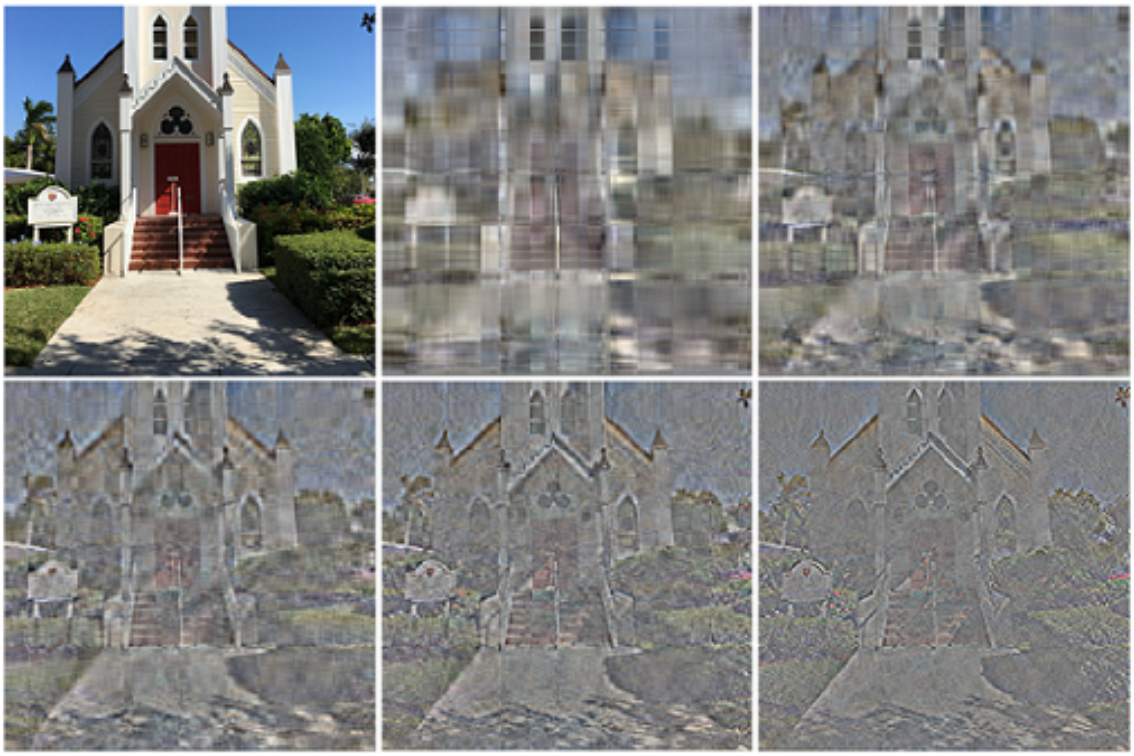

Figure 5.3: (Left to right, top to bottom) The original and reconstructed images by combining the left and right singular vectors from 10, 20, 30, 50, and 100 largest singular values. Singular vectors encode the multiscale correlation structure of the original image. It can be observed that large singular values are associated with large-scale variation (low frequency components) and vice versa for fine-scale variation (high frequency components).

Additionally, we propose a robust SVD algorithm (Algorithm 1) to generate quasi-distinct singular values from closely-separated ones. In doing so, the resultant singular vectors are more robust to input perturbations, the reconstructed images are used to detect strong and static adversarial attacks. First, the cumulative sum in ascending order of the singular values, $\hat{S}$ is divided into multiple bins with exponentially-distributed bin edges. The singular values and corresponding singular vectors that fall within a bin are summed up and linearly-combined respectively. This is in accordance with the theory of degenerate matrix SVD such that any normalized linear combination of singular vectors that share the same singular value is a valid singular vector of that singular value. By merging with TT-SVD (see Fig-

ure 5.4), the robust SVD algorithm can be extended to higher-order tensors such as 2D / 3D color images, videos, and hyperspectral images. The algorithm is efficient because the computational complexity increases linearly with the tensor mode size.

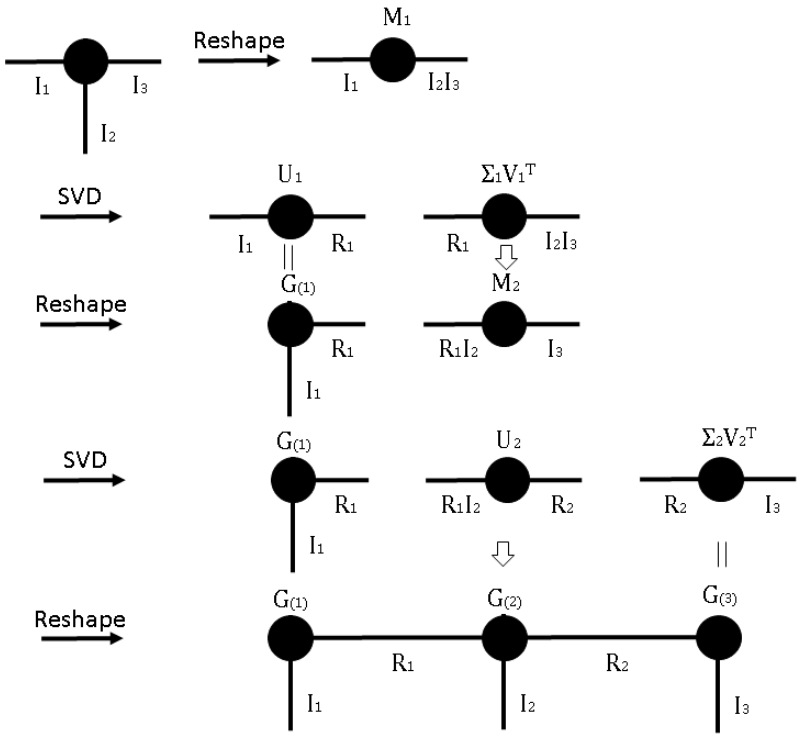

Figure 5.4: The TT-SVD algorithm for TT decomposition of a 3rd order tensor. M_k is the matricization of the subtensors. The ordering of indices I_k should be symmetric to get consistent SVD analyses (e.g. decay rate of singular values). For RGB color images, I_1: row indices, I_2: channels, and I_3: column indices. The decay rate is averaged over the sequences of SVD decomposition.

5.3 Adversarial Attacks and Defenses

Adversarial attacks can be targeted or untargeted; the choice / structure imposed on the input perturbations is typically shaped by an ℓ_p-norm distance metric and computed with gradient-based or optimization-based techniques with the objective to decrease the model performance. In our study, the adversarial strength is mea-

Algorithm 1 Robust SVD (can replace SVD in TT-SVD or TT-rounding algorithms, see Figure 5.4 for TT-SVD)

input: matrix A.

parameter: α, β of exponential distribution.

output: quasi-distinct singular values S and corresponding (linearly-combined) singular vectors U, V.

Initialize $\hat{S}_{binEdges} \leftarrow [0,\, \alpha \exp(\beta z)]$, $z \in \{0, 1, \cdots\}$

begin

$[U_0, S_0, V_0] \leftarrow svd(A)$

$\hat{S}_0 \leftarrow cumsum(S_0, \text{`}reverse'\text{)}$

$index \leftarrow bucketize(\hat{S}_0, \hat{S}_{binEdges})$

$S \leftarrow accumArray(S_0, index, \text{`}sum'\text{)}$

$U \leftarrow accumArray(U_0, index, \text{`}average'\text{)}$

$V \leftarrow accumArray(V_0, index, \text{`}average'\text{)}$

Rearrange S, U, V in descending order of S

Normalize $U \leftarrow \frac{U}{||U||_2}, V \leftarrow \frac{V}{||V||_2}$

(Optional) SVD rank truncation

Return S, U, V

end

sured by normalized ℓ_2-dissimilarity [125], the adversary is assumed to know only the CNN models but has no ability to influence them.

$$\ell_2 - dissimilarity = \frac{1}{N}\sum_{n=1}^{N}\frac{||x_n - x'_n||_2}{||x_n||_2} \tag{5.1}$$

where N is the sample size, x is the original image, and x' is the perturbed image.

Adversarial defenses can be broadly categorized into adversarial training and input transformations. Adversarial training requires prior knowledge of the type of attacks and train the models to differentiate them, therefore the amount of computational cost is much higher. Input transformations use either traditional image-processing techniques or generative models to remove adversarial examples; the technique is less expensive but susceptible to adaptive attacks who know the transformation techniques. We focus on input transformations that have been used against adversarial attacks in previous studies and explain the reason of their effectiveness based on our proposed theory.

5.4 Experiments

Datasets. MNIST [111] is a widely used dataset for digit classification that was introduced in 1998. It consists of 28×28 pixel grayscale images of handwritten digits. There are 10 classes (10 digits), $60,000$ training images, and $10,000$ testing images. Street View House Numbers (SVHN) [126] is an MNIST-like 32×32 pixel color images consists of $73,257$ training images and 26032 testing images for 10 classes (10 digits). They are taken from Google Street View images and usually corrupted by natural phenomena like severe blur, distortion, and illumination effects on top of wide style and font variations [126]. CIFAR-10 [112] is a dataset released in 2009 that consists of 32×32 pixel color images of 10 mutually exclusive classes with $50,000$ training images and $10,000$ test images. ImageNet [127] is used for large-scale evaluation, it was first introduced in 2010 and the dataset stabilized in 2012. ImageNet contains color images of at least 256×256 pixel with 1000 classes; only the validation set consists of $50,000$ images (50 per class) are used. Section A.2 benchmarks the TN storage complexity, algorithmic efficiency, and model perfor-

mance using these datasets. The privacy and security issues in image recognition are studied using 1000 development (color) images of 299×299 pixel released in NIPS 2017 Adversarial Attacks and Defenses Competition. This dataset is referred to as "ImageNet" [127] in Section 5.4.1 and 5.4.2 due to their similar task difficulty.

Experimental Setup. Adversarial defenses based on input transformation using color bit-depth reduction [128], cropping-rescaling, median, gaussian, and non-local means [129] filters are coded using Matlab functions and toolbox. Image quilting [130], total variance minimization (TVM) [131], and JPEG compression [132] are Python implementations by Guo et al. [125]. Adversarial attacks are generated using Tensorflow 1.4.0 [133, 134] and Cleverhans v2.1.0 [135]. The CNN model for MNIST, SVHN, and CIFAR-10 is an all convolutional net [136] taken from Cleverhans model zoo with 2 convolutional layers (64 filters), 1 average pooling layer, followed by classification layer. The ImageNet classification is using Inception v3 [137]. Fast Gradient Method (FGM) [138], Basic Iteration Method (BIM) [139], and Deep Fool (DF) [140] are gradient-based attacks whereas Carlini-Wagner (CW) [141] and Elastic Net Method (EAD) [142] are optimization-based adversarial attacks. We follow closely the method proposed by Guo et al. [125] to generate adversarial examples with increasing adversarial strength. FGM and BIM are done by adjusting the hyperparameters; DF and CW perturbations are amplified to increase the normalized ℓ_2-dissimilarity after successful attacks. Universal Perturbations (UP) [143] and Adversarial Patch (AP) [144] are strong static attacks which can be image- and network-agnostic. Only the image-agnostic case is considered here. UP is generated using BIM in each iteration; whereas AP is taken from the implementations by Brown et al. [144]. Different from other adversarial techniques that manipulate pixels within ℓ_p distance which may sometimes produce noticeable artifacts, spatially transformed adversarial example (stAdv) [145] is a new approach that generates realistic adversarial examples with smooth image deformation; their code is made publicly available by Dumont et al. [146]. In our study, AP and stAdv are targeted attacks with "toaster" and randomized targets respectively, others are untargeted attacks. Because stAdv deforms images smoothly, which stands in contrast to our proposed theory that adversarial examples tend to increase the image roughness,

we report the stAdv hyperparameter that regularizes the local distortion characterized by a flow field and adversarial loss used in our study, i.e., MNIST (10^{-2}), SVHN (10^{-3}), CIFAR-10 (10^{-3}), and ImageNet (10^{-6}). Notice that the regularization decreases with dataset complexity, this means that it is much harder to generate adversarial examples with smooth deformation for complex images.

5.4.1 Robustness against Adversarial Attacks

The decay rate of leading singular values determines the robustness of subspace approximation by CNNs. Table 5.1 tabulates the decay rate for 1000 randomly-selected images from MNIST, SVHN, CIFAR-10, and ImageNet datasets using the 5th-25th largest singular values. Coincidentally, the decay rate correlates well to the datasets' complexity. Figure 5.5 shows the robustness of the datasets to adversarial attacks. It can be observed that the steeper the dataset's TT-SVD slope, the more robust the dataset to a wide variety of different adversarial attacks. In particular, input perturbations by stAdv is done by smooth deformation; it requires much higher adversarial strength for successful attacks compared to other techniques. This agrees with our theory that adversarial examples tend to decrease the decay rate of singular values to increase sensitivity of the subspace approximation by CNNs, hence increase the image roughness as a result. Table 5.2 shows that strong adversarial attacks flatten the TT-SVD slope. Effectiveness of defenses based on input transformation has been studied before, our theory explains the reason why spatial smoothing techniques provide more resistance to adversarial attacks, as shown in [125, 128]. This is because spatial smoothing steepen the TT-SVD slope (see Table 5.3), hence reduce the sensitivity of subspace approximation by CNNs.

5.4.2 Detect Strong and Static Adversarial Attacks

As shown in Table 5.4, the detection of strong static attacks using our proposed robust TT-SVD algorithm only requires bounding the ℓ_2-norm between image before and after reconstruction. Currently, the proposed algorithm works well if the image consists of "simple" correlation structure (high SVD's decay rate), e.g., single object

Datasets	Image Size	Train/Test	TT-SVD Slope
MNIST (grayscale)	28x28x1	60,000/10,000	-0.4 ± 0.5
SVHN (color)	32x32x3	73,257/26,032	-0.38 ± 0.12
CIFAR10 (color)	32x32x3	50,000/10,000	-0.17 ± 0.04
CIFAR10 (grayscale)	32x32x1	50,000/10,000	-0.18 ± 0.04
ImageNet (color)	299x299x3	NA/50,000	-0.072 ± 0.019
ImageNet (grayscale)	299x299x1	NA/50,000	-0.073 ± 0.020
ImageNet (color) + 30% noise	299x299x3	NA/50,000	-0.061 ± 0.016

Table 5.1: Robustness of correlation structure of different datasets measured by the TT-SVD slope. Steeper slope means the separation between singular values are larger, hence the subspace approximation by CNNs is more robust to input perturbations. The standard deviation of the TT-SVD slope measures the variability of the estimated value. Notice that adding noise flattens the TT-SVD slope, hence decreases the robustness of correlation structure.

recognition. Images with complex variation or cluttered scene may consider pre-processing with spatial smoothing and cropping-rescaling [147] respectively before using the proposed algorithm. Existing adversarial detection techniques rely on (1) sample statistics, (2) prediction inconsistency, and (3) training a detector. Their shortcomings include (1) is ineffective, (2) needs to process a batch of images each time, (3) needs to have labeled data and model training takes time [128]. The robust TT-SVD algorithm provides a new way to detect adversarial examples directly on the input; the algorithm is adaptive in nature because the singular values / vectors are adaptively-derived.

5.5 Related Work

Adversarial examples have been extensively studied in recent years due to the wide-spread applications of deep learning, especially in safety-critical applications. There

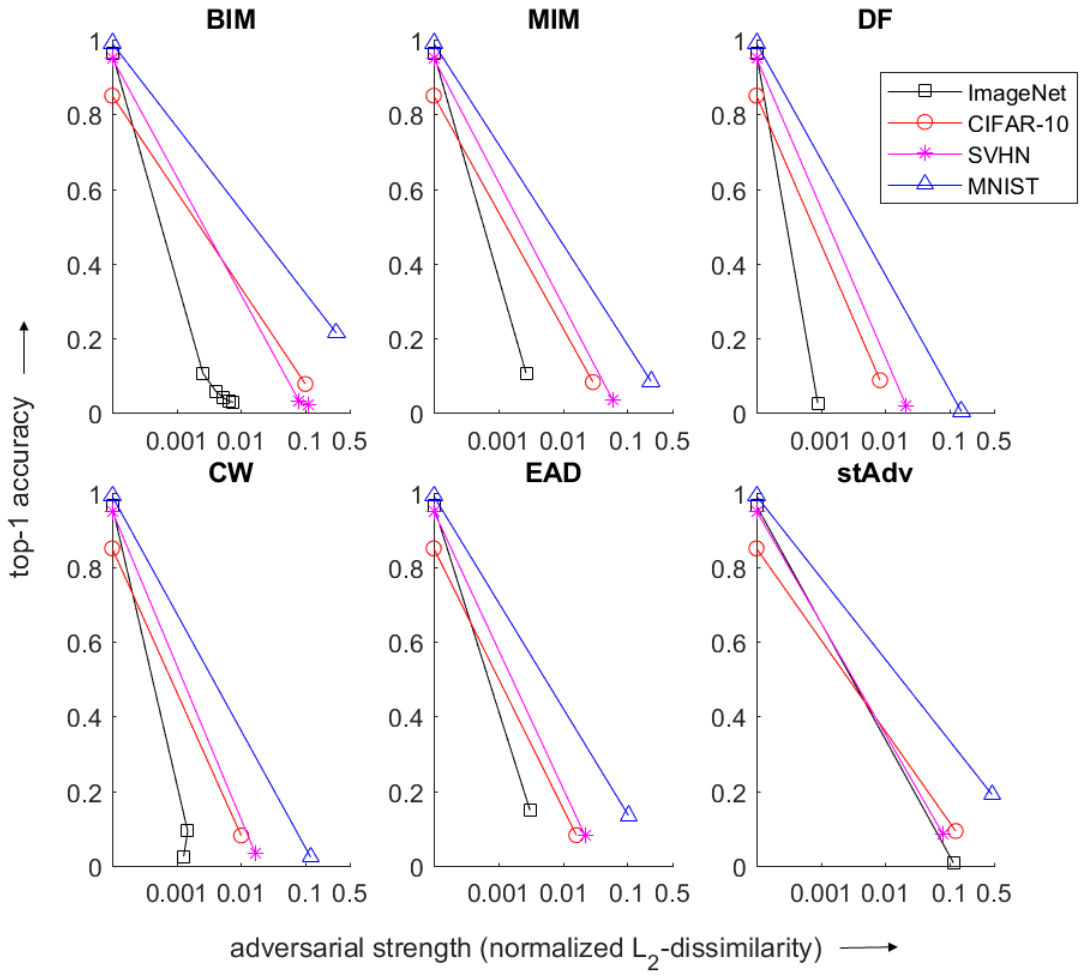

Figure 5.5: Model accuracy of datasets under adversarial attacks with increasing adversarial strength measured in normalized ℓ_2-dissimilarity. Notice the robustness of the datasets to adversarial attacks, i.e., MNIST > SVHN $\gtrsim$ CIFAR-10 > ImageNet.

are already a few review papers published in these areas including [148, 113, 149, 150, 151]. A number of theoretical explanations have been proposed for adversarial examples, such as boundary tilting, local or piecewise-linearity, concentration of measure in high dimensions, and statistical fluctuations in training data [152, 153, 154, 155, 156, 157]. Recent research also suggests that adversarial vulnerability may be due to the existence of non-robust features in the input data that are highly predictive, yet brittle and incomprehensible to humans [158]. However, there is currently no robust and adaptive adversarial defense solution for high-dimensional data such as ImageNet. For example, robust optimization proposed in [159] is useful

Adversarial Attacks (Strength)	**Top-1 Accuracy**	**TT-SVD Slope**
FGM, ℓ_∞ (0.0812)	0.25	$-0.068 \pm 0.017 \uparrow$
FGM, ℓ_2 (0.0773)	0.25	$-0.067 \pm 0.016 \uparrow$
BIM, ℓ_∞ (0.0778)	0.016	$-0.069 \pm 0.017 \uparrow$
BIM, ℓ_2 (0.0784)	0.014	$-0.068 \pm 0.016 \uparrow$
CW, ℓ_2 (0.0775)	0.17	$-0.067 \pm 0.017 \uparrow$
DF, ℓ_2 (0.0765)	0.134	$-0.066 \pm 0.016 \uparrow$
UP, ℓ_∞ (0.0832)	0.196	$-0.068 \pm 0.017 \uparrow$
UP, ℓ_2 (0.0787)	0.196	$-0.068 \pm 0.017 \uparrow$
AP (0.64)	0.0	$-0.062 \pm 0.008 \uparrow$

Table 5.2: Similar to Table 5.1 but for adversarial attacks. Adversarial strength is measured by normalized ℓ_2-dissimilarity. The upwards arrow means that the technique flattens the TT-SVD slope by more than 10% of the slope variability, vice versa for downwards arrow. The original slope value is -0.072 ± 0.019.

in protecting low-dimensional dataset such as MNIST and CIFAR-10, but it cannot be straightforwardly applied to protect high-dimensional inputs and large models from adversarial perturbations. Adversarial training [153, 139, 160] is costly in terms of computation in order to generate the adversarial examples needed to train a deep neural network that is robust to various kinds of adversarial attacks; new adversarial examples can still be computed based on the adversarially-trained network. Most adversarial defense techniques based on input transformations have been shown to be ineffective, especially in the white-box setting [161]. The state-of-the-art defense combines adversarial training with randomization and feature denoising techniques to provide the adversarial robustness [162, 163]. Another promising adversarial detection approach is based on feature squeezing [128]. To ensure the completeness of security evaluations, a principled approach in evaluating adversarial robustness is proposed in [164] to avoid the common pitfalls in this area.

Adversarial Defenses based on Input Transformation		TT-SVD Slope
Spatial Smoothing		
Cropping-Rescaling [147]		$-0.089 \pm 0.026 \downarrow$
Image Quilting [130]		$-0.079 \pm 0.019 \downarrow$
Median Filter [128]		$-0.080 \pm 0.020 \downarrow$
Gaussian Filter [128]		$-0.088 \pm 0.022 \downarrow$
Non-Local Means Filter [129]		$-0.081 \pm 0.021 \downarrow$
Total Variance Minimization [131]		$-0.081 \pm 0.020 \downarrow$
Amplitude Quantization		
Color Bit-Depth Reduction [128]	4-bit	-0.071 ± 0.018
	3-bit	$-0.069 \pm 0.017 \uparrow$
Frequency-based Compression		
JPEG [132] Compression	quality level	
	75	-0.072 ± 0.019
	50	-0.073 ± 0.019
	25	-0.073 ± 0.019
	5	-0.071 ± 0.017

Table 5.3: Similar to Table 5.2 but for adversarial defenses.

5.6 Discussion

At first glance, it may seems obvious that adversarial attacks increase the image roughness, therefore natural choices for adversarial defense based on input transformation should incorporate different levels of spatial smoothing, e.g., local, non-local, edge-preserving, etc. Further experiments show that the robustness against adversarial attacks differ between datasets with different decay rate of SVD singular values. This suggests that there is a deeper level connections between adversarial examples in deep learning models and SVD's decay rate. Perturbation theory of SVD shows that the closeness between singular values controls the sensitivity of subspace approximation by CNNs. Empirical results show that real-life images typically have

Adversarial Attacks	**Norm**	**Adversarial Strength**	**Detection Rate**
FGM [138]	ℓ_∞	0.071	0.997
	ℓ_2	0.043	0.956
BIM [139]	ℓ_∞	0.069	0.997
	ℓ_2	0.048	0.985
CW [141]	ℓ_2	0.046	0.944
DF [140]	ℓ_2	0.037	0.932
UP [143]	ℓ_∞	0.052	0.997
	ℓ_2	0.056	0.994
AP [144]	-	0.045	0.991

Table 5.4: Detection rate of strong and static adversarial attacks. The adversarial strength is measured by normalized ℓ_2-dissimilarity. The slope of cumulative distribution of TT-SVD is set to -0.03 and truncation error $\leqslant 0.03$. 337 out of 1000 development images from NIPS 2017 Adversarial Attacks and Defenses Competition are selected by setting the initial ℓ_2-norm < 1000. The ℓ_2-dissimilarity of the 337 samples is 0.01. Adversarial attacks are detected when ℓ_2-norm > 1000.

slow SVD's decay rate, which explains the cause of adversarial examples in CNNs. The subspace approximation is more stable to input perturbations if the approximation loss and rectification loss of CNNs can be reduced. However, this may not be possible because the non-linear activation units in CNNs or deep learning models in general are crucial in achieving the high model performance. Therefore, input transformation, adversarial training, and adversarial detection algorithms may provide better solutions to adversarial attacks.

Chapter 6

Protecting Big Data Privacy Using Randomized Tensor Network Decomposition and Dispersed Tensor Computation

Data privacy is an important issue for organizations and enterprises to securely outsource data storage, sharing, and computation on clouds / fogs. However, data encryption is complicated in terms of the key management and distribution; existing secure computation techniques are expensive in terms of computational / communication cost and therefore do not scale to big data computation. Tensor network decomposition and distributed tensor computation have been widely used in signal processing and machine learning for dimensionality reduction and large-scale optimization. However, the potential of distributed tensor networks for big data privacy preservation have not been considered before, this motivates the current study. Our primary intuition is that tensor network representations are mathematically non-unique, unlinkable, and uninterpretable; tensor network representations naturally support a range of multilinear operations for compressed and distributed / dispersed computation. Therefore, we propose randomized algorithms to decompose big data into randomized tensor network representations and analyze the privacy leakage for 1D to 3D data tensors. The randomness mainly comes from the com-

plex structural information commonly found in big data; randomization is based on controlled perturbation applied to the tensor blocks prior to decomposition. The distributed tensor representations are dispersed on multiple clouds / fogs or servers / devices with metadata privacy, this provides both distributed trust and management to seamlessly secure big data storage, communication, sharing, and computation. Experiments show that the proposed randomization techniques are helpful for big data anonymization and efficient for big data storage and computation.

6.1 Introduction

Big data generated from sensor networks or Internet-of-Things are essential for machine learning, in particular deep learning, in order to train cutting-edge intelligent systems for real-time decision making and precision analytics. However, big data may contain proprietary information or personal information such as location, health, emotion, and preference information of individuals which requires proper encryption and access control to protect users' privacy. Symmetric and asymmetric key cryptosystems work by adding entropy / disorderliness into data using encryption algorithms and (pseudo-)random number generator so that unauthorized users cannot find pattern from the ciphertext and decipher them, however, higher computational cost is usually incurred with added functionality such as secure operations (addition / multiplication) in homomorphic encryption and asymmetric key distribution in public key encryption. The pain point of encryption nowadays is the complicated key management and distribution especially when organizations or enterprises are undergoing digital transformation to complex computing environments such as multi- / hybrid-cloud and mobile environments. The field of secure multi-party computation (SMPC) originates from Yao's garbled circuit in the 1980s where untrusted parties jointly compute a function without disclosing their private inputs [165]. SMPC has evolved and adopts distributed trust paradigm in recent years given the complex computing environments, increasing attack surfaces, and recurring security breaches; the secret shares are now distributed among multiple computing nodes in order to be information-theoretically secure, i.e., secure against

adversary with unbounded computational resources. SMPC computing primitives include secret sharing, garbled circuit, and homomorphic encryption, the supported secure operations are arithmetic, boolean, comparison, and bitwise operations; other secure building blocks that are routinely being used in SMPC are oblivious transfer, commitment scheme, and zero-knowledge proof [166, 167]. It is well-known that fully homomorphic encryption [168] suffers from high computational complexity, making it not practical to compute complex functions during operational deployment; secret sharing and garbled circuit are expensive in terms of communication complexity and therefore routinely operate with low-latency networks, furthermore, garbled circuit involves symmetric encryption during the online phase. The communication complexity of existing SMPC protocols can incur runtime delay from an order of magnitude using local-area network (LAN) setting to several orders using wide-area network (WAN) setting.

The quest for scalability calls for innovative data security solutions which not only simplify privacy management, but also provide seamless integration between privacy-preserving big data storage / communication and computation / sharing. We believe this requires introducing a new secure computation primitive that is based on distributed / dispersed tensor network computation. However, TN increases the functionality and performance of multi-party computation at the expense of security. In contrast to classical encryption and SMPC techniques which are based on modular arithmetic and works on fixed-point representations; TN naturally supports both floating-point and fixed-point arithmetics / operations. Furthermore, TN representations allow further compression unlike encrypted computation techniques, which generally increase the storage and communication overhead. Therefore, this generally makes encrypted computation not scalable for big data processsing; whereas data compression prior to encryption usually makes the data representations lose some functionalities such as encrypted computation on the original data. With the impressive track records of distributed TNs in large-scale scientific computing and big data analytics, we propose a novel secret-sharing scheme based on tensor networks and investigate its feasibility for privacy-preserving big data distributed applications. Our contributions are as follows:

- Propose an arithmetic secret-sharing scheme based on randomized tensor network decomposition and dispersed tensor multilinear operations. The randomization is done by controlled perturbation applied to the data blocks prior to singular value decomposition (SVD), which results in randomized tensor blocks after decomposition due to the complex structural information in big data. The perturbation technique can be easily adapted in various TN decomposition algorithms to generate randomized TN representations.

- Empirically analyze the privacy leakage of the randomized TN representations for 1D to 3D datasets and propose mitigation techniques to reduce the privacy leakage. The data compressibility and algorithmic efficiency of the proposed randomized TN algorithms have also been investigated.

The organization of this work is as follows: Section 6.2 covers related work on state-of-the-art privacy-preserving techniques and secure tensor decompositions. Sections 6.3 and 6.4 explain the security model and our proposed randomized tensor dispersed computing approach for big data privacy preservation. Section 6.5 conducts experimental studies to benchmark the security, efficiency, and performance of the proposed approach. Section 6.6 discusses the implications, limitations, and potential extension of this research study.

6.2 Related Work

Secret-Sharing schemes provide information-theoretical security at the expense of high storage and communication cost. Here, we review practical secret-sharing schemes for big data protection that provide only computational security but offer high storage / computational efficiency. Krawczyk [169] proposes the first computational secret sharing scheme by encrypting the data using symmetric encryption with randomly-generated key, the encrypted data is divided into multiple blocks using Rabin's information dispersal algorithm; whereas the encryption / decryption key is split using Shamir's secret-sharing scheme such that collecting a certain threshold number of blocks is enough for secret reconstruction. Since then, many

variants of the computational secret-sharing scheme have been proposed to improve the data security, data redundancy / error resistance, performance, data integrity / authentication, fragment size, data deduplication, and location management with different machine trustworthiness [170, 171, 172]. Most notably, the key exposure problem is a practical issue to address due to usage of weak key for encryption, key reuse, or key leakage. The All-Or-Nothing Transform (AONT) introduced by Rivest [173] solves the key exposure problem by building dependency between the fragments such that acquiring only the key without all the fragments will not lead to immediate information leakage, a recent review on AONT can be found in [174]. Furthermore, access revocation is greatly simplified by re-encrypting only one data fragment with a fresh encryption key, which significantly reduces the transmission cost [175, 176]. However, utility of such encrypted data is quite limited such as search, update, and computation cannot be performed without reconstructing the original data [177, 178].

Database Fragmentation or Data Splitting [179] aim to provide functionality-preserving data protection for data storage on clouds. Sensitive data is fragmented in clear form in separate storage locations such that each data fragment does not reveal confidential information linked to a subject. Data splitting can be done at byte, semantic, or attribute level. Byte-level fragmentation splits the sensitive files and performs shifting and recombination of the bytes to form fixed data blocks before storing on different cloud locations, this is particularly suitable for binary or multimedia files, which are usually stored but not processed by cloud. Semantically-grounded splitting mechanism is well-suited for unstructured data such as textual data, it can provide keyword search for online document, email, and messaging applications. For example, a recent work by [180] automatically detects and splits the sets of textual entities that may disclose sensitive information by analysing the semantics they convey and their semantic dependencies. Attribute-level splitting such as vertical splitting [181] is very useful for statistical databases because usually is the combination of several risky attributes that may lead to personal re-identification. Computation on attributes stored on single fragment in vertial splitting is fast and straightforward, e.g., addition, updating, and uni-valued statistics such as mean and

variance. However, data splitting requires a proxy server to manage the locations, queries, and operations on the data fragments, this becomes the single point of failure if users cannot access the metadata stored at the proxy.

Data Anonymization is perhaps the simplest low-cost solution that is widely adopted nowadays for secure data sharing within and across enterprises for diverse applications, including machine learning. Data anonymization techniques cover both the removal of personally-idenfiable information (e.g., using hashing or masking techniques) and data randomization / perturbation techniques (e.g., random noise, permutation, transformation) [179]. The random components or functions have to be carefully designed to preserve important information in the training dataset and ensure model performance. A recent systematic survey of different privacy metrics that have been proposed over the years can be found in [182]. These privacy metrics are based on information theory, data similarity, indistinguishability measures, and adversary's success probability; to choose a suitable privacy metric for a particular setting depends on the adversarial model, data sources, information available to compute the metric and the properties to measure [182]. Differential privacy (DP) [183, 184] is a mathematical framework to rigorously quantify the amount of information leaked during operations on a statistical database or machine learning [185, 186, 187, 188, 189], DP is a proven privacy-preserving technique widely adopted by the industry. A recent promising data anonymization approach is to generate synthetic data [190] that resembles the statistical distribution or behavior observed in the original datasets using generative machine learning models such as generative adversarial networks [191] and computer simulations (e.g., [192]), however, these models / simulations are application-specific (i.e., depend on the training dataset or physical models) and any analysis on the synthetic data has to be verified over the real dataset for validation.

Although privacy-preserving matrix and tensor decomposition techniques have been well studied in the literature [193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203], distributed / dispersed TN representations and computation have not been proposed for big data privacy preservation, which motivates the current study. Different from data anonymization techniques, tensor decompositions are fully reversible and

compressible, the reconstruction accuracy can be either lossy or near-lossless [204]. Unlike data splitting, TN does not require proxy server to manage the metadata, but offers much better utility of the decomposed data at the expense of higher privacy leakage compared to computational secret-sharing schemes. To process big data, randomized mapping or projection techniques utilize a projection matrix such as Gaussian, Rademacher, and random orthonormal matrices [205, 36] to project the data tensor to much smaller tensor size before applying tensor decompositions. Randomized sampling techniques such as fiber subset selection or tensor cross approximation choose a small subset of tensor fibers that approximate the entire data tensor well, e.g., measured using quasi-optimal maximal volume or modulus determinant of the submatrix so that the matrix cross-approximation is close to the optimal SVD solution [37, 33]. Existing randomized mapping / projection and randomized sampling algorithms are useful for big data tensor decompositions to fit the data size into existing memory requirements, the decomposed tensor blocks are usually compressed with lossy reconstruction accuracy, which is different from our proposed randomized tensor decompositions. The randomness of the decomposed tensor blocks is also limited by the distribution of the projection matrix and sampling process to ensure small error bounds, whereas our proposed algorithms randomly disperse the complex structural information of big data into the tensor cores by applying large-but-controlled perturbations during the sequential matrix decomposition process in tensor decomposition algorithms. The time complexity is also much lower compared to randomized projection / mapping algorithms and can be easily adapted into existing TN algorithms. Nonetheless, the proposed tensor perturbation techniques can be easily combined with existing randomized projection / sampling algorithms for big data processing and privacy protection. Tensor decompositions have been widely used for dimensionality reduction of big data, however, research on tensor network coding schemes are lagging behind, only a few publications are found at the time of writing [204, 206, 207].

6.3 Threat Model and Security

The secure storage and computation by a client are outsourced to a set of untrusted but non-colluding servers $S_1, S_2, ..., S_n$, the client secret share their inputs among the servers in the initial setup phase, the servers then proceed to securely store (e.g., with encryption), compute and communicate using dispersed TN computation protocols. The servers run on different software stacks to minimize the chance that they all become vulnerable to the exploit available to malware attacks and can be operated under different sub-organizations to minimize insider threats. Given the cloud scenario, the secret shares can be distributed to multiple virtual instances provided by the same cloud service provider (CSP) or to different clouds (e.g., multi-cloud or hybrid-cloud environments). We assume a semi-honest adversary $\mathbb{A}$ (or so-called honest-but-curious adversary) who is able to corrupt any subset of the clients and at most $n-1$ servers at any point of time. Different from encrypted data processing, our security definition requires an adversary to learn only partial information of the client's input but not knowing the sensitive information from the process. The privacy leakage is measured based on information-theoretic and similarity-based privacy metrics. Secret-sharing scheme based on TN is asymmetric to each server, i.e., each server contains index-specific information. As shown in Sections 6.4 and 6.5, each of the TN representations requires high data complexity (or high tensor-rank complexity) to be privacy-preserving in multi-party setting.

6.4 Secret-Sharing Scheme Based on Distributed Tensor Networks

In this section, we propose a novel secret-sharing scheme based on dispersed TN representations / operations to seamlessly secure big data storage, communication, sharing, and computation. TN decomposes data chunk at the semantic level, each of the decomposed tensor block which contains latent information are randomly distributed among multiple non-colluding servers. The success of multi-way component analysis can be attributed to the existence of efficient algorithms for matrix and

tensor decomposition and the possibility to extract components with physical meaning by imposing constraints such as sparsity, orthogonality, smoothness, and non-negativity [34]. Our primary intuition is that higher-order tensor decompositions are in general non-unique, each tensor core or factor matrix contains index-specific information which are unlinkable and uninterpretable due to non-uniqueness of the decompositions, therefore they are commonly used for dimensionality reduction and compressed computation.

Several basic tensor models are described here within the multi-party computation setting to enhance the privacy protection of the original tensor. Here, we propose randomized algorithm based on perturbation technique to decompose each data chunk into randomized tensor blocks, each of the tensor blocks can be re-randomized again using tensor-rounding algorithm after performing tensor distributed, multilinear operations to reduce the tensor-rank complexity for storage and computational efficiency.

Tucker decomposition (TD) [208] is a natural extension of matrix Singular Value Decomposition (SVD) into high-dimensional tensor. TD captures the interactions between the latent factors $\mathbf{U}$ (from SVD of mode-n matricization of a tensor) using a core tensor $\mathcal{G}$ that reflects and ranks the major subspace variations in each mode of the original tensor. For a third-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, TD can be defined as follows using different tensor operations:

$$\begin{aligned} \mathcal{A}(i_1, i_2, i_3) &\cong \mathcal{G} \times_1 \langle \mathbf{U}_1 \rangle_1 \times_2 \langle \mathbf{U}_2 \rangle_2 \times_3 \langle \mathbf{U}_3 \rangle_3 \\ vec(\mathcal{A}) &\cong (\langle \mathbf{U}_3 \rangle_3 \otimes \langle \mathbf{U}_2 \rangle_2 \otimes \langle \mathbf{U}_1 \rangle_1)\, vec(\mathcal{G}) \end{aligned} \tag{6.1}$$

$\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ is a 3-dimensional core tensor, $\mathbf{U}_k \in \mathbb{R}^{I_k \times R_k}$, $k \in \{1, 2, 3\}$ are the factor matrices, $\times_n$ is the n-mode product, $\otimes$ is the Kronecker product, $vec(\cdot)$ is the vectorization operator (see the definitions in [208]), $\langle \cdot \rangle_\ell$ denotes the private share stored in server ℓ. Here, $\mathcal{G}$ is a shared core for exchange between servers to perform tensor computation schemes. TD is non-unique because the latent factors can be rotated without affecting the reconstruction error, however, TD yields a good low-rank approximation of a tensor in terms of squared error. Canonical Polyadic (CP) decomposition is a special case of TD when $\mathcal{G}$ is superdiagonal. CP is very popular in signal processing due to its uniqueness guerantee and ease of interpretation [16],

however, these properties also make CP unsuitable for privacy preservation.

Hierarchical Tucker (HT) decomposition [209, 210] was proposed to reduce the memory requirements of TD. HT approximates well higher-order tensors ($N >> 3$) without suffering from the curse of dimensionality. HT recursively splits the modes of a tensor based on a binary tree hierarchy such that each node contains a subset of the modes. Therefore, HT requires a priori knowledge of a binary tree of matricizations of the tensor, HT is defined as follows:

$$\mathbf{U}_t \cong (\mathbf{U}_{t_l} \otimes \mathbf{U}_{t_r}) \langle \mathbf{B}_t \rangle_t \tag{6.2}$$

$\mathbf{B}_t$ are the "transfer" core tensors (or internal nodes) reshaped into $R_{t_l} R_{t_r} \times R_t$ matrix, $\mathbf{U}_t$ contains the R_t left singular vectors of the original tensor, t_l and t_r correspond to the left and right child nodes respectively. The leaf nodes $\langle \mathbf{U}_1 \rangle_1, \langle \mathbf{U}_2 \rangle_2, \ldots, \langle \mathbf{U}_N \rangle_N$ contain the latent factors and should be stored distributedly to ensure privacy preservation. HT is particularly useful when the application provides an intuitive and natural hierarchy over the physical modes.

Tensor-Train (TT) [211] decomposes a given tensor into a series or cascade of connected core tensors, therefore TT can be interpreted as a special case of HT. TT core tensors are connected through a common reduced mode or TT-rank, R_k. TT is defined as follows:

$$\mathcal{A}(i_1, i_2, i_3) \cong \langle \mathbf{G}[i_1] \rangle_1 \times \langle \mathbf{G}[i_2] \rangle_2 \times \langle \mathbf{G}[i_3] \rangle_3 \tag{6.3}$$

where $\mathbf{G}[i_k]$ is a $R_{k-1} \times R_k$ matrix with $R_0 = R_3 = 1$, and $\times$ is the matrix multiplication operation. TT format and its variants are very useful owing to their flexibilty for a number of distributed, multilinear operations [4] and the possibility to convert other basic tensor models (e.g., CP, TD, HT) into TT format [34]. Similar properties apply to tensor chain or tensor ring format (TR) [212], which is a linear combination of TT formats, i.e., $R_1 = R_3 > 1$. TR representations are more generalized and powerful compared to TT representations [212]; whereas extended TT further decomposes the TT-cores into smaller blocks [213].

Shares Generation based on Randomized Tensor Decompositions. Algorithms 2, 3, 4, and 5 present our proposed randomized rTD, rHT, rTT-SVD, rTR-SVD algorithms

that decompose N-dimensional tensor into randomized secret shares by applying perturbations to randomly disperse the structural information in big data into the tensor cores. Algorithm 2 is based on Higher-Order Singular Value Decomposition (HOSVD) proposed in [214], HOSVD performs SVD on each mode of a tensor to extract the latent factors before obtaining the core tensor that captures the complex interactions between the latent factors. Algorithm 3 recursively applies rTD on each tensor node based on a binary tree matricizations of the input tensor. Algorithm 4 and 5 are based on the TT-SVD and TR-SVD algorithms proposed in [211] and [212] respectively, TT-SVD and TR-SVD perform sequential SVD decomposition on a tensor to obtain the TT and TR representations. Figures 6.1 and 6.2 show the graphical representations of the proposed rTD and rTT-SVD algorithms. The randomized dispersion is applied after performing each SVD step in Algorithms 2, 3, and 4. To balance between compression and randomness, the maximum (randomized) perturbation should be within certain threshold δ based on the magnitude of each singular value, and +ve/-ve sign difference from each singular vector. The share re-generation can be done with our proposed randomized TT-rounding algorithm based on [211] (see Algorithm 6) all carried out in TT format on distributed servers, however this is not recommended because computation with TN may leak private information (gradually) to the servers. The proposed secret-sharing scheme is asymmetric to the servers, each server stores only index-specific information based on the tensor core it receives. The perturbations are embedded inside existing tensor decomposition algorithms, therefore the computational complexity does not increase much, only a few more tensor core contractions (i.e., to apply perturbation and randomize 1^{st} core / factor) and an SVD are performed. The memory size to store the perturbation factors is considered negligible.

Privacy and Correctness. The correctness of secret sharing based on randomized TN formats is obvious; tensor representations are compressible if the data admits low-rank structure. The proposed randomized tensor decomposition algorithms simply split the complex structural information in big data randomly into different tensor cores or sub-blocks. The sensitivity of SVD decomposition subject to small perturbations is well-known for complex correlation structure, i.e., when the

Algorithm 2 Proposed randomized Tucker Decomposition (rTD) based on Higher-Order SVD (HOSVD) [214].

Input : Tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and ranks $R_1, R_2, \ldots, R_N$.

Output: Tucker core $\hat{\mathcal{G}} \in \mathbb{R}^{R_1, R_2, \ldots, R_N}$ and factor matrices $\hat{\mathbf{U}}_k \in \mathbb{R}^{I_k \times R_k}$ $s.t.$ $\mathcal{A} \cong \hat{\mathcal{G}} \times_1 \hat{\mathbf{U}}_1 \times_2 \hat{\mathbf{U}}_2 \ldots \times_N \hat{\mathbf{U}}_N$.

Initialization: $\mathcal{G}_1 = \mathcal{A}$;

Modified from multilinear SVD or N-mode SVD:

for $k = 1$ *to* N **do**

$[\mathbf{U}_k, \mathbf{S}_k, \mathbf{V}_k] \leftarrow tSVD(\mathbf{G}_{k(k)}, R_{trunc.} = R_k)$;

Generate diagonal perturbation matrix with uniform distribution bet. threshold δ and 1, $\mathbf{\Delta}_k \sim \mathcal{U}([\delta, 1])$;

Perturb the core tensor, $\mathcal{G}_{k+1} \leftarrow \mathcal{G}_k \times_k (\mathbf{U}_k^T \mathbf{\Delta}_k)$;

Update the factor matrix, $\hat{\mathbf{U}}_k \leftarrow \mathbf{\Delta}_k^{-1} \mathbf{U}_k$;

end

Randomize the 1^{st} TD factor matrix:

$\hat{\mathcal{G}} \leftarrow \mathcal{G}_{N+1}$; $\hat{\mathcal{G}} \leftarrow \hat{\mathcal{G}} \times_1 \hat{\mathbf{U}}_1$;

$[\mathbf{U}_1, \mathbf{S}_1, \mathbf{V}_1] \leftarrow tSVD(\hat{\mathbf{G}}_{(1)}, R_{trunc.} = R_1)$;

$\mathbf{\Delta}_1 \sim \mathcal{U}([\delta, 1])$; $\hat{\mathcal{G}} \leftarrow \hat{\mathcal{G}} \times_1 (\mathbf{U}_1^T \mathbf{\Delta}_1)$;

$\hat{\mathbf{U}}_1 \leftarrow \mathbf{\Delta}_1^{-1} \mathbf{U}_1$;

Algorithm 3 Proposed randomized Hierarchical Tucker (rHT) decomposition by recursive node-wise rTD (Algo. 2).

Input : Tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, ranks $R_1, R_2, \ldots, R_N$, and binary tree $\mathcal{T}$ of the matricizations of $\mathcal{A}$.

Output: HT factor matrices $\hat{\mathbf{U}}_1, \hat{\mathbf{U}}_2, \ldots, \hat{\mathbf{U}}_N$ and transfer cores $\hat{\mathcal{B}}_t$, $t \in$ nonleaf nodes of binary tree $\mathcal{T}$.

$\mathbf{U}_1 \leftarrow \mathbf{A}_{(1)}$;

Starting from the root node of tree $\mathcal{T}$, select a node t:

Set t_l and t_r to be the left and right child of t resp.;

If t_l is not singleton: $R_{t_l} \leftarrow R_{full}$;

If t_r is not singleton: $R_{t_r} \leftarrow R_{full}$;

$\mathcal{U}_t \leftarrow reshape(\mathbf{U}_t, [t_l, t_r, t])$;

$[\hat{\mathcal{B}}_t, \mathbf{U}_{t_l}, \mathbf{U}_{t_r}] \leftarrow rTD(\mathcal{U}_t, R_{trunc.} = [R_{t_l}, R_{t_r}])$;

If t_l is a singleton: $\hat{\mathbf{U}}_{t_l} \leftarrow \mathbf{U}_{t_l}$;

If t_r is a singleton: $\hat{\mathbf{U}}_{t_r} \leftarrow \mathbf{U}_{t_r}$;

Recurse on t_l and t_r until t_l and t_r are singletons.

Algorithm 4 Proposed randomized Tensor Train-Singular Value Decomposition (rTT-SVD) algorithm based on [211].

Input : Tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and error threshold ϵ.

Output: TT cores $\hat{\mathcal{A}} = \hat{\mathbf{G}}_1 \cdot \hat{\mathcal{G}}_2 \cdots \hat{\mathcal{G}}_{N-1} \cdot \hat{\mathbf{G}}_N$ such that the approximation error $\|\mathcal{A} - \hat{\mathcal{A}}\|_F \lesssim \epsilon \|\mathcal{A}\|_F$.

Initialization: TT-rank $R_0 = 1$; Perturbation threshold δ;

Truncation parameter $\delta_\epsilon = \frac{\epsilon}{\sqrt{N-1}}$;

Tensor shape, $[I_1, I_2, \ldots, I_N] \leftarrow shape(\mathcal{A})$;

Mode-1 matricization of tensor $\mathcal{A}$, $\mathbf{M}_1 \leftarrow \mathbf{A}_{(1)}$;

Sequential (SVD + randomized dispersion):

for $k = 1$ *to* $N - 1$ **do**

Truncated SVD, $[\mathbf{U}_k, \mathbf{S}_k, \mathbf{V}_k] \leftarrow tSVD(\mathbf{M}_k, \delta_{rel.} = \delta_\epsilon)$;

TT-rank, $R_k \leftarrow shape(\mathbf{S}_k, 1)$;

Generate diagonal perturbation matrix with uniform dist. between threshold δ and 1, $\mathbf{\Delta}_k \sim \mathcal{U}([\delta, 1])$;

Reshape the orthogonal matrix $\mathbf{U}_k$ divided by the perturbation factor $\mathbf{\Delta}_k$ into a third-order tensor $\hat{\mathcal{G}}_k \leftarrow reshape(\mathbf{U}_k \mathbf{\Delta}_k^{-1}, [R_{k-1}, I_k, R_k])$;

Matricize $\mathbf{S}_k \mathbf{V}_k^T$ multiplied by the perturbation factor $\mathbf{\Delta}_k$ $\mathbf{M}_{k+1} \leftarrow reshape(\mathbf{\Delta}_k \mathbf{S}_k \mathbf{V}_k^T, [R_k I_{k+1}, \prod_{p=k+2}^{N} I_p])$;

end

Construct the last core, $\hat{\mathbf{G}}_N \leftarrow reshape(\mathbf{M}_N, [R_{N-1}, I_N])$ **Randomize the** 1^{st} **TT-core:**

$\hat{\mathbf{G}}_1 \leftarrow reshape(\hat{\mathcal{G}}_1, [I_1, R_1])$;

$\hat{\mathbf{G}}_2 \leftarrow reshape(\hat{\mathcal{G}}_2, [R_1, I_2 R_2])$;

$[\mathbf{U}_1, \mathbf{S}_1, \mathbf{V}_1] \leftarrow tSVD(\hat{\mathbf{G}}_1 \hat{\mathbf{G}}_2, \delta_{rel.} = \delta_\epsilon)$;

$R_1 \leftarrow shape(\mathbf{S}_1, 1)$; $\mathbf{\Delta}_1 \sim \mathcal{U}([\delta, 1])$;

$\hat{\mathbf{G}}_1 \leftarrow reshape(\mathbf{U}_1 \mathbf{\Delta}_1^{-1}, [I_1, R_1])$;

$\hat{\mathcal{G}}_2 \leftarrow reshape(\mathbf{\Delta}_1 \mathbf{S}_1 \mathbf{V}_1^T, [R_1, I_2, R_2])$;

Algorithm 5 Proposed randomized Tensor Ring-Singular Value Decomposition (rTR-SVD) based on [212].

Input : Tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and error threshold ϵ.

Output: TR cores $\hat{\mathcal{A}} = \hat{\mathcal{G}}_1 \cdot \hat{\mathcal{G}}_2 \cdots \hat{\mathcal{G}}_N$ such that the approximation error $\|\mathcal{A} - \hat{\mathcal{A}}\|_F \lesssim \epsilon \|\mathcal{A}\|_F$.

Initialization: Perturbation threshold δ;

Truncation parameter $\delta_k = \begin{cases} \frac{\sqrt{2}\epsilon}{\sqrt{N}}, k = 1 \\ \frac{\epsilon}{\sqrt{N}}, k > 1 \end{cases}$;

Prepare the 1^{st} TR core:

Tensor shape, $[I_1, I_2, \ldots, I_N] \leftarrow shape(\mathcal{A})$;

Mode-1 matricization of tensor $\mathcal{A}$, $\mathbf{M}_1 \leftarrow \mathcal{A}_{(1)}$;

Truncated SVD, $[\mathbf{U}_1, \mathbf{S}_1, \mathbf{V}_1] \leftarrow tSVD(\mathbf{M}_1, \delta_{rel.} = \delta_1)$;

$\mathbf{\Delta}_1 \sim \mathcal{U}([\delta, 1])$; Split TT-ranks R_0, R_1:

$\min_{R_0, R_1} ||R_0 - R_1||$ s.t. $R_0 R_1 = shape(\mathbf{S}_1, 1)$;

Set TT-rank, $R_N \leftarrow R_0$;

$\hat{\mathcal{G}}_1 \leftarrow reshape(\mathbf{U}_1 \mathbf{\Delta}_1^{-1}, [R_0, I_1, R_1])$;

$\mathbf{M}_2 \leftarrow reshape(\mathbf{\Delta}_1 \mathbf{S}_1 \mathbf{V}_1^T, [R_1 I_2, \prod_{p=3}^{N} I_p R_N])$;

Sequential (SVD + randomized dispersion):

for $k = 2$ *to* $N - 1$ **do**

$[\mathbf{U}_k, \mathbf{S}_k, \mathbf{V}_k] \leftarrow tSVD(\mathbf{M}_k, \delta_{rel.} = \delta_k)$;

$R_k \leftarrow shape(\mathbf{S}_k, 1)$; $\mathbf{\Delta}_k \sim \mathcal{U}([\delta, 1])$;

$\hat{\mathcal{G}}_k \leftarrow reshape(\mathbf{U}_k \mathbf{\Delta}_k^{-1}, [R_{k-1}, I_k, R_k])$;

$\mathbf{M}_{k+1} \leftarrow reshape(\mathbf{\Delta}_k \mathbf{S}_k \mathbf{V}_k^T, [R_k I_{k+1}, \prod_{p=k+2}^{N} I_p R_N])$;

end

Construct the last core, $\hat{\mathcal{G}}_N \leftarrow reshape(\mathbf{M}_N, [R_{N-1}, I_N, R_N])$ **Randomize the 1^{st} TR core:**

$\hat{\mathbf{G}}_1 \leftarrow reshape(\hat{\mathcal{G}}_1, [R_0 I_1, R_1])$;

$\hat{\mathbf{G}}_2 \leftarrow reshape(\hat{\mathcal{G}}_2, [R_1, I_2 R_2])$;

$[\mathbf{U}_1, \mathbf{S}_1, \mathbf{V}_1] \leftarrow tSVD(\hat{\mathbf{G}}_1 \hat{\mathbf{G}}_2, \delta_{rel.} = \delta_1)$;

$R_1 \leftarrow shape(\mathbf{S}_1, 1)$; $\mathbf{\Delta}_1 \sim \mathcal{U}([\delta, 1])$;

$\hat{\mathcal{G}}_1 \leftarrow reshape(\mathbf{U}_1 \mathbf{\Delta}_1^{-1}, [R_0, I_1, R_1])$;

$\hat{\mathcal{G}}_2 \leftarrow reshape(\mathbf{\Delta}_1 \mathbf{S}_1 \mathbf{V}_1^T, [R_1, I_2, R_2])$

singular values are closely separated [124, 118]. Moreover, the proposed algorithms randomize TN decompositions by large-but-controlled perturbation that does not affect the reconstruction accuracy. The privacy leakage is limited by the tensor-rank complexity of each index, i.e., index that has sufficiently high rank complexity is privacy-preserving, whereas index that has only zeroes in the tensor cores implies that all the values that correspond to this index in the original tensor are zero. However, this can be easily overcome by padding the original tensor with random noise to increase the complexity before TN decomposition. With sufficiently high tensor-rank complexity, the magnitude, sign, and exact position of non-zero values are not leaked even with collusion by all-except-one servers. To further increase the uncertainty, we randomly permute the mode variables along each tensor dimension and store the random seeds for reconstruction. Random permutations can be performed after (block-wise) TN decomposition to ensure compressibility if the multi-dimensional data is highly-correlated. Each tensor core contains only index-specific information and therefore they are unlinkable in the event of massive data breach. The partition of more sophisticated TN structures into private and shared tensor cores can be done with hierarchical clustering based on pairwise network distance and randomized, dispersed tensor computation that minimize privacy leakage, communication, and computational cost.

Relationship with Additive Secret-Sharing Scheme. The classical additive secret-sharing scheme is defined as $x = \langle x_1 \rangle_1 + \langle x_2 \rangle_2 + \ldots$. The conversion from the classical scheme to secret-sharing scheme based on TN format is relatively straightforward, each party decomposes their individual share using the proposed randomized tensor decomposition algorithms and send to other parties the corresponding tensor cores. All parties perform an addition operations using their corresponding tensor cores based on tensor multilinear operations. The conversion from TN format to the additive secret-sharing scheme can be done by all-except-one parties generate randomized TN from randomly-generated share, distribute the generated tensor cores to the corresponding party and update all the tensor cores using distributed tensor operations, all-except-one parties pass their updated tensor cores to the remaining one (that didn't generate randomized tensor cores before) to generate his secret

share. Future work may consider how to prevent malicious servers from corrupting tensor network computing protocols.

6.4.1 Dispersed Storage, Sharing, and Communication For Big Data Protection

Encryption is complicated in terms of key management for big data distributed applications, encryption requires centralized management by a trusted authority to authenticate, authorize, and revoke access to prevent potential key leakage that may lead to massive data breach. Our proposal combines the secret-sharing scheme based on distributed TNs and metadata privacy to seamlessly secure big data storage, communication, and sharing. Distributed trust can be achieved by decentralizing the fragments / metadata encryption and access control mechanisms. Furthermore, the metadata management is flexible such that it can be done in a centralized or decentralized manner by using enterprise management systems, or in a distributed manner on the individual user's side. Any software applications can reconstruct the original data if granted access to the metadata information and shredded fragments. Data integrity can be ensured by cryptographic hashing; whereas data availability can be gueranteed by integrating in Hadoop Distributed File System (HDFS). The advantages of distributed TN representations for secure data storage / sharing include privacy protection, compression, granular access control, updatability, and compressed computation.

Metadata serves as the logical "map" for users to navigate through the information and data; metadata also helps auditors to carry out system review and post-breach damage assessment. After decomposing big data and distribute each tensor core or sub-block to multiple storage locations using our proposed randomized tensor decomposition algorithms, the master metadata files are updated with the locations and anonymized filenames of each tensor blocks, tensor structure, cryptographic hashes, random seeds used to permute the mode variables, and users' access permission; the storage locations and filenames of the tensor fragments can be routinely renewed to enhance the data privacy protection. The master metadata

files can be further encrypted and password-protected on the users' side. The metadata of each tensor core stored on the distributed storage locations contains only the anonymized filename and location such that they are unlinkable in the event of massive data breach; data encryption and access control based on role management policy can be implemented in a decentralized manner to protect the tensor cores. The system architecture and metadata organization is beyond the scope of this work but will be considered in future to take account of the various application scenarios, system performance, and requirements for different big data applications.

6.4.2 Big Data Dispersed Computation

Tensor network (TN) naturally supports distributed / dispersed computation using the smaller, interconnected tensor cores / blocks after big data decomposition [34, 43, 4]. Some basic arithmetic operations in Tucker format are derived in [4]. Let

$$\begin{aligned} \mathcal{A} &= [\![\mathcal{G}_A; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!] \\ \mathcal{B} &= [\![\mathcal{G}_B; \mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \mathbf{B}^{(N)}]\!] \end{aligned} \tag{6.4}$$

where $\mathcal{G}_L$, $L \in \{A, B\}$ and $\mathbf{A}^{(k)}/\mathbf{B}^{(k)}$, $k \in \{1, 2, \ldots, N\}$ correspond to the Tucker core tensors and factor matrices respectively,

$$\begin{aligned} &(a)\, \mathcal{A} + \mathcal{B} = [\![\mathcal{G}_A \oplus \mathcal{G}_B ; \mathbf{A}^{(1)} \boxplus \mathbf{B}^{(1)}, \ldots, \mathbf{A}^{(N)} \boxplus \mathbf{B}^{(N)}]\!] \\ &(b)\, \mathcal{A} \oplus \mathcal{B} = [\![\mathcal{G}_A \oplus \mathcal{G}_B ; \mathbf{A}^{(1)} \oplus \mathbf{B}^{(1)}, \ldots, \mathbf{A}^{(N)} \oplus \mathbf{B}^{(N)}]\!] \\ &(c)\, \mathcal{A} \circledast \mathcal{B} = [\![\mathcal{G}_A \otimes \mathcal{G}_B ; \mathbf{A}^{(1)} \boxtimes \mathbf{B}^{(1)}, \ldots, \mathbf{A}^{(N)} \boxtimes \mathbf{B}^{(N)}]\!] \\ &(d)\, \mathcal{A} \otimes \mathcal{B} = [\![\mathcal{G}_A \otimes \mathcal{G}_B ; \mathbf{A}^{(1)} \otimes \mathbf{B}^{(1)}, \ldots, \mathbf{A}^{(N)} \otimes \mathbf{B}^{(N)}]\!] \end{aligned} \tag{6.5}$$

The tensor operations expressed with the symbols $\oplus$, $\boxplus$, $\circledast$, $\otimes$, and $\boxtimes$ refer to the direct sum, partial direct sum, Hadamard product, Kronecker product, and partial Kronecker product respectively, the formal definitions can be found in [4]. Linear algebra operations for all tensor formats can be derived using the following rules [215]:

- separable components are added, or multiplied independently in each tensor core for all variables,

- all rank sums are added in linear operations, and multiplied in bilinear operations.

During iterative computations, the tensor rank grows quickly especially with the multiplications. Hence, another important operation called rank truncation should be provided with the tensor format.

TT format and its variants support wide range of multilinear operations such as addition, multiplication, matrix-by-matrix/vector multiplication, direct sum, Hadamard, Kronecker, and inner product [4, 34, 43]. As shown in Figure 6.3, multilinear operations in TT format can be performed naturally in dispersed (and compressed) manner, making it well-suited for big data processing and scientific computing. TT-rank grows with every multilinear operations and quickly become computationally prohibitive, the TT-rounding (or recompression) [211] procedure can be implemented to reduce the TT-ranks by first orthogonalizing the tensor cores using QR decomposition and then compress using SVD decomposition, all performed in TT format. The randomized TT-SVD algorithm proposed in Algorithm 4 can be easily adapted to the second step of TT-rounding procedure. Algorithm 6 shows an example of randomized rTT-rounding algorithm for an N^{th}-order tensor. To compute non-linear functions, TT cross-approximation can be used [33]. The idea of tensor cross or pseudo-skeleton approximation is to sample from the TN, reconstruct and compute arbitrary functions from the sample points, decompose the sample updates and update the original TN accordingly, but how to ensure the privacy preservation of tensor cross approximation is still a question remains.

Tensor network computing naturally supports a number of multilinear operations in floating- / fixed-point representations with minimal data pre-processing, unlike classical SMPC schemes that only support limited secure operations (e.g., addition and multiplication) and has to be pre-processed every time to carry out different operations. Therefore, SMPC generally requires many rounds of communication between the servers in order to compute complex functions. With TN representations, multilinear operations can be done in compressed and dispersed manner without the need to reconstruct the original tensor, this is the major advantage of tensor computation in overcoming the curse of dimensionality for large-scale optimization

Algorithm 6 Proposed randomized TT-rounding (rTT-rounding) based on [211] to reduce the size of TT-cores.

Input : TT cores of an N^{th}-order tensor stored on servers, $\mathcal{A} = \langle \mathbf{G}_1 \rangle_1 \langle \mathcal{G}_2 \rangle_2 \cdots \langle \mathcal{G}_{N-1} \rangle_{N-1} \langle \mathbf{G}_N \rangle_N$ and ϵ.

Output: Updated TT cores $\hat{\mathcal{A}} = \langle \hat{\mathbf{G}}_1 \rangle_1 \langle \hat{\mathcal{G}}_2 \rangle_2 \cdots \langle \hat{\mathbf{G}}_N \rangle_N$ such that $\|\mathcal{A} - \hat{\mathcal{A}}\|_F \leqslant \epsilon \|\mathcal{A}\|_F$.

Initialization: TT-rank $\langle R_0 \rangle_1 = 1$; $\langle R_N \rangle_N = 1$;

Perturbation threshold δ;

Truncation parameter $\delta_\epsilon = \frac{\epsilon}{\sqrt{N-1}}$ **Right-to-left QR orthogonalization:**

$[\langle I_1 \rangle_1, \langle R_1 \rangle_1] \leftarrow shape(\langle \hat{\mathbf{G}}_1 \rangle_1)$;

$[\langle R_{N-1} \rangle_N, \langle I_N \rangle_N] \leftarrow shape(\langle \hat{\mathbf{G}}_N \rangle_N)$;

$\langle \hat{\mathcal{G}}_1 \rangle_1 \leftarrow reshape(\langle \hat{\mathbf{G}}_1 \rangle_1, [R_0, I_1, R_1])$;

$\langle \hat{\mathcal{G}}_N \rangle_N \leftarrow reshape(\langle \hat{\mathbf{G}}_N \rangle_N, [R_{N-1}, I_N, R_N])$;

for $k = N$ *to* 2 **do**

$[\langle R_{k-1} \rangle_k, \langle I_k \rangle_k, \langle R_k \rangle_k] \leftarrow shape(\langle \mathcal{G}_k \rangle_k)$;

$\langle \mathbf{G}_k \rangle_k \leftarrow reshape(\langle \mathcal{G}_k \rangle_k, [R_{k-1}, I_k R_k])$;

QR decomposition: $\left[\langle \hat{\mathbf{Q}}_k \rangle_k, \langle \hat{\mathbf{R}}_k \rangle_k\right] \leftarrow QR(\langle \mathbf{G}_k \rangle_k)$;

$\langle \mathcal{G}_k \rangle_k \leftarrow reshape(\langle \hat{\mathbf{Q}}_k \rangle_k, [R_{k-1}, I_k, R_k])$;

$\langle \mathcal{G}_{k-1} \rangle_{k-1} \leftarrow \langle \mathcal{G}_{k-1} \times_3 \hat{\mathbf{R}}_k \rangle_{k-1}$;

end

Left-to-right (SVD compress + random disperse):

$[\langle R_0 \rangle_1, \langle I_1 \rangle_1, \langle R_1 \rangle_1] \leftarrow shape(\langle \mathcal{G}_1 \rangle_1)$;

for $k = 1$ *to* $N - 1$ **do**

$\langle \mathbf{G}_k \rangle_k \leftarrow reshape(\langle \mathcal{G}_k \rangle_k, [R_{k-1} I_k, R_k])$;

$[\langle \mathbf{U}_k \rangle_k, \langle \mathbf{S}_k \rangle_k, \langle \mathbf{V}_k \rangle_k] \leftarrow tSVD(\langle \mathbf{G}_k \rangle_k, \delta_{rel.} = \delta_\epsilon)$;

$\langle R_k \rangle_k \leftarrow shape(\langle \mathbf{S}_k \rangle_k, 1)$; $\langle \mathbf{\Delta}_k \rangle_k \sim \mathcal{U}(\delta, 1)$;

$\langle \hat{\mathcal{G}}_k \rangle_k \leftarrow reshape(\langle \mathbf{U}_k \mathbf{\Delta}_k^{-1} \rangle_k, [R_{k-1}, I_k, R_k])$;

$\langle \hat{\mathcal{G}}_{k+1} \rangle_{k+1} \leftarrow \langle \hat{\mathcal{G}}_{k+1} \times_1 \mathbf{\Delta}_k \mathbf{S}_k \mathbf{V}_k^T \rangle_{k+1}$;

end

$\langle \hat{\mathbf{G}}_1 \rangle_1 \leftarrow reshape(\langle \hat{\mathcal{G}}_1 \rangle_1, [I_1, R_1])$;

$\langle \hat{\mathbf{G}}_N \rangle_N \leftarrow reshape(\langle \hat{\mathcal{G}}_N \rangle_N, [R_{N-1}, I_N])$;

problems. Tensor multilinear operations generally require only computation on each tensor core, but some tensor computation schemes require communication between servers like the TT-rounding scheme mentioned before and the famous Density Matrix Renormalization Group (DMRG) scheme [216, 217]. Unlike SMPC schemes, the communication is mainly tensor cores instead of the secret shares of original tensor, which are generally much smaller in size. However, dispersed tensor computing leaks more information than SMPC schemes during communication, one way to overcome this is to continually ingest fresh entropy from complex data when performing dispersed tensor computation.

6.5 Experiments

Experimental Setup. The experiments are carried out using a workstation with 64-bit Intel® Xeon® W-2123 CPU 3.60GHz, 16.0GB RAM. Privacy metrics such as Pearson's correlation coefficient, histogram analysis, and normalized mutual information are used to measure the privacy leakage of the proposed randomized TN decompositions. Further comparisons are made between the original and the proposed randomized TN decompositions in terms of the computational speed, compression ratio, and distortion analysis of the reconstructed data from TN compression. For image data, the distortion as a result of the TN compression can be measured by the normalized L_2-dissimilarity, which is defined by

$$\frac{1}{N'}\sum_{n=1}^{N'}\frac{||\mathbf{x}_n-\mathbf{x}'_n||_2}{||\mathbf{x}_n||_2} \tag{6.6}$$

where $\mathbf{x}_n$, $n\in\{1,2,\ldots,N'\}$ are the set of original images and $\mathbf{x}'_n$, $n\in\{1,2,\ldots,N'\}$ are the set of reconstructed images after TN compression, $||\cdot||$ is the Euclidean norm. Here, we study the proposed rTT-SVD, rTR-SVD, and rTD algorithms only because rHT is based on recursive rTD, therefore showing rTD is privacy-preserving implies that rHT is also privacy-preserving for larger-scale tensor. The perturbation factor δ for randomized TN is set as 0.05 for all the experiments, hence the diagonal perturbation matrix $\mathbf{\Delta}$ falls within the range $[0.05, 1]$ uniformly.

Datasets. Table 6.1 tabulates all the datasets' sample size and mode size used

in this study. Experiments are carried out on 1D, 2D, and 3D biometric datasets to investigate thoroughly the proposed randomized TN algorithms across different data dimensions for privacy preservation. In general, vector and matrix data are reshaped into higher-order tensor before TN decomposition. The gait sensor database is recorded using smartphone's inertial sensors, the sampling frequency is 100Hz and the total walking distance is 640 meters per session [218]. The training images for real and fake face detection are provided by the Computational Intelligence and Photography Lab, Department of Computer Science, Yonsei University on Kaggle online data-sharing platform; only the real facial images are used in the experiments. The RGB channels of a facial image have very high spectral correlation, therefore the channels are stacked in 3D for tensor decomposition. Yale face database contains the GIF images of 15 human subjects, each with 11 different facial expressions or configurations [219]. Finally, we also generate a 3D super-diagonal tensor with ones on the (i, i, i) entries for our investigation studies.

Table 6.1: Datasets used in the experimental studies.

Dataset	**Subjects**	**Mode Size**
Human Gait (walking)	93	58 Features
Real & Fake Face Images	∼1000	600×600×3
Yale Face Database	15	320×243×11
Super-diagonal Tensor	N/A	10×10×10

Data Complexity for Randomized TN Decompositions. Figure 6.4 shows the effect on the TT decomposition before and after padding noise to a super-diagonal tensor (full rank), both approaches reproduce the same super-diagonal tensor after reconstruction because the noise can be truncated after decomposition. However, naive padding with noise usually results in high TN computation and storage cost due to the higher rank-complexity, whereas our proposed randomized TN algorithms simply make use of the complex correlation structure commonly found in big data to generate highly-randomized tensor blocks. Figure 6.5 shows the randomized TT

decomposition of human gait sensor data. To preserve important dataset features during the TN compression, each of the attributes is standardized to zero mean and variance equals to one, i.e., the z-score, before tensor decomposition. The translation and scaling factors have to be stored in the metadata for reconstruction.

Privacy Leakage Analysis. Figure 6.6 and 6.7 shows the TT decomposition of a facial image. The shape of the facial image is permuted to $600 \times 3 \times 600$ to have a balance shape of TT cores. In general, the histogram of the TN cores and factors are Gaussian or Laplacian distributed, which is very different from the histogram of the original data. Figure 6.8 and 6.9 show the reconstructed images from incomplete TN representations and measure the amount of information overlap with original images using normalized mutual information (NMI). The results show that if each of the TN cores or factors are large enough in terms of rank complexity or block size, the privacy leakage is minimal without having a complete TN representations for a data. In this case, the Tucker factor $\hat{\mathbf{U}}_{\mathbf{2}}$ is very small in size and therefore results in the highest NMI. Figures 6.10 and 6.11 show the correlation between the randomized and non-randomized TN cores for particular rank using the Yale Face Database. The correlation is higher for lower rank, this means it is harder to perturb correlation structure that contributes to higher variability within the data. One way to overcome this is to permute the mode variables along each dimension after TN compression to protect the privacy of the distribution of each tensor mode.

Data Compressibility and Algorithmic Efficiency. Table 6.2 measures the time efficiency of TN decomposition and reconstruction for Real and Fake Facial Image Database. The randomized TN decompositions generally take slightly longer time compared to the non-randomized TN decomposition mainly due to an extra SVD step needed to generate randomized tensor blocks. TR reconstruction is long ($\sim$1 min) because there is a loop in the TN structure. Figure 6.12 shows the image distortion analysis under different TN compression ratio for the Yale Face Database. Randomized TN algorithms result in slightly higher distortion in the reconstructed data compared to non-randomized TN algorithms. This is expected because randomized TN algorithms produce sub-optimal decomposition. Randomized TT decomposition generates the lowest distortion especially with high compression ratio

compared to randomized rTR and rTD decomposition. TT representation strikes a good balance between privacy preservation, computational, and storage efficiency.

Table 6.2: Comparison of the computational efficiency between the original and the proposed randomized TN algorithms. The dataset used is the Real & Fake Facial Images Database and the compression ratio is set as ~ 0.725.

TN Algorithms	**Tensor Rank**	**TN Decompose / Reconstruct Time**
HOSVD	$R_1 = R_3 = 350,\ R_2 = 3$	0.2794 / 0.0077 s
rTD	$R_1 = R_3 = 350,\ R_2 = 3$	0.3104 / 0.0081 s
TT-SVD	$R_1 = R_2 = 350$	0.1851 / 0.0054 s
rTT-SVD	$R_1 = R_2 = 350$	0.2817 / 0.0053 s
TR-SVD	$R_{0/1} = R_2 = 20,\ R_3 = 45$	0.3563 / 1.1746 s
rTR-SVD	$R_{0/1} = R_2 \approx 20,\ R_3 \approx 45$	0.3292 / 1.1150 s

6.6 Discussion

Scalability is an important consideration for both the success of big data analytics and widespread adoption of privacy-preserving techniques. We have proposed a simple perturbation technique that can be easily adapted for randomized decomposition of various tensor network structures. The proposed secret-sharing scheme based on dispersed TN representations / computation is very efficient in terms of storage, computational, and communication complexity due to natural support for dispersed tensor computation. Privacy leakage analysis is carried out to verify that the proposed scheme is secured against semi-honest adversary, however, privacy leakage may still happen when performing dispersed tensor operations, which requires further more investigation. One way is to ingest fresh entropy from complex data when performing tensor operations, hence increases the uncertainty of original tensor estimation. Nevertheless, the proposed scheme can be easily combined with

existing data-security solutions such as data anonymization, encryption, and secure-enclave technologies to provide layered protection. The potential extension of this work includes various applications of privacy-preserving big data analytics [34, 43] and large-scale numerical computing [12, 15, 13]. Another potential direction is extending our proposed secret-sharing scheme for federated machine learning and applying differential privacy to protect the privacy of individual items in the training dataset [220, 221, 189].

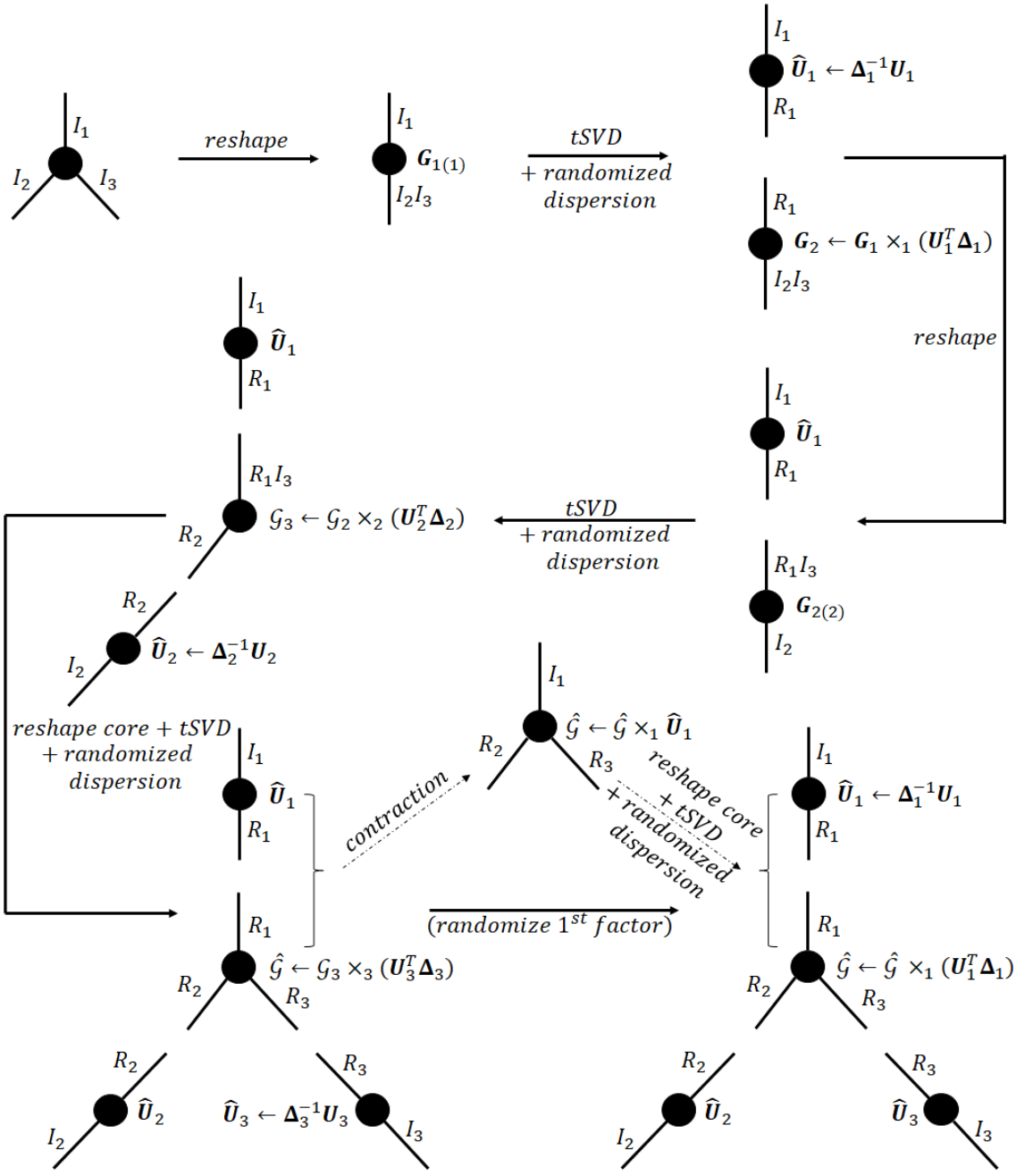

Figure 6.1: Graphical representation of the proposed rTD algorithm for a 3^{rd}-order tensor, see Algorithm 2 for the details.

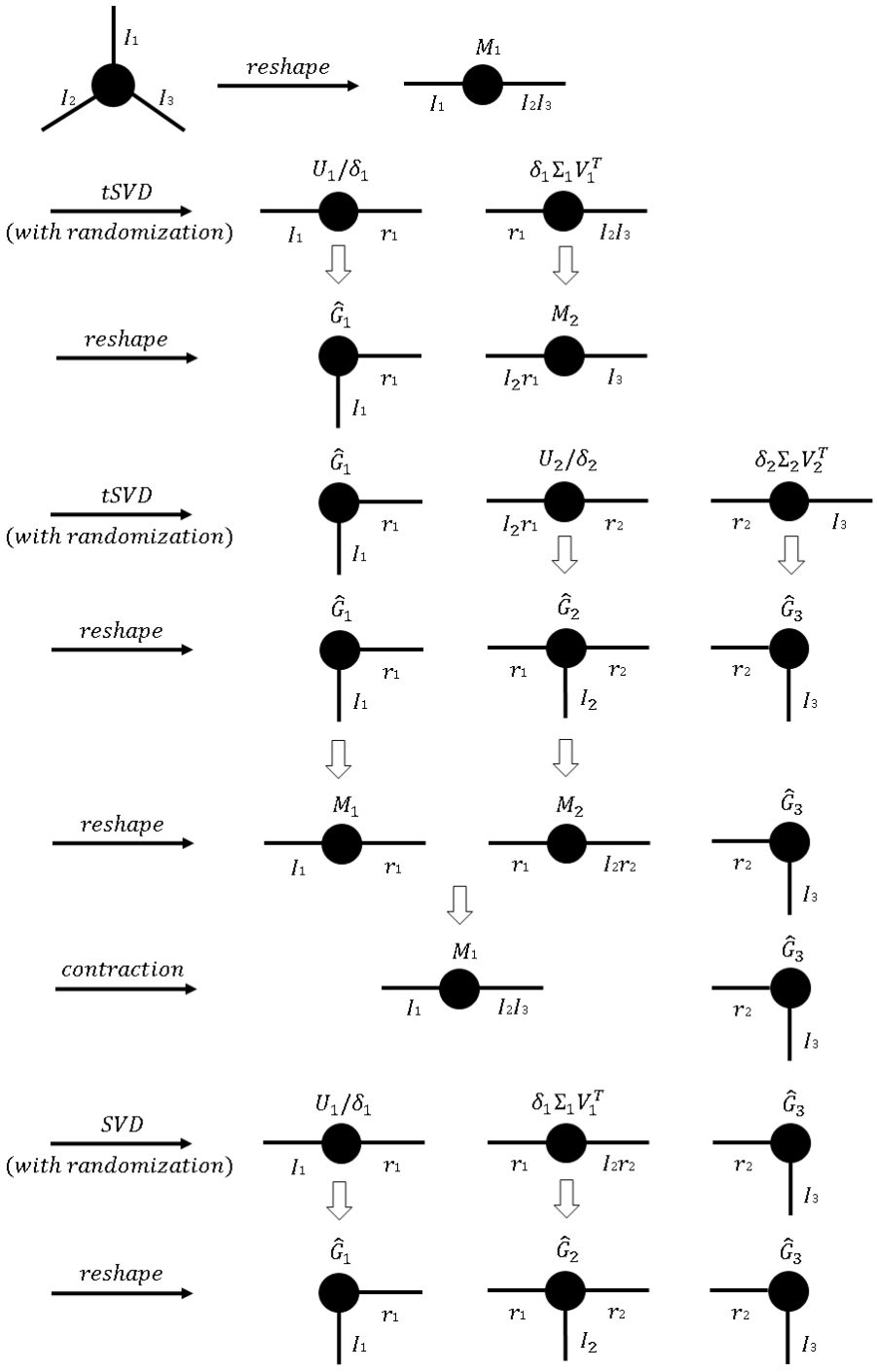

Figure 6.2: Graphical representation of the proposed rTT-SVD for a 3^{rd}-order tensor, see Algorithm 4 for the details.

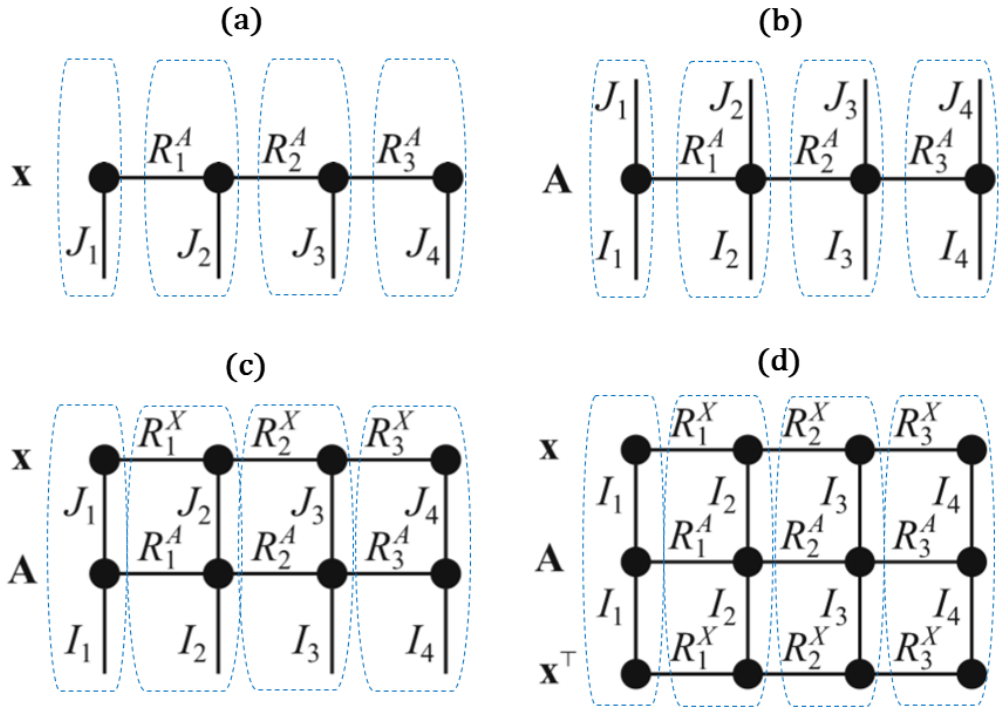

Figure 6.3: Tensor network diagrams of (a) a vector, $\mathbf{x} \in \mathbb{R}^{I_1 I_2 I_3 I_4}$ in vector TT format, (b) a matrix, $\mathbf{A} \in \mathbb{R}^{I_1 I_2 I_3 I_4 \times J_1 J_2 J_3 J_4}$ in matrix TT format, (b) matrix-by-vector multiplication $y = \mathbf{A}\mathbf{x}$, (c) quadratic form, $\mathbf{x}^T \mathbf{A}\mathbf{x}$ with $I_n = J_n$ [4]. The dashed blue boxes show each of the tensor blocks and multi-linear operations performed in multi-party computation setting.

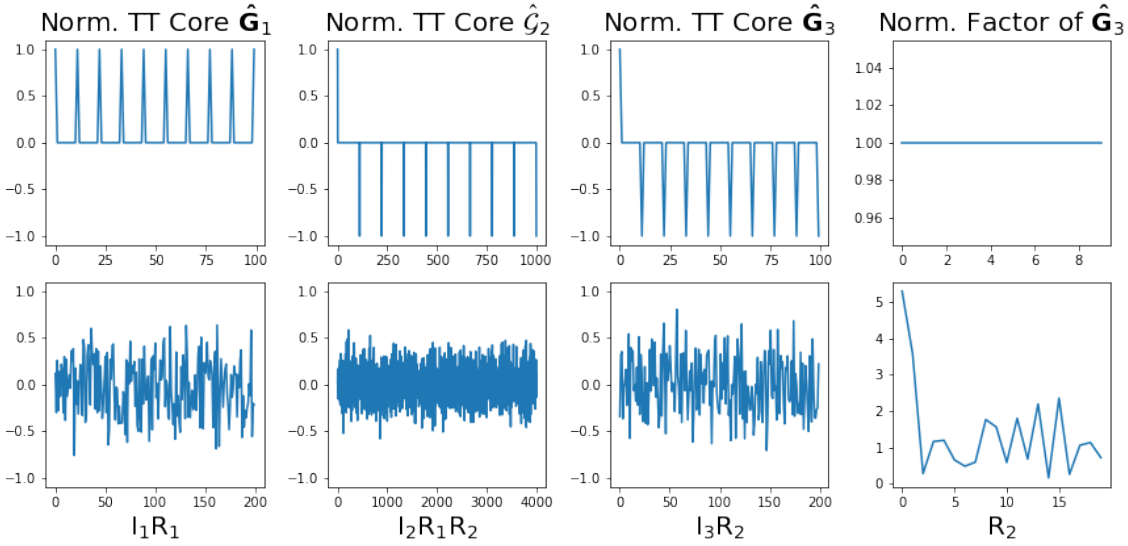

Figure 6.4: TT decomposition of a super-diagonal tensor using TT-SVD (top row) and a super-diagonal tensor padded with noise using rTT-SVD (bottom row).

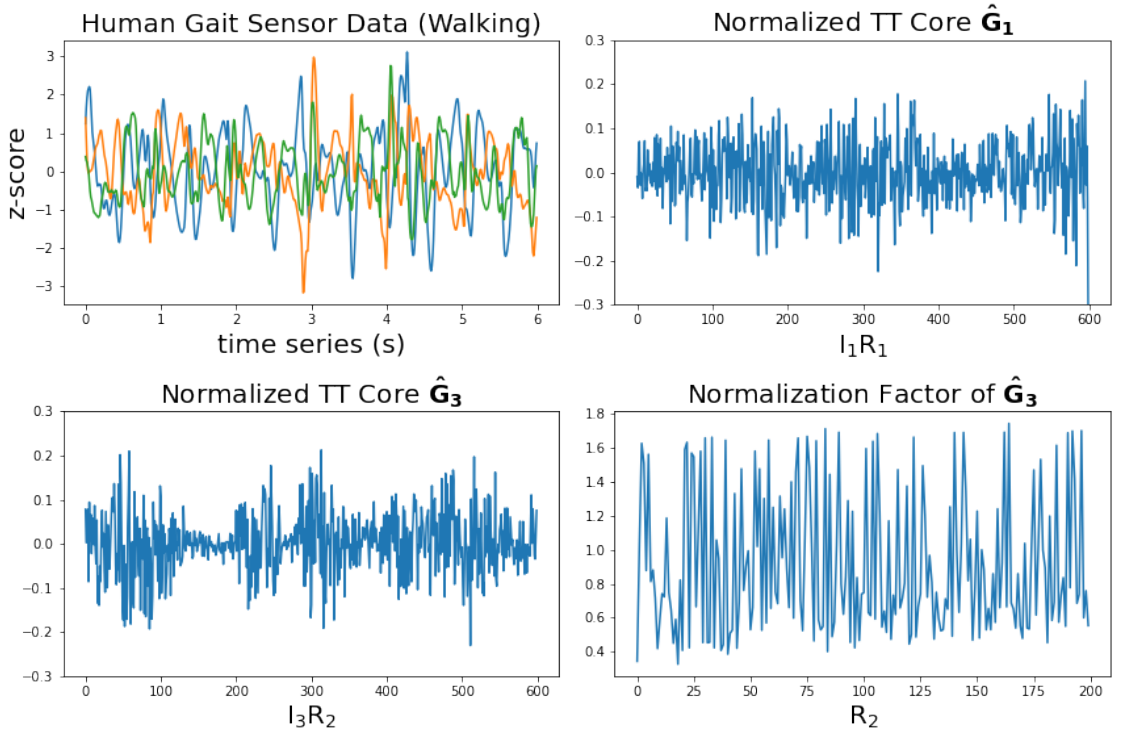

Figure 6.5: Top left: time series of the human gait sensor data (z-score) in walking mode. Top right and bottom left figures show the normalized TT cores $\hat{\mathbf{G}}_1$ and $\hat{\mathbf{G}}_3$ of the data decomposition; bottom right shows the normalization factor of $\hat{\mathbf{G}}_3$.

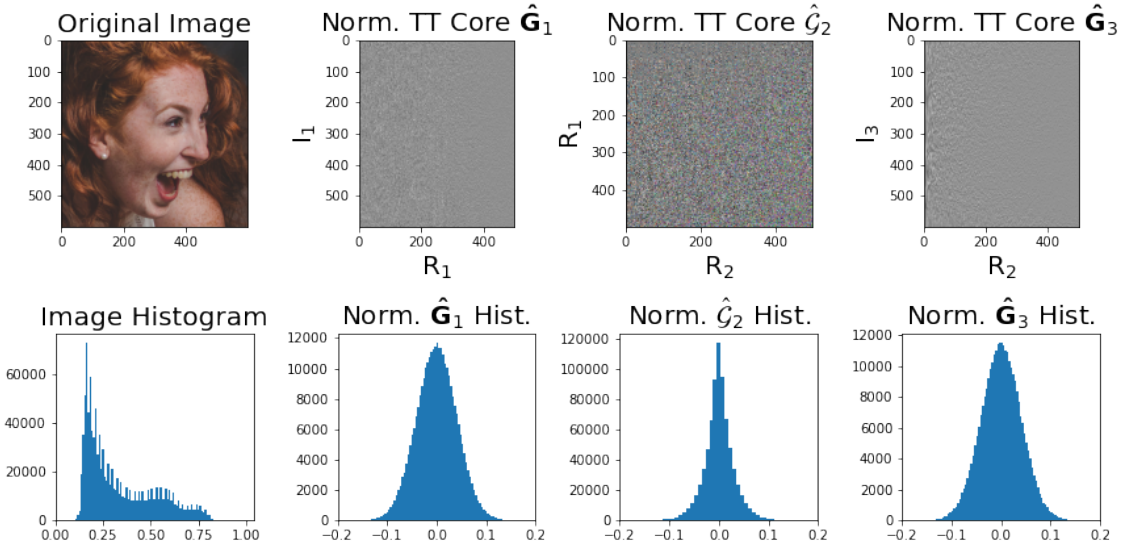

Figure 6.6: Histogram analysis of the distributed TT representations of a facial image. The normalized TT cores are either Gaussian- or Laplacian-distributed, which are usually different from the original image / data histogram distribution.

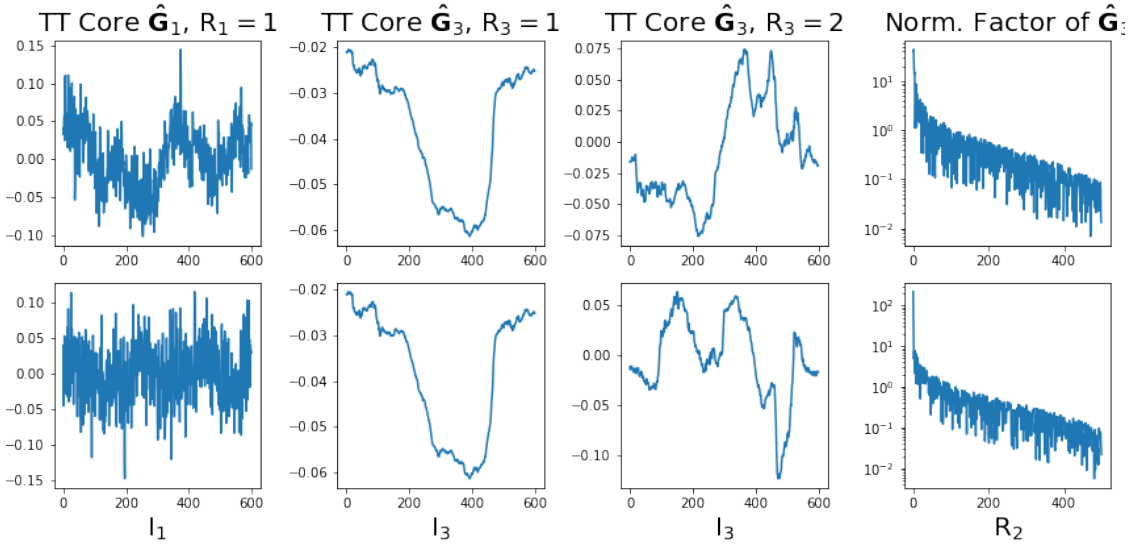

Figure 6.7: Normalized TT cores produced from two randomized rTT-SVD decompositions of a facial image using Algorithm 4 (top and bottom rows). Correlation structure that contributes higher variability (i.e., lower rank) is much harder to perturb and the normalization factor in the last TT core is mostly preserved in the randomized decomposition.

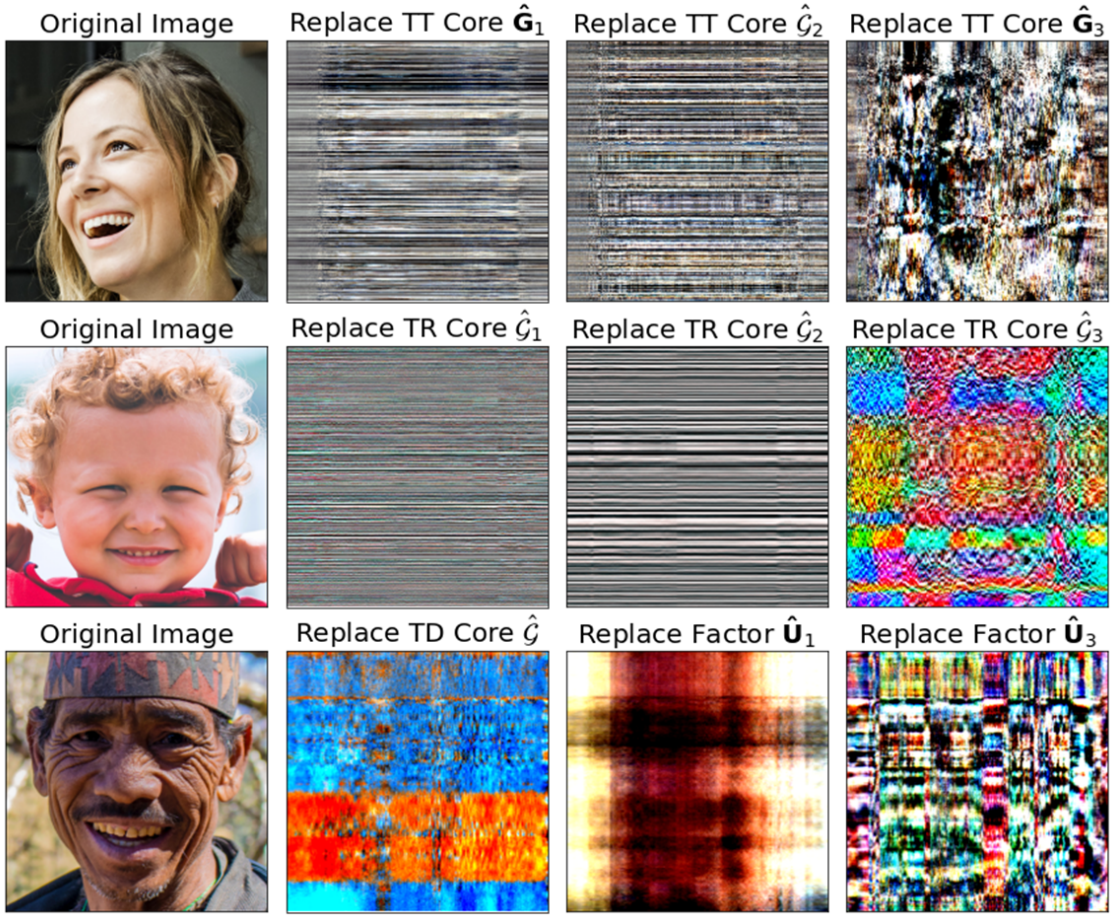

Figure 6.8: Reconstructed images from TN representations by replacing either a tensor core or factor matrix generated from a randomized TN decomposition process with another. First row corresponds to the rTT decomposition, second row the rTR, and third row the rTD respectively.

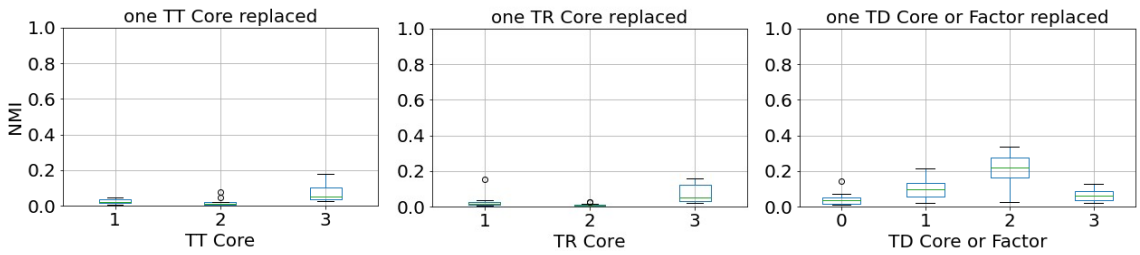

Figure 6.9: Normalized mutual information (NMI) between the original and reconstructed data from the randomized TN representations with one tensor core or factor replaced. Index 0 for the x-axis of the 3^{rd} plot refers to the TD core $\hat{\mathcal{G}}$.

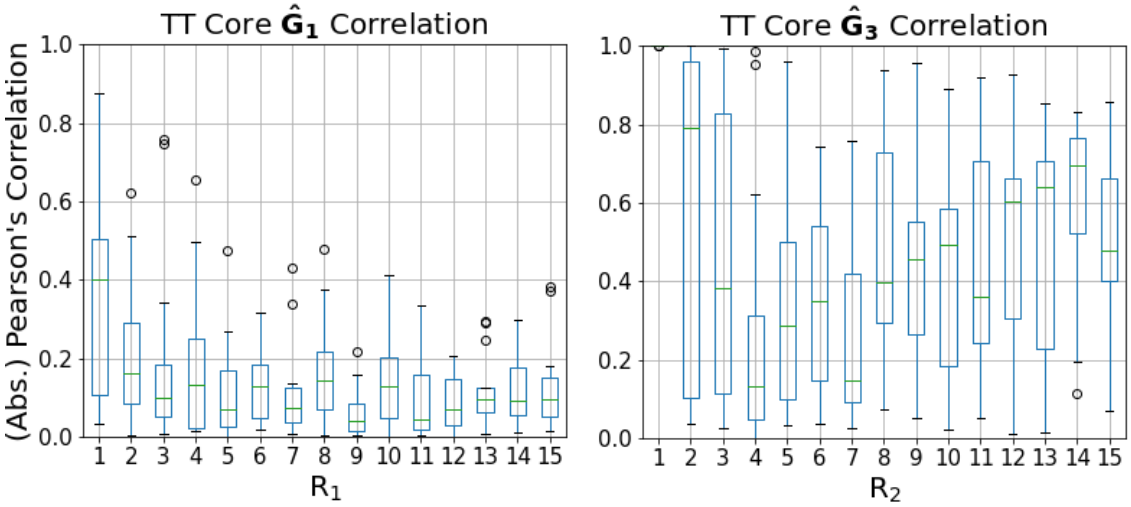

Figure 6.10: Absolute value of the Pearson's correlation between the normalized TT cores generated from the TT-SVD and rTT-SVD algorithms. Left: for TT core $\hat{\mathbf{G}}_1$. Right: for TT core $\hat{\mathbf{G}}_3$. The x-axes refer to R_1 and R_2 rank respectively.

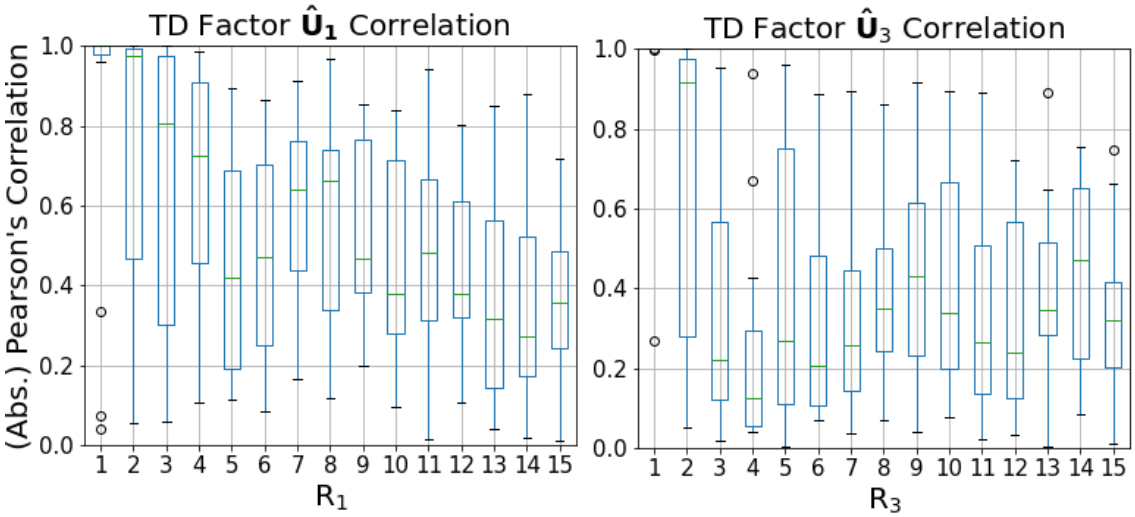

Figure 6.11: Absolute value of the Pearson's correlation between the Tucker factors generated from the HOSVD and rTD algorithms. Left: for TD factor $\hat{\mathbf{U}}_1$. Right: for TD factor $\hat{\mathbf{U}}_3$. The x-axes refer to R_1 and R_3 respectively.

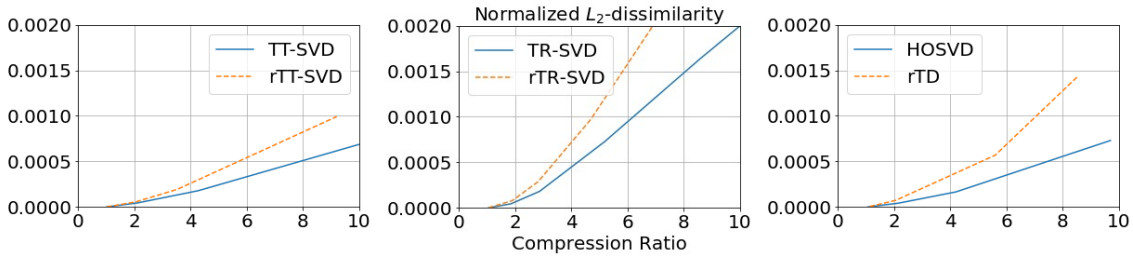

Figure 6.12: Normalized L_2-dissimilarity between the original and reconstructed data from the randomized and non-randomized TN representations for diff. compression ratio.

Chapter 7

Exploring Tensor Decompositions For CSI-based Vital Sign Detection Using Commodity WiFi

In this chapter, we explore different tensor decompositions to extract respiration signals from WiFi channel state information (CSI), the scenarios of our interest include single-person and multi-person situated in quasi-static environments. CSI contains detailed time variation of a number of subcarriers centered at different radio frequencies, which can be used for fine-grained sensing applications. However, the motion dynamics modulated in each subcarrier is different and cannot be expressed as linear mitures of the respiration signals. The theory of respiration sensing has been demonstrated using the Fresnel zone model, it can be shown that the CSI amplitude and phase are complementary in detecting respiration at different body locations with respect to the Fresnel zones. Therefore, we propose to model the complex time series made up of the complementary CSI amplitude and phase in the real and imaginary parts, respectively, as linear superposition of the multi-person respiration signals for the different subcarriers. The problem can then be formulated into blind source separation and solved using tensor decompositions in stochastic or deterministic manner for under-determined WiFi sensing systems.

7.1 Introduction

Occupancy information is an important part of a building management system to optimize the energy consumption, thermal comfort, air quality, security, facility planning and design, etc. For example, an intelligent system may switch on the lighting, heating or air-conditioning for selected areas where the active users are situated. With the worldwide pandemic of coronavirus 2019 (COVID-19), smart buildings may improve the air quality by adjusting the power level of their ventilation systems depending on the number of users and physical activity levels, e.g., activities inside the meeting rooms are usually quasi-static, whereas moderate or vigorous exercises are performed in gym rooms, etc. Hotels, hospitals, and emergency quarantine facilities may inspect the rooms remotely to check if the number of occupants stays within the maximum limit set by governments to prevent COVID-19 spreading and detect potential breach of quarantine orders. Apart from occupancy detection, vital signs such as breathing or heartbeat are important predictors of the health status of COVID-19 / chronic patients and aging population. Therefore, it is essential to provide remote monitoring services for these safety-critical situations in an ubiquitous, cost-effective, and privacy-preserving manner without leaking privacy of the individuals.

Existing indoor monitoring systems can be broadly grouped into device-based and device-free methods. Device-based methods including smart phones, smart watches, RFID tags, and other wearable sensors incur high hardware acquisition and maintenance cost that increase proportionally to the number of subjects involved. Furthermore, the subjects' comfort, consent, and cooperation to wear the devices and download / use the software applications are limiting factors that prevent the device-based methods from widespread deployment for continuous monitoring. Device-free methods such as biomedical radars, video-based, optical-based, and audio-based sensors have long history in the clinical settings and provide good sensing performance. In particular, surveillance cameras have been ubiquitously installed and may serve as a promising approach for remote monitoring. However, video-based methods may invade subjects' privacy and require them to stay within

the line of sight (LOS). On the other hand, commodity WiFi devices are less privacy-invasive, have no LOS issue, and cost-effective for ubiquitous sensing applications by re-using existing hardware infrastructure such as internet of things and commercial-off-the-shelf (COTS) routers that are already widely installed and equipped with transmitter and receiver antennas for wireless communications.

Metals and liquids reflect radio signals, therefore movements of human body (which consists of an average 60% water) are modulated in the surrounding wireless signals and may be extracted by analyzing the wireless data. Nowadays, both the received signal strength indicator (RSSI) and channel state information (CSI) can be obtained conveniently from commodity WiFi devices [222], e.g., via the Intel Wireless Link 5300 NIC or Atheros AR9580 chipset. RSSI is an average received signal power of the entire channel bandwidth, therefore it provides only coarse-grained information for LOS measurements. CSI represents the channel properties of each communication link with fine-grained measurements of multiple subcarriers centered at different frequencies. CSI time series contains both the environmental information and detailed variations over time for both LOS and non-LOS measurements of human movements. Wireless signal processing and analysis algorithms can be categorized into modeling-based and learning-based approaches [223]. Physical, theoretical, and statistical modeling need a lot of effort to build the models, tune the model parameters, and design and implement the signal processing pipeline for particular task or scenario / environment. Compared to learning-based approaches, modeling-based approaches do not require training data collection, ground-truth labeling, and high computational resources for the training of machine learning models. Learning-based approaches, especially deep learning models, are automated in terms of the feature engineering, evolvable with more training data, adaptive to new training instances, reusable in new application scenarios, and achieve impressive performance even with noisy input data like wireless signals, which are highly susceptible to environmental fluctuations and hardware imperfections.

Tensor decompositions or factorizations are multi-dimensional generalization of matrix decomposition techniques. Matrix and tensor algorithms have been developed for big data processing, e.g., missing data can be handled easily within the

framework of stochastic gradient descent, extracted components are less prone to environmental or hardware noise, incremental algorithms support the continuous analysis of streaming data. Unlike their matrix counterpart, specific tensor models such as canonical polyadic decomposition (CPD) [224, 225] and block term decomposition (BTD) [226, 227, 228] guerantee uniqueness under mild conditions without the need to impose any constraints such as non-negativity, orthogonality, and statistical independence on the latent factors. Blind source separation based on matrix decomposition such as independent component analysis allows stochastic separation with well-determined mixtures, i.e., the number of sensors to be equal or larger than the number of source signals. On the other hand, tensorization techniques or linear mapping to higher-order tensors such as Hankelization, Löwnerization, and Segmentation provide deterministic separation of sources that can be modeled by exponential polynomials, rational functions, and sum of Kronecker products, respectively [229]. Tensor decompositions uniquely separate the source signals for under-determined mixtures, given enough samples are collected. Although model-based and learning-based approaches produce outstanding WiFi sensing systems for single-person motion detection, there is a lack of research studies to extend their applications to multi-person and complex / mixed activities recognition comprised of multiple or different types of human motions, this severely limits WiFi sensing for real-life applications. Motivated by these observations, below are the contributions of this work:

- Propose a complex system made up of the complementary CSI amplitude and phase as the real and imaginary parts, respectively, and model the complex CSI time series as linear mixtures of respiration signals. The blind source separation problem is formulated by tensorization of the CSI signals and solved via tensor decompositions.

- Design and prototype a system using commodity 5GHz WiFi devices with 40MHz bandwidth, demonstrate the system performance in different environment settings, and show that reasonable performance can be achieved for vital sign detection.

This chapter is organized as follows. Section 7.2 reviews the related work on radio frequency-based sensing systems for single-person and multi-person vital sign and occupancy detection. Section 7.3 presents the technical background including Fresnel reflection model, CSI calibration methods, and blind source separation based on tensor decomposition. Section 7.4 explains the proposed software system architecture for multi-person sensing applications, the prototype system is experimentally evaluated in Section 7.5, followed by the discussion of this work in Section 7.6.

7.2 Related Work

This section reviews the existing literature on radio frequency (RF)-based sensing systems for vital sign detection with special focus on commodity WiFi for single-person and multi-person sensing scenarios due to its potential for ubiquitous deployment. Several survey papers have been published throughout the years with full or partial focus on vital sign detection using RF-based systems, including [230, 231, 232, 233, 234, 222, 235, 236, 237, 223]. The related work covered here are not exhaustive but updated with the latest research and focuses on the cutting-edge signal processing, fusion, analysis, and separation techniques that may be applicable across different wireless modalities to improve their systems performance.

RF-based sensing systems for vital sign detection. Research studies have explored a number of electromagnetic wave propagation systems to capture breathing / heart-modulated RF signals, which include frequency-modulated continuous wave (FMCW) radar [238, 239, 240, 241, 242, 243, 244, 245, 246], ultra-wide band (UWB) radar [247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259], Doppler radar [260, 261, 262, 263, 264, 265, 266, 267, 268, 269], radio-frequency identification (RFID) [270, 271, 272, 273, 274, 275, 276, 277, 278, 271], acoustic signal [279, 280, 281, 282, 283, 284, 285, 286, 287, 288], and hybrid-mode radars [289, 290]. These systems have different pros and cons during operations [230, 231, 232, 291], thanks to their dedicated hardware systems which result in high resolution or sensing range / performance, they have been deployed for various applications such

as indoor healthcare monitoring [292, 293, 294, 295, 296, 297, 298, 257, 273, 274, 256, 258, 299, 275, 276, 239, 242, 244, 245, 271, 300, 254, 301], in-vehicle monitoring [302, 278, 253, 285, 259, 303], and continuous user authentication [304, 284]. To improve multi-person sensing performance, existing approaches can be roughly grouped into innovative hardware designs / AoA estimation / beamforming techniques (e.g., [305, 306, 240, 307, 265, 308, 309, 266, 268, 310, 311]), software-based signal processing / separation techniques, and their combinations (e.g., [312, 313]). We focus on the signal processing / separation techniques, the techniques reviewed here may be applicable across different RF-based systems for vital sign detection. Lee et al. [243] enhance the range resolution of a 24-GHz bandwidth-limited FMCW radar by doubling the effective frequency bandwidth using a modified waveform. Fang et al. [314] use ensemble empirical mode decomposition (EEMD) algorithm to extract the intrinsic mode functions of RR and HR when two targets' distance is less than the range resolution of FMCW radar. Uysal et al. [315] use unmodulated carrier wave signal because it does not require time synchronization system, their system consists of a motion detector based on adaptive thresholding and a computationally-efficient, high resolution search-free spectrum estimation method (ESPRIT) to estimate the respiration rate even with limited number of data. The radar self-motion (RSM) may affect the vital sign detection and occurs when the radar platform cannot be maintained stationary such as in the automotive and industrial environments. Existing solutions reduce the RSM effect by deploying multiple sensors [316, 317, 318, 319, 320, 321] or using signal processing techniques such as empirical mode decomposition [322, 323], which requires high computational complexity and manual selection of the intrinsic mode functions. Cardillo et al. [324] propose a lightweight and compact solution by extracting the RSM from signals reflected by stationary clutters for small and large radar motions, their prototype system based on FMCW radar is effective against RSM effect for different applications, e.g., radar on moving platforms, vibrating tools, handheld devices, unmanned aerial vehicles, and cars. V^{2}iFi [259] proposes a novel Multi-Sequence Variational Mode Decomposition (MS-VMD) algorithm to leverage on the full UWB bandwidth to perform signal separation via optimization, their system can reliably

estimate the respiration rate, heart rate, and heart rate variability of the driver under driving condition and presence of multiple passengers. Yang et al. [278] attach multiple RFID tags to the seat belt and propose a tensor completion technique to recover missing readings from the phase data collected by the RFID reader and use tensor decomposition to extract the respiration signal of the driver, the proposed technique is effective against strong noises caused by frequency hopping, random sampling, vehicle vibration, and other environmental movements in a driving environment. Existing heart rate detection methods mainly employ fast Fourier transform (FFT), continuous Wavelet transform (CWT), and emerging signal processing techniques such as arctangent demodulation [325], ensemble empirical mode decomposition [326], adaptive noise cancelation [327], feature-based correlation [328], and machine learning [329, 330, 331, 332]. Because of the time criticality of vital sign detection applications, recent developments have considered real-time monitoring systems which reduce the computational complexity during operational deployment, e.g., [333, 334, 335]. Ye et al. [336] propose a zero-attracting sign least-mean-square (ZA-SLMS) algorithm to reconstruct the high-resolution heartbeat spectrum using stochastic gradient approach, they introduce an adaptive regularization parameter in the ZA-SLMS to reduce motion-induced noise and incorporate time-window-variation (TWV) technique for stable HR estimation. Nosrati et al. [267] propose a realistic mechanical heartbeat signal model based on modified Gaussian pulse and exploit the chest-wall acceleration measured by a continuous wave Doppler radar to amplify the heartbeat signal. Ye et al. [337] propose a blind source separation based on non-negative matrix factorization with sparseness constraints to mitigate the noise effect in received radar signal and incorporate the sparse reconstruction to achieve high-resolution heartbeat spectrum. Based on the Fresnel zone model, LungTrack [338] employs an RFID reader and multiple carefully-positioned RFID tags using optimization to address the "blind spot" or "dead zone" issue, their system can simultaneously monitor two human subjects when their locations and orientations are known *a priori*. RespTracker [339] uses the Zadoff-Chu sequence to distinguish different sound reflection paths with a high resolution of less than 10cm, by clustering the multi-dimensional / multipath reflection signals from differ-

ent distances that arrive at multiple microphones, RespTracker [339] can determine whether a given path contains the respiration signal and the user it belongs to up to 3m, which compares favourably against the 0.7-1.1m in conventional acoustic-based sensing systems. Independent component analysis (ICA) has been used to extract the normal, fast, and slow breathing patterns from microwave Doppler radar signals, the motion artifacts are removed using empirical mode decomposition to enhance the signal-to-noise ratio [264]. DeepBreath [340] utilizes an FMCW radio equipped with an antenna array to zoom in on the RF signal observations from different locations in space, they formulate the approximation problem as blind source separation and solve it using ICA to extract breathing patterns from multiple subjects with close separation up to zero distance. DeepBreath [340] uses a convolutional neural network as motion detector to identify stable period and an identity matching algorithm to stitch all reconstructed breathing signals that belong to the same person for continuous monitoring. Ding et al. [341] utilize variational mode decomposition to decompose the respiration signals of multiple targets located at the same distance from a UWB radar into different sub-signals, the time-varying respiration rates can then be tracked via Hilbert transform. Ultra-Wide Band (UWB) radar can accurately measure the travelling distance of the signals, therefore Multi-Breath [342] proposes to transform the UWB radar signal matrices of multiple persons as different RGB images and analyzes using image processing algorithms to identify the breathing cycles. DeepMining [343] uses a single-antenna, narrowband Doppler radar system and extracts the vital signals using frequency separation algorithms based on successive signal cancellation. Wang et al. [313] propose the concept of sensing capability, which enables the theoretical analysis based on the angle, range, and source division multiplexing mechanisms to evaluate the condition in which two targets can be sensed simultaneously given the sensing system parameters. Lee et al. [289] propose a system that distinguishes multiple targets located less than the limit of the theoretical range resolution of a 24-GHz FMCW Doppler Radar by combining the phase information formed by the fast Fourier transform (FFT) method and the range information obtained by the parametric spectral estimation method.

WiFi-based systems for single-person vital sign detection. UbiBreathe [344]

shows that the WiFi received signal strength (RSS) can capture the breathing signal at the line of sight (LOS) and proposes a system that first denoises the RSS signals, detects human motion to identify stable period, and estimates the respiration rate in order to detect cessation of breathing event. Existing solutions based on WiFi RSS signals [345, 346, 347, 344] are limited in terms of their sensing range and performance due to coarse-grained RSS measurements. Wi-Sleep [348, 349] and PhaseBeat [350, 351] systems are among the earlier work to observe the periodic, sinuisoidal-like pattern of vital signals modulated in the WiFi CSI amplitude and phase difference, respectively. Theoretical models based on Fresnel reflection have been proposed for indoor vital sign detection in recent years, it has been shown repeatedly in theoretical and empirical studies that the CSI amplitude and phase are complementary in terms of the vital sign detectability at different subject's locations with respect to the WiFi Fresnel zones, this complementarity property holds in different environments with different placement of the transceiver pair [352, 5, 353, 354, 355, 356]. In the first Fresnel zone (FFZ), radio wave diffraction dominates and hence the Fresnel diffracion model has been used to simulate the diffraction gain of the CSI amplitude signal with respect to the subject's positions and postures, results show that the vital sign detectability in the FFZ varies with the subject's body size, posture, and location [357, 358]. Existing CSI signal fusion methods mostly select subcarriers with larger variance [352, 359], larger mean absolute deviation [350], or periodicity of the time sequence [348] in order to quantify each subcarrier's sensitivity to minute movements. CardioFi [360] empirically observes that the heart rate estimation error of each subcarrier is highly correlated to the variance of the estimation, therefore they propose a novel subcarrier selection method based on spectral stability, which is the inverse of the variance of heart rate estimation from each subcarrier, the results show that CardioFi can estimate the heart rate using commodity WiFi devices equipped with omnidirectional antennas with low estimation error comparable to the systems proposed before [350, 359] that require bulky directional antennas to increase the signal-to-noise ratio (SNR) for heartbeat detection. The extracted vital signals can be split into respiration and heartbeat signals using band-pass filtering because their normal rates fall into dif-

ferent frequency range, i.e., 0.167-0.617Hz (10-37 beats per minute) and 1-1.333Hz (60-80 beats per minute), respectively. WiCare system [361] distinguishes breathing from the micro motions (e.g., reading, writing, talking, etc.) for in-situ monitoring by modeling the breathing signals as periodic sinuisoidal waves and use curve fitting to select subcarriers that exhibit good fit, ICA is then used to isolate the breathing signals from the selected subcarriers. Khan et al. [362] propose to use convolutional neural network to classify the complex CSI time series data if it contains breathing activity, followed by a random forest estimator to determine the breathing rate. To improve the sensing capabilities of WiFi systems, different CSI calibration methods have been proposed to remove the noise from raw CSI measurements and combine the complementary CSI amplitude and phase information for better analysis of the body movements. BreathTrack [363, 364] calibrates the time-invariant phase offset by the hardware correction using cables and splitters and removes the time-varying phase distortions by the phase difference between the CSI at the receiver antennas, BreathTrack utilizes the sparse recovery method to find the dominant path and extract the detailed breath status and the breath rate. CSI ratio between two adjacent antennas of a receiver device can effectively remove both the amplitude impulse noise due to hardware imperfection and the complex phase offset due to unsynchronized clocks of the network devices, most importantly, CSI ratio preserves the correlation between the target's movements and the time variation of the true CSI value [365]. The use of CSI ratio has resulted in higher sensing range and accuracy for several fine-grained applications (e.g., FingerDraw [366] and FarSense [367]) using different wireless modalities (e.g., LTE [368] and LoRa [369]). In particular, FarSense [367] enlarges the sensing range of respiration detection using commodity WiFi from the state-of-the-art 2-4m to 8-9m, the system performance is robust through a 10cm-thick wall which are normally found in residential environments. To preserve the scale of the CSI measurements, CSI conjugate multiplication of two CSI streams from adjacent receiver antennas have been used in different WiFi sensing applications to remove the phase offset, however, it still left with the amplitude impulse noise in the CSI measurements and therefore have to be denoised in the CSI pre-processing stage [353, 370]. The performance of CSI-based vital sign detection systems has

improved over the years and results in many applications such as stationary person detection [371], derivation of the respiration biometrics for continuous user verification/authentication [372, 373], in-vehicle breathing rate monitoring [374, 375, 309], mobile WiFi sensing using wearables / smartphones [376, 377], and indoor healthcare monitoring [378] such as abnormal respiration events detection [379, 380, 381, 291] and contactless sleep monitoring [382, 292, 359, 383, 384].

WiFi-based systems for multi-person vital sign detection. Liu et al. [383] perform Fast Fourier Transform (FFT) on the denoised CSI amplitude from selected subcarriers to find the dominant frequency in the spectral domain that corresponds to the respiration rates of multiple subjects. PhaseBeat [350] employs the Root-MUSIC (Multiple Signal Classification) algorithm to estimate the frequency components in the denoised CSI phase difference. Inspired by the WiFi Fresnel zone model, TinySense [385] and Yang et al. [386] both optimize the deployment of multiple transceiver pairs such that each transmission is affected by only one subject, however, this requires knowing the location of each person in advance for accurate detection. To amplify the minute variations caused by breathing, TR-BREATH [387, 388] projects the CSIs into the Time-Reversal Resonating Strength (TRRS) feature space and extracts the candidates of breathing rates by Root-MUSIC algorithm, TR-BREATH then performs affinity propagation to partition the candidates into clusters corresponding to the breathing of different persons. To extract the detailed waveform of multi-person respiration, TensorBeat [389] and MultiSense [390] systems formulate the problem as blind source separation and solve it using tensor decomposition and ICA, respectively. CSI ratio is not a linear combination of respiration signals, therefore MultiSense proposes a modified version of CSI ratio by minimizing the respiration energy ratio in the denominator of CSI ratio. MultiSense utilizes the ICA and requires the WiFi sensing system to be well-determined to extract multi-person respiration signals, i.e., the number of received signals is equal or larger than the number of source signals. Existing systems require either moving subjects or extensive training to achieve crowd counting, therefore Wang et al. [391] propose to transform the problem into quasi-static, continuous multi-person breathing rate estimation, their system models the natural breathing using Markov chain

assuming that one's breathing rate does not change abruptly within a short time period and employ iterative dynamic programming to extract the breathing traces of multiple subjects in a successive cancellation manner, the breathing traces from different time windows are concatenated using a novel similarity measure provided if the calculated value is above a predefined threshold. State-of-the-art solutions require prior knowledge of the crowd numbers and assume distinct respiration rates of different subjects, therefore Wang et al. [392] further propose an adaptive subcarrier combination method to boost the signal-to-noise ratio of breathing signals and achieve people counting up to four persons, their system can continuously track the breathing rates of multiple users even if some of them merge together for a short time period and recognize particular person based on hypothesis testing on the breathing distribution of extracted traces. In smart car environments, people counting using breathing traces and respiration rates estimation for health monitoring have been considered in order to determine both the driver's state and passengers' health, a lightweight vital sign monitoring system has been proposed in [393] and shown to work even when the engine is running as long as the car is not moving. When the respiration rates of different subjects are close to each other, existing methods could not resolve them using only the frequency domain analysis. Motivated by the fact that WiFi devices nowadays are equipped with multiple antennas, Gao et al. [394] propose a super resolution method to build the two-dimensional Doppler AoA map (DAM) to separate and estimate the respiration rate of each person by clustering and analyzing the DAM from both the Doppler and AoA domain. Mtrack [395] further implements a multi-antenna wideband system that can provide high-resolution AoA and Time of Flight (ToF) information to determine the location of a target, and utilize a 2D beamformer to transform the raw radio signals into the AoA-ToF domain in order to address both the signal separation and mapping of the multiple identified vital signs to the corresponding persons. To alleviate the multipath interference, Mtrack [395] proposes a spatial-temporal path selection method to track the trajectories of moving persons and a correlation-based method to extract the corresponding vital signals of static persons. The feasibility of mmWave signals for vital sign monitoring have been validated under different facing conditions [396, 397, 398].

ViMo [399] employs the emerging 60-GHz WiFi (e.g., 802.11ad) to detect multiple moving / stationary subjects and perform vital sign monitoring. Different from 2.4/5-GHz WiFi, commodity 60-GHz mmWave devices offer high directionality with large phased arrays in small size and provide precise time-of-flight measurements, which allows higher spatial and range resolution making it possible to monitor respiration / heart rate for multiple persons simultaneously.

The closest to our work is TensorBeat, the system also models the CSI source signals with sum of exponentials / exponential polynomials and solve the blind source separation problem in a deterministic manner using tensor decomposition. The major difference is that TensorBeat uses the real-valued CSI phase difference while our work uses the complex-valued CSI conjugate multiplication [400] and CSI ratio [390], which contain the complementary CSI amplitude and phase information for analysis. In our experiments, we also use a more general tensor decomposition technique, namely block-term decomposition, in addition to the Canonical Polyadic decomposition utilized in the TensorBeat system, which allows us to model both the sinusoidal-like vital signals and smooth polynomial-like motion signals in the complex plane. Unlike ICA employed by the MultiSense system [390], tensor decomposition works even for blind signal separation with under-determined system, it does not require the input of the number of subjects but instead estimate the number of subjects based on tensor-rank optimization.

7.3 Technical Preliminary

This section starts with a brief introduction of wireless channel state information in Section 7.3.1. Fresnel reflection model is presented in Section 7.3.2 to explain the complementarity property of CSI amplitude and phase information for respiration and motion signals detection. Section 7.3.3 summarizes the mathematical-based CSI calibration methods to remove the amplitude impulse noise and phase offsets in raw CSI measurements. The problem of extracting individual respiration and motion signals in multi-person sensing scenarios are formulated in the blind source separation described in Section 7.3.4 and solved using tensor decomposition for

under-determined WiFi CSI sensing systems, where the number of received signals from the antenna array of a receiver is less than the number of source signals.

7.3.1 Wireless Channel State Information (CSI)

Channel state information (CSI) represents the channel properties of the communication link between the transmitter and the receiver. The WiFi signals received at the receiver are superposition of the transmitted signals traversed across multiple paths, some of which are caused by the human subjects' movements in the environment. Therefore, both the environment information and variations are modulated in the WiFi signals and can be extracted by analyzing the CSI time series. The WiFi 802.11 protocol has adopted the orthogonal frequency-division multiplexing (OFDM) and multiple-input-multiple-output (MIMO) techniques. With the OFDM, the CSI is measured at multiple subcarriers centered at different frequencies, which provides fine-grained measurements for both macro-scale and micro-scale human motions. With MIMO, the signal is transmitted using transmitter and receiver devices with multiple antennas to exploit the spatial diversity in order to increase the diversity, array, and multiplexing gain. These latest wireless technologies, combined with the computational resources provided by cloud big data processing platforms, create much opportunity to extract high-quality motion signals from WiFi CSI data for real-time analysis and machine learning by using advanced signal processing and fusion techniques to enhance the signal-to-noise ratio.

Mathematically, the received signals at the receiver device can be written as $\mathbf{y}_i = \mathbf{H}_i\mathbf{x}_i + \mathbf{n}_i$, where $\mathbf{x}_i \in \mathbb{R}^{N_T}$ and $\mathbf{y}_i \in \mathbb{R}^{N_R}$ are the transmitted and the received signal for subcarrier i, respectively. N_T is the number of transmitter antennas, N_R is the number of receiver antennas, and $\mathbf{n}_i$ is the noise vector. $\mathbf{H}_i \in \mathbb{C}^{N_R \times N_T}$ denotes the CSI matrix

$$\mathbf{H}_i = \begin{bmatrix} H_i^{11} & H_i^{12} & H_i^{13} & \cdots & H_i^{1N_T} \\ H_i^{21} & H_i^{22} & H_i^{23} & \cdots & H_i^{2N_T} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ H_i^{N_R1} & H_i^{N_R2} & H_i^{N_R3} & \cdots & H_i^{N_RN_T} \end{bmatrix} \tag{7.1}$$

where H_i^{mn} is the CSI of the link between the m-th receiver antenna and the n-th

transmitter antenna for subcarrier i. $H_i^{mn} = |H_i^{mn}| \exp(j \angle H_i^{mn})$ is a complex number where $|H_i^{mn}|$ and $\angle H_i^{mn}$ denote the amplitude and phase, respectively. Due to various reasons, such as hardware imperfections of the commodity WiFi devices and the interference from background electromagnetic noise, most of the CSI measurements are corrupted and unusable without removing the noise. Therefore, in order to extract useful information from the CSI, the CSI sanitization or pre-processing steps are required, which include CSI data calibration, denoising, outlier removal, interpolation to map the unevenly-spaced time series to uniform time interval [222].

7.3.2 Fresnel Reflection Model

Fresnel zone model characterizes the RF signal-propagation properties for a pair of transmitter and receiver antenna which are physically separated at two sites. Fresnel boundaries refer to a series of concentric ellipsoids with foci at the pair of transceivers. As shown in Figure 7.1, the Fresnel zones are numbered from $1, 2, 3, \ldots$ according to the Fresnel boundaries each zone is bounded in between, e.g., the zone bounded by the $(n-1)$-th and n-th Fresnel boundaries is numbered the n-th Fresnel zone. Fresnel zones are caused by radio waves following multiple paths (direct path at the LOS and indirect path due to reflection at NLOS) as they propagate in free space, which result in constructive and destructive interference at the receiver side as the difference in path lengths travelled by the direct and indirect radio waves go in and out of phase. For a given radio wavelength λ, the n-th Fresnel boundary or ellipsoid can be constructed according to the following equation,

$$|P_1 Q_n| + |Q_n P_2| - |P_1 P_2| = \frac{n\lambda}{2} \tag{7.2}$$

where P_1 and P_2 are the positions of the transmitter and receiver, respectively, and Q_n is a point on the n-th ellipsoid. Fresnel model has been used in the design of radio communication systems to prevent significant signal attenuation or obstruction due to obstacles. Ideally, at least 60-80% of the primary Fresnel zones should be clear of obstruction to obtain received signal with high SNR. Let d_1 and d_2 be the distances between the antennas and the intersection point between the LOS and the perpendicular line formed by a point on the n-th Fresnel boundary and the LOS, as

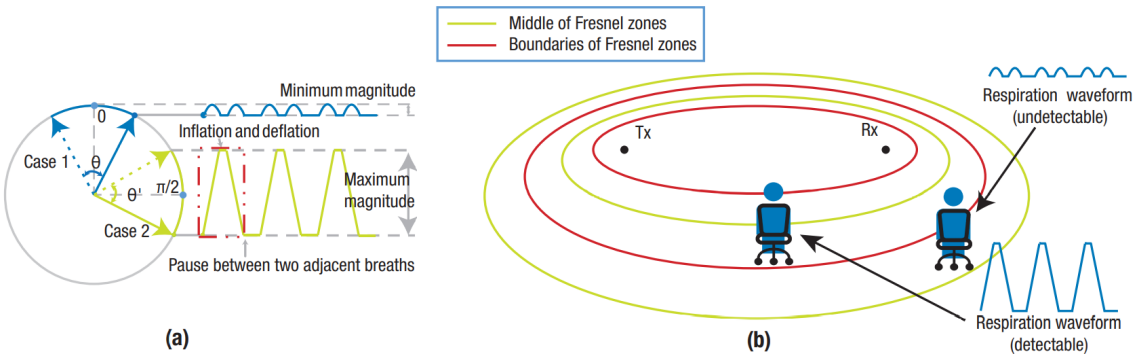

Figure 7.1: Fresnel zone model and respiration sensing with CSI amplitude at different locations [5].

shown in Figure 7.1. The radius of the Fresnel Zones can be calculated using the following approximation when the radius r_n is much smaller than d_1 and d_2 [401]:

$$r_n \approx \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}} \tag{7.3}$$

Respiration sensing for a stationary person. Fresnel reflection model has been demonstrated in the indoor environments to understand the factors affecting detectability of respiration using WiFi sensing systems [352, 5, 353]. Based on the Fresnel reflection model, the CSI signal H (shorthand for H^{11}) can be represented by the superposition of the static component H_s and dynamic component H_d that depend on continuous variables such as the frequency f and time t as follows,

$$H(f,t) = H_s(f) + H_d(f,t) = H_s(f) + A(f,t)e^{-j2\pi\frac{d(t)}{\lambda}} \tag{7.4}$$

where $A(f,t)$ is a complex-valued representation of the signal attenuation and initial phase offset of the dynamic component due to chest movement, $e^{-j2\pi\frac{d(t)}{\lambda}}$ is the phase shift along the dynamic path length $d(t)$. Let $\theta(t)$ be the phase difference between the static and dynamic components, the function of CSI amplitude with respect to $\theta(t)$ can be derived using the relationship between the complex vector components as shown in Figure 7.1 and applying the Pythagoras' theorem,

$$\begin{aligned} |H(f,\theta)|^2 &= (|H_s(f)| + |H_d(f)|\cos\theta)^2 + (|H_d(f)|\sin\theta)^2 \\ &= |H_s|^2 + |H_d(f)|^2 + 2|H_s(f)||H_d(f)|\cos\theta \end{aligned} \tag{7.5}$$

where $|H_d|$ can be considered constant with time when the object moves only a short distance, such as within several wavelengths [352]. Equation 7.5 shows that

the CSI amplitude is a function of $\cos\theta(t)$ that changes with time. For normal and deep breath, the chest displacement Δd is about 5-12mm [352]. Using a 5.24-GHz WiFi, the radio wavelength $\lambda = \frac{speed\ of\ light}{frequency} = 5.725cm$, the phase change due to the human's respiration can be approximated by [352]

$$\Delta\theta \approx 2\pi\frac{2\Delta d}{\lambda}, \rightarrow \Delta\theta : 60^\circ - 150^\circ \tag{7.6}$$

As shown in Figure 7.1, owing to the cosine function with respect to the phase change $\theta_0 \pm \frac{1}{2}\Delta\theta$ in the expression of CSI amplitude in Equation 7.5, the best subject locations for detection happen when $\theta_0 = \frac{\pi}{2}$ or $\frac{3\pi}{2}$, which correspond to the middle of each Fresnel zone; the worst are at the Fresnel boundaries when $\theta_0 = 0$ or π. The chest displacement can be decomposed into two components: one is the effective displacement along the normal line of the Fresnel zone, which causes the reflection path length to change and another along the tangent line, which causes no change in reflection path length. Furthermore, it can be shown that the subcarriers' frequency diversity improves the detectability in the outer Fresnel zones where the Fresnel zones of shorter and longer wavelength do not overlap but more evenly distributed in space [352]. However, the power loss in the reflected path is another determining factor when the subjects are far away from the transceivers in the outer Fresnel zones. The function of CSI phase with respect to the phase difference $\theta(t)$ between the static and dynamic components can be derived using the phase-angular relationship as shown in Figure 7.2 [353],

$$\begin{aligned}\angle H(f,\theta) = \beta' - \alpha' &= \angle H_s(f) - \frac{P}{|H(f,\theta)|} \\ &= \angle H_s(f) - \arcsin\frac{|H_d(f)|\sin\theta}{\sqrt{|H_s(f)|^2 + |H_s(f)|^2 + 2|H_s(f)||H_d(f)|\cos\theta}}\end{aligned} \tag{7.7}$$

Assume the LOS signal is not blocked or attenuated by objects, the static component is always larger than the dynamic component, i.e., $|H_s(f)| >> |H_d(f)|\sin\theta$, therefore the CSI phase can be approximated by

$$\rightarrow \angle H(f,\theta) \approx \angle H_s(f) - \underbrace{\frac{|H_d(f)|}{\sqrt{|H_s(f)|^2 + |H_d(f)|^2 + 2|H_s(f)||H_d(f)|\cos\theta}}}_{constant}\sin\theta \tag{7.8}$$

Because the denominator is much larger than the numerator in the second term, i.e., $|H_s(f)|^2 + |H_d(f)|^2 + 2|H_s(f)||H_d(f)|\cos\theta >> |H_d(f)|^2$, therefore the second

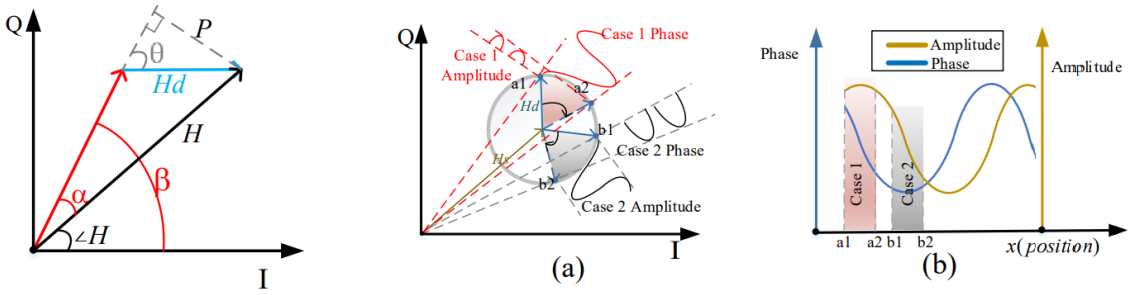

Figure 7.2: CSI amplitude and phase variation for respiration sensing at different subject's locations with respect to the Fresnel zones [5].

term depends on the variation caused by $\sin\theta$ only. In contrast to the CSI amplitude, the best subject locations for respiration sensing using CSI phase happen at the Fresnel boundaries and the worst locations happen at the middle of Fresnel zones. As shown in Figure 7.2, the orthogonality of the sine and cosine bases means that respiration cannot be detectable in all the locations when either CSI amplitude or phase is used, which results in the "blind spot" issue observed in previous work [353]. The complementarity of CSI phase and amplitude is irrelevant to the subjects position, this property is resilient to the environment changes and holds with different placement of the transceiver [353]. In the First Fresnel Zone (FFZ), radio wave diffraction dominates and it has been shown in simulation using Fresnel diffraction model and empirical studies that the respiration sensing capabilities in the FFZ vary with the subject's posture (e.g., sitting or lying) and body size [401], in addition to the subject's location and orientation outside the FFZ based on Fresnel reflection model. These results show that respiration sensing in the FFZ is much more complicated and challenging, therefore we rely solely on empirical results to evaluate the performance of our proposed system in the FFZ.

7.3.3 CSI Calibration Methods

Empirical measurement of wireless CSI signals are heavily corrupted by random complex noise. The CSI measurement can be expressed as a function of the true

CSI signal as shown in previous work [367, 365],

$$\hat{H} = \delta(t)e^{-j\phi(t)}H = \delta(t)e^{-j\phi(t)}(H_s + H_d) = \delta(t)e^{-j\phi(t)}\Big(H_s + \sum_{k\in K} A_k e^{-j2\pi\frac{d_k(t)}{\lambda}}\Big) \quad (7.9)$$

where $\delta(t)$ and $\phi(t)$ are time-varying amplitude impulse noise due to hardware imperfections and complex phase offsets due to unsynchronized clocks between the transmitter and receiver devices, respectively. K denotes the set of dynamic paths due to the targets' movements, A_k denotes the signal attenuation and initial phase shift, and $d_k(t)$ is the path length of the k-th propagation path. The phase offsets consist of the time-varying phase offsets such as carrier frequency offset (CFO), sampling frequency offset (SFO), and packet detection delay (PDD) and constant phase offset such as phase-locked-loop (PLL) initial phase [402]. When accurate phase information is required, the transmitter and the receiver can be connected with coaxial directly to obtain the reference signal if they are close together and under control [363, 403]. Linear transformation on the raw CSI phase can be used to remove significant portion of the random phase offsets based on observations that there are linear correlations among different subcarriers [404]. Phase difference between two adjacent antennas of a receiver has been utilized to eliminate the time-varying phase offsets from CSI phase measurements because the data packets received by adjacent antennas that share the same network card have the same CFO, SFO, and PDD phase offsets over time [405, 350, 389]. However, advanced CSI calibration methods should provide both the complementary CSI amplitude and phase information for better analysis instead of pre-processing the CSI amplitude and phase streams separately, which result in loss of information during the process. CSI conjugate multiplication and CSI ratio are two recent proposals that produce promising results for fine-grained human sensing applications.

CSI conjugate multiplication has been used for phase calibration in a number of WiFi sensing applications in recent years, e.g., [353, 400, 406, 380, 370]. Let $\hat{H}_1$ and $\hat{H}_2$ be the measured CSI signals from two adjacent antennas of a receiver device,

the CSI conjugate multiplication is defined as follows,

$$
\begin{aligned}
H_{cm} \coloneqq \hat{H}_1\hat{H}_2^* &= \Big(\delta(t)e^{-j\phi(t)}\Big(H_{s1} + \sum_{k1\in K1} A_{k1}e^{-j2\pi\frac{d_{k1}(t)}{\lambda}}\Big)\Big)\Big(\delta(t)e^{j\phi(t)}\Big(H_{s2}^* + \sum_{k2\in K2} A_{k2}^*e^{j2\pi\frac{d_{k2}(t)}{\lambda}}\Big)\Big) \\
&= \delta^2(t)\Big(H_{s1} + \sum_{k1\in K1} A_{k1}e^{-j2\pi\frac{d_{k1}(t)}{\lambda}}\Big)\Big(H_{s2}^* + \sum_{k2\in K2} A_{k2}^*e^{j2\pi\frac{d_{k2}(t)}{\lambda}}\Big) \\
&= \delta^2(t)\Big(\underbrace{H_{s1}H_{s2}^*}_{constant} + H_{s1}\sum_{k2\in K2} A_{k2}^*e^{j2\pi\frac{d_{k2}(t)}{\lambda}} + H_{s2}^*\sum_{k1\in K1} A_{k1}e^{-j2\pi\frac{d_{k1}(t)}{\lambda}} + \\
&\qquad \underbrace{\sum_{k1\in K1}\sum_{k2\in K2} A_{k1}A_{k2}^*e^{j2\pi\frac{d_{k2}(t)-d_{k1}(t)}{\lambda}}}_{negligible}\Big)
\end{aligned}
\tag{7.10}
$$

Equation 7.10 shows that H_{cm} removes the phase offsets but amplifies the amplitude impulse noise. The first term is the conjugate multiplication of the background static signals, which is constant over time. The second and third terms are the dynamic signals multiplied by the background signals, which vary with time. The last term is the conjugate multiplication of the dynamic signals, which can be treated as negligible when compared to the magnitude of static signals that come from large number of static paths. The dynamic signals can be further decomposed into the respiration signals from stationary persons and motion signals from moving persons. Suppose that K_S is the set of respiration signals from the stationary subjects that may be modeled by sinusoidal functions,

$$
\begin{aligned}
H_{cm} &\approx \delta^2(t)\Big(H_{s1}H_{s2}^* + H_{s1}\sum_{k2\in K_S} A_{k2}^*e^{j2\pi\frac{d_{k2}(t)}{\lambda}} + H_{s2}^*\sum_{k1\in K_S} A_{k1}e^{-j2\pi\frac{d_{k1}(t)}{\lambda}} + \\
&\qquad H_{s1}\sum_{k2\in K2\setminus K_S} A_{k2}^*e^{j2\pi\frac{d_{k2}(t)}{\lambda}} + H_{s2}^*\sum_{k1\in K1\setminus K_S} A_{k1}e^{-j2\pi\frac{d_{k1}(t)}{\lambda}}\Big) \\
&= \delta^2(t)\Big(H_{s1}H_{s2}^* + H_{s1}\sum_{k1,k2\in K_S} A_{k2}^*\underbrace{e^{j2\pi\frac{d_{k2}(t)-d_{k1}(t)}{\lambda}}}_{constant}e^{j2\pi\frac{d_{k1}(t)}{\lambda}} + H_{s2}^*\sum_{k1\in K_S} A_{k1}e^{-j2\pi\frac{d_{k1}(t)}{\lambda}} + \\
&\qquad H_{s1}\sum_{k2\in K2\setminus K_S} A_{k2}^*e^{j2\pi\frac{d_{k2}(t)}{\lambda}} + H_{s2}^*\sum_{k1\in K1\setminus K_S} A_{k1}e^{-j2\pi\frac{d_{k1}(t)}{\lambda}}\Big)
\end{aligned}
\tag{7.11}
$$

The difference in dynamic path length between adjacent antennas, i.e., $\Delta d_{k1/k2} = d_{k2}(t) - d_{k1}(t)$, depends only on the angle of arrival and distance between the antennas, therefore it does not change when the stationary subject breathes. Assume the

dynamic signals from moving subjects can be grouped into sets of consistent movements $K_M \subseteq (K_1 \cup K_2) \setminus K_S$, e.g., hand, leg, and body movements due to walking, performing exercises, turning around, etc. that may be modeled by smooth polynomial functions,

$$H_{cm} = \delta^2(t)\Big(\underbrace{H_{s1}H_{s2}^*}_{static\ signal} + \sum_{k1,k2\in K_S} \underbrace{\Big(H_{s1}A_{k2}^* e^{j2\pi\frac{\Delta d_{k1/k2}}{\lambda}} e^{j2\pi\frac{d_{k1}(t)}{\lambda}} + H_{s2}^* A_{k1} e^{-j2\pi\frac{d_{k1}(t)}{\lambda}}\Big)}_{respiration\ signal} + \sum_{k1,k2\in K_M} \underbrace{\Big(H_{s1}A_{k2}^* e^{j2\pi\frac{d_{k2}(t)}{\lambda}} + H_{s2}^* A_{k1} e^{-j2\pi\frac{d_{k1}(t)}{\lambda}}\Big)}_{motion\ signal}\Big) \tag{7.12}$$

Equation 7.12 shows that H_{cm} is a linear combination of terms made up of the background static signals, respiration signals of the stationary persons, and motion signals of the moving subjects, therefore it can be used in blind source separation of linear mixtures after removing the multiplicative impulse noise in the CSI data preprocessing stage. For individual's respiration signal, H_{cm} is a superposition of two dynamic vectors with constant magnitudes but opposite rotation direction. When a subject breathes, the locus of H_{cm} is an arc on the ellipse instead of the circle in the ideal CSI signal H. The complementarity property of CSI amplitude and phase for respiration sensing described in Section 7.3.2 still holds for ellipse in the H_{cm}. For individual's motion signal, the dynamics in the complex plane of H_{cm} are much more complicated due to the sum of two dynamic vectors, both with time-dependent amplitudes and phases.

CSI ratio and its modified version. CSI ratio between two adjacent antennas of a receiver device has been utilized to remove the amplitude impulse noise and complex phase offsets, hence increasing the sensing range and accuracy for fine-grained human sensing applications [365]. Formally, CSI ratio is defined as follows for single-person respiration sensing,

$$H_r \coloneqq \frac{\hat{H}_1}{\hat{H}_2} = \frac{\delta(t)e^{-j\phi(t)}(A_1 e^{-j2\pi\frac{d_1(t)}{\lambda}} + H_{s1})}{\delta(t)e^{-j\phi(t)}(A_2 e^{-j2\pi\frac{d_2(t)}{\lambda}} + H_{s2})} = \frac{A_1 e^{-j2\pi\frac{d_1(t)}{\lambda}} + H_{s1}}{A_2 e^{-j2\pi\frac{\Delta d_{1/2}}{\lambda}} e^{-j2\pi\frac{d_1(t)}{\lambda}} + H_{s2}} = \frac{\mathcal{A}\mathcal{Z} + \mathcal{B}}{\mathcal{C}\mathcal{Z} + \mathcal{D}} \tag{7.13}$$

where $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ are complex constants and $\mathcal{Z} = e^{-j2\pi\frac{d_1(t)}{\lambda}}$ is the phase shift due to the subject's respiration. Note that the difference in dynamic path lengths between

adjacent receiver antennas, $\Delta d_{1/2} = d_2(t) - d_1(t)$ can be considered constant as long as the AoA from the stationary subject remains the same. The CSI ratio in Equation 7.13 is a Möbius transformation that can be re-written as follows,

$$H_r = \frac{\mathcal{BC} - \mathcal{AD}}{\mathcal{C}^2} \frac{1}{\mathcal{Z} + \frac{\mathcal{D}}{\mathcal{C}}} + \frac{\mathcal{A}}{\mathcal{C}} = \frac{\beta e^{j\Theta}}{\mathcal{Z} + \alpha} + \gamma \tag{7.14}$$

where β, Θ are real numbers and α, γ are complex numbers. The Möbius transformation can be decomposed into four basic transformation: (1) $\mathcal{Z} \leftarrow \mathcal{Z} + \alpha$ (translation), (2) $\mathcal{Z} \leftarrow \frac{1}{\mathcal{Z}}$ (complex inversion), (3) $\mathcal{Z} \leftarrow \beta e^{j\theta}\mathcal{Z}$ (scaling and rotation), (4) $\mathcal{Z} \leftarrow \mathcal{Z} + \gamma$ (another translation). Note that translation, scaling, and rotation preserve the geometry and rotational properties of $\mathcal{Z}$, the complex inversion preserves the geometry but may change the rotation direction if $\alpha < 1$. Fortunately, this is not the case as long as the static component is larger than the dynamic component, i.e., the LOS signal is not attenuated by objects. Therefore, the CSI ratio preserves the phase relationship of the ideal CSI for stationary-person sensing scenarios, it has been demonstrated in several experimental studies in different fine-grained applications (e.g., FingerDraw [366] and FarSense [367]) and wireless modalities (e.g., LTE [368] and LoRa [369]). However, it is noted that the CSI ratio loses the scale or magnitude of change of the motion dynamics, which may contain useful information for data analysis. For multi-person respiration sensing, the CSI ratio can be expressed as follows,

$$\begin{aligned} H_r \coloneqq \frac{\hat{H}_1}{\hat{H}_2} &= \frac{\delta(t)e^{-j\phi(t)}\left(H_{s1} + \sum_{k1\in K} A_{k1}e^{-j2\pi\frac{d_{k1}(t)}{\lambda}}\right)}{\delta(t)e^{-j\phi(t)}\left(H_{s2} + \sum_{k2\in K} A_{k2}e^{-j2\pi\frac{d_{k2}(t)}{\lambda}}\right)} \\ &= \frac{H_{s1} + \sum_{k1\in K} A_{k1}e^{-j2\pi\frac{d_{k1}(t)}{\lambda}}}{H_{s2} + \sum_{k1,k2\in K} A_{k2}e^{-j2\pi\frac{\Delta d_{k1/k2}}{\lambda}}e^{-j2\pi\frac{d_{k1}(t)}{\lambda}}} \neq H_s + \sum_{k\in K} A_k e^{-j2\pi\frac{d_k(t)}{\lambda}} \end{aligned} \tag{7.15}$$

where $\Delta d_{k1/k2} = d_{k2}(t) - d_{k1}(t)$ can be considered constant for stationary subjects with the same AoA from the receiver antennas. However, Equation 7.15 shows that H_r is not a linear combination of the multi-person respiration signals, it does not preserve the correlation between the subjects' movements and the ideal CSI in general. Zeng et al. [390] propose a modified version of CSI ratio by optimizing the coefficients of linear combination of the CSI from multiple links such that the respiration energy ratio is minimized in the dominator, where the respiration energy

ratio (RER) is defined as the ratio of respiration energy to the overall energy in the CSI amplitude, e.g., the RER calculation can be based on the spectral power of Fast Fourier Transform. Mathematically, the modified version of CSI ratio is defined as follows,

$$\begin{aligned} H_r^{mn} &:= \frac{\hat{H}^{mn}}{\sum_{m'}\sum_{n'} g^*_{m'n'}\hat{H}^{m'n'}} \\ &= \frac{\delta(t)e^{-j\phi(t)}\Big(H_s^{mn} + \sum_{k\in K} A_k^{mn} e^{-j2\pi\frac{d_k^{mn}(t)}{\lambda}}\Big)}{\delta(t)e^{-j\phi(t)}\sum_{m'}\sum_{n'} g^*_{m'n'}\Big(H_s^{m'n'} + \sum_{k\in K} A_k^{m'n'} e^{-j2\pi\frac{d_k^{m'n'}(t)}{\lambda}}\Big)} \\ &\approx \frac{1}{\mu}\Big(\sum_{k\in K} A_k^{mn} e^{-j2\pi\frac{d_k^{mn}(t)}{\lambda}} + H_s^{mn}\Big) \end{aligned} \tag{7.16}$$

where $\mu = \sum_{m'}\sum_{n'} g^*_{m'n'}(H_s^{m'n'} + \sum_{k\in K} A_k^{m'n'} e^{-j2\pi\frac{d_k^{m'n'}(t)}{\lambda}}) \approx \sum_{m'}\sum_{n'} g^*_{m'n'} H_s^{m'n'}$ is approximately constant with time provided if the optimal solution of the linear coefficients $g^*_{m'n'}$ can be found. Such condition happens only when the number of antenna pairs or communication links is larger than the number of respiration signals [390]. Equation 7.16 shows that the modified version of CSI ratio is a linear combination of the multi-person respiration signals and background static signals, therefore it can be used in blind source separation of linear mixtures for quasi-static environments that involve sensing of the minute chest movement of stationary persons.

7.3.4 Blind Source Separation Based On Tensor Decomposition

Tensor decompositions. Canonical polyadic decomposition (CPD) approximates a third-order tensor $\mathcal{X} \in \mathbb{K}^{I_1\times I_2\times I_3}$ with the sum of R rank-one components:

$$\mathcal{X} \approx \sum_{r=1}^{R} \tilde{\boldsymbol{a}}_r \circ \tilde{\boldsymbol{b}}_r \circ \tilde{\boldsymbol{c}}_r \tag{7.17}$$

where $\circ$ is the outer product of two tensors $\tilde{\mathcal{A}} \in \mathbb{K}^{I_1,I_2,\cdots,I_{\tilde{A}}}$ and $\tilde{\mathcal{B}} \in \mathbb{K}^{J_1,J_2,\cdots,J_{\tilde{B}}}$ defined by $(\tilde{\mathcal{A}} \circ \tilde{\mathcal{B}})_{i_1\cdots i_{\tilde{A}} j_1\cdots j_{\tilde{B}}} = \tilde{a}_{i_1\cdots i_{\tilde{A}}}\tilde{b}_{j_1\cdots j_{\tilde{B}}}$. The latent factors of the r-th component are $\tilde{a}_r$, $\tilde{b}_r$, and $\tilde{c}_r$. Figure 7.3 shows the CPD decomposition of a third-order

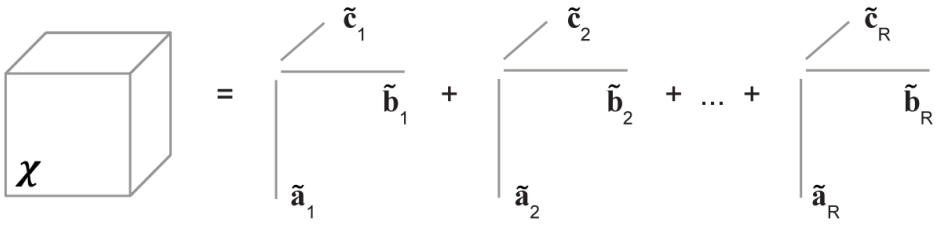

Figure 7.3: CPD of a third-order tensor into sum of R rank-1 terms.

tensor. The rank of the tensor is defined as the smallest R that yields equality in Equation 7.17. The advantage of CPD is its uniqueness up to scaling and permutation indeterminacies under mild conditions, i.e., the rank-one components are sufficiently different from each other and the tensor rank R not unreasonably large. Let $\tilde{\boldsymbol{A}} = \begin{bmatrix} \tilde{\boldsymbol{a}}_1 & \tilde{\boldsymbol{a}}_2 & \cdots & \tilde{\boldsymbol{a}}_R \end{bmatrix} \in \mathbb{K}^{I_1 \times R}$, $\tilde{\boldsymbol{B}} = \begin{bmatrix} \tilde{\boldsymbol{b}}_1 & \tilde{\boldsymbol{b}}_2 & \cdots & \tilde{\boldsymbol{b}}_R \end{bmatrix} \in \mathbb{K}^{I_2 \times R}$, and $\tilde{\boldsymbol{C}} = \begin{bmatrix} \tilde{\boldsymbol{c}}_1 & \tilde{\boldsymbol{c}}_2 & \cdots & \tilde{\boldsymbol{c}}_R \end{bmatrix} \in \mathbb{K}^{I_3 \times R}$, the k-rank of a matrix $\tilde{\boldsymbol{A}}$ (denoted as $k_{\tilde{A}}$) is defined such that any subset of $k_{\tilde{A}}$ columns of matrix $\tilde{A}$ are linearly independent. Kruskal's result shows that a sufficient condition for the uniqueness of CPD is given by

$$k_{\tilde{A}} + k_{\tilde{B}} + k_{\tilde{C}} \geqslant 2R + 2 \tag{7.18}$$

A more general framework has extended the Kruskal's result to multi-way tensors higher than third order [407]. Furthermore, the CPD rank can be selected using the Core Consistency Diagnostic [408].

Block term decomposition, BTD-(L_r,L_r,1) approximates a third-order tensor $\mathcal{X} \in \mathbb{K}^{I_1 \times I_2 \times I_3}$ with a sum of rank-(L_r,L_r,1) terms:

$$\mathcal{X} \approx \sum_{r=1}^{R} (\tilde{\boldsymbol{A}}_r \cdot \tilde{\boldsymbol{B}}_r^T) \circ \tilde{\boldsymbol{c}}_r \tag{7.19}$$

where $\tilde{\boldsymbol{A}}_r \in \mathbb{K}^{I_1 \times L_r}$ and $\tilde{\boldsymbol{B}}_r \in \mathbb{K}^{I_2 \times L_r}$ have linearly-independent columns, and $\tilde{\boldsymbol{c}}_r \in \mathbb{K}^{I_3}$ are non-zero for all $r \in 1, 2, \cdots, R$. Figure 7.4 shows the BTD-(L_r,L_r,1) decomposition of a third-order tensor, it can be observed that BTD-(L_r,L_r,1) is a generalisation of CPD for a third-order tensor by comparing Equations 7.17 and 7.19, and Figures 7.3 and 7.4. When both the matrices $\begin{bmatrix} \tilde{\boldsymbol{A}}_1 & \tilde{\boldsymbol{A}}_2 & \cdots & \tilde{\boldsymbol{A}}_R \end{bmatrix}$ and $\begin{bmatrix} \tilde{\boldsymbol{B}}_1 & \tilde{\boldsymbol{B}}_2 & \cdots & \tilde{\boldsymbol{B}}_R \end{bmatrix}$ have full column rank, and the matrix $\begin{bmatrix} \tilde{\boldsymbol{c}}_1 & \tilde{\boldsymbol{c}}_2 & \cdots & \tilde{\boldsymbol{c}}_R \end{bmatrix}$ does not contain collinear columns, the BTD-(L_r,L_r,1) decomposition is unique up

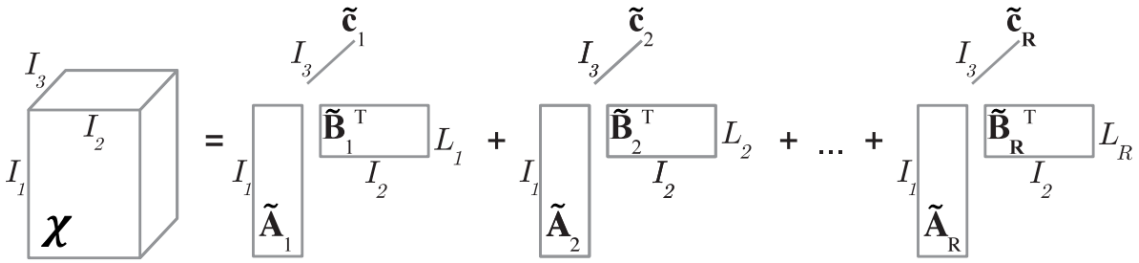

Figure 7.4: BTD-(L_r,L_r,1) of a third-order tensor into sum of rank-(L_r,L_r,1) terms ($1 \leqslant r \leqslant R$).

to scaling and permutation indeterminacies, which are similar to CPD decomposition. An additional indeterminacy is the factors $\tilde{\boldsymbol{A}}_r$ may be postmultiplied by any non-singular matrix $\tilde{\boldsymbol{F}}_r \in \mathbb{K}^{L_r \times L_r}$, provided $\tilde{\boldsymbol{B}}_r$ is premultiplied by the inverse of $\tilde{\boldsymbol{F}}_r$, i.e., $\tilde{\boldsymbol{A}}_r \cdot \tilde{\boldsymbol{B}}_r^T = (\tilde{\boldsymbol{A}}_r \tilde{\boldsymbol{F}}_r) \cdot (\tilde{\boldsymbol{F}}_r^{-1} \tilde{\boldsymbol{B}}_r^T)$.

Tensor decomposition algorithms. Alternating least squares (ALS) algorithm is commonly used to calculate the CPD by updating the factors alternatively during the optimization. For BTD, the ALS with enhanced line search [228] and non-linear least squares methods [409] have been proposed to improve the convergent rate. Additionally, sparsity constraint can be imposed on the latent factors to improve the computational efficiency especially for big data processing applications [410, 411]. Previous studies have also shown that the sparsity-inducing regularization helps in promoting low-rank estimation of the number of BTD blocks (R) and size (L_r) and avoiding collinear blocks with no physical interpretation [412, 413].

Problem formulation for blind source separation. Let the number of received signals be $\tilde{K}$ and the total number of data collection time steps be $2\tilde{T}$, consider the following data model in which the noisy measurements, $\boldsymbol{X} \in \mathbb{K}^{\tilde{K} \times 2\tilde{T}}$ obtained from experiments are linear combinations of the source signals that we are interested to acquire,

$$\boldsymbol{X} = \boldsymbol{M} \cdot \boldsymbol{S}^T + \boldsymbol{N} = \sum_{r=1}^{R} \boldsymbol{m}_r \boldsymbol{s}_r^T + \boldsymbol{N} \tag{7.20}$$

where $\boldsymbol{M} = \begin{bmatrix} \boldsymbol{m}_1 & \boldsymbol{m}_2 & \cdots & \boldsymbol{m}_R \end{bmatrix} \in \mathbb{K}^{\tilde{K} \times R}$ is the unknown mixing matrix, $\boldsymbol{S} = \begin{bmatrix} \boldsymbol{s}_1 & \boldsymbol{s}_2 & \cdots & \boldsymbol{s}_R \end{bmatrix} \in \mathbb{K}^{2\tilde{T} \times R}$ represents the unknown sources in each column, and $\boldsymbol{N} \in \mathbb{K}^{\tilde{K} \times 2\tilde{T}}$ is an additive white Gaussian noise, which is considered as a perturba-

tion in the equation and will be ignored in the presentation later for convenience. Since the matrix decomposition $\boldsymbol{M} \cdot \boldsymbol{S}^T$ is generally non-unique, therefore some assumptions on the structure of $\boldsymbol{M}$ and/or $\boldsymbol{S}$ are usually necessary. For example, the Principal Component Analysis (PCA) assumes orthogonality for both the source and mixing matrices. Existing blind source separation techniques can be grouped into stochastic and deterministic separation. Stochastic separation aims to separate the source signals in a probabilistic manner based on temporal correlation or statistical independence assumptions, whereas deterministic separation assumes the source models to be in certain functional forms.

Stochastic separation. Independent component analysis (ICA) is a popular blind source separation technique for linear mixtures. ICA has been applied to RF-based sensing systems for vital sign detection [390, 340], but previous systems require well-determined mixtures, however, this may not be applicable in practice due to the limited communication links. ICA assumes that the source signals are statistically-independent and non-Gaussian, iterative methods are usually used to optimize the cost function that measures the non-Gaussianity of the source signals, e.g., kurtosis and differential entropy. Here, tensor decomposition techniques are applied for subcarriers signal fusion, blind source separation based on ICA, and incremental analysis for streaming data. Cumulants have interesting properties such that the cumulants higher than second order of Gaussian sources are zero and the covariance between statistically-independent sources is zero. Consider the fourth-order cumulant of $\boldsymbol{X}$ with row mean removed and non-Gaussian, mutually independent source signals, $\mathcal{C}_{\boldsymbol{X}}^{(4)} \in \mathbb{K}^{\tilde{K}\times\tilde{K}\times\tilde{K}\times\tilde{K}}$ can be expressed as follows,

$$\begin{aligned}\mathcal{C}_{\boldsymbol{X}}^{(4)} &\coloneqq \mathbb{E}(\boldsymbol{X}\boldsymbol{X}^H\boldsymbol{X}^H\boldsymbol{X}) - 2\,\mathbb{E}(\boldsymbol{X}\boldsymbol{X}^H)\,\mathbb{E}(\boldsymbol{X}^H\boldsymbol{X}) - \mathbb{E}(\boldsymbol{X}\boldsymbol{X})\,\mathbb{E}(\boldsymbol{X}^H\boldsymbol{X}^H) \\ &= \mathcal{C}_{\boldsymbol{S}}^{(4)} \cdot_1 \boldsymbol{M} \cdot_2 \boldsymbol{M}^* \cdot_3 \boldsymbol{M}^* \cdot_4 \boldsymbol{M} \\ &= \sum_{r=1}^{R} \kappa_r \circ \boldsymbol{m}_r \circ \boldsymbol{m}_r^* \circ \boldsymbol{m}_r^* \circ \boldsymbol{m}_r\end{aligned} \tag{7.21}$$

where $\cdot^H$ is the Hermitian transpose / conjugate transpose, $\cdot^*$ is the complex conjugate, and $\cdot_i$ is the tensor-matrix multiplication on the i-th tensor mode, $\mathcal{C}_{\boldsymbol{S}}^{(4)} \in \mathbb{K}^{R\times R\times R\times R}$ is the fourth-order cumulant of the source $\boldsymbol{S}$ and noise $\boldsymbol{N}$, respectively. $\mathcal{C}_{\boldsymbol{S}}^{(4)}$ is a super-diagonal tensor that contains the kurtosis of the r-th source / compo-

nent, κ_r at the diagonal positions and $\mathcal{C}^N = 0$. Equation 7.21 is in fact decomposition of a symmetric fourth-order tensor into a sum of symmetric rank-1 terms, which can be identified uniquely via the CPD decomposition.

Consider the lagged covariance defined as $\boldsymbol{C}_{\boldsymbol{X}}(\tau) = \mathbb{E}[\boldsymbol{X}(t)\boldsymbol{X}(t+\tau)^H] \in \mathbb{K}^{\tilde{K}\times\tilde{K}}$, the stacked third-order tensor for multiple time lags $\tau_1, \tau_2, \ldots, \tau_L$ is shown as follows,

$$\begin{cases} \boldsymbol{C}_{\boldsymbol{X}}(\tau_1) & = \boldsymbol{M}\boldsymbol{C}_{\boldsymbol{S}}(\tau_1)\boldsymbol{M}^H + \boldsymbol{C}_{\boldsymbol{N}}(\tau_1) \\ \boldsymbol{C}_{\boldsymbol{X}}(\tau_2) & = \boldsymbol{M}\boldsymbol{C}_{\boldsymbol{S}}(\tau_2)\boldsymbol{M}^H + \boldsymbol{C}_{\boldsymbol{N}}(\tau_2) \\ & \vdots \\ \boldsymbol{C}_{\boldsymbol{X}}(\tau_L) & = \boldsymbol{M}\boldsymbol{C}_{\boldsymbol{S}}(\tau_L)\boldsymbol{M}^H + \boldsymbol{C}_{\boldsymbol{N}}(\tau_L) \end{cases} \tag{7.22}$$

The working hypothesis of the ICA based on second-order statistics is that the source signals are mutually independent but individually correlated for different time lags, hence the lagged covariance of the source signals are diagonal matrices [414]. Suppose the noise level is low ($\boldsymbol{C}_{\boldsymbol{N}} \approx 0$), a CPD decomposition emerges by collecting the autocovariance $\sigma_r^2(\tau_\ell), \ell = 1, 2, \ldots, L$ of each source into a vector [415],

$$\mathcal{C}_{\boldsymbol{X}} = \sum_{r=1}^{R} \boldsymbol{m}_r \circ \boldsymbol{m}_r^* \circ \boldsymbol{\sigma}_r^2 \tag{7.23}$$

For non-stationary sources, the covariance of different time segments can be stacked into third-order tensor to improve the identifiability of the mixing vectors [416]. The source signals are estimated based on maximum likelihood because the system may be under-determined in the inner zones, in which all the Fresnel zones of different subcarriers overlap and result in only slight phase shift of the received signal. In the outer zones, the Fresnel zones are more evenly distributed and hence more variations in the received signals.

Deterministic separation. Assume that the source signals can be modeled as linear combination of complex exponentials or more generally exponential polynomials, which are functions that can be written as sums and/or products of exponentials, sinusoids and/or polynomials. Each row of $\boldsymbol{X}$ is first linearly mapped to a $\tilde{T} \times \tilde{T}$ Hankel matrix and stacked into a third-order tensor $\mathcal{X} \in \mathbb{K}^{\tilde{T}\times\tilde{T}\times\tilde{K}}$, more formally,

$$\mathcal{X}_{ijk} = \boldsymbol{X}_{k,i+j-1},\ 1 \leqslant i \leqslant \tilde{T},\ 1 \leqslant j \leqslant \tilde{T},\ 1 \leqslant k \leqslant \tilde{K} \tag{7.24}$$

The $(\tilde{T} \times \tilde{T})$ slices of $\mathcal{X}$ are linear combinations of the Hankel representations of the sources with the coefficients correspond to the entries of the mixing matrix $\boldsymbol{M}$,

$$\mathcal{X} = \sum_{r=1}^{R} \tilde{\boldsymbol{H}}_r \circ \boldsymbol{m}_r \tag{7.25}$$

where $\tilde{\boldsymbol{H}}_r \in \mathbb{K}^{\tilde{T} \times \tilde{T}}$ is the Hankel matrix derived from the r-th column of the source matrix $\boldsymbol{S}$, $1 \leqslant r \leqslant R$. A detailed proof of the the theory underlying the blind separation of exponential polynomials by BTD-(L_r,L_r,1) decomposition can be found in [417]. Suppose the source signals can be expressed as sum of exponentials,

$$s_{r,t+1} = \sum_{l_r=1}^{L_r} c_{l_r,r} z_{l_r,r}^{t}, \;\; 0 \leqslant t \leqslant \tilde{T} - 1, 1 \leqslant r \leqslant R \tag{7.26}$$

where $c_{l_r,r}$ are the complex coefficients and $z_{l_r,r} = e^{\tilde{\alpha}_{l_r,r} + j\tilde{\omega}_{l_r,r}}$ are the poles / exponentials. This model subsumes that the sources might be exponentially-damped sinusoids (e.g., $e^{-\tilde{\alpha}t}\cos(\tilde{\omega}t + \tilde{\phi}) = cz^t + c^*(z^*)^t$ where $c = \frac{1}{2}e^{j\tilde{\phi}}$ and $z = e^{-\tilde{\alpha}+j\tilde{\omega}}$) and hyperbolic sines and cosines (e.g., $\sinh(\tilde{\alpha}t) = \frac{e^{\tilde{\alpha}t} - e^{-\tilde{\alpha}t}}{2}$ and $\cosh(\tilde{\alpha}t) = \frac{e^{\tilde{\alpha}t} + e^{-\tilde{\alpha}t}}{2}$). The Hankel matrices associated with the sources that can be modeled by Equation 7.26 have low rank and admit the Vandermonde decomposition as follows,

$$\tilde{\boldsymbol{H}}_r = \boldsymbol{V}_r \cdot diag(c_{1,r}, c_{2,r}, \cdots, c_{L_r,r}) \cdot \hat{\boldsymbol{V}}_r^T \tag{7.27}$$

If the $\tilde{\boldsymbol{H}}_r$ are square Hankel matrices, then the Vandermonde matrices are given by

$$\boldsymbol{V}_r = \hat{\boldsymbol{V}}_r = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_{1,r} & z_{2,r} & \cdots & z_{L_r,r} \\ \vdots & \vdots & \vdots & \vdots \\ z_{1,r}^{\tilde{T}-1} & z_{2,r}^{\tilde{T}-1} & \cdots & z_{L_r,r}^{\tilde{T}-1} \end{bmatrix} \in \mathbb{K}^{\tilde{T} \times L_r} \tag{7.28}$$

Assuming that $\tilde{T} \geqslant \max(L_1, L_2, \ldots, L_R)$ and all the poles $z_{l_r,r}$ are distinct ($1 \leqslant l_r \leqslant L_r$, $1 \leqslant r \leqslant R$), then the Hankel matrix $\tilde{\boldsymbol{H}}_r$ has full L_r rank. Therefore, the BTD-(L_r,L_r,1) decomposition can solve the blind source separation problem by providing unique decomposition of $\mathcal{X}$. Finally, the poles are distinct if and only if the subjects have distinct respiration rates.

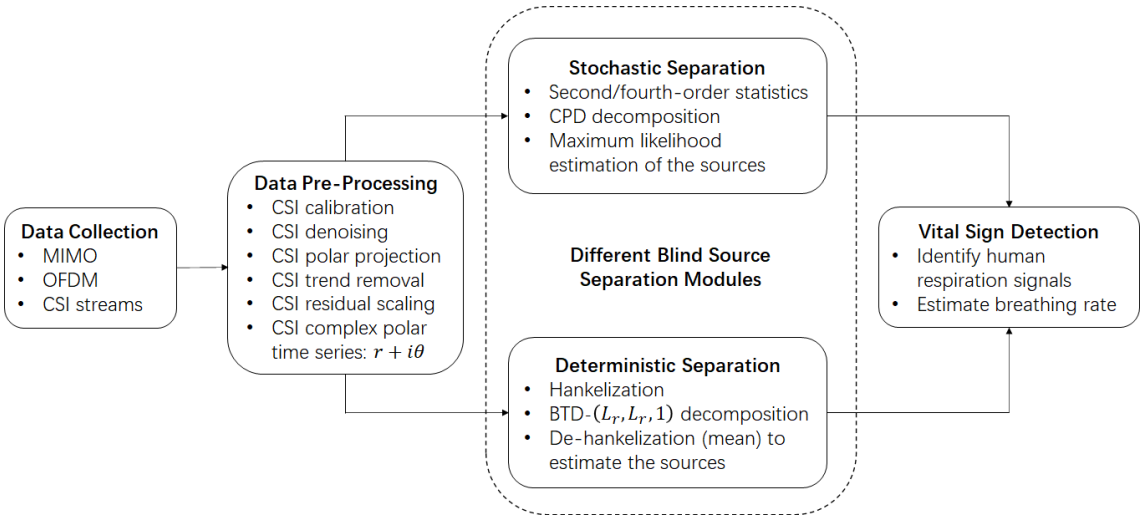

Figure 7.5: System architecture for multi-person vital sign detection in quasi-static environments.

7.4 Proposed System Design And Implementation

Figure 7.5 shows the proposed software system architecture for continous vital sign monitoring of multiple persons in quasi-static environments, e.g., meeting, dining, or resting. This section explains the technical details of the software implementation. The system consists of the CSI data collection, CSI data pre-processing, blind source separation, and detection modules.

Data collection module. The raw CSI data is first collected from the MIMO-OFDM wireless system with physically-separated transmitter and receiver devices, both can be equipped with multiple antennas that form multiple communication links. Each communication link further comprised of multiple subcarriers centered at different radio frequencies, as described in Section 7.3.1.

Data pre-processing module. CSI conjugate multiplication between two adjacent antennas of the receiver device is first computed to remove the random phase offsets due to unsynchronized clocks of network devices as shown in Section 7.3.3. Savitzky-Golay filter is used to smooth the CSI calibrated data by fitting successive subset of data points with low-degree polynomials, this method keeps the shape of respiration waveforms and introduces less distortion than band-pass filtering [353, 418, 419].

The denoised CSI data is projected into the polar coordinates and the trend is removed from the amplitude and phase of CSI conjugate multiplication using Hampel filter. Each subcarrier's residual or detrended CSI amplitude and phase are scaled by their respective standard deviations and combined into a complex polar time series, i.e., $r(t) + i\theta(t)$. this is because the CSI amplitude and phase are complementary and form the orthogonal bases in enhancing the respiration detectability as mentioned in Section 7.3.2.

Blind source separation modules. For the stochastic separation, the second-order and fourth-order statistics are computed for the CSI complex polar time series of each subcarrier and the source signals of (possibly) under-determined mixtures are estimated by maximum likelihood. For the deterministic separation, the time series of each subcarriers is linearly mapped to a set of Hankel matrices and stacked to form a third-order tensor. BTD-$(L_r, L_r, 1)$ decomposition is applied on each CSI time segments. The source signals are retrieved via the de-hankelization process by taking the average along the anti-diagonals of the hankel matrices corresponding to each extracted components.

Detection module. The respiration signals are identified based on the respiration energy ratio, which is defined as the ratio of respiration energy to the total energy [390]. The respiration rate is estimated using peak detection algorithm.

7.5 Experimental Evaluation

In this section, the proposed system is evaluated using a public CSI dataset for single-person sleep monitoring [420] and a set of experimental data collected from commodity WiFi devices. The experimental setup consists of a single person sitting in front of the WiFi transmitter and receiver, which are either in parallel or in-line with each other as shown in Figure 7.6. There are altogether three male subjects, the subject is allowed to change his orientation and talk with another person sitting close-by. The ground-truth measurements of the subject's respiration are collected using the NeuLog respiration sensors. The system performance is evaluated using the Pearson's correlation between the extracted signals and ground truth.

Figure 7.6: Experimental setup with the WiFi transmitter and receiver positioned in aligned (left) or parallel (right) with respect to each other.

Figures 7.7 and 7.8 show the CSI data pre-processing for two subcarriers. The results show that the denoised CSI amplitude and phase signals resemble the ground-truth respiration signals. Figures 7.9 and 7.10 show the extracted source signals using the stochastic and deterministic separation, respectively. Preliminary results show that the deterministic separation produces smoother source signals compared to stochastic separation. Based on the Pearson's correlation score, both methods are useful in extracting the source signals that are highly-correlated with the ground-truth respiration signals.

7.6 Discussion

The proposed system has been demonstrated for single-person sleeping and sitting scenarios. We will collect more data to evaluate the system for multi-person respiration sensing and benchmark with previous systems. The tensor-based approach works for under-determined WiFi sensing systems and supports incremental analysis of streaming CSI data. The limitation of this work is that the respiration signals have not been mapped to the corresponding persons, future work may consider AoA estimation to locate individual subject in the environment [394, 421].

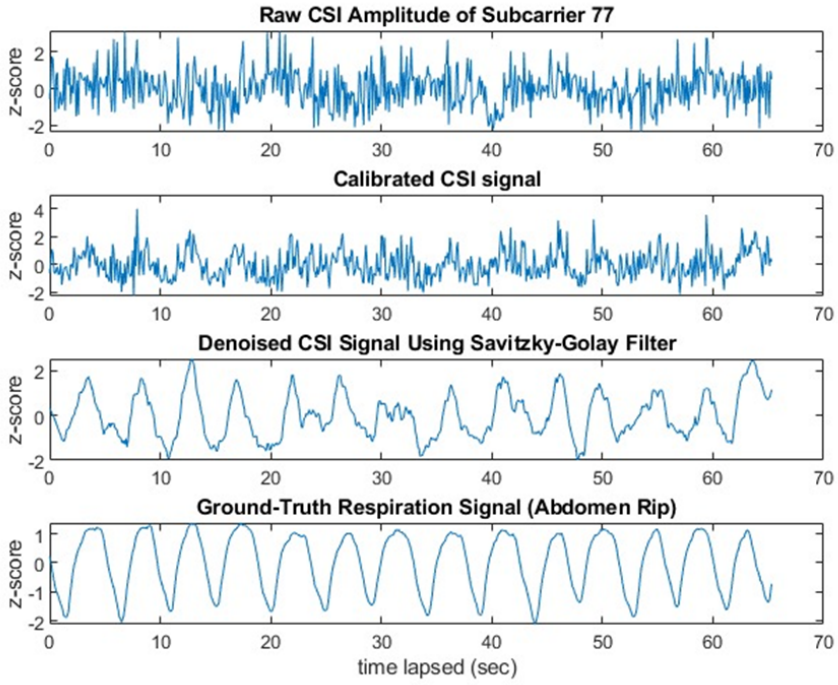

Figure 7.7: Data pre-processing for CSI amplitude signal.

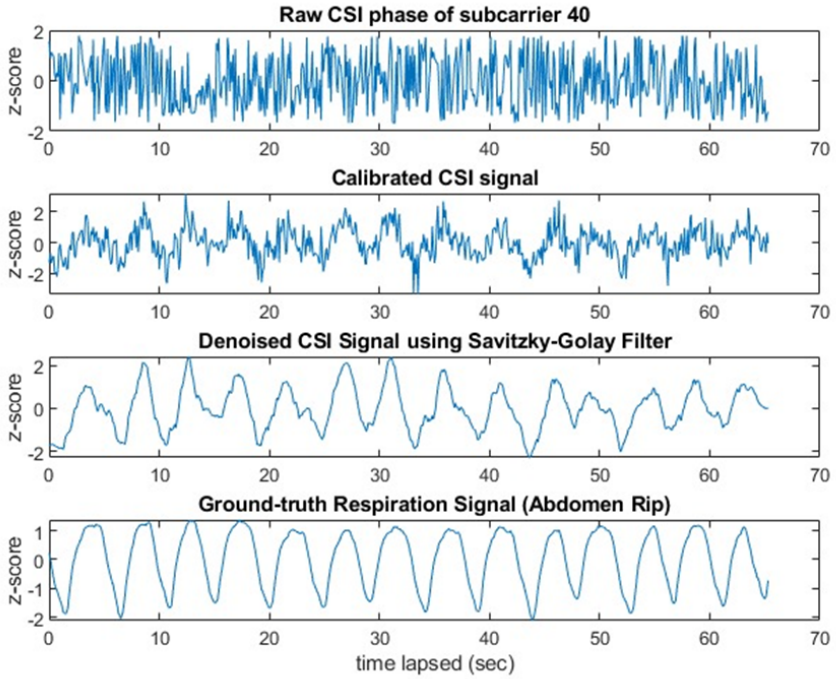

Figure 7.8: Data pre-processing for CSI phase signal.

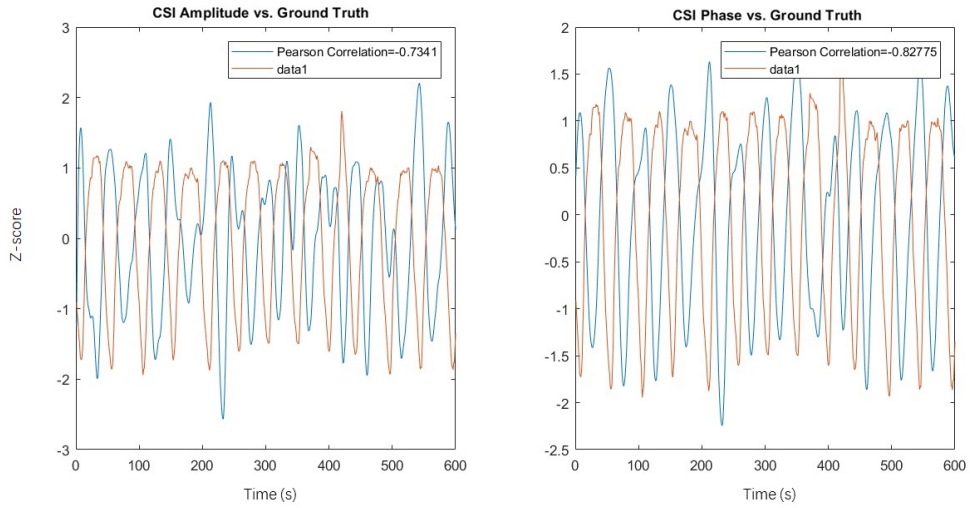

Figure 7.9: Extracted source signal using stochastic separation. Z-score of the extracted (left) CSI amplitude and (right) CSI phase signals. Red curves are the ground-truth respiration signals.

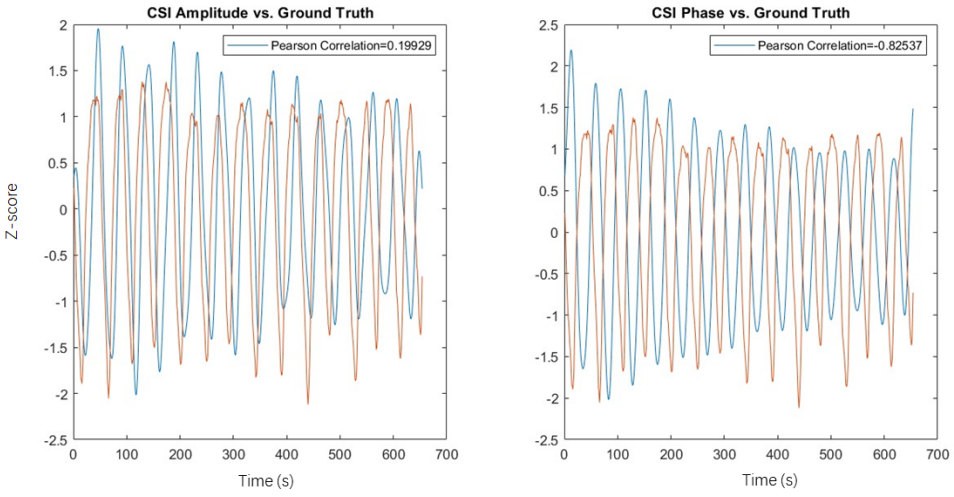

Figure 7.10: Extracted source signal using deterministic separation. Z-score of the extracted (left) CSI amplitude and (right) CSI phase signals. Red curves are the ground-truth respiration signals.

Chapter 8

Conclusion

Tensor decomposition has been widely used in many applications such as signal processing, machine learning, and recommender systems. The capability of tensor decomposition in extracting latent relationship has made tensor techniques useful in data mining, blind source separation, feature extraction, classification, and component analysis. Furthermore, tensor network is able to capture complex correlation structure in high-dimensional data and therefore provide compact representations of big data for analysis. However, the intrinsic non-uniqueness and lack of interpretability makes tensor network less attractive for latent factor analyses, but has been used in different ways such as data imputation and compressed computation. In particular, tensor network computing has found applications in large-scale numerical computing and optimization problems. Tensor network is able to represent high-dimensional data in parsimonious manner (i.e., by-passing the curse of dimensionality) and naturally supports compressed computation on the distributed tensor representations. Therefore, it is natural to ask the question whether tensor network can be used in secure multi-party computation setting for big data privacy preservation. Our findings show that this is indeed feasible by turning tensor network computing from encoding to randomization technique. Encryption requires randomness to generate enough entropy to protect data privacy, the randomness mainly comes from the pseudorandom number generator and encryption algorithm. Our proposed tensor network algorithm harnesses the randomness in complex data to randomize the tensor network representations in order to generate secret shares for

secure multi-party computation, hence the randomness in tensor network is data-dependent and not uniformly distributed. Our results show that tensor network computing is a promising approach in securing big data storage, communication, sharing, and computation when performed in multi-party computation setting.

8.1 Future Work

The next phase of this work will consider big data networking to speed up tensor network computation and broaden the privacy-preserving applications:

- *Tensor-based software-defined networking (SDN) for big data* have recently been proposed to represent and process unstructured, semi-structured, and structured data [76, 78, 422]. Combining SDN and tensor network improves the efficiency of big data applications and security of the architectures [423], however, networking of parallel distributed tensor network computing is still an open problem where latency and throughput are important considerations.

- *Tensor-structured scientific computing.* The use of tensor-structured data formats was recognized as the basic concept for breaking the curse of dimensionality in multidimensional numerical simulations, tensor computing provides an alternative option for efficient design space exploration [71]. Future studies will consider tensor-structured numerical methods in secure multi-party computation setting to enhance the privacy of input data and model parameters, e.g., electronic design automation and advanced materials modeling.

Bibliography

[1] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, and Danilo P. Mandic. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends in Machine Learning*, 9(4-5):249–429, 2016.

[2] Kevin Bache and Moshe Lichman. UCI Machine Learning Repository. *University of California Irvine School of Information*, 2013.

[3] Namgil Lee and Andrzej Cichocki. Fundamental tensor operations for large-scale data analysis using tensor network formats. *Multidimensional Systems and Signal Processing*, Mar 2017.

[4] Namgil Lee and Andrzej Cichocki. Fundamental tensor operations for large-scale data analysis using tensor network formats. *Multidimensional Systems and Signal Processing*, 29(3):921–960, 2018.

[5] D. Zhang, H. Wang, and D. Wu. Toward centimeter-scale human activity sensing with wi-fi signals. *Computer*, 50(1):48–57, 2017.

[6] Andrzej Cichocki, Anh-Huy Phan, Qibin Zhao, Namgil Lee, Ivan Oseledets, Masashi Sugiyama, and Danilo P. Mandic. Tensor networks for dimensionality reduction and large-scale optimization: Part 2 applications and future perspectives. *Foundations and Trends in Machine Learning*, 9(6):431–673, 2017.

[7] Manlio De Domenico, Albert Solé-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason A Porter, Sergio Gómez, and Alex Arenas. Mathe-

matical formulation of multilayer networks. *Physical Review X*, 3(4):041022, 2013.

[8] Yuan Luo, Fei Wang, and Peter Szolovits. Tensor factorization toward precision medicine. *Briefings in bioinformatics*, 18(3):511–514, 2016.

[9] Jonathan Marchini, V Hore, A Vinuela, A Buil, J Knight, M McCarthy, and K Small. Tensor decomposition for multi-tissue gene expression experiments. *Nature Genetics*, 2016.

[10] Esin Karahan, Pedro A Rojas-Lopez, Maria L Bringas-Vega, Pedro A Valdes-Hernandez, and Pedro A Valdes-Sosa. Tensor analysis and fusion of multimodal brain images. *Proceedings of the IEEE*, 103(9):1531–1559, 2015.

[11] Fengyu Cong, Qiu-Hua Lin, Li-Dan Kuang, Xiao-Feng Gong, Piia Astikainen, and Tapani Ristaniemi. Tensor decomposition of eeg signals: a brief review. *Journal of neuroscience methods*, 248:59–69, 2015.

[12] Boris N Khoromskij. *Tensor numerical methods in scientific computing*, volume 19. Walter de Gruyter GmbH & Co KG, 2018.

[13] Boris N Khoromskij. Tensors-structured numerical methods in scientific computing: Survey on recent advances. *Chemometrics and Intelligent Laboratory Systems*, 110(1):1–19, 2012.

[14] Román Orús. Tensor networks for complex quantum systems. *Nature Reviews Physics*, 1(9):538–550, 2019.

[15] Lars Grasedyck, Daniel Kressner, and Christine Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36(1):53–78, 2013.

[16] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2):145–163, 2015.

[17] Nicholas D Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017.

[18] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

[19] Dana Lahat, Tulay Adali, and Christian Jutten. Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects, 2015.

[20] Laurent Sorber, Marc Van Barel, and Lieven De Lathauwer. Structured Data Fusion. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):586–600, 2015.

[21] Guoxu Zhou, Qibin Zhao, Yu Zhang, Tülay Adalı, Shengli Xie, and Andrzej Cichocki. Linked component analysis from matrices to high-order tensors: Applications to biomedical data. *Proceedings of the IEEE*, 104(2):310–331, 2016.

[22] Qingquan Song, Hancheng Ge, James Caverlee, and Xia Hu. Tensor completion algorithms in big data analytics. *arXiv preprint arXiv:1711.10105*, 2017.

[23] Jimeng Sun, Dacheng Tao, Spiros Papadimitriou, Philip S Yu, and Christos Faloutsos. Incremental tensor analysis: Theory and applications. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(3):11, 2008.

[24] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383. ACM, 2006.

[25] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos. Tensor decomposition for signal processing and machine

learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, July 2017.

[26] Evangelos E. Papalexakis, Christos Faloutsos, and Nicholas D. Sidiropoulos. Tensors for Data Mining and Data Fusion. *ACM Transactions on Intelligent Systems and Technology*, 8(2):1–44, 2016.

[27] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2):145–163, 2015.

[28] Animashree Anandkumar, Rong Ge, Daniel J Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

[29] Lars Grasedyck, Daniel Kressner, and Christine Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM Mitteilungen*, 36(1):53–78, 2013.

[30] Wolfgang Hackbusch and Stefan Kühn. A new scheme for the tensor representation. *Journal of Fourier analysis and applications*, 15(5):706–722, 2009.

[31] Lars Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054, 2010.

[32] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[33] Ivan Oseledets and Eugene Tyrtyshnikov. Tt-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432(1):70–88, 2010.

[34] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, Danilo P Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5):249–429, 2016.

[35] Cesar F Caiafa and Andrzej Cichocki. Stable, robust, and super fast reconstruction of tensors using multi-way projections. *IEEE Transactions on Signal Processing*, 63(3):780–793, 2015.

[36] Joel A Tropp, Alp Yurtsever, Madeleine Udell, and Volkan Cevher. Randomized single-view algorithms for low-rank matrix approximation. 2017.

[37] Michael W Mahoney and Petros Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, pages pnas–0803205106, 2009.

[38] Mario Bebendorf. Adaptive cross approximation of multivariate functions. *Constructive approximation*, 34(2):149–179, 2011.

[39] Sergei A Goreinov, Eugene E Tyrtyshnikov, and Nickolai L Zamarashkin. A theory of pseudoskeleton approximations. *Linear algebra and its applications*, 261(1-3):1–21, 1997.

[40] Sergei A Goreinov, Nikolai Leonidovich Zamarashkin, and Evgenii Evgenevich Tyrtyshnikov. Pseudo-skeleton approximations by matrices of maximal volume. *Mathematical Notes*, 62(4):515–519, 1997.

[41] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, 2014.

[42] Glen Evenbly and Guifre Vidal. Tensor network renormalization yields the multiscale entanglement renormalization ansatz. *Physical review letters*, 115(20):200401, 2015.

[43] Andrzej Cichocki, Anh-Huy Phan, Qibin Zhao, Namgil Lee, Ivan Oseledets, Masashi Sugiyama, Danilo P Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 2 applications and future perspectives. *Foundations and Trends® in Machine Learning*, 9(6):431–673, 2017.

[44] Sinduja T and Dr. K. Thippeswamy. A Survey on Secure Cloud Data in Decomposition of Encrypted Subtensor Using Homomorphic Encryption Scheme. *International Journal of Trend in Research and Development*, 3(2):492–496, 2016.

[45] Liwei Kuang, Laurence Yang, Jun Feng, and Mianxiong Dong. Secure tensor decomposition using fully homomorphic encryption scheme. *IEEE Transactions on Cloud Computing*, 2015.

[46] Yining Wang and Animashree Anandkumar. Online and Differentially-Private Tensor Decomposition. *NIPS*, pages 1–20, 2016.

[47] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[48] Anh Huy Phan and Andrzej Cichocki. Tensor decompositions for feature extraction and classification of high dimensional datasets. *Nonlinear theory and its applications, IEICE*, 1(1):37–68, 2010.

[49] Johann A Bengua, Ho N Phien, and Hoang D Tuan. Optimal feature extraction and classification of tensors via matrix product state decomposition. In *Big Data (BigData Congress), 2015 IEEE International Congress on*, pages 669–672. IEEE, 2015.

[50] Qingquan Song, Hancheng Ge, James Caverlee, and Xia Hu. Tensor completion algorithms in big data analytics. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(1):1–48, 2019.

[51] Johann A Bengua, Phien N Ho, Hoang Duong Tuan, and Minh N Do. Matrix product state for higher-order tensor compression and classification. *IEEE Transactions on Signal Processing*, 65(15):4019–4030, 2017.

[52] Evgeny Frolov and Ivan Oseledets. Tensor methods and recommender systems. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(3):e1201, 2017.

[53] Adrian R Groves, Christian F Beckmann, Steve M Smith, and Mark W Woolrich. Linked independent component analysis for multimodal data fusion. *Neuroimage*, 54(3):2198–2217, 2011.

[54] Yongxin Yang and Timothy Hospedales. Deep multi-task representation learning: A tensor factorisation approach. *ICLR*, 2016.

[55] Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. In *Conference on Learning Theory*, pages 698–728, 2016.

[56] Nadav Cohen and Amnon Shashua. Convolutional rectifier networks as generalized tensor decompositions. In *International Conference on Machine Learning*, pages 955–963, 2016.

[57] Valentin Khrulkov, Alexander Novikov, and Ivan Oseledets. Expressive power of recurrent neural networks. *arXiv preprint arXiv:1711.00811*, 2017.

[58] Edwin Stoudenmire and David J Schwab. Supervised learning with tensor networks. In *Advances in Neural Information Processing Systems*, pages 4799–4807, 2016.

[59] E Miles Stoudenmire. Learning relevant features of data with multi-scale tensor networks. *Quantum Science and Technology*, 3(3):034003, 2018.

[60] Alexander Novikov, Dmitrii Podoprikhin, Anton Osokin, and Dmitry P Vetrov. Tensorizing neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 442–450. Curran Associates, Inc., 2015.

[61] Yong-Deok Kim, Eunhyeok Park, Sungjoo Yoo, Taelim Choi, Lu Yang, and Dongjun Shin. Compression of deep convolutional neural networks for fast and low power mobile applications. *ICLR*, 2015.

[62] Cheng Tai, Tong Xiao, Yi Zhang, Xiaogang Wang, et al. Convolutional neural networks with low-rank regularization. *International Conference on Learning Representations*, 2015.

[63] Vadim Lebedev, Yaroslav Ganin, Maksim Rakhuba, Ivan Oseledets, and Victor Lempitsky. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. *ICLR*, 2015.

[64] Wenqi Wang, Yifan Sun, Brian Eriksson, Wenlin Wang, and Vaneet Aggarwal. Wide compression: Tensor ring nets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9329–9338, 2018.

[65] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. Compressing recurrent neural network with tensor train. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 4451–4458. IEEE, 2017.

[66] Chen Yunpeng, Jin Xiaojie, Kang Bingyi, Feng Jiashi, and Yan Shuicheng. Sharing residual units through collective tensor factorization in deep neural networks. *IJCAI*, 2017.

[67] Jen-Tzung Chien and Yi-Ting Bao. Tensor-factorized neural networks. *IEEE transactions on neural networks and learning systems*, 2017.

[68] Jean Kossaifi, Aran Khanna, Zachary Lipton, Tommaso Furlanello, and Anima Anandkumar. Tensor contraction layers for parsimonious deep nets. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1940–1946. IEEE, 2017.

[69] Boris N. Khoromskij. Tensors-structured numerical methods in scientific computing: Survey on recent advances. *Chemometrics and Intelligent Laboratory Systems*, 110(1):1 – 19, 2012.

[70] Nico Vervliet, Otto Debals, Laurent Sorber, and Lieven De Lathauwer. Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis. *IEEE Signal Processing Magazine*, 31(5):71–79, 2014.

[71] Z. Zhang, K. Batselier, H. Liu, L. Daniel, and N. Wong. Tensor computation: A new framework for high-dimensional problems in eda. *IEEE Transactions*

on Computer-Aided Design of Integrated Circuits and Systems, 36(4):521–536, April 2017.

[72] Shui Yu, Meng Liu, Wanchun Dou, Xiting Liu, and Sanming Zhou. Networking for big data: A survey. *IEEE Communications Surveys & Tutorials*, 19(1):531–549, 2017.

[73] L. Kuang, L. T. Yang, X. Wang, P. Wang, and Y. Zhao. A tensor-based big data model for qos improvement in software defined networks. *IEEE Network*, 30(1):30–35, January 2016.

[74] Liwei Kuang, Laurence T. Yang, Seungmin (Charlie) Rho, Zheng Yan, and Kai Qiu. A tensor-based framework for software-defined cloud data center. *ACM Trans. Multimedia Comput. Commun. Appl.*, 12(5s):74:1–74:23, December 2016.

[75] Laurence T Yang, Liwei Kuang, Jinjun Chen, Fei Hao, and Changqing Luo. A holistic approach to distributed dimensionality reduction of big data. *IEEE Transactions on Cloud Computing*, 2015.

[76] Liwei Kuang, Fei Hao, Laurence T Yang, Man Lin, Changqing Luo, and Geyong Min. A tensor-based approach for big data representation and dimensionality reduction. *IEEE transactions on emerging topics in computing*, 2(3):280–291, 2014.

[77] Evrim Acar, Animashree Anandkumar, Lenore Mullin, Sebnem Rusitschka, and Volker Tresp. Tensor computing for internet of things. *Dagstuhl Reports*, 6:57–79, 2016.

[78] L. Kuang, L. T. Yang, and K. Qiu. Tensor-based software-defined internet of things. *IEEE Wireless Communications*, 23(5):84–89, October 2016.

[79] Xiaokang Wang, Laurence T Yang, Huazhong Liu, and M Jamal Deen. A big data-as-a-service framework: State-of-the-art and perspectives. *IEEE Transactions on Big Data*, 4(3):325–340, 2018.

[80] Xiaokang Wang, Laurence T Yang, Liwei Kuang, Xingang Liu, Qingxia Zhang, and M Jamal Deen. A tensor-based big-data-driven routing recommendation approach for heterogeneous networks. *IEEE Network*, 33(1):64–69, 2019.

[81] Kim Batselier, Zhongming Chen, and Ngai Wong. Tensor network alternating linear scheme for mimo volterra system identification. *Automatica*, 84:26–35, 2017.

[82] Kim Batselier, Zhongming Chen, and Ngai Wong. A tensor network kalman filter with an application in recursive mimo volterra system identification. *Automatica*, 84:17–25, 2017.

[83] Gérard Favier, C Alexandre R Fernandes, and André LF de Almeida. Nested tucker tensor decomposition with application to mimo relay systems using tensor space–time coding (tstc). *Signal Processing*, 128:318–331, 2016.

[84] Gérard Favier and André LF de Almeida. Tensor space-time-frequency coding with semi-blind receivers for mimo wireless communication systems. *IEEE Transactions on Signal Processing*, 62(22):5987–6002, 2014.

[85] Ítalo Vitor Cavalcante, André LF de Almeida, and Martin Haardt. Tensor-based approach to channel estimation in amplify-and-forward mimo relaying systems. In *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 445–448. IEEE, 2014.

[86] Leandro R Ximenes, Gerard Favier, Andre LF de Almeida, and Yuri CB Silva. Parafac-paratuck semi-blind receivers for two-hop cooperative mimo relay systems. *IEEE Transactions on Signal Processing*, 62(14):3604–3615, 2014.

[87] André LF de Almeida, Gérard Favier, and Leandro R Ximenes. Space-time-frequency (stf) mimo communication systems with blind receiver based on a generalized paratuck2 model. *IEEE Transactions on Signal Processing*, 61(8):1895–1909, 2013.

[88] Fredrik Kjolstad, Shoaib Kamil, Stephen Chou, David Lugato, and Saman Amarasinghe. The tensor algebra compiler. *Proceedings of the ACM on Programming Languages*, 1(OOPSLA):1–29, 2017.

[89] Evangelos E Papalexakis, Christos Faloutsos, and Nicholas D Sidiropoulos. Parcube: Sparse parallelizable tensor decompositions. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 521–536. Springer, 2012.

[90] U Kang, Evangelos Papalexakis, Abhay Harpale, and Christos Faloutsos. Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 316–324. ACM, 2012.

[91] Inah Jeon, Evangelos E Papalexakis, Uksong Kang, and Christos Faloutsos. Haten2: Billion-scale tensor decompositions. In *2015 IEEE 31st International Conference on Data Engineering*, pages 1047–1058. IEEE, 2015.

[92] Alex Beutel, Partha Pratim Talukdar, Abhimanu Kumar, Christos Faloutsos, Evangelos E Papalexakis, and Eric P Xing. Flexifact: Scalable flexible factorization of coupled tensors on hadoop. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 109–117. SIAM, 2014.

[93] Ramakrishnan Kannan, Grey Ballard, and Haesun Park. Mpi-faun: an mpi-based framework for alternating-updating nonnegative matrix factorization. *IEEE Transactions on Knowledge and Data Engineering*, 30(3):544–558, 2017.

[94] Nicholas D Sidiropoulos, Evangelos E Papalexakis, and Christos Faloutsos. Parallel randomly compressed cubes: A scalable distributed architecture for big tensor decomposition. *IEEE Signal Processing Magazine*, 31(5):57–70, 2014.

[95] Rong Ge, Qingqing Huang, and Sham M. Kakade. Learning Mixtures of Gaussians in High Dimensions. *STOC '15*, 2015.

[96] Santosh S. Vempala. Technical Perspective Modeling high-Dimensional Data. *Communications of the ACM*, 55(2):112, 2012.

[97] Oren Barkan and Amir Averbuch. Robust Mixture Models for Anomaly Detection. *IEEE Internation Workshop on Machine Learning for Signal Processing*, 2016.

[98] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

[99] D Peel and G J McLachlan. Robust mixture modelling using the t distribution. *Statistics and Computing*, 10(4):339–348, 2000.

[100] Greg Hamerly and Charles Elkan. Learning the k in k-means. *Neural Information Processing Systems*, pages 281–288, 2003.

[101] Mingzhou Song and Hongbin Wang. Highly efficient incremental estimation of Gaussian mixture models for online data stream clustering. *Intelligent Computing: Theory and Applications*, 5803:174–183, 2005.

[102] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*, 10(1-3):19–41, 2000.

[103] Stephane Lafon. Diffusion Maps and Geometric Harmonics. *PhD thesis, Yale University, U.S.A.*, page 97, 2004.

[104] Christopher M Bishop. Pattern Recognition and Machine Learning. *Pattern Recognition*, 4(4):738, 2006.

[105] Kamalika Chaudhuri, Sanjoy Dasgupta, and Andrea Vattani. Learning Mixtures of Gaussians using the k-means Algorithm. *Arxiv preprint arXiv09120086*, pages 1–22, 2009.

[106] Mohammad Sazzadul Hoque, Md Abdul Mukit, Md Abu Naser Bikas, and Mohammad Sazzadul Hoque. An Implementation of Intrusion Detection Sys-

tem Using Genetic Algorithm. *International Journal of Network Security Its Applications*, 4(2):109–120, 2012.

[107] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[108] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S. Emer. Efficient processing of deep neural networks: A tutorial and survey. *CoRR*, abs/1703.09039, 2017.

[109] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, 1989.

[110] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[111] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The mnist database of handwritten digits. 1998.

[112] A. Krizhevsky, V. Nair, and G. Hinton. Learning multiple layers of features from tiny images. 2009.

[113] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *arXiv preprint arXiv:1801.00553*, 2018.

[114] Manish Narwaria and Weisi Lin. Svd-based quality metric for image and video using machine learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):347–364, 2012.

[115] M Alex O Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *European Conference on Computer Vision*, pages 447–460. Springer, 2002.

[116] Xiang Zhu and Peyman Milanfar. Automatic parameter selection for denoising algorithms using a no-reference measure of image content. *IEEE transactions on image processing*, 19(12):3116–3132, 2010.

[117] Gilbert W Stewart. Stochastic perturbation theory. *SIAM review*, 32(4):579–610, 1990.

[118] Jun Liu, Xiangqian Liu, and Xiaoli Ma. First-order perturbation analysis of singular vectors in singular value decomposition. *IEEE Transactions on Signal Processing*, 56(7):3044–3049, 2008.

[119] C-C Jay Kuo. The cnn as a guided multilayer recos transform [lecture notes]. *IEEE Signal Processing Magazine*, 34(3):81–89, 2017.

[120] C-C Jay Kuo and Yueru Chen. On data-driven saak transform. *Journal of Visual Communication and Image Representation*, 50:237–246, 2018.

[121] C-C Jay Kuo, Min Zhang, Siyang Li, Jiali Duan, and Yueru Chen. Interpretable convolutional neural networks via feedforward design. *arXiv preprint arXiv:1810.02786*, 2018.

[122] Yueru Chen, Zhuwei Xu, Shanshan Cai, Yujian Lang, and C-C Jay Kuo. A saak transform approach to efficient, scalable and robust handwritten digits recognition. In *2018 Picture Coding Symposium (PCS)*, pages 174–178. IEEE, 2018.

[123] A-J Van Der Veen, ED F Deprettere, and A Lee Swindlehurst. Subspace-based signal analysis using singular value decomposition. *Proceedings of the IEEE*, 81(9):1277–1308, 1993.

[124] Gilbert W Stewart. Perturbation theory for the singular value decomposition. *In SVD and Signal Processing, II: Algorithms, Analysis and Applications*, 1991.

[125] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. *ICLR*, 2017.

[126] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, page 5, 2011.

[127] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[128] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *In The Network and Distributed System Security Symposium (NDSS)*, 2017.

[129] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.

[130] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001.

[131] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[132] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.

[133] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[134] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh

Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[135] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016.

[136] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[137] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[138] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *In Proc. ICLR*, 2015.

[139] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[140] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016.

[141] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[142] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. *arXiv preprint arXiv:1709.04114*, 2017.

[143] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1765–1773, 2017.

[144] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *In Machine Learning and Computer Security Workshop, Neural Information Processing Systems*, 2017.

[145] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *ICLR*, 2018.

[146] Beranger Dumont, Simona Maggio, and Pablo Montalvo. Robustness of rotation-equivariant networks to adversarial perturbations. *arXiv preprint arXiv:1802.06627*, 2018.

[147] Abigail Graese, Andras Rozsa, and Terrance E Boult. Assessing threat of adversarial examples on deep neural networks. In *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*, pages 69–74. IEEE, 2016.

[148] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019.

[149] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

[150] Shilin Qiu, Qihe Liu, Shijie Zhou, and Chunjiang Wu. Review of artificial intelligence adversarial attack and defense technologies. *Applied Sciences*, 9(5):909, 2019.

[151] Han Xu Yao Ma Hao-Chen, Liu Debayan Deb, Hui Liu Ji-Liang Tang Anil, and K Jain. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 17(2):151–178, 2020.

[152] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018.

[153] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[154] Ali Shafahi, W Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? *arXiv preprint arXiv:1809.02104*, 2018.

[155] Thomas Tanay and Lewis Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690*, 2016.

[156] Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. Adversarial vulnerability for any classifier. In *Advances in neural information processing systems*, pages 1178–1187, 2018.

[157] Saeed Mahloujifar, Dimitrios I Diochnos, and Mohammad Mahmoody. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4536–4543, 2019.

[158] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019.

[159] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[160] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[161] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, July 2018.

[162] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

[163] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019.

[164] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[165] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 162–167. IEEE, 1986.

[166] Ronald Cramer, Ivan Damgard, and J Buus Nielsen. Secure multiparty computation and secret sharing-an information theoretic approach book draft, 2012.

[167] David Evans, Vladimir Kolesnikov, Mike Rosulek, et al. A pragmatic introduction to secure multi-party computation. *Foundations and Trends® in Privacy and Security*, 2(2-3):70–246, 2018.

[168] Craig Gentry et al. Fully homomorphic encryption using ideal lattices. In *Stoc*, volume 9, pages 169–178, 2009.

[169] Hugo Krawczyk. Secret sharing made short. In *Annual international cryptology conference*, pages 136–146. Springer, 1993.

[170] Katarzyna Kapusta and Gerard Memmi. Data protection by means of fragmentation in distributed storage systems. In *2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS)*, pages 1–8. IEEE, 2015.

[171] Gérard Memmi, Katarzyna Kapusta, and Han Qiu. Data protection: Combining fragmentation, encryption, and dispersion. In *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*, pages 1–9. IEEE, 2015.

[172] Katarzyna Kapusta and Gerard Memmi. Data protection by means of fragmentation in various different distributed storage systems-a survey. *arXiv preprint arXiv:1706.05960*, 2017.

[173] Ronald L Rivest. All-or-nothing encryption and the package transform. In *International Workshop on Fast Software Encryption*, pages 210–218. Springer, 1997.

[174] Han Qiu, Katarzyna Kapusta, Zhihui Lu, Meikang Qiu, and Gerard Memmi. All-or-nothing data protection for ubiquitous communication: Challenges and perspectives. *Information Sciences*, 502:434–445, 2019.

[175] K. Kapusta, H. Qiu, and G. Memmi. Poster abstract: Secure data sharing by means of fragmentation, encryption, and dispersion. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1051–1052, 2019.

[176] K. Kapusta, H. Qiu, and G. Memmi. Secure data sharing with fast access revocation through untrusted clouds. In *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5, 2019.

[177] Benjamin Fabian, Tatiana Ermakova, and Philipp Junghanns. Collaborative and secure sharing of healthcare data in multi-clouds. *Information Systems*, 48:132–150, 2015.

[178] Buket Yüksel, Alptekin Küpçü, and Öznur Özkasap. Research issues for privacy and security of electronic health services. *Future Generation Computer Systems*, 68:1–13, 2017.

[179] Josep Domingo-Ferrer, Oriol Farràs, Jordi Ribes-González, and David Sánchez. Privacy-preserving cloud computing on sensitive data: A survey of methods, products and challenges. *Computer Communications*, 140:38–60, 2019.

[180] David Sánchez and Montserrat Batet. Privacy-preserving data outsourcing in the cloud via semantic data splitting. *Computer Communications*, 110:187–201, 2017.

[181] Gagan Aggarwal, Mayank Bawa, Prasanna Ganesan, Hector Garcia-Molina, Krishnaram Kenthapadi, Rajeev Motwani, Utkarsh Srivastava, Dilys Thomas, and Ying Xu. Two can keep a secret: A distributed architecture for secure database services. *CIDR 2005*, 2005.

[182] Isabel Wagner and David Eckhoff. Technical privacy metrics: a systematic survey. *ACM Computing Surveys (CSUR)*, 51(3):57, 2018.

[183] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[184] Cynthia Dwork. Differential privacy. *Encyclopedia of Cryptography and Security*, pages 338–340, 2011.

[185] Anand D Sarwate and Kamalika Chaudhuri. Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE signal processing magazine*, 30(5):86–94, 2013.

[186] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

[187] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

[188] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

[189] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321. ACM, 2015.

[190] Sergey I Nikolenko. Synthetic data for deep learning. *arXiv preprint arXiv:1909.11512*, 2019.

[191] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[192] Edgar Alonso Lopez-Rojas and Stefan Axelsson. A review of computer simulation for fraud detection research in financial datasets. In *2016 Future Technologies Conference (FTC)*, pages 932–935. IEEE, 2016.

[193] Valeria Nikolaenko, Stratis Ioannidis, Udi Weinsberg, Marc Joye, Nina Taft, and Dan Boneh. Privacy-preserving matrix factorization. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 801–812. ACM, 2013.

[194] Ziqi Liu, Yu-Xiang Wang, and Alexander Smola. Fast differentially private matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 171–178. ACM, 2015.

[195] Arnaud Berlioz, Arik Friedman, Mohamed Ali Kaafar, Roksana Boreli, and Shlomo Berkovsky. Applying differential privacy to matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 107–114. ACM, 2015.

[196] Arik Friedman, Shlomo Berkovsky, and Mohamed Ali Kaafar. A differential privacy framework for matrix factorization recommender systems. *User Modeling and User-Adapted Interaction*, 26(5):425–458, 2016.

[197] Feng Zhang, Victor E Lee, and Kim-Kwang Raymond Choo. Jo-dpmf: Differentially private matrix factorization learning through joint optimization. *Information Sciences*, 467:271–281, 2018.

[198] Yining Wang and Anima Anandkumar. Online and differentially-private tensor decomposition. In *Advances in Neural Information Processing Systems*, pages 3531–3539, 2016.

[199] Hafiz Imtiaz and Anand D Sarwate. Distributed differentially private algorithms for matrix and tensor factorization. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1449–1464, 2018.

[200] Jing Ma, Qiuchen Zhang, Jian Lou, Joyce C Ho, Li Xiong, and Xiaoqian Jiang. Privacy-preserving tensor factorization for collaborative health data analysis. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1291–1300. ACM, 2019.

[201] Yejin Kim, Jimeng Sun, Hwanjo Yu, and Xiaoqian Jiang. Federated tensor factorization for computational phenotyping. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 887–895. ACM, 2017.

[202] Jun Feng, Laurence T Yang, Qing Zhu, and Kim-Kwang Raymond Choo. Privacy-preserving tensor decomposition over encrypted data in a federated cloud environment. *IEEE Transactions on Dependable and Secure Computing*, 2018.

[203] Liwei Kuang, Laurence T Yang, Jun Feng, and Mianxiong Dong. Secure tensor decomposition using fully homomorphic encryption scheme. *IEEE Transactions on Cloud Computing*, 6(3):868–878, 2015.

[204] Justin Dauwels, K Srinivasan, M Ramasubba Reddy, and Andrzej Cichocki. Near-lossless multichannel eeg compression based on matrix and tensor decompositions. *IEEE journal of biomedical and health informatics*, 17(3):708–714, 2012.

[205] Cesar F Caiafa and Andrzej Cichocki. Stable, robust, and super fast reconstruction of tensors using multi-way projections. *IEEE Transactions on Signal Processing*, 63(3):780–793, 2014.

[206] Azam Karami, Mehran Yazdi, and Grégoire Mercier. Compression of hyperspectral images using discerete wavelet transform and tucker decomposition. *IEEE journal of selected topics in applied earth observations and remote sensing*, 5(2):444–450, 2012.

[207] Rafael Ballester-Ripoll, Peter Lindstrom, and Renato Pajarola. Tthresh: Tensor compression for multidimensional visual data. *IEEE transactions on visualization and computer graphics*, 2019.

[208] Evangelos E Papalexakis, Christos Faloutsos, and Nicholas D Sidiropoulos. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(2):1–44, 2016.

[209] Wolfgang Hackbusch and Stefan Kühn. A new scheme for the tensor representation. *Journal of Fourier analysis and applications*, 15(5):706–722, 2009.

[210] Lars Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054, 2010.

[211] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[212] Qibin Zhao, Guoxu Zhou, Shengli Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*, 2016.

[213] Wolfgang Hackbusch. *Tensor spaces and numerical tensor calculus*, volume 42. Springer, 2012.

[214] Göran Bergqvist and Erik G Larsson. The higher-order singular value decomposition: Theory and an application [lecture notes]. *IEEE Signal Processing Magazine*, 27(3):151–154, 2010.

[215] Sergey Dolgov and Boris Khoromskij. Two-level qtt-tucker format for optimized tensor calculus. *SIAM Journal on Matrix Analysis and Applications*, 34(2):593–623, 2013.

[216] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of physics*, 326(1):96–192, 2011.

[217] Ulrich Schollwöck. The density-matrix renormalization group. *Reviews of modern physics*, 77(1):259, 2005.

[218] Amir Vajdi, Mohammad Reza Zaghian, Saman Farahmand, Elham Rastegar, Kian Maroofi, Shaohua Jia, Marc Pomplun, Nurit Haspel, and Akram Bayat. Human gait database for normal walk collected by smart phone accelerometer. *arXiv preprint arXiv:1905.03109*, 2019.

[219] Peter N. Belhumeur, João P Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

[220] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth.

Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191. ACM, 2017.

[221] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[222] Ying He, Yan Chen, Yang Hu, and Bing Zeng. Wifi vision: Sensing, recognition, and detection with commodity mimo-ofdm wifi. *IEEE Internet of Things Journal*, 7(9):8296–8317, 2020.

[223] Yongsen Ma, Gang Zhou, and Shuangquan Wang. Wifi sensing with channel state information: A survey. *ACM Comput. Surv.*, 52(3), June 2019.

[224] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[225] Evangelos E. Papalexakis, Christos Faloutsos, and Nicholas D. Sidiropoulos. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Trans. Intell. Syst. Technol.*, 8(2), October 2016.

[226] Lieven De Lathauwer. Decompositions of a higher-order tensor in block termspart i: Lemmas for partitioned matrices. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1022–1032, 2008.

[227] Lieven De Lathauwer. Decompositions of a higher-order tensor in block termspart ii: Definitions and uniqueness. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1033–1066, 2008.

[228] Lieven De Lathauwer and Dimitri Nion. Decompositions of a higher-order tensor in block termspart iii: Alternating least squares algorithms. *SIAM journal on Matrix Analysis and Applications*, 30(3):1067–1083, 2008.

[229] Otto Debals and Lieven De Lathauwer. Stochastic and deterministic tensorization for blind signal separation. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 3–13. Springer, 2015.

[230] Mamady Kebe, Rida Gadhafi, Baker Mohammad, Mihai Sanduleanu, Hani Saleh, and Mahmoud Al-Qutayri. Human vital signs detection methods and potential using radars: A review. *Sensors*, 20(5):1454, 2020.

[231] Ameen Bin Obadi, Ping Jack Soh, Omar Aldayel, Muataz Hameed Al-Doori, Marco Mercuri, and Dominique Schreurs. A survey on vital signs detection using radar techniques and processing with fpga implementation. *IEEE Circuits and Systems Magazine*, 21(1):41–74, 2021.

[232] Emanuele Cardillo and Alina Caddemi. A review on biomedical mimo radars for vital sign detection and human localization. *Electronics*, 9(9), 2020.

[233] Hugo Saner, Samuel Elia Johannes Knobel, Narayan Schuetz, and Tobias Nef. Contact-free sensor signals as a new digital biomarker for cardiovascular disease: chances and challenges. *European Heart Journal - Digital Health*, 1(1):30–39, 11 2020.

[234] Changzhi Li, Victor M. Lubecke, Olga Boric-Lubecke, and Jenshan Lin. A review on recent advances in doppler radar sensors for noncontact healthcare monitoring. *IEEE Transactions on Microwave Theory and Techniques*, 61(5):2046–2060, 2013.

[235] Zhengjie Wang, Kangkang Jiang, Yushan Hou, Zehua Huang, Wenwen Dou, Chengming Zhang, and Yinjing Guo. A survey on csi-based human behavior recognition in through-the-wall scenario. *IEEE Access*, 7:78772–78793, 2019.

[236] Zhengjie Wang, Zehua Huang, Chengming Zhang, Wenwen Dou, Yinjing Guo, and Da Chen. Csi-based human sensing using model-based approaches: a survey. *Journal of Computational Design and Engineering*, 02 2021.

[237] Zhengjie Wang, Kangkang Jiang, Yushan Hou, Wenwen Dou, Chengming Zhang, Zehua Huang, and Yinjing Guo. A survey on human behavior recognition using channel state information. *IEEE Access*, 7:155986–156024, 2019.

[238] Fadel Adib, Hongzi Mao, Zachary Kabelac, Dina Katabi, and Robert C. Miller. Smart homes that monitor breathing and heart rate. In *Proceedings of the 33rd*

Annual ACM Conference on Human Factors in Computing Systems, CHI '15, page 837846, New York, NY, USA, 2015. Association for Computing Machinery.

[239] Marco Mercuri, Yao-Hong Liu, Sunil Sheelavant, Salvatore Polito, Tom Torfs, and Chris Van Hoof. Digital linear discrete fmcw radar for healthcare applications. In *2019 IEEE MTT-S International Microwave Symposium (IMS)*, pages 144–147, 2019.

[240] Shu Xu and Wei Kang. Multi-target vital sign monitoring in the same beam using a fmcw radar based on tags. In *Proceedings of the 2020 4th International Conference on Electronic Information Technology and Computer Engineering*, EITCE 2020, page 243247, New York, NY, USA, 2020. Association for Computing Machinery.

[241] Yong Wang, Wen Wang, Mu Zhou, Aihu Ren, and Zengshan Tian. Remote monitoring of human vital signs based on 77-ghz mm-wave fmcw radar. *Sensors*, 20(10), 2020.

[242] fa Alizadeh, George Shaker, Joo Carlos Martins De Almeida, Plinio Pelegrini Morita, and Safeddin Safavi-Naeini. Remote monitoring of human vital signs using mm-wave fmcw radar. *IEEE Access*, 7:54958–54968, 2019.

[243] Hyunjae Lee, Byung-Hyun Kim, Jin-Kwan Park, Sung Woo Kim, and Jong-Gwan Yook. A resolution enhancement technique for remote monitoring of the vital signs of multiple subjects using a 24 ghz bandwidth-limited fmcw radar. *IEEE Access*, 8:1240–1248, 2020.

[244] Sungwon Yoo, Shahzad Ahmed, Sun Kang, Duhyun Hwang, Jungjun Lee, Jungduck Son, and Sung Ho Cho. Radar recorded child vital sign public dataset and deep learning-based age group classification framework for vehicular application. *Sensors*, 21(7), 2021.

[245] Emmi Turppa, Juha M. Kortelainen, Oleg Antropov, and Tero Kiuru. Vital sign monitoring using fmcw radar in various sleeping scenarios. *Sensors*, 20(22), 2020.

[246] Guochao Wang, Jos-Mara Muoz-Ferreras, Changzhan Gu, Changzhi Li, and Roberto Gmez-Garca. Application of linear-frequency-modulated continuous-wave (lfmcw) radars for tracking of vital signs. *IEEE Transactions on Microwave Theory and Techniques*, 62(6):1387–1399, 2014.

[247] M.Y.W. Chia, S.W. Leong, C.K. Sim, and K.M. Chan. Through-wall uwb radar operating within fcc's mask for sensing heart beat and breathing rate. In *European Radar Conference, 2005. EURAD 2005.*, pages 267–270, 2005.

[248] Mario Leib, Wolfgang Menzel, Bernd Schleicher, and Hermann Schumacher. Vital signs monitoring with a uwb radar based on a correlation receiver. In *Proceedings of the Fourth European Conference on Antennas and Propagation*, pages 1–5, 2010.

[249] Yogesh Nijsure, Wee Peng Tay, Erry Gunawan, Fuxi Wen, Zhang Yang, Yong Liang Guan, and Ai Ping Chua. An impulse radio ultrawideband system for contactless noninvasive respiratory monitoring. *IEEE Transactions on Biomedical Engineering*, 60(6):1509–1517, 2013.

[250] Bernd Schleicher, Ismail Nasr, Andreas Trasser, and Hermann Schumacher. Ir-uwb radar demonstrator for ultra-fine movement detection and vital-sign monitoring. *IEEE Transactions on Microwave Theory and Techniques*, 61(5):2076–2085, 2013.

[251] Qiuchi Jian, Jian Yang, Yinan Yu, Peter Bjrkholm, and Tomas McKelvey. Detection of breathing and heartbeat by using a simple uwb radar system. In *The 8th European Conference on Antennas and Propagation (EuCAP 2014)*, pages 3078–3081, 2014.

[252] Dag T. Wisland, Kristian Granhaug, Jan Roar Pleym, Nikolaj Andersen, Stig Sta, and Hkon A. Hjortland. Remote monitoring of vital signs using a cmos

uwb radar transceiver. In *2016 14th IEEE International New Circuits and Systems Conference (NEWCAS)*, pages 1–4, 2016.

[253] Seong Kyu Leem, Faheem Khan, and Sung Ho Cho. Vital sign monitoring and mobile phone usage detection using ir-uwb radar for intended use in car crash prevention. *Sensors*, 17(6), 2017.

[254] Faheem Khan and Sung Ho Cho. A detailed algorithm for vital sign monitoring of a stationary/non-stationary human through ir-uwb radar. *Sensors*, 17(2):290, 2017.

[255] Zhicheng Yang, Maurizio Bocca, Vivek Jain, and Prasant Mohapatra. Contactless breathing rate monitoring in vehicle using uwb radar. In *Proceedings of the 7th International Workshop on Real-World Embedded Wireless Systems and Networks*, RealWSN'18, page 1318, New York, NY, USA, 2018. Association for Computing Machinery.

[256] Sun Kang, Yonggu Lee, Young-Hyo Lim, Hyun-Kyung Park, Sung Ho Cho, and Seok Hyun Cho. Validation of noncontact cardiorespiratory monitoring using impulse-radio ultra-wideband radar against nocturnal polysomnography. *Sleep and Breathing*, pages 1–8, 2019.

[257] Sun Kang, Dong-Kyu Kim, Yonggu Lee, Young-Hyo Lim, Hyun-Kyung Park, Sung Ho Cho, and Seok Hyun Cho. Non-contact diagnosis of obstructive sleep apnea using impulse-radio ultra-wideband radar. *Scientific reports*, 10(1):1–7, 2020.

[258] Hyun Bin Kwon, Sang Ho Choi, Dongseok Lee, Dongyeon Son, Heenam Yoon, Mi Hyun Lee, Yu Jin Lee, and Kwang Suk Park. Attention-based lstm for non-contact sleep stage classification using ir-uwb radar. *IEEE Journal of Biomedical and Health Informatics*, pages 1–1, 2021.

[259] Tianyue Zheng, Zhe Chen, Chao Cai, Jun Luo, and Xu Zhang. V2ifi: In-vehicle vital sign monitoring via compact rf sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(2), June 2020.

[260] Changzhi Li, Jenshan Lin, and Yanming Xiao. Robust overnight monitoring of human vital signs by a non-contact respiration and heartbeat detector. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2235–2238, 2006.

[261] Amy D. Droitcour, Olga Boric-Lubecke, and Gregory T. A. Kovacs. Signal-to-noise ratio in doppler radar system for heart and respiratory rate measurements. *IEEE Transactions on Microwave Theory and Techniques*, 57(10):2498–2507, 2009.

[262] Changzhi Li, Jun Ling, Jian Li, and Jenshan Lin. Accurate doppler radar noncontact vital sign detection using the relax algorithm. *IEEE Transactions on Instrumentation and Measurement*, 59(3):687–695, 2010.

[263] Mari Zakrzewski, Antti Vehkaoja, Atte S. Joutsen, Karri T. Palovuori, and Jukka J. Vanhala. Noncontact respiration monitoring during sleep with microwave doppler radar. *IEEE Sensors Journal*, 15(10):5683–5693, 2015.

[264] Shekh M. M. Islam and Victor M. Lubecke. Extracting individual respiratory signatures from combined multi-subject mixtures with varied breathing pattern using independent component analysis with the jade algorithm. In *2020 IEEE Asia-Pacific Microwave Conference (APMC)*, pages 734–736, 2020.

[265] Shekh M. M. Islam, O. Boric-Lubecke, and V. M. Lubecke. Comparative analysis of phase-comparison monopulse and music algorithm methods for direction of arrival (doa) of multiple-subject respiration measured with doppler radar. In *2020 IEEE Asia-Pacific Microwave Conference (APMC)*, pages 968–970, 2020.

[266] Mehrdad Nosrati, Shahram Shahsavari, and Negar Tavassolian. Multi-target vital-signs monitoring using a dual-beam hybrid doppler radar. In *2018 IEEE International Microwave Biomedical Conference (IMBioC)*, pages 58–60, 2018.

[267] Mehrdad Nosrati and Negar Tavassolian. Accurate doppler radar-based cardiopulmonary sensing using chest-wall acceleration. *IEEE Journal of Electromagnetics, RF and Microwaves in Medicine and Biology*, 3(1):41–47, 2019.

[268] Mehrdad Nosrati, Shahram Shahsavari, Sanghoon Lee, Hua Wang, and Negar Tavassolian. A concurrent dual-beam phased-array doppler radar using mimo beamforming techniques for short-range vital-signs monitoring. *IEEE Transactions on Antennas and Propagation*, 67(4):2390–2404, 2019.

[269] Zi-Kai Yang, Heping Shi, Sheng Zhao, and Xiang-Dong Huang. Vital sign detection during large-scale and fast body movements based on an adaptive noise cancellation algorithm using a single doppler radar sensor. *Sensors*, 20(15), 2020.

[270] Xiangmao Chang, Jiahua Dai, Zhiyong Zhang, Kun Zhu, and Guoliang Xing. Rf-rvm: Continuous respiratory volume monitoring with cots rfid tags. *IEEE Internet of Things Journal*, pages 1–1, 2021.

[271] Chuyu Wang, Lei Xie, Wei Wang, Yingying Chen, Yanling Bu, and Sanglu Lu. Rf-ecg: Heart rate variability assessment based on cots rfid tag array. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(2), July 2018.

[272] Zawar Hussain, Subhash Sagar, Wei Emma Zhang, and Quan Z. Sheng. A cost-effective and non-invasive system for sleep and vital signs monitoring using passive rfid tags. In *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MobiQuitous '19, page 153161, New York, NY, USA, 2019. Association for Computing Machinery.

[273] Chao Yang, Xuyu Wang, and Shiwen Mao. Unsupervised detection of apnea using commodity rfid tags with a recurrent variational autoencoder. *IEEE Access*, 7:67526–67538, 2019.

[274] Xiaohui Tao, Thanveer Basha Shaik, Niall Higgins, Raj Gururajan, and Xujuan Zhou. Remote patient monitoring using radio frequency identification

(rfid) technology and machine learning for early detection of suicidal behaviour in mental health facilities. *Sensors*, 21(3):776, 2021.

[275] Kagome Naya, Xiaouan Hu, Toshiaki Miyazaki, Peng Li, and Kun Wang. Non-invasive and quick respiratory-rate monitoring at bedtime using passive rfids. In *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 244–249, 2019.

[276] Chao Yang, Xuyu Wang, and Shiwen Mao. Autotag: Recurrent variational autoencoder for unsupervised apnea detection with rfid tags. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7, 2018.

[277] Yanni Yang and Jiannong Cao. Robust rfid-based respiration monitoring in dynamic environments. In *2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9, 2020.

[278] Chao Yang, Xuyu Wang, and Shiwen Mao. Respiration monitoring with rfid in driving environments. *IEEE Journal on Selected Areas in Communications*, 39(2):500–512, 2021.

[279] Kun Qian, Chenshu Wu, Fu Xiao, Yue Zheng, Yi Zhang, Zheng Yang, and Yunhao Liu. Acousticcardiogram: Monitoring heartbeats using acoustic signals on smart devices. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 1574–1582, 2018.

[280] Anran Wang, Jacob E. Sunshine, and Shyamnath Gollakota. Contactless infant monitoring using white noise. In *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, New York, NY, USA, 2019. Association for Computing Machinery.

[281] Xuyu Wang, Runze Huang, Chao Yang, and Shiwen Mao. Smartphone sonar-based contact-free respiration rate monitoring. *ACM Trans. Comput. Healthcare*, 2(2), February 2021.

[282] X. Wang, R. Huang, and S. Mao. Sonarbeat: Sonar phase for breathing beat monitoring with smartphones. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–8, 2017.

[283] Xingzhe Song, Boyuan Yang, Ge Yang, Ruirong Chen, Erick Forno, Wei Chen, and Wei Gao. Spirosonic: Monitoring human lung function via acoustic sensing on commodity smartphones. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, MobiCom '20, New York, NY, USA, 2020. Association for Computing Machinery.

[284] Jagmohan Chauhan, Yining Hu, Suranga Seneviratne, Archan Misra, Aruna Seneviratne, and Youngki Lee. Breathprint: Breathing acoustics-based user authentication. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '17, page 278291, New York, NY, USA, 2017. Association for Computing Machinery.

[285] Xiangyu Xu, Jiadi Yu, Yingying Chen, Yanmin Zhu, Linghe Kong, and Minglu Li. Breathlistener: Fine-grained breathing monitoring in driving environments utilizing acoustic signals. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '19, page 5466, New York, NY, USA, 2019. Association for Computing Machinery.

[286] Tian Hao, Guoliang Xing, and Gang Zhou. Runbuddy: A smartphone system for running rhythm monitoring. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, page 133144, New York, NY, USA, 2015. Association for Computing Machinery.

[287] Rajalakshmi Nandakumar, Shyamnath Gollakota, and Nathaniel Watson. Contactless sleep apnea detection on smartphones. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '15, page 4557, New York, NY, USA, 2015. Association for Computing Machinery.

[288] Yanzhi Ren, Chen Wang, Jie Yang, and Yingying Chen. Fine-grained sleep monitoring: Hearing your breathing with smartphones. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1194–1202, 2015.

[289] Hyunjae Lee, Byung-Hyun Kim, Jin-Kwan Park, and Jong-Gwan Yook. A novel vital-sign sensing algorithm for multiple subjects based on 24-ghz fmcw doppler radar. *Remote Sensing*, 11(10), 2019.

[290] Lingyun Ren, Haofei Wang, Krishna Naishadham, Ozlem Kilic, and Aly E. Fathy. Phase-based methods for heart rate detection using uwb impulse doppler radar. *IEEE Transactions on Microwave Theory and Techniques*, 64(10):3319–3331, 2016.

[291] X. Wang, X. Wang, and S. Mao. Rf sensing in the internet of things: A general deep learning framework. *IEEE Communications Magazine*, 56(9):62–67, 2018.

[292] Chen Liu, Jie Xiong, Lin Cai, Lin Feng, Xiaojiang Chen, and Dingyi Fang. Beyond respiration: Contactless sleep sound-activity recognition using rf signals. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(3), September 2019.

[293] Yanni Yang, Jiannong Cao, and Xiulong Liu. Er-rhythm: Coupling exercise and respiration rhythm using lightweight cots rfid. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(4), December 2019.

[294] Aboajeila Milad Ashleibta, Qammer H. Abbasi, Syed Aziz Shah, Muhammad Arslan Khalid, Najah Abed AbuAli, and Muhammad Ali Imran. Non-invasive rf sensing for detecting breathing abnormalities using software defined radios. *IEEE Sensors Journal*, 21(4):5111–5118, 2021.

[295] Q. Wang, J. Zhao, S. Xu, and R. Wang. Retype: Your breath tells your mind! *IEEE Internet of Things Journal*, pages 1–1, 2021.

[296] Gaper Slapniar, Wenjin Wang, and Mitja Lutrek. Classification of hemodynamics scenarios from a public radar dataset using a deep learning approach. *Sensors*, 21(5), 2021.

[297] Mingmin Zhao, Fadel Adib, and Dina Katabi. Emotion recognition using wireless signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, MobiCom '16, page 95108, New York, NY, USA, 2016. Association for Computing Machinery.

[298] Ahsan Noor Khan, Achintha Avin Ihalage, Yihan Ma, Baiyang Liu, Yujie Liu, and Yang Hao. Deep learning framework for subject-independent emotion detection using wireless signals. *PLOS ONE*, 16(2):1–16, 02 2021.

[299] Sumit Kumar Rai, Chetna Sharma, Vikash Shaw, Ranjan Kumar Jha, and Sanjeev Kumar. A non-contact approach for detection of sleep apnea using doppler phenomena. In *Proceedings of 6th International Conference on Recent Trends in Computing: ICRTC 2020*, page 99. Springer Nature.

[300] Phuc Nguyen, Xinyu Zhang, Ann Halbower, and Tam Vu. Continuous and fine-grained breathing volume monitoring from afar using wireless signals. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, 2016.

[301] Wenda Li, Bo Tan, and Robert Piechocki. Passive radar for opportunistic monitoring in e-health applications. *IEEE Journal of Translational Engineering in Health and Medicine*, 6:1–10, 2018.

[302] Zhicheng Yang, Maurizio Bocca, Vivek Jain, and Prasant Mohapatra. Contactless breathing rate monitoring in vehicle using uwb radar. In *Proceedings of the 7th International Workshop on Real-World Embedded Wireless Systems and Networks*, RealWSN'18, page 1318, New York, NY, USA, 2018. Association for Computing Machinery.

[303] Chuanwei Ding, Rachel Chae, Jing Wang, Li Zhang, Hong Hong, Xiaohua Zhu, and Changzhi Li. Inattentive driving behavior detection based on

portable fmcw radar. *IEEE Transactions on Microwave Theory and Techniques*, 67(10):4031–4041, 2019.

[304] Feng Lin, Chen Song, Yan Zhuang, Wenyao Xu, Changzhi Li, and Kui Ren. Cardiac scan: A non-contact and continuous heart-based user authentication system. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, MobiCom '17, page 315328, New York, NY, USA, 2017. Association for Computing Machinery.

[305] Yichao Yuan, Chunchi Lu, Austin Ying-Kuang Chen, Chao-Hsiung Tseng, and Chung-Tse Michael Wu. Multi-target concurrent vital sign and location detection using metamaterial-integrated self-injection-locked quadrature radar sensor. *IEEE Transactions on Microwave Theory and Techniques*, 67(12):5429–5437, 2019.

[306] Yichao Yuan, Chunchi Lu, Austin Ying-Kuang Chen, Chao-Hsiung Tseng, and Chung-Tse Michael Wu. Noncontact multi-target vital sign detection using self-injection-locked radar sensor based on metamaterial leaky wave antenna. In *2019 IEEE MTT-S International Microwave Symposium (IMS)*, pages 148–151, 2019.

[307] Masashi Muragaki, Shigeaki Okumura, Katsutoshi Maehara, Takuya Sakamoto, Mototaka Yoshioka, Kenichi Inoue, Takeshi Fukuda, Hiroyuki Sakai, and Toru Sato. Noncontact respiration monitoring of multiple closely positioned patients using ultra-wideband array radar with adaptive beamforming technique. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1118–1122, 2017.

[308] Wei-Chih Su, Pin-Hsun Juan, De-Ming Chian, Tzyy-Sheng Jason Horng, Chao-Kai Wen, and Fu-Kang Wang. 2-d self-injection-locked doppler radar for locating multiple people and monitoring their vital signs. *IEEE Transactions on Microwave Theory and Techniques*, 69(1):1016–1026, 2021.

[309] Jin-Kwan Park, Yunseog Hong, Hyunjae Lee, Chorom Jang, Gi-Ho Yun, Hee-Jo Lee, and Jong-Gwan Yook. Noncontact rf vital sign sensor for continuous

monitoring of driver status. *IEEE Transactions on Biomedical Circuits and Systems*, 13(3):493–502, 2019.

[310] Junjun Xiong, Hongqiang Zhang, Hong Hong, Heng Zhao, Xiaohua Zhu, and Changzhi Li. Multi-target vital signs detection using simo continuous-wave radar with dbf technique. In *2020 IEEE Radio and Wireless Symposium (RWS)*, pages 194–196, 2020.

[311] Zhicheng Sang and Wei Kang. A fsk radar with frequency-scanned array for moving and stationary human subjects detection. In *Proceedings of the 2020 4th International Conference on Electronic Information Technology and Computer Engineering*, EITCE 2020, page 248252, New York, NY, USA, 2020. Association for Computing Machinery.

[312] Guan-Wei Fang, Ching-Yao Huang, and Chin-Lung Yang. Switch-based low intermediate frequency system of a vital sign radar for simultaneous multi-target and multidirectional detection. *IEEE Journal of Electromagnetics, RF and Microwaves in Medicine and Biology*, 4(4):265–272, 2020.

[313] J. Wang, X. Bai, Q. Gao, X. Li, X. Bi, and M. Pan. Multi-target device-free wireless sensing based on multiplexing mechanisms. *IEEE Transactions on Vehicular Technology*, 69(9):10242–10251, 2020.

[314] Guan-Wei Fang, Ching-Yao Huang, and Chin-Lung Yang. Simultaneous detection of multi-target vital signs using eemd algorithm based on fmcw radar. In *2019 IEEE MTT-S International Microwave Biomedical Conference (IMBioC)*, volume 1, pages 1–4, 2019.

[315] Can Uysal and Tansu Filik. Rf-based noncontact respiratory rate monitoring with parametric spectral estimation. *IEEE Sensors Journal*, 19(21):9841–9849, 2019.

[316] Robert Nakata, Scott Clemens, Alex Lee, and Victor Lubecke. Rf techniques for motion compensation of an unmanned aerial vehicle for remote radar life

sensing. In *2016 IEEE MTT-S International Microwave Symposium (IMS)*, pages 1–4, 2016.

[317] Robert H. Nakata, Brian Haruna, Takashi Yamaguchi, Victor M. Lubecke, Shigeru Takayama, and Kiyotsugu Takaba. Motion compensation for an unmanned aerial vehicle remote radar life sensor. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 8(2):329–337, 2018.

[318] Ashikur Rahman, Ehsan Yavari, Aditya Singh, Victor M. Lubecke, and Olga-Boric Lubecke. A low-if tag-based motion compensation technique for mobile doppler radar life signs monitoring. *IEEE Transactions on Microwave Theory and Techniques*, 63(10):3034–3041, 2015.

[319] Aditya Singh and Victor Lubecke. Adaptive noise cancellation for two frequency radars using frequency doubling passive rf tags. In *2012 IEEE/MTT-S International Microwave Symposium Digest*, pages 1–3, 2012.

[320] Isar Mostafanezhad, Olga Boric-Lubecke, Victor Lubecke, and Anders Host-Madsen. Cancellation of unwanted motion in a handheld doppler radar used for non-contact life sign monitoring. In *2008 IEEE MTT-S International Microwave Symposium Digest*, pages 1171–1174, 2008.

[321] Eugene Ferguson Greneker III and Daren Joseph Zywicki. Stabilizing motion in a radar detection system using ultrasonic radar range information, 2005.

[322] Isar Mostafanezhad, Ehsan Yavari, Olga Boric-Lubecke, Victor M. Lubecke, and Danilo P. Mandic. Cancellation of unwanted doppler radar sensor motion using empirical mode decomposition. *IEEE Sensors Journal*, 13(5):1897–1904, 2013.

[323] Isar Mostafanezhad, Olga Boric-Lubecke, Victor Lubecke, and Danilo P. Mandic. Application of empirical mode decomposition in removing fidgeting interference in doppler radar life signs monitoring devices. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 340–343, 2009.

[324] Emanuele Cardillo, Changzhi Li, and Alina Caddemi. Vital sign detection and radar self-motion cancellation through clutter identification. *IEEE Transactions on Microwave Theory and Techniques*, 69(3):1932–1942, 2021.

[325] Byung-Kwon Park, Olga Boric-Lubecke, and Victor M. Lubecke. Arctangent demodulation with dc offset compensation in quadrature doppler radar receiver systems. *IEEE Transactions on Microwave Theory and Techniques*, 55(5):1073–1079, 2007.

[326] Zhaohua Wu and Norden E Huang. Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Advances in adaptive data analysis*, 1(01):1–41, 2009.

[327] Guohua Lu, Fang Yang, Xijing Jing, and Jianqi Wang. Contact-free measurement of heartbeat signal via a doppler radar using adaptive filtering. In *2010 International Conference on Image Analysis and Signal Processing*, pages 89–92, 2010.

[328] Takuya Sakamoto, Ryohei Imasaka, Hirofumi Taki, Toru Sato, Mototaka Yoshioka, Kenichi Inoue, Takeshi Fukuda, and Hiroyuki Sakai. Feature-based correlation and topological similarity for interbeat interval estimation using ultrawideband radar. *IEEE Transactions on Biomedical Engineering*, 63(4):747–757, 2016.

[329] Kohei Yamamoto and Tomoaki Ohtsuki. Non-contact heartbeat detection by heartbeat signal reconstruction based on spectrogram analysis with convolutional lstm. *IEEE Access*, 8:123603–123613, 2020.

[330] Kohei Yamamoto, Ryosuke Hiromatsu, and Tomoaki Ohtsuki. Ecg signal reconstruction via doppler sensor by hybrid deep learning model with cnn and lstm. *IEEE Access*, 8:130551–130560, 2020.

[331] Neboja Maleevi, Vladimir Petrovi, Minja Beli, Christian Antfolk, Veljko Mihajlovi, and Milica Jankovi. Contactless real-time heartbeat detection via 24

ghz continuous-wave doppler radar using artificial neural networks. *Sensors*, 20(8), 2020.

[332] Pengfei Wang, Fugui Qi, Miao Liu, Fulai Liang, Huijun Xue, Yang Zhang, Hao Lv, and Jianqi Wang. Noncontact heart rate measurement based on an improved convolutional sparse coding method using ir-uwb radar. *IEEE Access*, 7:158492–158502, 2019.

[333] Can Uysal, Altan Onat, and Tansu Filik. Non-contact respiratory rate estimation in real-time with modified joint unscented kalman filter. *IEEE Access*, 8:99445–99457, 2020.

[334] Yipeng Ding, Xiali Yu, Chengxi Lei, Yinhua Sun, Xuemei Xu, and Juan Zhang. A novel real-time human heart rate estimation method for noncontact vital sign radar detection. *IEEE Access*, 8:88689–88699, 2020.

[335] Vladimir L. Petrovi, Milica M. Jankovi, Anita V. Lupi, Veljko R. Mihajlovi, and Jelena S. Popovi-Boovi. High-accuracy real-time monitoring of heart rate variability using 24 ghz continuous-wave doppler radar. *IEEE Access*, 7:74721–74733, 2019.

[336] Chen Ye, Kentaroh Toyoda, and Tomoaki Ohtsuki. A stochastic gradient approach for robust heartbeat detection with doppler radar using time-window-variation technique. *IEEE Transactions on Biomedical Engineering*, 66(6):1730–1741, 2019.

[337] Chen Ye, Kentaroh Toyoda, and Tomoaki Ohtsuki. Blind source separation on non-contact heartbeat detection by non-negative matrix factorization algorithms. *IEEE Transactions on Biomedical Engineering*, 67(2):482–494, 2020.

[338] Lili Chen, Jie Xiong, Xiaojiang Chen, Sunghoon Ivan Lee, Daqing Zhang, Tao Yan, and Dingyi Fang. Lungtrack: Towards contactless and zero dead-zone respiration monitoring with commodity rfids. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(3), September 2019.

[339] Haoran Wan, Shuyu Shi, Wenyu Cao, Wei Wang, and Guihai Chen. Resptracker: Multi-user room-scale respiration tracking with commercial acoustic devices. 2021.

[340] Shichao Yue, Hao He, Hao Wang, Hariharan Rahul, and Dina Katabi. Extracting multi-person respiration from entangled rf signals. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(2), July 2018.

[341] Chuanwei Ding, Jiaming Yan, Li Zhang, Heng Zhao, Hong Hong, and Xiaohua Zhu. Noncontact multiple targets vital sign detection based on vmd algorithm. In *2017 IEEE Radar Conference (RadarConf)*, pages 0727–0730, 2017.

[342] Y. Yang, J. Cao, X. Liu, and X. Liu. Multi-breath: Separate respiration monitoring for multiple persons with uwb radar. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 840–849, 2019.

[343] D. M. Chian, C. K. Wen, F. K. Wang, and K. K. Wong. Signal separation and tracking algorithm for multi-person vital signs by using doppler radar. *IEEE Transactions on Biomedical Circuits and Systems*, 14(6):1346–1361, 2020.

[344] Heba Abdelnasser, Khaled A. Harras, and Moustafa Youssef. Ubibreathe: A ubiquitous non-invasive wifi-based breathing estimator. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '15, page 277286, New York, NY, USA, 2015. Association for Computing Machinery.

[345] Ossi Kaltiokallio, Hseyin Yiitler, Riku Jntti, and Neal Patwari. Non-invasive respiration rate monitoring using a single cots tx-rx pair. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, pages 59–69, 2014.

[346] Neal Patwari, Lara Brewer, Quinn Tate, Ossi Kaltiokallio, and Maurizio Bocca. Breathfinding: A wireless network that monitors and locates breathing

in a home. *IEEE Journal of Selected Topics in Signal Processing*, 8(1):30–42, 2014.

[347] Neal Patwari, Joey Wilson, Sai Ananthanarayanan, Sneha K. Kasera, and Dwayne R. Westenskow. Monitoring breathing via signal strength in wireless networks. *IEEE Transactions on Mobile Computing*, 13(8):1774–1786, 2014.

[348] X. Liu, J. Cao, S. Tang, and J. Wen. Wi-sleep: Contactless sleep monitoring via wifi signals. In *2014 IEEE Real-Time Systems Symposium*, pages 346–355, 2014.

[349] X. Liu, J. Cao, S. Tang, J. Wen, and P. Guo. Contactless respiration monitoring via off-the-shelf wifi devices. *IEEE Transactions on Mobile Computing*, 15(10):2466–2479, 2016.

[350] X. Wang, C. Yang, and S. Mao. Phasebeat: Exploiting csi phase data for vital sign monitoring with commodity wifi devices. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 1230–1239, 2017.

[351] Xuyu Wang, Chao Yang, and Shiwen Mao. On csi-based vital sign monitoring using commodity wifi. *ACM Trans. Comput. Healthcare*, 1(3), May 2020.

[352] Hao Wang, Daqing Zhang, Junyi Ma, Yasha Wang, Yuxiang Wang, Dan Wu, Tao Gu, and Bing Xie. Human respiration detection with commodity wifi devices: Do user location and body orientation matter? In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, page 2536, New York, NY, USA, 2016. Association for Computing Machinery.

[353] Youwei Zeng, Dan Wu, Ruiyang Gao, Tao Gu, and Daqing Zhang. Fullbreathe: Full human respiration detection exploiting complementarity of csi phase and amplitude of wifi signals. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(3), September 2018.

[354] Dan Wu, Daqing Zhang, Chenren Xu, Hao Wang, and Xiang Li. Device-free wifi human sensing: From pattern-based to model-based approaches. *IEEE Communications Magazine*, 55(10):91–97, 2017.

[355] X. Wang, C. Yang, and S. Mao. Resbeat: Resilient breathing beats monitoring with realtime bimodal csi data. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6, 2017.

[356] X. Wang, C. Yang, and S. Mao. Resilient respiration rate monitoring with realtime bimodal csi data. *IEEE Sensors Journal*, 20(17):10187–10198, 2020.

[357] Fusang Zhang, Daqing Zhang, Jie Xiong, Hao Wang, Kai Niu, Beihong Jin, and Yuxiang Wang. From fresnel diffraction model to fine-grained human respiration sensing with commodity wi-fi devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1), March 2018.

[358] Kai Niu, Fusang Zhang, Zhaoxin Chang, and Daqing Zhang. A fresnel diffraction model based human respiration detection system using cots wi-fi devices. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, UbiComp '18, page 416419, New York, NY, USA, 2018. Association for Computing Machinery.

[359] J. Liu, Y. Chen, Y. Wang, X. Chen, J. Cheng, and J. Yang. Monitoring vital signs and postures during sleep using wifi signals. *IEEE Internet of Things Journal*, 5(3):2071–2084, 2018.

[360] Abdelwahed Khamis, Chun Tung Chou, Branislav Kusy, and Wen Hu. Cardiofi: Enabling heart rate monitoring on unmodified cots wifi devices. MobiQuitous '18, page 97106, New York, NY, USA, 2018. Association for Computing Machinery.

[361] Jin Zhang, Weitao Xu, Wen Hu, and Salil S. Kanhere. Wicare: Towards in-situ breath monitoring. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*,

MobiQuitous 2017, page 126135, New York, NY, USA, 2017. Association for Computing Machinery.

[362] Usman Mahmood Khan, Zain Kabir, Syed Ali Hassan, and Syed Hassan Ahmed. A deep learning framework using passive wifi sensing for respiration monitoring. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6, 2017.

[363] D. Zhang, Y. Hu, Y. Chen, and B. Zeng. Breathtrack: Tracking indoor human breath status via commodity wifi. *IEEE Internet of Things Journal*, 6(2):3899–3911, 2019.

[364] D. Zhang, Y. Hu, and Y. Chen. Breath status tracking using commodity wifi. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2018.

[365] Y. Zeng, D. Wu, J. Xiong, and D. Zhang. Boosting wifi sensing performance via csi ratio. *IEEE Pervasive Computing*, 20(1):62–70, 2021.

[366] Dan Wu, Ruiyang Gao, Youwei Zeng, Jinyi Liu, Leye Wang, Tao Gu, and Daqing Zhang. Fingerdraw: Sub-wavelength level finger motion tracking with wifi signals. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(1), March 2020.

[367] Youwei Zeng, Dan Wu, Jie Xiong, Enze Yi, Ruiyang Gao, and Daqing Zhang. Farsense: Pushing the range limit of wifi-based respiration sensing with csi ratio of two antennas. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(3), September 2019.

[368] W. Chen, K. Niu, D. Zhao, R. Zheng, D. Wu, W. Wang, L. Wang, and D. Zhang. Robust dynamic hand gesture interaction using lte terminals. In *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 109–120, 2020.

[369] Fusang Zhang, Zhaoxin Chang, Kai Niu, Jie Xiong, Beihong Jin, Qin Lv, and Daqing Zhang. Exploring lora for long-range through-wall sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(2), June 2020.

[370] Y. T. Xu, X. Chen, X. Liu, D. Meger, and G. Dudek. Pressense: Passive respiration sensing via ambient wifi signals in noisy environments. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4032–4039, 2020.

[371] C. Wu, Z. Yang, Z. Zhou, X. Liu, Y. Liu, and J. Cao. Non-invasive detection of moving and stationary human with wifi. *IEEE Journal on Selected Areas in Communications*, 33(11):2329–2342, 2015.

[372] J. Liu, Y. Chen, Y. Dong, Y. Wang, T. Zhao, and Y. D. Yao. Continuous user verification via respiratory biometrics. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 1–10, 2020.

[373] Syed W. Shah and Salil S. Kanhere. Smart user identification using cardiopulmonary activity. *Pervasive and Mobile Computing*, 58:101024, 2019.

[374] M. Hussain, A. Akbilek, F. Pfeiffer, and B. Napholz. In-vehicle breathing rate monitoring based on wifi signals. In *2020 50th European Microwave Conference (EuMC)*, pages 292–295, 2021.

[375] Weijia Jia, Hongjian Peng, Na Ruan, Zhiqing Tang, and Wei Zhao. Wifind: Driver fatigue detection with fine-grained wi-fi signal features. *IEEE Transactions on Big Data*, 6(2):269–282, 2020.

[376] Jinyi Liu, Youwei Zeng, Tao Gu, Leye Wang, and Daqing Zhang. Wiphone: Smartphone-based respiration monitoring using ambient reflected wifi signals. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5(1), March 2021.

[377] Biyi Fang, Nicholas D. Lane, Mi Zhang, Aidan Boran, and Fahim Kawsar. Bodyscan: Enabling radio-based sensing on wearable devices for contactless activity and vital sign monitoring. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys

'16, page 97110, New York, NY, USA, 2016. Association for Computing Machinery.

[378] Usman Mahmood Khan, Zain Kabir, and Syed Ali Hassan. Wireless health monitoring using passive wifi sensing. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1771–1776, 2017.

[379] Muhammad Imran Khan, Mian Ahmad Jan, Yar Muhammad, Dinh-Thuan Do, Ateeq ur Rehman, Constandinos X Mavromoustakis, and Evangelos Pallis. Tracking vital signs of a patient using channel state information and machine learning for a smart healthcare system. *Neural Computing and Applications*, pages 1–15, 2021.

[380] Xiaopeng Niu, Shengjie Li, Yue Zhang, Zhaopeng Liu, Dan Wu, Rahul C. Shah, Cagri Tanriover, Hong Lu, and Daqing Zhang. Wimonitor: Continuous long-term human vitality monitoring using commodity wi-fi devices. *Sensors*, 21(3), 2021.

[381] Z. He, L. Guo, Z. Lu, X. Wen, W. Zheng, and S. Zhou. Contact-free in-home health monitoring system with commodity wi-fi. In *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6, 2019.

[382] F. Zhang, C. Wu, B. Wang, M. Wu, D. Bugos, H. Zhang, and K. J. R. Liu. Smars: Sleep monitoring via ambient radio signals. *IEEE Transactions on Mobile Computing*, 20(1):217–231, 2021.

[383] Jian Liu, Yan Wang, Yingying Chen, Jie Yang, Xu Chen, and Jerry Cheng. Tracking vital signs during sleep leveraging off-the-shelf wifi. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '15, page 267276, New York, NY, USA, 2015. Association for Computing Machinery.

[384] Yu Gu, Yifan Zhang, Jie Li, Yusheng Ji, Xin An, and Fuji Ren. Sleepy: Wireless channel data driven sleep monitoring via commodity wifi devices. *IEEE Transactions on Big Data*, 6(2):258–268, 2020.

[385] P. Wang, B. Guo, T. Xin, Z. Wang, and Z. Yu. Tinysense: Multi-user respiration detection using wi-fi csi signals. In *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–6, 2017.

[386] Y. Yang, J. Cao, X. Liu, and K. Xing. Multi-person sleeping respiration monitoring with cots wifi devices. In *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 37–45, 2018.

[387] C. Chen, Y. Han, Y. Chen, H. Q. Lai, F. Zhang, B. Wang, and K. J. R. Liu. Tr-breath: Time-reversal breathing rate estimation and detection. *IEEE Transactions on Biomedical Engineering*, 65(3):489–501, 2018.

[388] Chen Chen, Yi Han, Yan Chen, and K. J. Ray Liu. Multi-person breathing rate estimation using time-reversal on wifi platforms. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1059–1063, 2016.

[389] Xuyu Wang, Chao Yang, and Shiwen Mao. Tensorbeat: Tensor decomposition for monitoring multiperson breathing beats with commodity wifi. *ACM Trans. Intell. Syst. Technol.*, 9(1), September 2017.

[390] Youwei Zeng, Dan Wu, Jie Xiong, Jinyi Liu, Zhaopeng Liu, and Daqing Zhang. Multisense: Enabling multi-person respiration sensing with commodity wifi. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(3), September 2020.

[391] Fengyu Wang, Feng Zhang, Chenshu Wu, Beibei Wang, and K. J. Ray Liu. Passive people counting using commodity wifi. In *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, pages 1–6, 2020.

[392] F. Wang, F. Zhang, C. Wu, B. Wang, and K. J. R. Liu. Respiration tracking for people counting and recognition. *IEEE Internet of Things Journal*, 7(6):5233–5245, 2020.

[393] Q. Xu, B. Wang, F. Zhang, D. S. Regani, F. Wang, and K. J. R. Liu. Wireless ai in smart car: How smart a car can be? *IEEE Access*, 8:55091–55112, 2020.

[394] Q. Gao, J. Tong, J. Wang, Z. Ran, and M. Pan. Device-free multi-person respiration monitoring using wifi. *IEEE Transactions on Vehicular Technology*, 69(11):14083–14087, 2020.

[395] D. Zhang, Y. Hu, and Y. Chen. Mtrack: Tracking multiperson moving trajectories and vital signs with radio signals. *IEEE Internet of Things Journal*, 8(5):3904–3914, 2021.

[396] Zhicheng Yang, Parth H. Pathak, Yunze Zeng, Xixi Liran, and Prasant Mohapatra. Vital sign and sleep monitoring using millimeter wave. *ACM Trans. Sen. Netw.*, 13(2), April 2017.

[397] Huey-Ru Chuang, Hsin-Chih Kuo, Fu-Ling Lin, Tzuen-Hsi Huang, Chi-Shin Kuo, and Ya-Wen Ou. 60-ghz millimeter-wave life detection system (mlds) for noncontact human vital-signal monitoring. *IEEE Sensors Journal*, 12(3):602–609, 2012.

[398] Zhicheng Yang, Parth H. Pathak, Yunze Zeng, Xixi Liran, and Prasant Mohapatra. Monitoring vital signs using millimeter wave. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '16, page 211220, New York, NY, USA, 2016. Association for Computing Machinery.

[399] F. Wang, F. Zhang, C. Wu, B. Wang, and K. J. R. Liu. Vimo: Multiperson vital sign monitoring using commodity millimeter-wave radio. *IEEE Internet of Things Journal*, 8(3):1294–1307, 2021.

[400] Shengjie Li, Zhaopeng Liu, Yue Zhang, Qin Lv, Xiaopeng Niu, Leye Wang, and Daqing Zhang. Wiborder: Precise wi-fi based boundary sensing via through-

wall discrimination. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(3), September 2020.

[401] Fusang Zhang, Daqing Zhang, Jie Xiong, Hao Wang, Kai Niu, Beihong Jin, and Yuxiang Wang. From fresnel diffraction model to fine-grained human respiration sensing with commodity wi-fi devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1), March 2018.

[402] M. Speth, S.A. Fechtel, G. Fock, and H. Meyr. Optimum receiver design for wireless broad-band systems using ofdm. i. *IEEE Transactions on Communications*, 47(11):1668–1677, 1999.

[403] Dongheng Zhang, Yang Hu, Yan Chen, and Bing Zeng. Calibrating phase offsets for commodity wifi. *IEEE Systems Journal*, 14(1):661–664, 2020.

[404] Kun Qian, Chenshu Wu, Zheng Yang, Yunhao Liu, and Zimu Zhou. Pads: Passive detection of moving targets with dynamic speed using phy layer information. In *2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pages 1–8, 2014.

[405] Hao Wang, Daqing Zhang, Yasha Wang, Junyi Ma, Yuxiang Wang, and Shengjie Li. Rt-fall: A real-time and contactless fall detection system with commodity wifi devices. *IEEE Transactions on Mobile Computing*, 16(2):511–526, 2017.

[406] Xiang Li, Daqing Zhang, Qin Lv, Jie Xiong, Shengjie Li, Yue Zhang, and Hong Mei. Indotrack: Device-free indoor human tracking with commodity wi-fi. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(3), September 2017.

[407] Nicholas D Sidiropoulos and Rasmus Bro. On the uniqueness of multilinear decomposition of n-way arrays. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 14(3):229–239, 2000.

[408] Rasmus Bro and Henk AL Kiers. A new efficient method for determining the number of components in parafac models. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 17(5):274–286, 2003.

[409] Laurent Sorber, Marc Van Barel, and Lieven De Lathauwer. Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(l_r,l_r,1)$ terms, and a new generalization. *SIAM Journal on Optimization*, 23(2):695–720, 2013.

[410] Ioannis C. Tsaknakis, Paris V. Giampouras, Athanasios A. Rontogiannis, and Konstantinos D. Koutroumbas. A computationally efficient tensor completion algorithm. *IEEE Signal Processing Letters*, 25(8):1266–1270, 2018.

[411] Bo Yang, Gang Wang, and Nicholas D. Sidiropoulos. Tensor completion via group-sparse regularization. In *2016 50th Asilomar Conference on Signals, Systems and Computers*, pages 1750–1754, 2016.

[412] Athanasios A. Rontogiannis, Eleftherios Kofidis, and Paris V. Giampouras. Block-term tensor decomposition: Model selection and computation. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 1976–1980, 2021.

[413] Jose Henrique de Morais Goulart, Pedro Marinho Ramos de Oliveira, Rodrigo Cabral Farias, Vicente Zarzoso, and Pierre Comon. Alternating group lasso for block-term tensor decomposition and application to ecg source separation. *IEEE Transactions on Signal Processing*, 68:2682–2696, 2020.

[414] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on Signal Processing*, 45(2):434–444, 1997.

[415] Lieven De Lathauwer. A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization. *SIAM journal on Matrix Analysis and Applications*, 28(3):642–666, 2006.

[416] Dinh-Tuan Pham and J.-F. Cardoso. Blind separation of instantaneous mixtures of nonstationary sources. *IEEE Transactions on Signal Processing*, 49(9):1837–1848, 2001.

[417] Lieven De Lathauwer. Blind separation of exponential polynomials and the decomposition of a tensor in rank-(l_r,l_r,1) terms. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1451–1474, 2011.

[418] Khadija Baba, Lahcen Bahi, and Latifa Ouadif. Enhancing geophysical signals through the use of savitzky-golay filtering method. *Geofsica Internacional*, 53(4):399–409, 2014.

[419] Dan Wu, Daqing Zhang, Chenren Xu, Yasha Wang, and Hao Wang. Widir: Walking direction estimation using wireless signals. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, page 351362, New York, NY, USA, 2016. Association for Computing Machinery.

[420] Peter Hillyard, Anh Luong, Alemayehu Solomon Abrar, Neal Patwari, Krishna Sundar, Robert Farney, Jason Burch, Christina Porucznik, and Sarah Hatch Pollard. Experience: Cross-technology radio respiratory monitoring performance study. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, MobiCom '18, page 487496, New York, NY, USA, 2018. Association for Computing Machinery.

[421] Yaxiong Xie, Jie Xiong, Mo Li, and Kyle Jamieson. Md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, New York, NY, USA, 2019. Association for Computing Machinery.

[422] Shui Yu, Meng Liu, Wanchun Dou, Xiting Liu, and Sanming Zhou. Networking for Big Data: A Survey. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, 2016.

[423] Laizhong Cui, F. Richard Yu, and Qiao Yan. When big data meets software-defined networking: SDN for big data and big data for SDN. *IEEE Network*, 30(1):58–65, 2016.

[424] Nico Vervliet, Otto Debals, and Lieven De Lathauwer. Tensorlab 3.0 - numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization. In *Signals, Systems and Computers, 2016 50th Asilomar Conference on*, pages 1733–1738. IEEE, 2016.

[425] Daniel Kressner and Christine Tobler. htucker - a matlab toolbox for tensors in hierarchical tucker format. *Mathicse, EPF Lausanne*, 2012.

[426] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[427] Joel Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, 6(1):7–12, 1960.

Appendix A

Research Commercialization

Tensor network decomposition, originated from quantum physics to model entangled many-particle quantum systems, turns out to be a promising mathematical technique to efficiently represent and process big data in parsimonious manner. In this study, we show that tensor networks can systematically partition structured and un-structured data for distributed storage and communication in privacy-preserving manner. Leveraging the sea of big data and metadata privacy, empirical results show that neighboring subtensors with intrinsic information stored in tensor network formats cannot be identified for data reconstruction. This technique complements the existing encryption and randomization techniques which store explicit data representation at one place and highly susceptible to adversarial attacks such as side-channel attacks and de-anonymization. This chapter explores the potential commercialization pathways of tensor network computing in this big data era. Through extensive market survey and business model analysis, we identify potential applications, market competitors, and implementation details of the proposed technology. The unique selling points of tensor network compression include working seamlessly with various data structures, can be combined with existing privacy-preserving technology, and can be seamlessly integrated into existing data storage, communication, and computation processes. Most importantly, the proposed technology provide distributed trust and distributed management for big data privacy. The target market segments include individuals, enterprises, government bodies, cloud providers and data centers with a total of 16 billions annual spending on data security solutions.

A.1 Introduction

Big data serves as the fuel in driving deep learning models that create tremendous value for various applications, ranging from science, business, to government. Deep learning automates the process of feature extraction and exploits their compositionality to construct high-level features that achieve human-level performance in many designated tasks such as classification and prediction [107]. The tradeoff, however, involves storage and processing of large amount of (labeled) data and models with millions to billions of parameters. The data explosion growth is expected to outpace the development of storage and processing technology, therefore domain-specific hardware acceleration and algorithmic codesign all aim to improve throughput and energy-efficiency without compromising model performance and hardware costs to cater for widespread deployment of deep learning models [108]. Sharing of personal and confidential data across organizations demand cutting-edge privacy-preserving technology. Current privacy-preserving technologies such as encryption and randomization techniques share a common drawback that any security breaches such as leakage of decryption key or the data content during the storage, communication, or computation phases expose the individual records that contain explicit information. Therefore, it is timely and essential to explore new data structures which provide not only efficient and distributed storage and computation, but also privacy preservation such that data leakage provides the adversary partial (intrinsic or latent) information of individual records and reconstruction is difficult without knowledge of the data structure. In this chapter, we propose to use tensor network representations for distributed storage of input data for machine learning models. In particular, the model performance, storage, and compression / decompression efficiency are benchmarked for CNNs. Empirical results based on information theory show that neighboring subtensors with intrinsic / latent information cannot be identified for data reconstruction. The robustness of tensor network representations subject to perturbation of the subtensors is also investigated.

A.2 Technology Disclosure

The proposed research commercialization receives financial support from the NTUitive Gap Fund. The project ID is 2019-109-01-SG PRV, dated from Sep 2019 to Aug 2021. The principal investigators include Assoc. Prof. Ng Wee Keong, Assoc. Prof. Wang Huaxiong, and myself. A PCT patent application is submitted for examination based on the proposed big data shredding technology. The PCT Application Number is PCT/SG2020/050404 and the International Filing Date is 13 July 2020. Through the Lean LaunchPad Programme organized by NTUitive, we have mapped out the value proposition and business model canvas to commercialize the proposed technology as shown in Figures A.1 and A.2. Extensive market survey also suggests that multi-cloud management Software-as-a-Service providers are very interested in adopting the shredding technology to protect big data privacy, some interviewees also suggest combining our technology with blockchain for secure data sharing in multi-clouds or hybrid-clouds environment. Big data processing software such as Hadoop is highlighted as a very promising commercial pathway where backend integration of the proposed big data privacy solution can be done. It would further attract commercial interest if query, operations, and analytics in big database or data warehouse can be performed with the proposed technology.

Background of the invention. The development of cutting-edge artificial intelligent (AI) systems requires lots of data collected from sensors and Internet-of-Thing (IoT) devices to achieve high performance in many tasks ranging from business decision-making to personalized services for end customers. Big data requires storage and processing power beyond traditional computing resources, therefore enterprises and government bodies undergoing digital transformation often need to extend their computing capability to the cloud and mobile environments. However, big data contains sensitive information that can be exploited once placed in the public cloud resources. For example, videos taken by a network of surveillance camera contain personal information such as individuals locations and preferences. As mentioned in Thales Data Threat Report 2018 and 2019, the growing attack surfaces from digital transformation calls for simpler data-security solutions such as

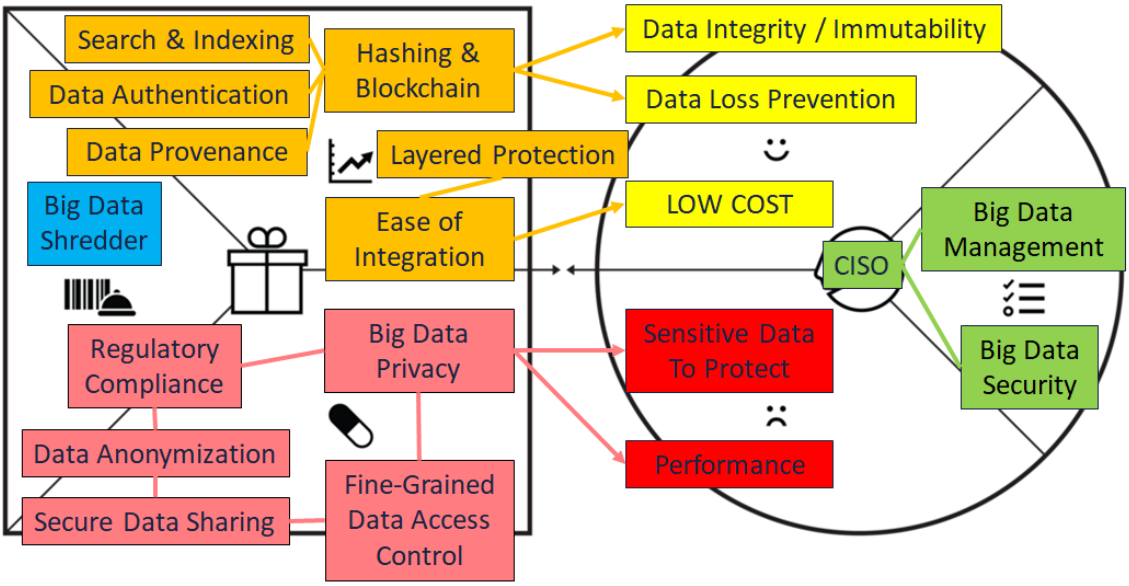

Figure A.1: Value proposition of the proposed technology. The target customer is chief information security officers (CISO) who is interested in data security products.

encryption and access control. Classical cybersecurity solutions such as end-point security, network security, and digital vault are not scalable and not cost-effective to protect data privacy and security. The state-of-the-art privacy-preserving big data storage, communication, computation, and sharing techniques are reviewed below:

- *Encryption* is a proven technique to protect confidential data by encoding the information in such a way that only authorized parties with the decryption key can decipher it. Encryption is susceptible to side-channel attacks and many existing public key encryption schemes are susceptible to quantum-computing attackers in future. Big data encryption is complicated in terms of the key management and distribution; the re-encryption of large amount of data is a bottleneck to scalability of encryption technique. Therefore, commercial applications encrypt only confidential data at much smaller scale on clouds, e.g., relational databases which contain customer and proprietary information. Furthermore, encrypted data computation such as homomorphic encryption involves very high computational complexity can incur up to several orders of performance overhead, hence making it not practical for big data processing.

- *Classical Secure Multi-Party Computation* involves multiple rounds of com-

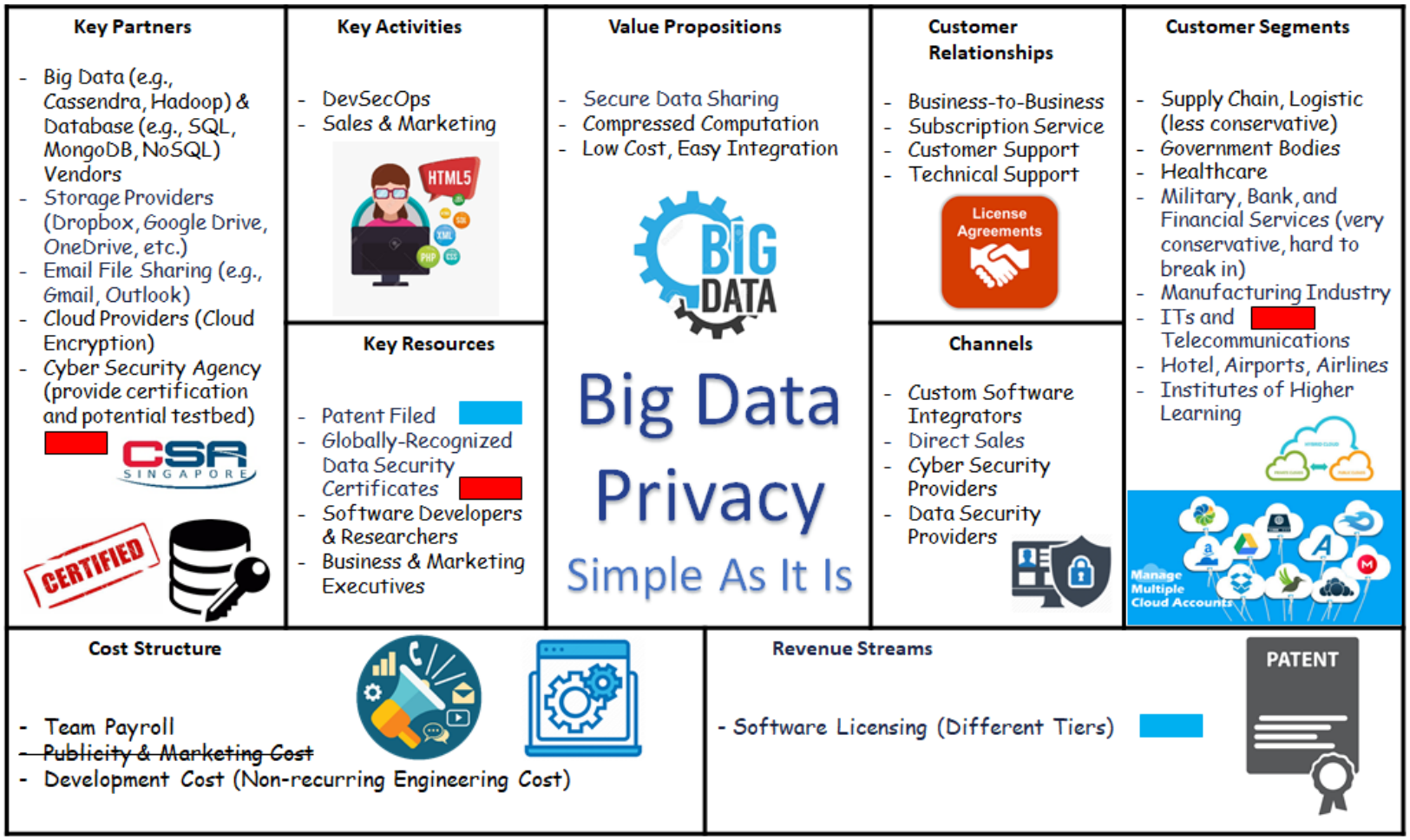

Figure A.2: Business model canvas.

munications and computation using the secret shares or random splitting of a piece of data among multiple servers to securely compute a function (e.g. addition and multiplication). The security model requires the servers to be controlled by mutually-untrusted parties to achieve information-theoretical security. Existing schemes such as garbled circuit requires oblivious transfer and symmetric cryptographic operations in the online phase; secret sharing scheme requires high round / communication complexity and only practical / efficient enough with high-speed networking such as LAN connections. These techniques are typically based on modular arithmetic and works on discrete value such as integers and fixed-point representations. Protocols and algorithms have to be redeveloped for continuous value such as floating-point arithmetic, hence making classical techniques inefficient and not scalable for modern big data and machine learning applications.

- *Hardware enclaves* or secure enclave is hardware-enforced isolated execution environment (or processor with security support) that allow general-purpose computing on confidential or sensitive data, i.e., encrypted data can be decrypted inside the hardware enclaves and computation can be done on the plaintext. Intels SGX and ARMs TrustZone are paving the way towards realizing hardware enclaves for various applications. Nonetheless, hardware enclaves are still costly and experimental at the time being, most of the processors such as cloud resources, commercial workstations and AI chips nowadays do not have such high-level of security support. Furthermore, hardware enclaves become the single point of attack failure, we believe a new paradigm of distributed trust is needed to ensure the privacy of big data distributed applications such as secure multi-party computation.

- *Data anonymization* techniques such as data perturbation and differential privacy inject noise into the data to prevent privacy leakage, however it is difficult to control the noise threshold in order to balance usability and privacy of confidential or sensitive data. These transformation techniques are not reversible therefore result in information loss. Therefore, data anonymization does not

provide very useful solution for big data storage, communication, sharing, and computation on public cloud resources.

- *Data splitting* techniques partition data into randomized blocks at byte, attribute, or semantic level. Data splitting is widely used in the industry to provide layered data protection. However, data splitting releases true values and requires centralized server to keep track of the splitting criterion to ensure that each block does not leak privacy.

Other data-security techniques include tokenization, hashing, and blockchain are effective approaches and will be combined with our proposed innovation for privacy and security enhancements for practical commercial applications and operations.

Core techniques. We propose randomized tensor network decomposition to efficiently decompose big data into fragments (smaller blocks of tensor) with partial information that are randomized, un-linkable, and not interpretable. As such, the proposed technology is a randomized information dispersal algorithm. These fragments (distributed tensor network representations) are distributed among multiple virtual instances, devices, servers, nodes or clouds controlled by non-colluding parties or one party with multiple authentication factors to provide distributed trust; the fragments are protected by metadata privacy such that only authorized user is able to recover the original record with the metadata that stores the fragments location and reconstruction algorithms. The distributed tensor network representations naturally support compressed and distributed / dispersed computation, making it well-suited for big data processing. Furthermore, we propose randomized and distributed / dispersed tensor network computation such that the fragments are randomized before and after performing mathematical operation. Sophisticated hackers would have to gain access to all or most of the communication routes, storage or computing nodes / servers in order to recover the original and processed information. We also propose incremental update scheme of the randomized tensor network representations to cater for real-time streaming data. Furthermore, we propose conversion to-and-fro and operations between the tensor network representations and classical secret-sharing scheme to increase the range of supported secure operations.

As a mathematical technique that has been well-studied for big data processing, the proposed randomized tensor network algorithms can decompose and process various kind of data structures such as tabular, graphical, discrete, or continuous data (e.g., relational databases, graphical databases, structured, unstructured, and semi-structured databases), pre-process big data for data integration and cleaning, the tensor representations can be updated incrementally or dynamically, and the proposed technique can be easily integrated into existing computing platforms, environments, and processes (e.g., mobile-cloud environments).

Implementation framework. The proposed invention makes use of the distributed storage and processing of tensor network representations to seamlessly provide privacy-preserving big data storage, communication, computation, and sharing. Privacy and security of distributed / dispersed tensor network representations and computation can be enhanced significantly within the multi-party computation setting by distributing the tensor blocks (or distributed tensor network representations) to different virtual instances, nodes, servers, or clouds with metadata privacy. Access control of the fragments (tensor blocks) of the tensor network representations has to be given to non-colluding parties or one party with different authentication factors on different portions of the fragments in order to provide distributed trust for data protection. Data-secure implementations of the proposed invention will require combining the proposed invention with traditional data-security technologies such as data anonymization, differential privacy, data splitting, and encryption to provide layered protection, perform the conversion to-and-fro / operations between distributed tensor network and classical secure multi-party computation / secret-sharing scheme or perform computation with the aid of secure-enclave technology to increase the flexibility of secure computing circuits / functionality and computational / communication efficiency, implement digital signatures and hashing / blockchain technology to authenticate the data fragments and ensure data availability and integrity, combine with MAC (message authentication code) and digital signatures to provide verifiable computation that is secure against malicious parties, and to use the proposed invention to compute zero-knowledge proof. Machine-learning models can be compressed and trained in tensor network representations with differential

privacy; both the model training and inference can be performed with distributed / dispersed tensor network computation. Metadata of the data fragments contains fragments location that correspond to particular record and the reconstruction algorithms to recover the records. Metadata privacy and security can be achieved with classical data-security technologies such as encryption or secret sharing schemes. Both constrained and unconstrained optimization techniques can be used to speed up and compress the distributed tensor network decomposition / computation for various big data applications. The tensor network decomposition can be performed in plaintext, encrypted data computation, or with data perturbation techniques. Randomization can be achieved with randomization of the hyperparameters during tensor network decomposition, initialization, and computation. The proposed invention works for both randomized and non-randomized tensor network decomposition and computation, but it should be noted that randomization increases the privacy guarantee. We anticipate the proposed invention to be implemented at filesystem, database, or application levels of existing software architecture. The proposed invention can operate on the byte, attribute, or semantic level of data. The data can be compressed or approximated using tensor network with lossy or lossless accuracy. Near-lossless data accuracy can be achieved with tensor network lossy compression and residual coding. Further compression of distributed tensor network representations can be done with existing codec such as dictionary-based compression, run-length encoding, and arithmetic coding.

Software Architecture. Figure A.3 shows an example software architecture of the proposed secret sharing scheme based on distributed tensor network computation. The data is first ingested from multiple databases and pre-processed in Hadoop parallel processing framework (i.e., MapReduce) for various big data applications, Hadoop MapReduce framework ensures the data availability by duplicating the data to multiple copies and stored in different computing nodes. Accordingly, our proposed method of distributed data management provides a big data shredder or dispersal service to distribute the randomized tensor blocks to multiple databases hosted on multiple public clouds. The metadata that comprises the identity information and location information of the randomized tensor blocks is stored within

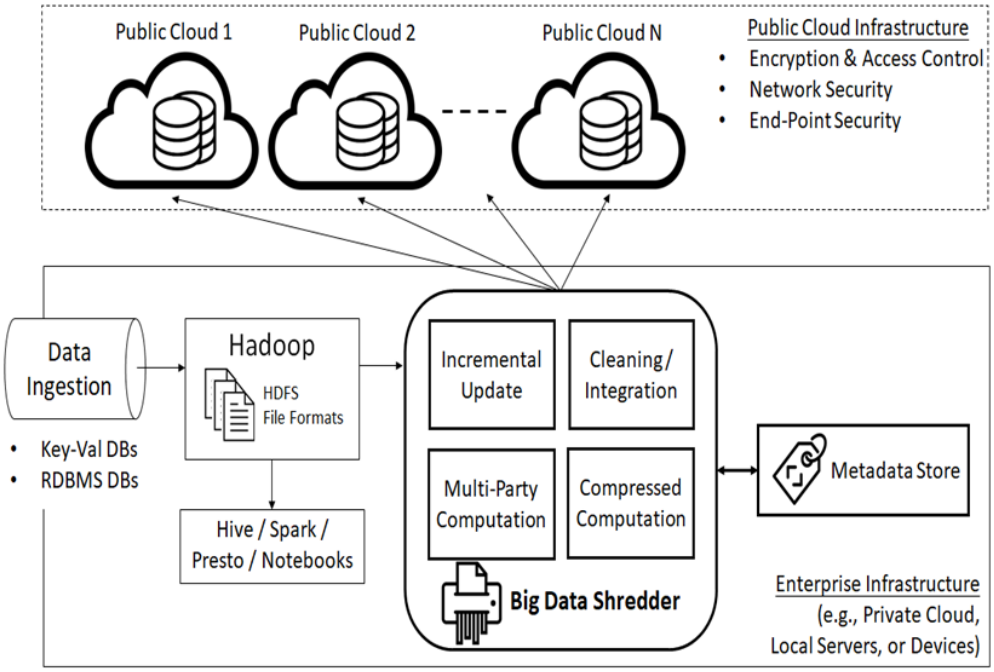

Figure A.3: An example software architecture for the proposed secret-sharing scheme based on distributed tensor network computation.

the enterprise infrastructure.

Advantages and improvements over existing methods. Tensor networks decompose big data into fragments with partial information which are un-linkable and not interpretable; the fragments are then communicated and stored on single/multiple devices/clouds with metadata privacy. The proposed technology simplifies the encryption key management and distribution by providing distributed trust for big data applications in complex mobile-cloud environments nowadays. For example, enterprise can share data by transferring the metadata and data fragments through secure communication channels without exchanging decryption keys or doing re-encryption. The proposed invention is promising for big data decomposition and compression and inherently supports compressed and distributed computation. Therefore, this simplifies the big data processing by removing the need to decompress or preprocess the data compared to classical privacy-preserving computation techniques. Furthermore, the distributed nature of the proposed invention naturally supports secure multi-party computation based on fixed-point or floating-point

Comparative Analysis– Private Storage / Sharing

	Tensor Shredding	Secret Sharing	Symmetric Encryption	Public Key Encryption	Data Anonymization	Data Splitting
Data Privacy Approach	Randomized Information Dispersion	Information-Theoretical Security	Computational Hardness	Computational Hardness	Masking / Randomization	Random Partition, Release True Values
Trusted Authority	Distributed Management	Distributed Management	Centralized Management	Centralized Management	Centralized / Distributed	Centralized Management
Data Threat Model	Distributed Trust	Distributed Trust	Centralized Trust	Centralized Trust	Centralized Trust	Centralized Trust
Keyed	Metadata Privacy	Metadata Privacy	Keyed	Keyed	Non-Keyed	Metadata Privacy
Byte / Attribute / Semantic Level	Semantic Level	Byte / Semantic Level	Byte / Semantic Level	Byte / Semantic Level	Semantic Level	Byte / Attribute/ Semantic Level
Storage / Comm.	Compressed	2 – 3x Larger	Same	Same	Same	Same
Encoding Speed	Mod. / Fast	Fast	Fast	Mod. / Slow	Mod. / Fast	Fast
Decoding Speed	Fast	Fast	Fast	Mod. / Slow	Fast	Fast
Data Accuracy	Lossy, Lossless	Lossless	Lossless	Lossless	Lossy	Lossless
Re-Encrypt / Re-Sharing	Randomized Re-Sharing	Re-Sharing	Re-Encryption (Bottleneck)	Re-Encryption (Bottleneck)	Re-Anonymize lose more info.	Observe Splitting Criterion

Figure A.4: Comparison of secure storage techniques across technical parameters.

arithmetic and reduces the communication overhead between multiple computing nodes. The proposed invention provides layered protection against sophisticated hackers with the sea of big data to search for fragments that correspond to particular record. As a mathematical technique, the proposed invention can be easily combined with existing privacy-preserving techniques and seamlessly integrated into exiting computing environments, platforms, or processes. Figures below systematically compares our big data shredding technology with existing data-security solutions based on the technical parameters. Notice that these privacy-preserving techniques are not mutually exclusive with each other but can be combined to provide layered data protection.

Dimensionality Reduction of Input Data To compute the TN decomposition, Matlab 2017a and several toolboxes are used: Tensorlab 3.0 [424], htucker 1.2 [425], and TT-toolbox 2.2.2. The compression and decompression time are benchmarked using Intel(R) Xeon(R) CPU E5-1650 v4 @ 3.60GHz 3.60GHz. CP and TD are computed using Tensorlab "cpd_gevd" and "mlsvd" functions, HT using htucker "htensor.truncate_ltr" function, and TT using TT-toolbox "tt_tensor" function. These functions use generalized eigenvalue or SVD to speed up the decomposition. Ta-

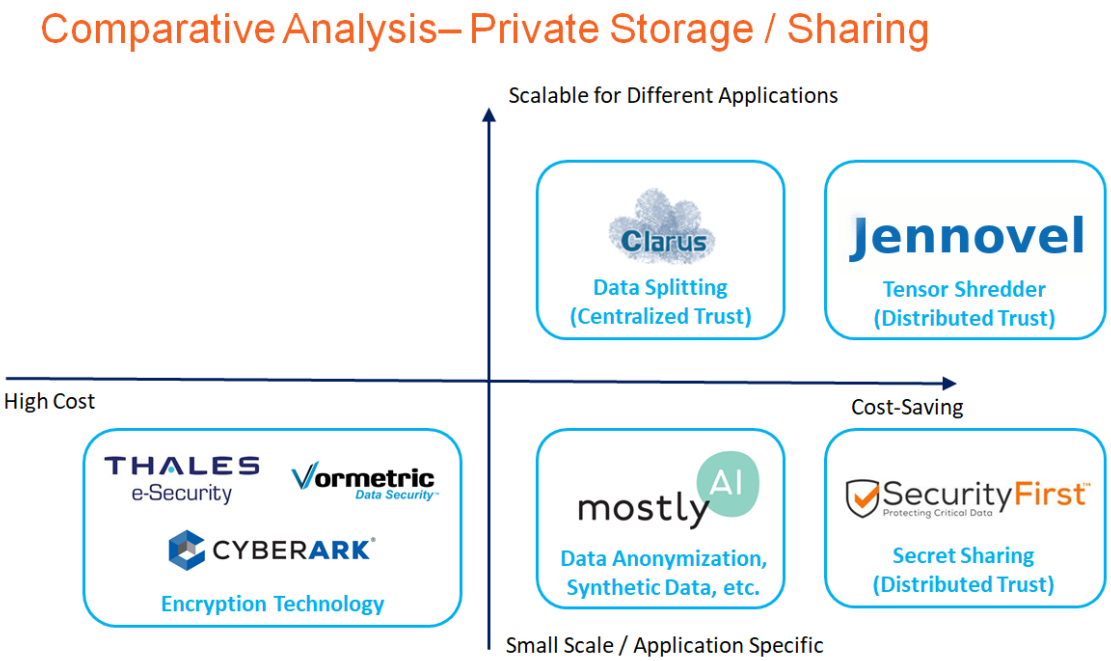

Figure A.5: Comparison chart of secure storage techniques.

Comparative Analysis – Private Computation

	Tensor Dispersed Computation	Multi-Party Computation	Secure Enclave	Data Anonymization / Data Perturbation	Data Splitting
General-Purpose Computing	Arithmetic (✓), Boolean (?)	Arithmetic, Boolean	General-Purpose	General-Purpose / Application-Specific	General-Purpose (Attribute Level)
Dispersed Computation	Distributed Trust	Distributed Trust	Centralized Trust	Centralized Trust	Centralized Trust
Computational Complexity	Exact, Compressed Computation	2 – 3x Larger, Pre-Processing	High Performance	Low Complexity / Optimized	Low Complexity
Communication Complexity	Compressed / Optimized	High Complexity	Low Complexity	Low Complexity / Optimized	Low Complexity
Data Security Level	Non-Cryptographic, Randomization	Cryptographic	Secure Hardware	Non-Cryptographic, Randomization	Non-Cryptographic
Need Interactions	Yes	Yes	No	Yes / No	No
Data Representations	Fixed-Point / Floating-Point	Fixed-Point	Fixed-Point / Floating-Point	Fixed-Point / Floating-Point	Fixed-Point / Floating-Point
Scalability	Large-Scale, Structured / Un-Structured Data	Small-Scale	Small-Scale (Costly Usage)	Large-Scale (Application-Specific)	Large-Scale (Application-Specific)
Data Accuracy	Exact / Approximate	Exact	Exact	Exact / Approximate	Exact

Figure A.6: Comparison of secure computation across technical parameters.

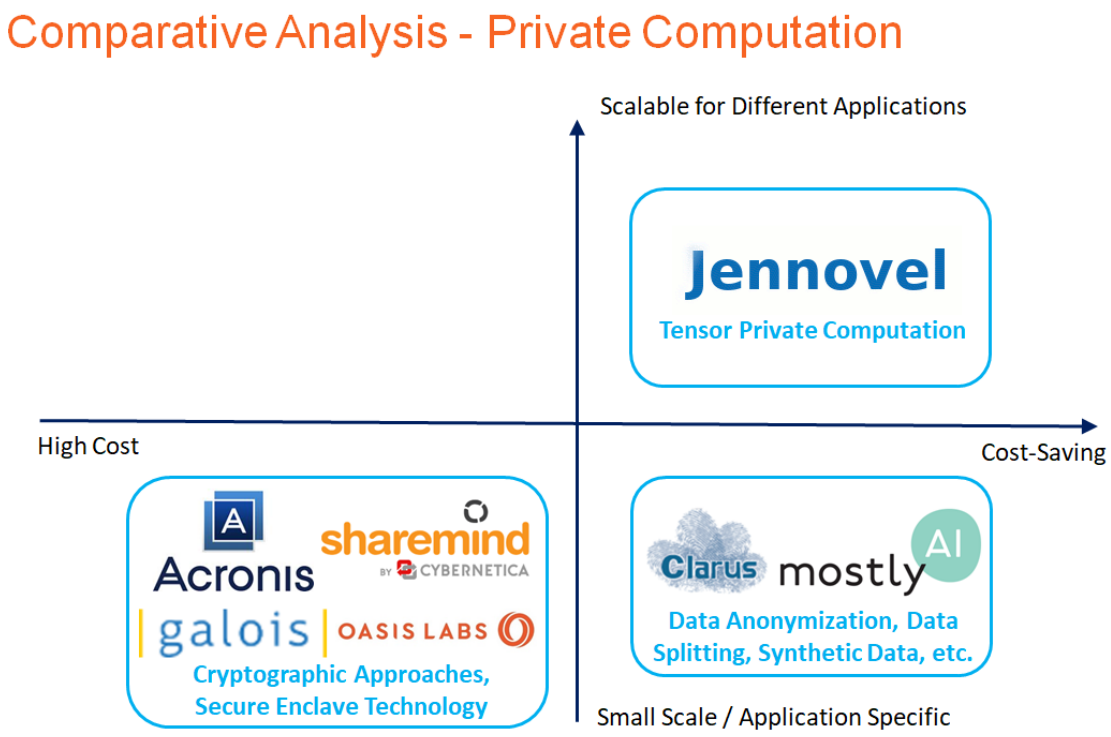

Figure A.7: Comparison chart of secure computation techniques.

ble A.1 benchmarks the compression / decompression time for TN with different compression ratio. The time needed generally increases with TN size. CP decomposition is about 4× longer than other TN decomposition. TN decompression time is generally much faster compared to compression time. After TN decomposition, the cores and latent factors are quantized to 8-bit depth. Some of the subtensors can be uniformly quantized but some require non-uniform quantization using Lloyd's algorithm [426, 427] to reduce the image distortion, e.g., TD's core G. It can be observed that TN generally retains the features for image classification by CNNs without the need to retrain the model; at least half of the storage size can be saved using TN for data compression.

Privacy-Preserving Distributed Data Storage and Communication. Figure A.8 shows the image distortion as a result of adding noise to randomly-selected TN subtensor, the effect is larger if the perturbations is applied on the singular vectors corresponding to the leading singular values, however this information is usually unknown to the adversary. CP's distortion is larger because the format is more

TN	Time (ms)	Compression Ratio	Top-1 Accuracy
MNIST Dataset			0.993
CP	N/A	N/A	N/A
TD	0.6 / 0.3	0.390	0.988
HT	2.5 / 0.5	0.422	0.988
TT	0.39 / 0.07	0.439	0.990
SVHN Dataset			0.951
CP	7.9 / 0.14	0.327	0.945
TD	1.6 / 0.37	0.309	0.950
HT	2.8 / 0.56	0.342	0.950
TT	0.97 / 0.07	0.150	0.929
CIFAR-10 Dataset			0.858
CP	7.8 / 0.13	0.334	0.811
TD	1.6 / 0.39	0.309	0.8105
HT	2.9 / 0.58	0.342	0.809
TT	1.2 / 0.09	0.455	0.833
ImageNet Dataset			0.748
CP	214 / 2.2	0.460	0.535
TD	53.9 / 4.3	0.335	0.693
HT	53.8 / 3.2	0.372	0.703
TT	48.3 / 2.7	0.501	0.655

Table A.1: Model performance using TN for input data compression. The time for compression / decompression per image is measured in milliseconds. CP decomposition is not available (N/A) for MNIST dataset because of algorithmic instability.

compact compared to other TNs. Due to the diverse possible topology structure of decomposition for a given tensor, Wang et al. [79] propose three different security models to process data generated by cyber-physical-social systems, i.e., open model, half-open model, and encrpted model to process data with different level of sensitivity and privacy requirements. The tensor formats and topology structure are made private to selected users for half-open and encrypted models. Here, we experimentally verify that the neighboring subtensors could not be identified based on information theory. Mutual information is commonly used to cross-examine the information content between subtensors [1]. Figure A.9 shows the normalized mutual information (NMI) between two subtensors of particular TN for one image, two images, and random noise, the results show that they are indistinguishable from each other. NMI is a universal metric such that any other distance measure judges two random variables close-by, NMI will also judge them close. As shown in Figure A.9. the NMI variation is largely attributed to the variation in subtensor's value distribution, if the variation in particular subtensor is high (i.e., entropy is high), its NMI with other subtensor is likely to be smaller.

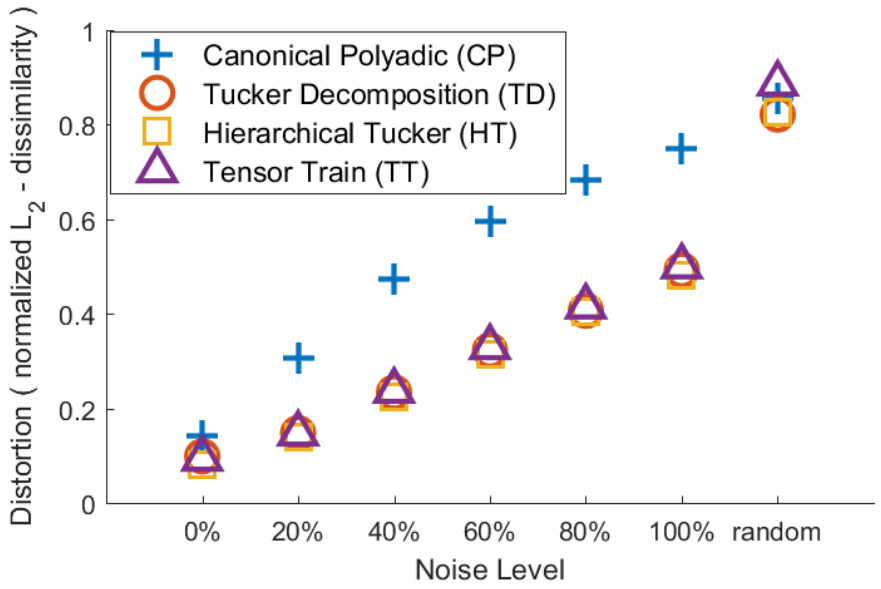

Figure A.8: Image distortion resulted from adding noise to a randomly-selected core of the TN. Note that "random" label in the x-axis means randomize the sequence in the selected core.

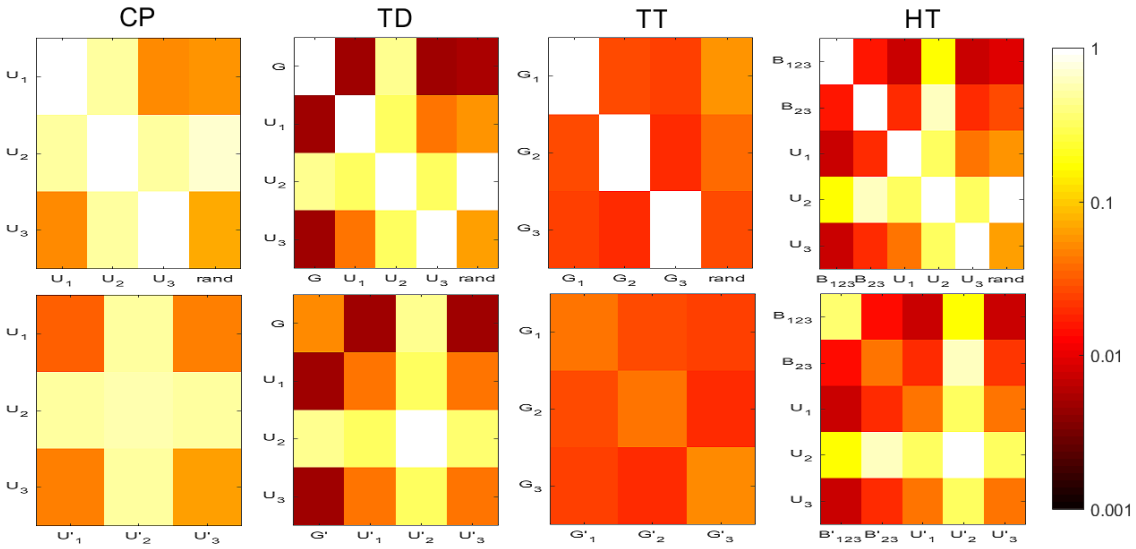

Figure A.9: Normalized mutual information between cores and latent factors of one image (top row) and two different images (bottom row) for different TNs. Note that "rand" label in the x-axis means cores with uniformly-distributed noise.

Embodiment of the invention. Below we describe the technical implementations of the proposed invention:

1) *Big Data At-Rest and In-Transit Security:* Data owner shreds / decomposes / approximates their data using tensor network decomposition and distribute the smaller tensor blocks to multiple clouds, hybrid clouds, multiple virtual instances of a single cloud, servers, or devices (see Figure. A.10) . The shredding / decomposition process can be performed in plaintext, anonymized / perturbed data, or encrypted data. The tensor blocks can be further compressed using existing codec before or after distribution to multiple storage points. There are a few ways to realize the secure multi-party computation setting to provide distributed trust depending on the security model. Firstly, the data owner can store the tensor blocks on multi-clouds or hybrid-clouds environments. Secondly, the data owner can store the tensor blocks to multiple virtual instances with different authentication factors, however this can potentially leak the data privacy to cloud administrator but still resists hacking by removing single point of failure. Thirdly, an organization can give access to different sub-organization portions of the tensor blocks that correspond to particular record, this can be implemented in single cloud, multiple or hybrid clouds. A data

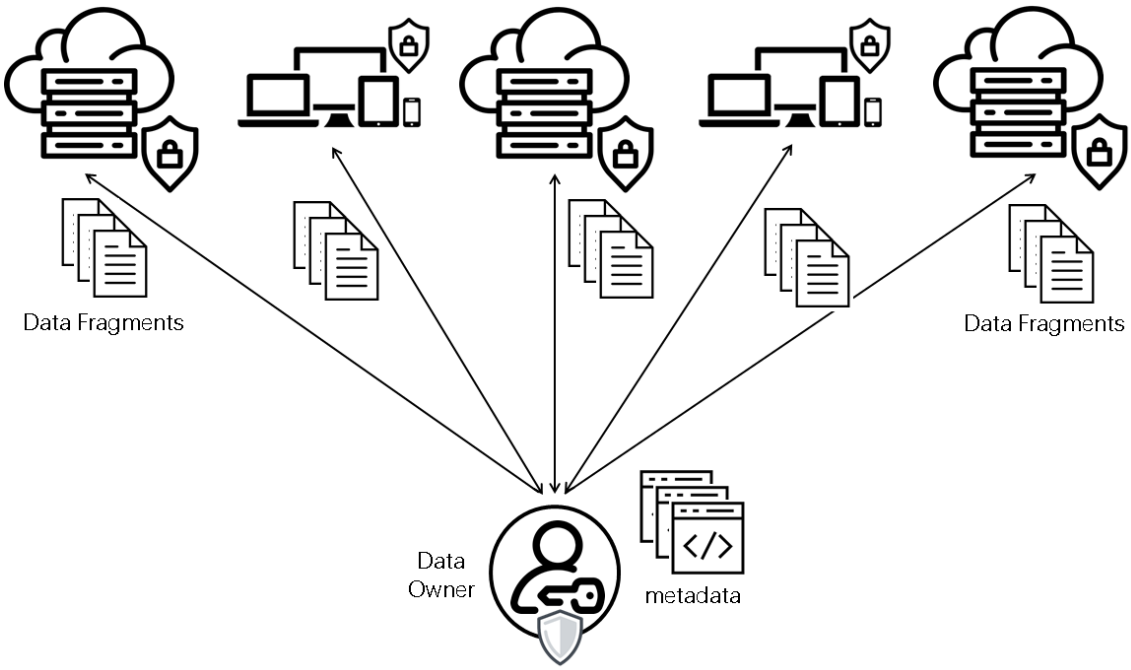

Figure A.10: Secure big data storage and communication.

owner can also distribute the tensor blocks to multiple devices and mobile-cloud environments. Metadata of the data fragments correspond to particular record can be protected by encryption or secret sharing scheme. The metadata stores the fragments location and reconstruction algorithm in order to recover the original data. The reconstructed data accuracy can be lossy, lossless, or near-lossless. The data owner or organization can first retrieve the metadata from the storage points, locate and download the data fragments / tensor blocks from the clouds, server, or devices, reconstruct the original data using the reconstruction algorithm stored in the metadata. Note that the proposed invention can be easily combined with existing privacy-preserving technologies such as anonymization, secret sharing, encryption, and secure enclave to provide layered protection and enhance the functionality or efficiency.

2)*Secure Big Data Sharing:* Data or content owner instructs the clouds or devices that store the fragments corresponding to particular record to give access to the intended user to share the record. The data or content owner also share the metadata corresponding to the record to the intended user so that the user can reconstruct the original record using the metadata and data fragments. All the communication can

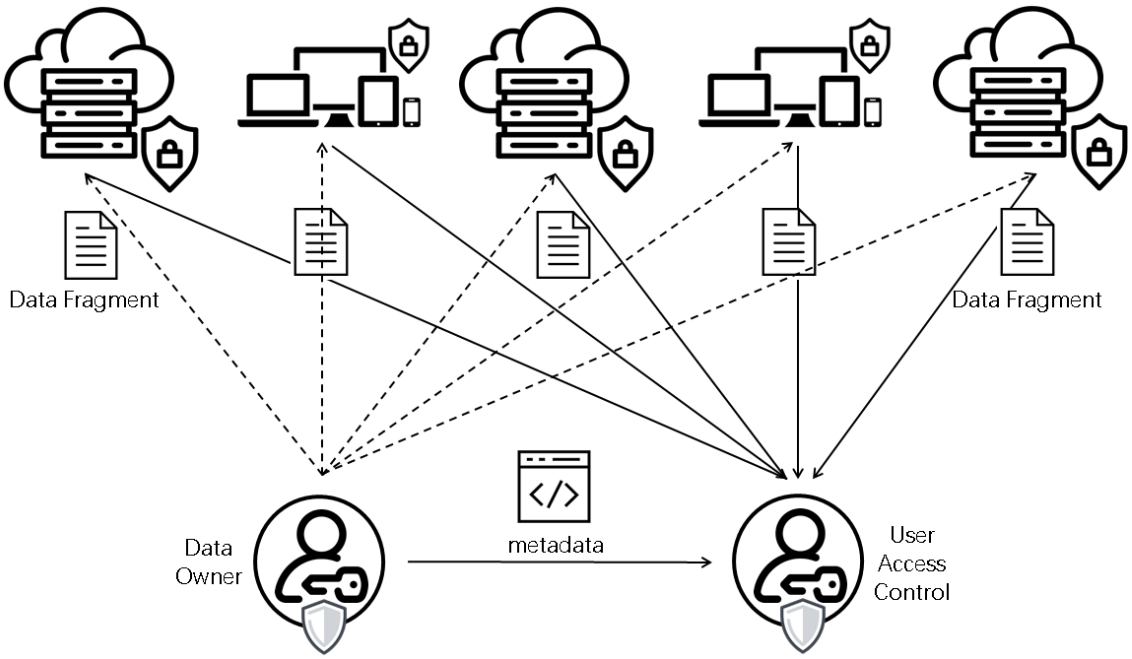

Figure A.11: Secure big data sharing.

be done on multiple secure communication channels to provide distributed trust. Compression and granular data access control on the dataset or database can be simultaneously achieved with the proposed innovation. See Figure A.11 for the technical implementation.

3)*Privacy-Preserving Big Data Computation:* Data or content owner instructs the clouds, servers, or devices that contain the tensor blocks / data fragments corresponding to particular record to perform secure multi-party computation based on distributed tensor network representations and operations (see Figure A.12). The secure computation can be combined with existing privacy-preserving techniques such as secret sharing, encryption, and secure enclave. Each cloud, server, or device contains and communicates only partial information and therefore hackers would have to gain access to multiple routes, storage, and computing nodes in order to reconstruct the original record. The data owner can also send incremental updates to the storage / computing nodes to update the tensor blocks using distributed / dispersed tensor network computation to ensure the updated data remains compressed.

3)*Secure Multi-Party Computation:* Multiple data or content owners instruct the

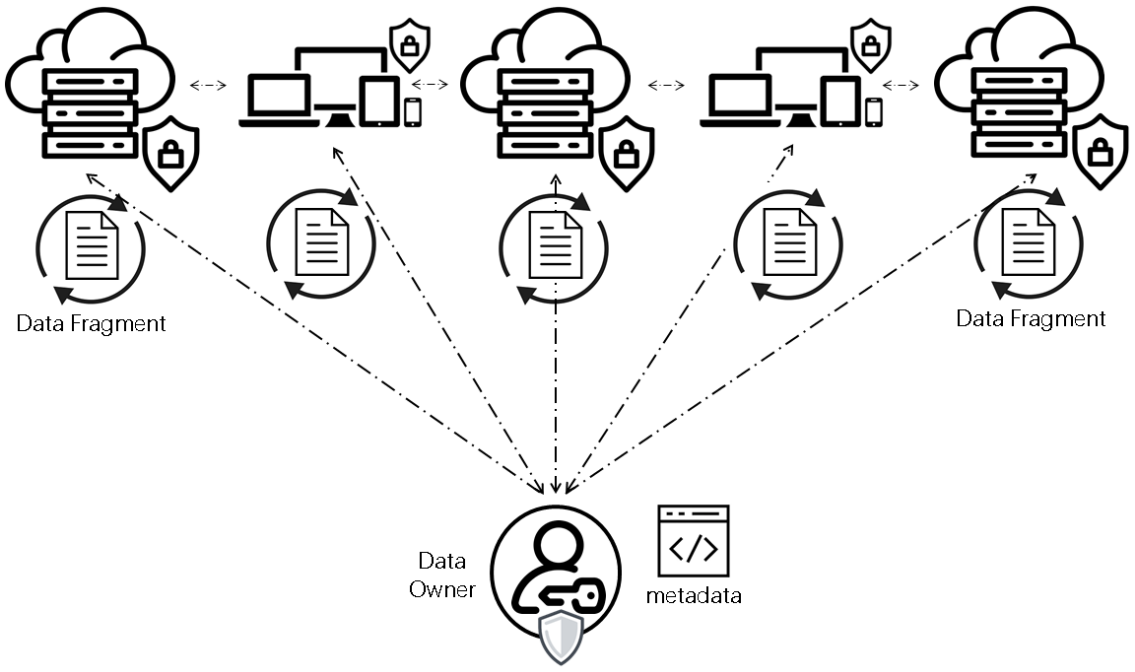

Figure A.12: Secure big data computation.

clouds that contain the data fragments or tensor blocks corresponding to respective records to perform secure multi-party computation based on distributed tensor network representations and computation (see Figure A.13). Each data owner has control on different portions of the shared fragments to ensure privacy and fairness of joint computation. The distributed tensor blocks of the computed function are retrieved by the data or content owners. Real implementations require combining secret sharing, encryption, and secure enclave to increase the functionality and improve the computational or communication efficiency of the proposed secure computation.

Commercial applications. As more enterprises undergoing digital transformation, cloud computing becomes inevitable for their daily commercial operations. Sharing of data collected between multiple parties increases mutual benefits (e.g. banks want to combine their data for fraud detection, hospitals want to increase the size of their database for more accurate diagnosis or predictions, etc.); but this also incurs privacy-preservation issues, our proposed invention provides efficient and secure big data sharing, secure multi-party computation, and scalability for privacy-preserving big data computation. The proposed invention can help to secure big data ap-

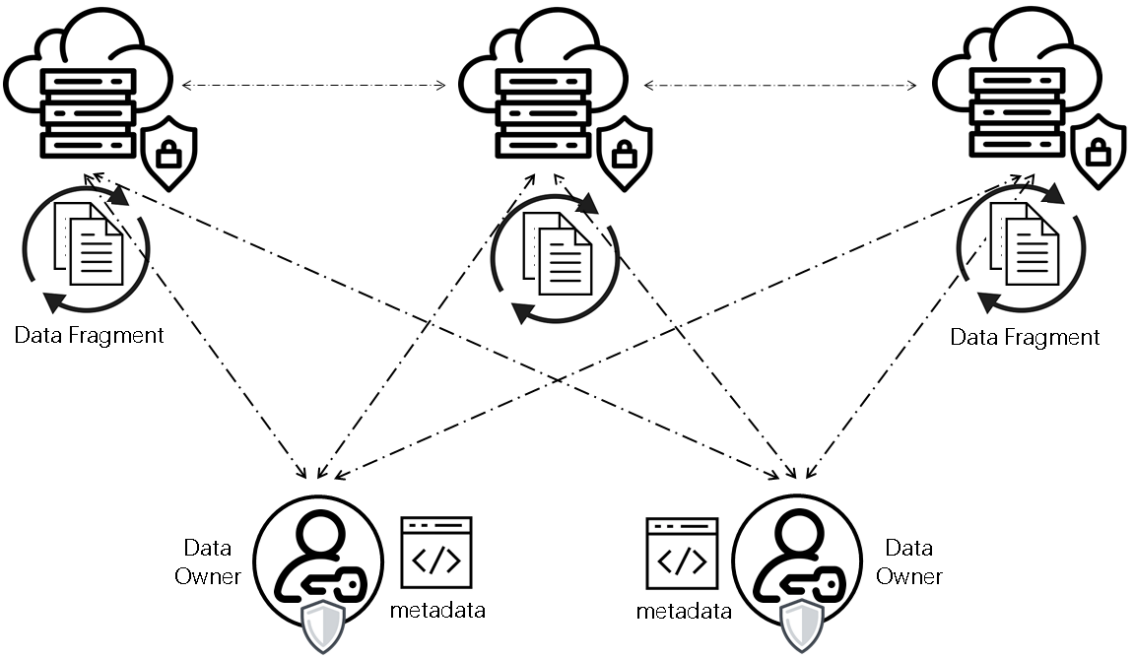

Figure A.13: Secure multi-party computation.

plications such as data warehouse (e.g., data cleaning and integration), database / database query, operations and analytics, and filesystem privacy for distributed software applications on clouds, fogs, edges, and devices to facilitate the digital transformation and digital data sharing within and cross enterprises ranging from healthcare, smart manufacturing, and smart cities applications. Furthermore, the proposed invention can be used for compressed and private computation of machine learning models and large-scale numerical computing.