9-2021

Secure and verifiable outsourced data dimension reduction on dynamic data

Zhenzhu CHEN

Anmin FU

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Ximeng LIU

Yang YANG

See next page for additional authors

Citation

CHEN, Zhenzhu; FU, Anmin; DENG, Robert H.; LIU, Ximeng; YANG, Yang; and ZHANG, Yinghui. Secure and verifiable outsourced data dimension reduction on dynamic data. (2021). *Information Sciences*. 573, 182-193.
Available at: https://ink.library.smu.edu.sg/sis_research/6738

Information Sciences 573 (2021) 182–193

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Secure and verifiable outsourced data dimension reduction on dynamic data

Zhenzhu Chen [a,b,c], Anmin Fu [a,b,*], Robert H. Deng [c], Ximeng Liu [c,d], Yang Yang [c,d], Yinghui Zhang [c,e]

[a] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
[b] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541010, China
[c] School of Information Systems, Singapore Management University, 178902, Singapore
[d] College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China
[e] School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

ARTICLE INFO

Article history:
Received 25 July 2019
Received in revised form 25 May 2021
Accepted 26 May 2021
Available online 28 May 2021

Keywords:
Outsourcing computation
Data privacy
Non-negative matrix factorization
Dimensionality reduction

ABSTRACT

Dimensionality reduction aims at reducing redundant information in big data and hence making data analysis more efficient. Resource-constrained enterprises or individuals often outsource this time-consuming job to the cloud for saving storage and computing resources. However, due to inadequate supervision, the privacy and security of outsourced data have been a serious concern to data owners. In this paper, we propose a privacy-preserving and verifiable outsourcing scheme for data dimension reduction, based on incremental Non-negative Matrix Factorization (NMF) method. We emphasize the importance of incremental data processing, exploiting the properties of NMF to enable data dynamics in consideration of data updating in reality. Besides, our scheme can also maintain data confidentiality and provide verifiability of the computation result. Experiment evaluation has shown that the proposed scheme achieves high efficiency, saving about more than 80% computation time for clients.

1. Introduction

In recent years, big data has attracted people's attention with the development of the Internet and information technology. Companies in all trades and fields are more interested in harnessing big data to conduct business, such as personalized recommendations, intelligent text input, and intelligence vision surveillance, etc [1,2]. However, due to the large volume and diversity, big data requires enormous storage resources. Besides, irrelevant or redundant attributes also increase the difficulty of data processing and computational consumption. Therefore, before mining valuable knowledge and hidden information, small companies with limited storage and computational capabilities are more likely to outsource data to the cloud server to perform data dimension reduction [3].

However, the exposure of original data to the cloud server has caused clients' concerns for data security and privacy [4,5]. Although the cloud server with powerful computing and storage capacity can provide a solid foundation for the rise of outsourcing computing, it is also a profit-seeking and unsupervised business [6]. More specifically, once a client outsources his data to a cloud for completing the calculation, he also loses direct control of the data [7]. On the one hand, the outsourced

* Corresponding author.
E-mail address: fuam@njust.edu.cn (A. Fu).

https://doi.org/10.1016/j.ins.2021.05.066
0020-0255/

data and final results may contain private information, such as medical records, facial features, consumer information, which has significant commercial value in targeted advertising. Therefore, the cloud server may retain the data furtively for commercial use without clients' permission [8,9]. On the other hand, computation-intensive tasks often require more computational resources. Lack of supervision, the cloud may slack off and return error results for saving costs [10]. If there is not an effective way to detect the cloud's misbehaviors, it will cause a loss for the data owner [11]. Therefore, the challenging problem is *how to enable privacy-preserving dimension reduction over mass data while ensuring that the computing results meet clients' requirements.*

Non-negative Matrix Factorization (NMF) is an important data integration algorithm in multivariate analysis. Compared with the traditional methods, such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA), NMF not only can lower data space dimensions but also retain interpretation of the original data when processing non-negative data. Specifically, traditional methods can approximate a high dimensional matrix $X \in \mathbb{R}^{n\times m}$ with two low-rank matrices $W \in \mathbb{R}^{n\times r}$ and $H \in \mathbb{R}^{r\times m}$, where $r \ll n, m$, while NMF also puts the nonnegativity constraints on both factors, which means $W \geqslant 0$ and $H \geqslant 0$. Due to its ability to automatically extract sparse and easily interpreted factors, NMF can be used for large-scale non-negative data to effectively decrease resource waste in subsequent machine learning algorithm processing, such as image processing, neural computing, speech recognition, and computational biology [12,13]. Considering the non-negative of real data, in this paper, we are interested in security issues of outsourcing incremental Non-negative Matrix Factorization to a public cloud.

Many works have been done to address data security concerns in outsourcing computing, which ranges from different fundamental mathematical functions to application-oriented tasks, such as matrix computation [14], machine learning [15–17]. In recent years, some studies [18,19] have researched the secure problems of outsourcing NMF to a public cloud. However, their proposed schemes only focus on conventional NMF, which handles static data. At the same time, big data is often thought of as a data stream, moving at very high transmission speeds. Thus, data outsourced to the cloud in reality tend to be dynamic rather than static, which means incremental data is more common in data processing [20]. When dealing with incremental data, the conventional NMF method needs to decompose historical data and newly added data, consume a large amount of computing and storage resources, and significantly reduce the method's efficiency. Considering the storage and computing overhead, clients prefer to real-time data processing to periodically collect data and discard part of the data to save service costs rather than store large amounts of data for a long time. Therefore, in this paper, we investigate secure issues faced by outsourcing incremental NMF to a public cloud. We will address two main issues. The first one is how to preserve data privacy while enabling the cloud server to perform incremental NMF. The other is how to achieve results verifiability, which protects clients from fraud.

1.1. Contributions

Focusing on the above challenges, we design a new secure dynamic data outsourcing scheme for incremental NMF. In the proposed scheme, a resource-constrained client can hand over incremental NMF computing to the cloud server, without the fear of privacy leaks and deceptive results. The main contributions are summarized below:

- We propose a secure and efficient outsourcing scheme for incremental NMF. The proposed scheme is equipped with easy-to-implement encryption and verification mechanisms, which can not only preserve data privacy but also provide verifiability of the result returned by the cloud server.
- Our scheme exploits the properties of incremental NMF to support data dynamics in the outsourcing process. To the best of our knowledge, this is the first effort on dynamic data processing in outsourced NMF.
- Extensive experiments on real-world and synthetic datasets confirm that our scheme achieves a high efficiency for clients, saving about more than 80% computation time.

1.2. Organization

The rest of the paper is organized as follows. Section 2 gives an overview of related works. Some preliminaries of this paper are briefly introduced in Section 3. Section 4 outlines the system model and security requirements. Then we go into details about our scheme in Section 5. Security analyses and performance evaluation are assessed in Section 6 and Sections 7. Finally, we conclude our paper in Section 8.

2. Related work

Data security research on outsourced computing has been conducted for a long time. At present, most of the research focuses on the design of applicability solutions for specific computing problems, which can be roughly divided into two categories: basic computing operations and application-specific computing [21].

2.1. Outsourcing basic computing operations

The outsourcing research of basic operations mainly involves that how to conduct some basic mathematical operations under encryption, such as rational number calculation, matrix operation, linear equation, and mathematical optimization. Typically, cryptography is used to convert outsourced data into encrypted ones, such as homomorphic encryption. Liu et al. [22] proposed an efficient and privacy-preserving framework for outsourcing rational number calculation, called POCR. Their scheme utilizes additive homomorphic property to achieve rational number calculation without privacy leakage. Benjamin et al. [23] designed a protocol for secure outsourcing multiplication of matrices, which is also based on additive homomorphic cryptosystem. However, due to homomorphic encryption's computational complexity, it is not a good choice for extensive data, especially high-dimensional matrices. Therefore, Wang et al. [24] later applied matrix transformation to hide data privacy, which achieves appropriate security/efficiency tradeoffs. After that, matrix transformation technique is widely used in the outsourcing matrix research, such as matrix multiplication [25], matrix inversion [26], matrix decomposition [27] and so on. Duan et al. [18] first considered secure problems in outsourcing NMF. They also employed a matrix transformation technique to construct outsourcing protocols for NMF. Later, Pan et al. [19] pointed out that their scheme has design flaws that cannot preserve data privacy and proposed a new framework for outsourcing NMF with enhanced privacy protection. However, both proposed schemes are only designed for static data and overlooked the importance of the initial value of matrix factors to the final result.

2.2. Outsourcing application-specific computing

Application-specific computing outsourcing research involves how to design outsourcing solutions for complex application scenarios, such as support vector machines (SVM), Bayesian classification, artificial neural network, and optimization algorithms [28]. Homomorphic encryption is an essential technique used to preserve data privacy. Rahulamathavan et al. [15] proposed the first outsourcing protocol of SVM, which is based on Pailler homomorphic encryption and secure two-party computation. Their protocol achieves the same classification accuracy as the ordinary non-encrypted domain. Li et al. [29] constructed a privacy-preserving outsourcing scheme for the Naive Bayes classification. Later studies focused more on neural network algorithms, which have a further wide field of application. These works put emphasis on model training security while others are more concerned with cooperation training with multi-party. For example, Li et al. [30] proposed two privacy-preserving deep learning schemes based on multi-key fully homomorphic encryption. Their schemes focus on data privacy during the model training. Phong et al. [31] also paid attention to the model security, which was guaranteed by homomorphic encryption in their proposed scheme. Esposito et al. [32] targeted at internal security issues for multiple users in collaborative deep learning and used game theory to solve the interaction problem between users and the cloud. However, currently due to the limitations of the cryptographic algorithm, works on outsourcing machine learning can achieve data confidentiality but cost large computation and communication, not practical in application.

3. Preliminaries

In this section, we will introduce the preliminaries related to our work. Before that, we first illustrate some symbols used in this paper. All processed data is presented in the form of a matrix, which is denoted by capital letters such as A. The augmented matrix is represented by $[AB]$. A matrix or vector transpose is represented by $(\cdot)^T$. We denote encrypted A by $\widetilde{A}$, and $A^{(k)}$ as the result in the k-th iteration. Besides, there is no operator sign in matrix products.

3.1. Incremental NMF

NMF is a new matrix decomposition method, which is inconspicuous by Lee and Seung's non-negative matrix research in *Nature* [33]. Similar to other matrix decomposition methods, it decomposes a high-dimensional matrix into two low-rank matrices. Specifically, given an original large non-negative matrix $X \in \mathbb{R}_+^{n\times m}$, NMF algorithm decomposes it into matrices $W \in \mathbb{R}_+^{n\times r}$ and $H \in \mathbb{R}_+^{r\times m}$, which is

$$X \approx WH. \tag{1}$$

The column vector in the original matrix X can be interpreted as a weighted sum of all column vectors in the left matrix W, called base vectors, and the weight coefficients are the corresponding column vector in the right matrix H.

Incremental NMF algorithm involves new data processing. Specifically, matrix $A \in \mathbb{R}_+^{n\times m}$ and $B \in \mathbb{R}_+^{n\times p}$ are both non-negative. Assume that A has been decomposed by NMF, i.e., $A \approx W_1H_1$, while B is the incremental part. We want to update the matrix factors after adding the incremental part B, which can be view as a new matrix $C = [AB]$. Incremental NMF algorithm will exploit the basis matrix W_1 to obtain the new basis matrix of C rather than performing NMF on C again.

As illustrated in Fig. 1, the target of Incremental NMF is to find appropriate W and H to minimize the loss function $f(W,H) = \frac{1}{2}||C - WH||^2$. The method to obtain W and H follows the idea in [34], which is described as following:

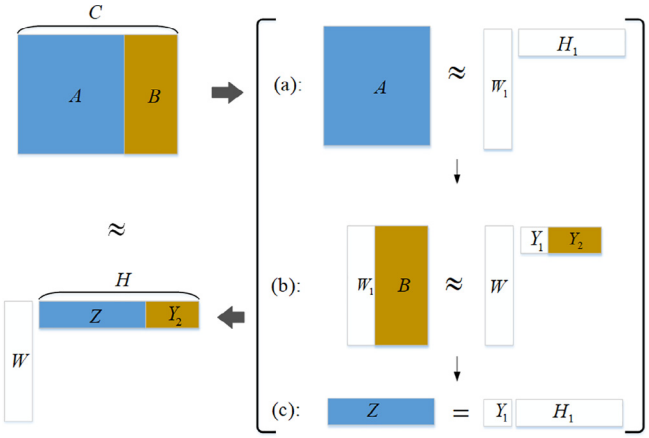

Fig. 1. Incremental NMF.

a) Let W_1 and H_1 be the matrix factors of A, which is obtained by NMF;
b) Joint the basis matrix W_1 with the increment matrix B as $D = [W_1 B]$, then perform NMF on the matrix D to obtain new basis matrix W. Denote the first r columns of the coefficient matrix as Y_1 and the last p columns as Y_2, then $D \approx W[Y_1 Y_2]$.
c) Computing new coefficient matrix of A under the new basis matrix using the transformation matrix Y_1, which is $Z = Y_1 H_1$. Then we can obtain the coefficient matrix $H = [ZY_2]$ of C when joining the matrix Z and Y_2. Finally, we get final results of incremental NMF, which is $C \approx WH$.

There are many different NMF algorithms to get two non-negative matrices [35–37]. In this paper, we present multiplication iterative update algorithm for Euclidian distance measure as follows, more details of which can be referred to in [38].

$$H \leftarrow H \frac{W^T X}{W^T W H} \tag{2}$$

$$W \leftarrow W \frac{X H^T}{W H H^T} \tag{3}$$

3.2. Matrix transformation

In this paper, we use matrix transformation technology to protect data by balancing practicality and security. In general, matrix transformation can be achieved by multiplying left or right by an invertible sparse matrix, where each row and column of the matrix have only one element. The matrix is generated based on two mathematical functions:

Kronecker delta function: is a function of two variables. Given two input numbers x and y, the output value is 1 if the two values are equal; otherwise, it is 0. Formally, a Kronecker delta function is

$$\delta_{ij} = \begin{cases} 1 & if\ x = y; \\ 0 & otherwise. \end{cases} \tag{4}$$

Permutation: is a bijective function that maps all elements of a set to other elements in the set. Denote $\varphi : S \to S$, where $S = \{s_1, s_2, \ldots, s_n\}$. φ actually rearrange the order of the elements in the set, which is written as

$$\begin{pmatrix} s_1, s_2, \ldots, s_n \\ s'_1, s'_2, \ldots, s'_n \end{pmatrix}. \tag{5}$$

where $S = \{s'_1, s'_2, \ldots, s'_n\}$. Namely, $\varphi(s_i) = s_j$. Its inverse permutation is $\varphi^{-1}(s_j) = s_i$. A permutation matrix in our scheme can be constructed by following steps:

a) Taken a parameter λ, generate a species space $\mathscr{K}$ on a non-empty finite field $\mathbb{F}$, and a random permutation φ of the integers $\{1, 2, \cdots, n\}$;
b) Select a set of non-null random number α from $\mathscr{K}$ where $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_n\}$;
c) Get a permutation matrix like $M(i,j) = \alpha_i \delta_{\varphi(i),j}$, and its inverse is $M^{-1}(i,j) = \delta_{\varphi^{-1}(j),i}/\alpha_j$.

4. Problem statement

In this section, we first formulate the problem in the architecture of our outsourced scheme, then identify adversary threats and design goals.

4.1. Problem definition

To better illustrate our scheme, we consider a scenario where a client want to perform the NMF method on a non-negative dataset, while the processed data is continuously updated. In order to save computing overhead, the client decides to outsource this task to a cloud service provider for processing. Thus, our system consists of two entities: a client *CL* and a cloud service provider *CSP*, as illustrated in Fig. 2.

Driven by some economic motivations, *CSP* may behave as a malicious party who is not only interested in inferring sensitive information from outsourced data but also tries to cheat in computation to save cost. As illustrated in Fig. 2, to preserve data privacy, *CL* will encrypt data before outsourcing to *CSP*, and carry out verification after obtaining returned results. Details of each entity are described as follows.

- *CL*: launches the outsourced scheme, leveraging cloud computing resources to solve problems. To protect data privacy, *CL* will encrypt its original data set before outsourcing and check on the validity of the results returned by the *CSP*.
- *CSP*: provides *CL* with unlimited storage space and massive computing power. Upon receiving computing task, *CSP* performs the calculation as requested by *CL* and returns the result. Actually, driven by some economic advantages, *CSP* may behave as a malicious attacker which not only steals private data but also compromises the computational integrity.

4.2. Adversary threats

In this paper, we assume that *CSP* is malicious. A malicious *CSP* can be viewed as an active adversary $\mathscr{A}$. Threats from $\mathscr{A}$ are as follows:

- $\mathscr{A}$ can access all ciphertext data sent by *CL*, and is interested in obtaining all plaintexts, including inputs and final results.
- There is no way to ask $\mathscr{A}$ to expose computing details. Thus, $\mathscr{A}$ can skip necessary calculations and fake final results to save computing cost.

In this system, we only focus on secure challenges issued by *CSP*. External attacks, such as attacking *CL* directly, refer to hardware and system security, which is out of the scope of this work. Besides, the communication channels are reliable in our system.

4.3. Design goals

Considering the mentioned threats, we present three main design goals essential for outsourced computing in this paper.

- **Correctness.** As long as *CL* and *CSP* both follow the computing steps correctly can the final results obtained by *CL* be correct.
- **Input/Output Privacy.** The data outsourced by *CL* usually contains a large amount of private information. When uploaded to *CSP*, *CL* lose control of the data, but there is no guarantee that the sensitive information will not be abused by *CSP*. Therefore, *CL* needs to encrypt the data to ensure that *CSP* cannot get any sensitive information from the outsourced data. In addition, *CSP* should not have access to any information about the execution results, including intermediate and final results.

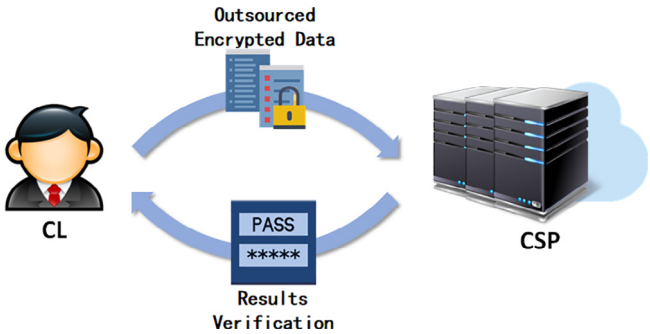

Fig. 2. Architecture of our outsourced scheme.

- **Cheating Resistance.** Ensuring that *CSP* returns the correct results is one of the most important aspects of outsourced computing services. If *CSP* returns the correct result, the validation passes. Conversely, the likelihood of any erroneous outcome passing validation is negligible.

5. Proposed outsourcing scheme

In this section, we construct a privacy-preserving scheme for incremental NMF to process dynamic data.

5.1. System architecture

Assuming that a matrix A composed of m samples has been decomposed by NMF algorithm, *CL* already gets the matrix factors W_1 and H_1. Suppose *CL* gathers new dataset B after outsourcing A to *CSP* for NMF, and he wants to update the base matrix W_1. Thus, we can construct the new processed data $[W_1 B]$ according to the incremental NMF algorithm. The overall procedure is depicted in Fig. 3 and the simplified overview is outlined below, which mainly consists of the following four phases: *Preparing*, *Data uploading*, *Computing* and *Results Verification*.

- *Preparing*. This phase contains two steps. First, the client *CL* outsources encrypted matrix $\widetilde{A}$ to *CSP* and obtains the final results W_1 and H_1. Next, *CL* generates some secret key matrices for the incremental matrix B and other data to protect privacy.
- *Data uploading*. In this phase, *CL* encrypts all inputs into ciphertexts, then uploads them to *CSP* to process.
- *Computing*. This phase is conducted on *CSP* side. When receiving the encrypted data, *CSP* runs the NMF algorithm to obtain the encrypted output and sends them to *CL*.
- *Results Verification*. In this phase, upon receiving returned results, *CL* conducts verification mechanism to detect its correctness and decrypts results to obtain the plaintext result if the verification is positive; otherwise it rejects them.

5.2. Detailed description

In this section, we will present details of these four phases.

5.2.1. Preparing

In this phase, *CL* obtains matrix factors W_1 and H_1 of matrix A (Step 1) and prepares some secret key matrices to protect data privacy (Step 2).

Step 1. To outsource data A, *CL* generates three permutation matrices P, Q and R and keeps them as secret encryption/decryption keys $\mathscr{SK} = \{P, Q, R\}$. Then choosing an appropriate r, *CL* initializes $W_1^{(0)} \geqslant 0, H_1^{(0)} \geqslant 0$ and transforms data into encrypted form, which are

a. $\widetilde{A} \longleftarrow PAQ^{-1}$
b. $\widetilde{W}_1^{(0)} \longleftarrow PW_1^{(0)}R^{-1}$
c. $\widetilde{H}_1^{(0)} \longleftarrow RH_1^{(0)}Q^{-1}$.

CL sends these encrypted data to *CSP* for NMF, and obtains final results $\widetilde{W_1}$ and $\widetilde{H_1}$. Because our scheme focus on incremental NMF outsourcing rather than NMF, we skip verification in this step and assume that $\widetilde{W_1}$ and $\widetilde{H_1}$ is correct. Note that if *CL* keeps sending incremental data to *CSP* for process, this step can be skipped.

Step 2. CL constructs three new permutation matrices S and T and adds them into the secret encryption/decryption keys $\mathscr{SK}$. Then *CL* initializes $W^{(0)} \geqslant 0, H^{(0)} \geqslant 0$. To be clear, $W^{(0)} \in \mathbb{R}_+^{n \times r}$ and $H^{(0)} \in \mathbb{R}_+^{r \times (r+p)}$. We also denote the first r columns of $H^{(0)}$ as $Y_1^{(0)}$ and the last p columns as $Y_2^{(0)}$.

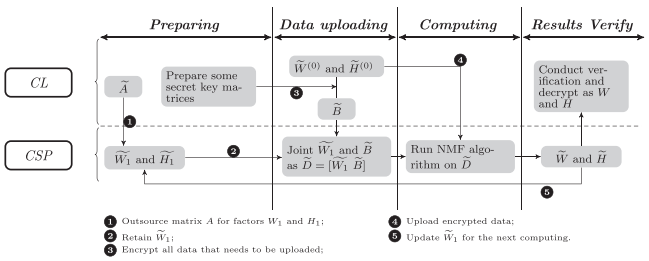

Fig. 3. The overall procedure of our scheme.

5.2.2. Data uploading

After *Preparing*, *CL* first encrypts all data that needs to be send to *CSP* using $\mathscr{SK}$, including

a. $\widetilde{B} \longleftarrow PBS^{-1}$
b. $\widetilde{W}^{(0)} \longleftarrow PW^{(0)}T^{-1}$
c. $\widetilde{Y_1}^{(0)} \longleftarrow TY_1^{(0)}R^{-1}$
d. $\widetilde{Y_2}^{(0)} \longleftarrow TY_2^{(0)}S^{-1}$

Next, before sending to *CSP*, *CL* needs to join the block matrices into the augmented matrix:

a. $\widetilde{H}^{(0)} \longleftarrow \left[\widetilde{Y_1}^{(0)}\widetilde{Y_2}^{(0)}\right]$
b. $U^{-1} \longleftarrow \begin{bmatrix} Q^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix}$

Then, *CL* sends $\widetilde{B}, \widetilde{W}^{(0)}, \widetilde{H}^{(0)}$ to *CSP*.

5.2.3. Computing

This phase takes place in *CSP*. Because *CSP* keeps $\widetilde{W_1}$ in case *CL* sends new data for process. Thus, after receiving encrypted data from *CL*, *CSP* joints $\widetilde{W_1}$ and $\widetilde{B}$ as $\widetilde{D} = \left[\widetilde{W_1}\widetilde{B}\right]$. Then *CSP* will runs NMF algorithm to decompose the matrix $\widetilde{D}$ with $\widetilde{W}^{(0)}, \widetilde{H}^{(0)}$, and obtain optimal solution $\widetilde{W}, \widetilde{H}$ and return the final results to *CL*. Besides, *CSP* keeps $\widetilde{W}$ for another incremental data processing.

5.2.4. Results verification

To verify the correctness of the results, we put forward an algorithm which incorporates the stop condition in ANLS method [36]. Generally, $\widetilde{W}^{(k)} = \widetilde{W}^{(k-1)}$ and $\widetilde{H}^{(k)} = \widetilde{H}^{(k-1)}$ imply the solution are found. However, due to the particularity of iterative method, a dishonest *CSP* may reply the result of the previous $(k-1)$-th iteration for the k-th iteration to save computation cost. This misbehaviour cannot be detected by only checking whether $\widetilde{W}^{(k)} = \widetilde{W}^{(k-1)}$ and $\widetilde{H}^{(k)} = \widetilde{H}^{(k-1)}$ are hold. Besides, $\|\widetilde{D} - \widetilde{W}\widetilde{H}\| < \epsilon$is often used to detect forged results, but this condition does not reveal whether a solution is close to a stationary point. It fails to detect results that are not fully calculated. This defect can be rectified by the stop condition in ANLS method, which is

$$\|\nabla \bar{f}\left(\widetilde{W}, \widetilde{H}\right)\|_F \leqslant \epsilon \|\nabla \bar{f}\left(\widetilde{W}^{(0)}, \widetilde{H}^{(0)}\right)\|_F \tag{6}$$

where $\widetilde{W}$ and $\widetilde{H}$ are the final solutions. Therefore, our verification steps can be concluded as shown in **Algorithm 1**.

Algorithm 1: Results Verification

```
while receiving W̃ and H̃ do
    if ‖∇f̄(W̃, H̃)‖_F ≤ ε‖∇f̄(W̃^(0), H̃^(0))‖_F then
        Verification pass;
        Get the first r columns of H̃ as Ỹ_1;
        Get the last p columns of H̃ as Ỹ_2;
        Compute Z̃ ⟵ Ỹ_1 H̃_1;
        Joint H̃ ⟵ [Z̃ Ỹ_2];
        Compute W ⟵ P^{-1} W̃ T;
        Compute H ⟵ T^{-1} H̃ U;
    else
        reject results.
    end
end
```

6. Analysis on the proposed scheme

In this section, we will show that our proposed scheme has achieved all design goals with the theoretical analysis.

6.1. Correctness guarantee

Theorem 1. *In our scheme, as long as CL and CSP both follow the scheme honestly, the final results W, H satisfy $C \approx WH$.*

Proof. When *CSP* runs NMF algorithm to decompose the matrix $\widetilde{D}$ amd obtain $\widetilde{W}$ and $\left[\widetilde{Y_1}\widetilde{Y_2}\right]$, we have

$$\left[\widetilde{W_1}\widetilde{B}\right] \approx \widetilde{W}\left[\widetilde{Y_1}\widetilde{Y_2}\right]. \tag{7}$$

Thus,

$$\widetilde{W_1} = \widetilde{W}\widetilde{Y_1}. \tag{8}$$

Since $\widetilde{A} \approx \widetilde{W_1}\widetilde{H_1}$, we have

$$\widetilde{A} \approx \widetilde{W}\widetilde{Y_1}\widetilde{H_1}, \tag{9}$$

which implies

$$\left[\widetilde{A}\widetilde{B}\right] \approx \widetilde{W}\left[\widetilde{Y_1}\widetilde{H_1}\widetilde{Y_2}\right] = \widetilde{W}\left[\widetilde{Z}\widetilde{Y_2}\right] = \widetilde{W}\widetilde{H}. \tag{10}$$

After decryption, we can obtain

$$C \approx P^{-1}\widetilde{W}TT^{-1}\widetilde{H}U = WH \qquad \square \tag{11}$$

6.2. Privacy preserving

Definition 1. Let $f(x,y) = (f_{CL}(x,y); f_{CSP}(x,y))$ be the protocol outputs of *CL* and *CSP* who run the protocol π, respectively, where x is the private input of *CL* and y is the private input of *CSP*. The view of *CL* during an execution of π is denoted by $\text{VIEW}^{\pi}_{CL}(x; m^{CL}_1, \ldots, m^{CL}_t)$ is and that of *CSP* is $\text{VIEW}^{\pi}_{CSP}(y;\ m^{CSP}_1, \ldots, m^{CSP}_l)$, where m_j represents the jth message *CL* or *CSP* received. We say that the protocol π is securely computes f against semi-trusted adversaries if there exist polynomial time simulators $\mathscr{S}_{CL}$ and $\mathscr{S}_{CSP}$ such that

$$\mathscr{S}_{CL}(x; f_{CL}(x,y)) \equiv \text{VIEW}^{\pi}_{CL}(12)$$
$$\mathscr{S}_{CSP}(x; f_{CSP}(x,y)) \equiv \text{VIEW}^{\pi}_{CSP}(13)$$

where the symbol $\equiv$ represents computational indistinguishability.

Theorem 2. *(Input/Output Privacy) Our scheme is secure in the semi-trusted model.*

Proof. We present the security proof of *Data uploading* and *Computing* due to page limitation. Security proof for *Preparing* can be obtained similarly, while security proof for *Results Verification* is presented in Theorem 3.

There is non-interactive between *CL* and *CSP* when *CL* encrypts all data and *CSP* joints $\widetilde{W_1}$ and $\widetilde{B}$, and matrix encryption is secure under brute-force attack (the probability is $\frac{1}{|\kappa|^{n\|r\|}}$. Then we first create simulator for *CL*. The view of *CL* is $VIEW_{CL} = \left(\left(B, W^{(0)}, H^{(0)}\right), m^{CL}_1\right)$, where m^{CL}_1 is the message received from *CSP* in Computing. To simulates the view VIEW_{CL} of *CL*, we need to construct the simulator $\mathscr{S}_{CL}\left(\left(\widetilde{B}, \widetilde{W}^{(0)}, \widetilde{H}^{(0)}\right); \left(\widetilde{W}, \widetilde{H}\right)\right)$. According to Theorem 1, $\mathscr{S}_{CL}$ can construct the simulator to simulate m^{CL}_1. Therefore, $\mathscr{S}_{CL} \equiv \text{VIEW}_{CL}$. Similarly, the view of *CSP* is $\text{VIEW}_{CSP} = \left(\widetilde{W_1}; m^{CSP}_1\right)$ wherem^{CSP}_1 is the message received from *CSP* in Data uploading. To simulates the view $\text{VIEW}_{CSP}$of *CL*, we need to construct the simulator $\mathscr{S}_{CSP}\left(\widetilde{W_1}; \left(\widetilde{B}, \widetilde{W}^{(0)}, \widetilde{H}^{(0)}\right)\right)$. Because the output $f_{CSP}(x,y)$ is $\left(\widetilde{W}, \widetilde{H}\right)$, $\mathscr{S}_{CSP}$ can construct the simulator to simulate m^{CSP}_1, which means $\mathscr{S}_{CSP} \equiv \text{VIEW}_{CSP}$. According to Definition 1 and the above analysis, we prove that our scheme is secure in the presence of semi-honest adversaries. $\square$

6.3. Cheating resistance

Definition 2. Let $f(x,y)=(f_{CL}(x,y);f_{CSP}(x,y))$ be the protocol outputs of *CL* and *CSP* who run the protocol π, respectively, where x is the private input of *CL* and y is the private input of *CSP*. Let $\mathscr{A}$ be a non-uniform probabilistic polynomial-time adversary (control *CSP*) for the real world, and the real execution of π is denoted by $\mathrm{REAL}_{\mathscr{A}(Z)}(x,y)$. Let $\mathscr{S}$ be a non-uniform probabilistic polynomial-time adversary for the ideal world, and the ideal execution of π is denoted by $\mathrm{IDEAL}_{\mathscr{S}(Z)}(x,y)$. We say that the protocol π is securely computes f against malicious adversaries (*CSP*) if there exist a non-uniform probabilistic polynomial-time adversary $\mathscr{S}$ such that for *CSP*,

$$\mathrm{IDEAL}_{f,\ \mathscr{S}(Z),CSP}(x,y)_{x,y}\equiv \mathrm{REAL}_{\pi,\mathscr{A}(Z),CSP}(x,y)_{x,y} \tag{14}$$

Theorem 3. *(Cheating resistance) Our scheme is secure in the malicious model.*

Proof. Due to *CSP* is the malicious party in our scheme, we focus on the security proof of *Results Verification* (Algorithm 1), which is related to the malicious behavior of *CSP*.

Suppose *CSP* is corrupted by $\mathscr{A}$. According to the stop condition in ANLS method, the output of the real world is continuing when $\left(\widetilde{W},\widetilde{H}\right)$ is correct and reject when $\mathscr{A}$ chose a valid result $\left(\widetilde{W},\widetilde{H}\right)$ to reply *CL*. In the ideal world, there is a trusted party. *CL* sends $\widetilde{W^{(0)}},\widetilde{H^{(0)}}$ to the trusted party. Assume there is a non-uniform probabilistic polynomial-time adversary $\mathscr{S}$ invokes $\mathscr{A}$ and internally receives the message $\left(\widetilde{W},\widetilde{H}\right)$ that $\mathscr{A}$ sends to *CL*. $\mathscr{S}$ to the trusted party $\left(\widetilde{W},\widetilde{H}\right)$. If $\mathscr{A}$ sends correct $\left(\widetilde{W},\widetilde{H}\right)$, the trusted party computes $f\left(\left(\widetilde{W}^{(0)},\widetilde{H}^{(0)}\right),\left(\widetilde{W},\widetilde{H}\right)\right)=\|\nabla\bar{f}\left(\widetilde{W}\widetilde{H}\right)\|_F\leqslant\epsilon\|\nabla\bar{f}\left(\widetilde{W}^{(0)}\widetilde{H}^{(0)}\right)\|_F$, and outputs continue to *CL* and $\mathscr{S}$; If $\mathscr{A}$ chose a valid result $\left(\widetilde{W},\widetilde{H}\right)$ to reply, the trusted party outputs reject to *CL* and $\mathscr{S}$. Thus,

$$\begin{aligned}&\mathrm{IDEAL}_{f,\ \mathscr{S},CSP}\left(\left(\widetilde{W}^{(0)},\widetilde{H}^{(0)}\right),\left(\widetilde{W},\widetilde{H}\right)\right)_{\left(\widetilde{W}^{(0)},\widetilde{H}^{(0)}\right),\left(\widetilde{W},\widetilde{H}\right)}\\&\equiv\mathrm{REAL}_{\pi,\mathscr{A},CSP}\left(\left(\widetilde{W}^{(0)},\widetilde{H}^{(0)}\right),\left(\widetilde{W},\widetilde{H}\right)\right)_{\left(\widetilde{W}^{(0)},\widetilde{H}^{(0)}\right),\left(\widetilde{W},\widetilde{H}\right)}\end{aligned} \tag{15}$$

□

7. Performance evaluation

In this section, we present experimental results to show the practicability and efficiency of our new scheme. Our experiment environment includes *CL* side and *CSP* side. *CL* side is conducted on a laptop with an Intel Core i7-8565U 1.8 GHz CPU and 8 GB memory, while *CSP* side is conducted on a computer with an Intel Core i7-4790 3.6 GHz CPU and 16 GB memory.

7.1. Practicability analysis

As discussed before, our scheme can preserve the privacy of *CL*'s outsourced data. To illustrate this fact more intuitively, we first conduct the experiment on AT&T database [39] to evaluate security performance, using Python programming language.

AT&T database contains 400 images, and each of them has 92×112 pixels. Nine images are randomly selected to display in Fig. 4(a). According to our scheme, *CL* encrypts these images before outsourcing. After receiving returned results, *CL* can also recover images by *WH*. We choose half the images as experimental data and another half as the incremental data to conduct our scheme. We use $\widetilde{W}\widetilde{H}$ as the encrypted images, which is shown in Fig. 4(a) and the corresponding encrypted and reconstructed images of those 9 images are and Fig. 4(c). It is quite clear that compared to the original images, images in *CSP* are totally masked after encryption, which indicates that the data privacy is well preserved.

Comparing Fig. 4(a) and (c), it is observed that the overall structure and appearance of the faces are well maintained, which also indicates that *CSP* can obtain correct solutions when performing NMF on encrypted data.

7.2. Efficiency analysis

As presented in the scheme, the data matrix is $n\times m$, and the incremental matrix is $n\times p$, while the dimension of the basis matrix W is r. The maximum size of data need to be processed using NMF is $n\times(m+p)$, while that for incremental NMF is $n\times(r+p)$. Considering that r is much less than n and m, it saves considerable storage resources during operation. This superiority also can be observed in computing cost.

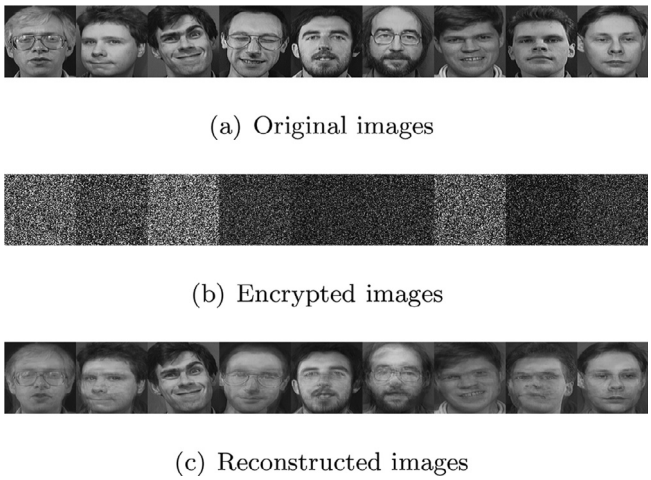

Fig. 4. Randomly selected 9 images of dataset.

In order to confirm this claim, we conduct simulation experiments and compare the time cost of our scheme and Pan's scheme [19] with the different number of samples. We set $m = 5000$ and increase sample size p. To be clear, the sample size in Pan's scheme is equal to the sample size $m + p$ in our scheme. The comparison of time cost with $n = 5000, 8000, 10,000$ is depicted in Fig. 5. The comparison results suggest that compared to the computation cost of Pan's scheme, that of ours is much less no matter how big n is. Especially, with the increase of data size, the superiority is even more apparent.

Additionally, in our scheme, the most time-consuming phase is *Computing*. Fortunately, this phase occurs in *CSP* side. Thus, *CL* only needs to conduct some simple computation. To demonstrate clearly that *CL* can save computing overhead

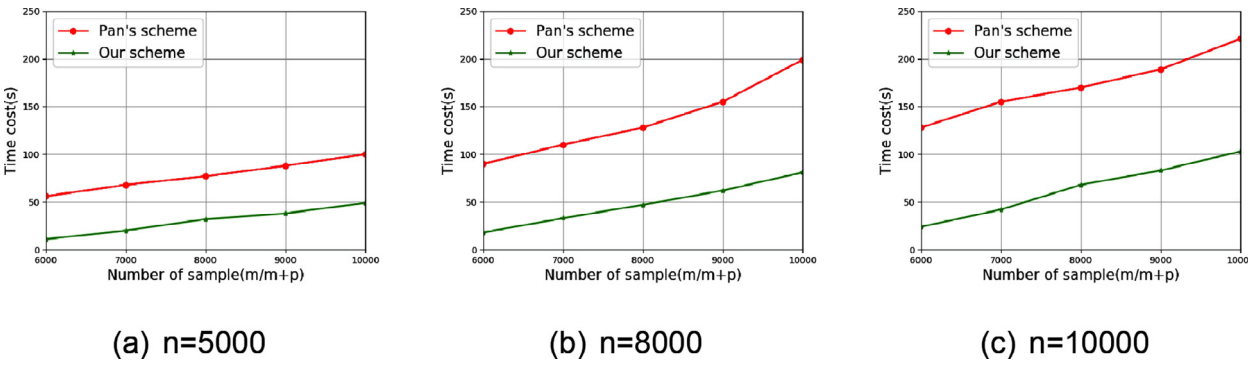

Fig. 5. Time cost Comparison with different n.

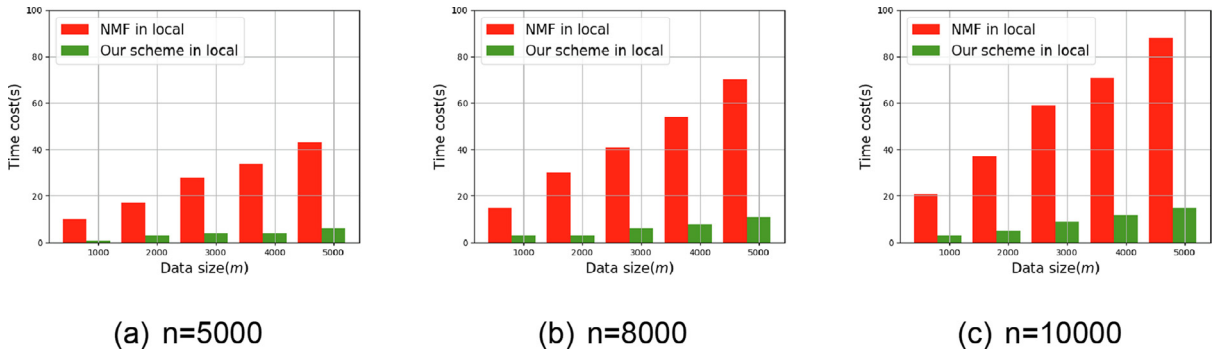

Fig. 6. Time cost Comparison in local.

Table 1
Time cost for CL computation comparison.

Data dimension ($m = 5000$)	$n = 5000$			$n = 8000$			$n = 10,000$		
	NMF in local (s)	Ours (s)	Cost saving	NMF in local (s)	Ours (s)	Cost saving	NMF in local (s)	Ours (s)	Cost saving
$p = 1000$	10.21	1.12	89.03%	15.41	3.34	78.33%	21.21	3.47	83.64%
$p = 2000$	17.64	3.02	82.87%	30.11	3.50	88.38%	37.62	5.23	86.10%
$p = 3000$	28.19	4.33	84.64%	41.86	6.78	83.80%	59.43	9.41	84.17%
$p = 4000$	37.33	4.67	87.49%	54.23	8.21	84.86%	71.35	12.26	82.82%
$p = 5000$	43.89	6.55	85.08%	70.26	11.43	83.73%	88.89	15.76	82.27%

when outsourced data to *CSP*, we compare the computation *CL* solving NMF problem in local and the computation *CL* spend in outsourcing. We also set $m = 5000$ and increase sample size p. In order to have a better presentation of the efficiency, we depict the results with a histogram in Fig. 6. Numerical results are also presented in Table 1. As expected, the computation cost for *CL* in our scheme is much less than that of *CL* solving NMF problem locally, which can save about more than 80% computation time. Significantly, the dominance is more evident with the increase of p and n, which indicates that it can save lots of time and computational resources for *CL*, especially when the data gets to the thousands or even millions.

8. Conclusion

Privacy-preserving outsourcing of data dimension reduction can help resource-constrained clients to use cloud resources for data integration without exposing data privacy. Aiming at the dynamic characteristics of data in reality, in this paper, we proposed a secure and efficient outsourcing scheme for data dimensions reduction, based on incremental NMF. Exploiting the properties of NMF, our scheme is capable of supporting dynamic data processing while maintaining data privacy and resisting cheating by malicious *CSP*. Without employing any cryptography algorithms, our scheme also achieves a high efficiency for *CL*.

CRediT authorship contribution statement

Zhenzhu Chen: Conceptualization, Methodology, Investigation, Writing - original draft. **Anmin Fu:** Supervision, Project administration, Funding acquisition. **Robert H. Deng:** Supervision, Resources, Writing - review & editing. **Ximeng Liu:** Validation, Writing - review & editing. **Yang Yang:** Formal analysis, Writing - review & editing. **Yinghui Zhang:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by National Natural Science Foundation of China (62072239, 61872091, 62072369, 62072109, U1804263), the Guangxi Key Laboratory of Trusted Software (KX202029), the Innovation Capability Support Program of Shaanxi (2020KJXX-052) and the Open Fund Project of Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University) (MJUKF-IPIC201908).

References

[1] L. Zhou, A. Fu, S. Yu, M. Su, B. Kuang, Data integrity verification of the outsourced big data in the cloud environment: a survey, J. Netw. Comput. Appl. 122 (2018) 1–15.
[2] L. Zhou, A. Fu, G. Yang, H. Wang, Y. Zhang, Efficient certificateless multi-copy integrity auditing scheme supporting data dynamics, IEEE Trans. Dependable Secure Comput. (2020), https://doi.org/10.1109/TDSC.2020.3013927, 1–1.
[3] M. Su, B. Zhou, A. Fu, Y. Yu, G. Zhang, PRTA: a proxy re-encryption based trusted authorization scheme for nodes on cloudiot, Inf. Sci. .
[4] Y. Zhang, R.H. Deng, X. Liu, D. Zheng, Blockchain based efficient and robust fair payment for outsourcing services in cloud computing, Inf. Sci. 462 (2018) 262–277.
[5] Z. Chen, A. Fu, Y. Zhang, Z. Liu, R.H. Deng, Secure collaborative deep learning against gan attacks in the internet of things, IEEE Inter. Things J. (2020), https://doi.org/10.1109/JIOT.2020.3033171, 1–1.
[6] Y. Yang, X. Zheng, W. Guo, X. Liu, V. Chang, Privacy-preserving fusion of iot and big data for e-health, Future Gen. Comput. Syst. 86 (2018) 1437–1455.
[7] C. Wang, K. Ren, J. Wang, Q. Wang, Harnessing the cloud for securely outsourcing large-scale systems of linear equations, IEEE Trans. Parallel Distrib. Syst. 24 (6) (2013) 1172–1181.
[8] A. Fu, Y. Zhu, G. Yang, S. Yu, Y. Yu, Secure outsourcing algorithms of modular exponentiations with optimal checkability based on a single untrusted cloud server, Cluster Comput. 21 (4) (2018) 1933–1947.

[9] X.D. Wang, W.Z. Meng, Y. Liu, Lightweight privacy-preserving data aggregation protocol against internal attacks in smart grid, J. Inform. Security Appl. (2020) 1, https://doi.org/10.1016/j.jisa.2020.102628.
[10] P. Li, J. Li, Z. Huang, C.-Z. Gao, W.-B. Chen, K. Chen, Privacy-preserving outsourced classification in cloud computing, Cluster Comput. 21 (1) (2018) 277–286.
[11] Y. Yang, X. Liu, R. Deng, Expressive query over outsourced encrypted data, Inf. Sci. 442 (2018) 33–53.
[12] Y. Wang, Y. Zhang, Nonnegative matrix factorization: a comprehensive review, IEEE Trans. Knowl. Data Eng. 25 (6) (2013) 1336–1353.
[13] Y.N. Liu, Q. Zhong, M. Xie, Z.B. Chen, A novel multiple-level secret image sharing scheme, Multimedia Tools Appl. 77 (5) (2018) 6017–6031.
[14] Z. Chen, A. Fu, K. Xiao, M. Su, Y. Yu, Y. Wang, Secure and verifiable outsourcing of large-scale matrix inversion without precondition in cloud computing, in: Proceedings of 2018 IEEE International Conference on Communications (ICC) IEEE, 2018, pp. 1–6.
[15] Y. Rahulamathavan, R.C.-W. Phan, S. Veluru, K. Cumanan, M. Rajarajan, Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud, IEEE Trans. Dependable Secure Comput. 11 (5) (2014) 467–479.
[16] M. Kim, J. Lee, L. Ohno-Machado, X. Jiang, Secure and differentially private logistic regression for horizontally distributed data, IEEE Trans. Inform. Foren. Sec.https://doi.org/10.1109/TIFS.2019.2925496. .
[17] A. Fu, X. Zhang, N. Xiong, Y. Gao, H. Wang, VFL: a verifiable federated learning with privacy-preserving for big data in industrial IoT, IEEE Trans. Ind. Inf. (2020) 1, https://doi.org/10.1109/TII.2020.3036166.
[18] J. Duan, J. Zhou, Y. Li, Secure and verifiable outsourcing of nonnegative matrix factorization (NMF), in: Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security (IH& MMSec), ACM, 2016, pp. 63–68.
[19] S. Pan, F. Zheng, W.-T. Zhu, Q. Wang, Harnessing the cloud for secure and efficient outsourcing of non-negative matrix factorization, in: Proceedings of 2018 IEEE Conference on Communications and Network Security (CNS), IEEE, 2018, pp. 1–9.
[20] A. Fu, Y. Li, S. Yu, Y. Yu, G. Zhang, DIPOR: an ida-based dynamic proof of retrievability scheme for cloud storage systems, J. Netw. Comput. Appl. 104 (2018) 97–106.
[21] Z. Shan, K. Ren, M. Blanton, C. Wang, Practical secure computation outsourcing: a survey, ACM Comput. Surv. 51 (2) (2018) 31.
[22] X. Liu, K.-K.R. Choo, R.H. Deng, R. Lu, J. Weng, Efficient and privacy-preserving outsourced calculation of rational numbers, IEEE Trans. Dependable Secure Comput. 15 (1) (2016) 27–39.
[23] D. Benjamin, M.J. Atallah, Private and cheating-free outsourcing of algebraic computations, in: Proceedings of 2008 Sixth Annual Conference on Privacy, Security and Trust (PST), IEEE, 2008, pp. 240–245.
[24] C. Wang, K. Ren, J. Wang, Secure and practical outsourcing of linear programming in cloud computing, in, in: Proceedings of 2011 IEEE International Conference on Computer Communications (INFOCOM), IEEE, 2011, pp. 820–828.
[25] M. Nassar, A. Erradi, Q.M. Malluhi, Practical and secure outsourcing of matrix computations to the cloud, in, in: Proceedings of 2013 IEEE International Conference on Distributed Computing Systems Workshops (ICDCS), IEEE, 2013, pp. 70–75.
[26] C. Hu, A. Alhothaily, A. Alrawais, X. Cheng, C. Sturtivant, H. Liu, A secure and verifiable outsourcing scheme for matrix inverse computation, in: Proceedings of 2017 IEEE International Conference on Computer Communications (INFOCOM), IEEE, 2017, pp. 1–9.
[27] L. Zhou, C. Li, Outsourcing eigen-decomposition and singular value decomposition of large matrix to a public cloud, IEEE Access 4 (2016) 869–879.
[28] C. Zhang, M. Ahmad, Y. Wang, ADMM based privacy-preserving decentralized optimization, IEEE Trans. Inf. Forensics Secur. 14 (3) (2019) 565–580.
[29] T. Li, J. Li, Z. Liu, P. Li, C. Jia, Differentially private naive bayes learning over multiple data sources, Inf. Sci. 444 (2018) 89–104.
[30] P. Li, J. Li, Z. Huang, T. Li, C. Gao, S. Yiu, K. Chen, Multi-key privacy-preserving deep learning in cloud computing, Future Gener. Comput. Syst. 74 (2017) 76–85.
[31] L.T. Phong, Y. Aono, T. Hayashi, L. Wang, S. Moriai, Privacy-preserving deep learning via additively homomorphic encryption, IEEE Trans. Inf. Forensics Secur. 13 (5) (2018) 1333–1345.
[32] C. Esposito, X. Su, S.A. Aljawarneh, C. Choi, Securing collaborative deep learning in industrial applications within adversarial scenarios, IEEE Trans. Ind. Inf. 14 (11) (2018) 4972–4981.
[33] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788.
[34] L. Guo, S. Zhang, W. Wang, B. Shi, Incremental non-negative matrix factorization algorithm, Comput. Eng. 36 (2010) 66–68.
[35] E.F. Gonzalez, Y. Zhang, Accelerating the lee-seung algorithm for nonnegative matrix factorization, Tech. rep. (2005). .
[36] C. Lin, Projected gradient methods for nonnegative matrix factorization, Neural Comput. 19 (10) (2007) 2756–2779.
[37] S.Z. Li, X. Hou, H. Zhang, Q. Cheng, Learning spatially localized, parts-based representation, in, in: Proceedings of 2001 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2001, pp. 207–212.
[38] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing Systems, 2000, pp. 556–562.
[39] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: Proceedings of 1994 IEEE Workshop on Applications of Computer Vision (WACV) , IEEE, 1994, pp. 138–142.