Federated Multi-View Spectral Clustering

HONGTAO WANG1,2, ANG LI1, BOLIN SHEN1, YUYAN SUN3, AND HONGMEI WANG1

1School of Control and Computer Engineering, North China Electric Power University, Baoding 071051, China
2Engineering Research Center of Intelligent Computing for Complex Energy Systems, Ministry of Education, North China Electric Power University, Baoding 071051, China
3Beijing Key Laboratory of IoT Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

Corresponding author: Hongmei Wang (wanghm@ncepu.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61802124, in part by the National Key Research and Development Program of China under Grant 2017YFB0802805, and in part by the Fundamental Research Funds for the Central Universities under Grant 2019MS126.

ABSTRACT Multi-view spectral clustering (MVSC) has become a popular approach to harvest knowledge about group information from multiple views of data, owned by different parties. A high quality MVSC approach usually requires collecting massive amount of data from each view party, in order to perform MVSC algorithm in a centralized manner. However, such centralized MVSC approach raises serious privacy concerns, not only in terms of the sensitivity property of many real-world data such as medical or financial records, but also in terms of the regulations from authorities to preclude centralized operations. Hence it is crucial to design new paradigm for training spectral clustering model on multi-view data in industrial scenarios. In this article, we propose a distributed and secure framework named Federated Multi-view Spectral Clustering (FMSC), in which a group of view parties collaboratively perform a MVSC model, but couldn’t learn the data of other participants. FMSC is inspired by the concept of federated learning, and utilize Homomorphic Encryption (HE) and Differential Privacy (DP) to achieve secure and private clustering. We conduct a series of extensive experiments to verify the effectiveness of FMSC on both synthetic and real-world datasets. Evaluations show that FMSC achieves respectable clustering results over conventional centralized approaches.

INDEX TERMS Multi-view clustering, spectral clustering, distributed learning, federated learning, differential privacy, honest-but-curious, perturbation.

I. INTRODUCTION

Multi-view clustering has been receiving considerable attentions in recent years. The reason comes from the fact that objects are usually presented in different views. These views, held by same or different parties, often contain partial information of different domains of features associated with the same objects. For example, a webpage can be organized by images, texts and audios. In the real world, human activity may be captured by street cameras, GPS sensors, transportation card records, tweets on tweeter, etc. On the scenario of Internet of Things (IoT), multi-view data is more likely observed, collected and even stored by local edge devices.

Performing clustering algorithms on multi-view data usually requires centralizing these data and exploiting all features of diverse views. Unlike traditional single-view clustering (SVC) approaches to concatenate all views of features together, multi-view clustering (MVC) aims to cluster data directly from diverse views. Compared to SVC, MVC can leverage more structural and complementary information hidden in multi-view data to improve the clustering performance [1]. Existing MVC approaches can be categorized into two classes, namely, projection-based approaches and similarity-based approaches. Projection-based approaches [2]–[4] project each view with a compatible matrix into a common low-dimensional feature subspace, and then apply a standard clustering method (e.g., k-means) in the projected subspace. More extensive studies have focused on similarity-based approaches [5]–[7], which learn a clustering structure for different views by considering both within-view and between-view similarities. Among these approaches, multi-view spectral clustering (MVSC) [7]–[11] has earned a lot of interests, and has shown encouraging results in comparison to others. These pioneering approaches demonstrate the promise of multi-view clustering task, but suffer from a major limitation:
multi-view data should be collected to a central server to perform the clustering algorithms, without considering the possible leakage of privacy from sensitive data owned by each view party. In fact, in many applications multi-view data are collected on distributed parties, each with one view data [12], [13]. For example, a user may have credit accounts in a number of banks. User’s financial activities in different banks are very sensitive to be collected into a server to get a more robust clustering model. In recent years, several regulations have been proposed to prohibit privacy leakage in commercial scenarios, such as European Union General Data Protection Regulation (GDPR) [14]. Thus, these approaches raise a critical question: how to derive a global multi-view spectral clustering structure in each view-data party if the view specific data are located on diverse distributed parties? To mitigate this issue, we should keep the following regulations which are very challenging:

1) No party should share its private view data to others, including, but not limited to other multi-view data owners and the central server.
2) To minimize information leakage, sensitive local parameters should also be impeded to directly share to others without any privacy preserving mechanism.
3) The performance of model proposed shouldn’t degenerate too much to the model trained in a centralized manner.

Inspired by the growing popularity of Federated Learning [15], which investigates a distributed architecture that a group of view data parties jointly train a machine learning model and nobody can learn the data of other participants, we propose a new approach named Federated Multi-view Spectral Clustering (FMSC) that addresses the aforementioned challenges in a principled way. Our approach has several contributions as follows.

- FMSC exploits a central server only to compute aggregated statistics of sensitive view specific parameters but in a secure manner: each party can encrypt individual parameters, namely, the discriminative eigenvectors of each view data, with Homomorphic Encryption. Every party receives and decrypts the encrypted aggregation of global eigenvectors, and then computes its local clustering structure at each iteration until convergence.
- We also utilize a Differential Privacy (DP) mechanism to prevent the information leakage between view parties when there are few parties involved. Under FMSC, we demonstrate that no private information has leaked to the central server and little view specific information has leaked to other view parties with theoretical assurance.
- We perform extensive experiments on both synthetic and real world benchmark datasets. Experiment results show that predictive accuracy of FMSC is almost approaching to the traditional centralized methods.

The remainder of this article is organized as follows. Section II gives a brief review of related work on multi-view clustering and federated learning. In section III, we propose the architecture of our FMSC approach in details and address the privacy challenges. To evaluate the proposed approach, we conduct extensive experiments on a variety of datasets and demonstrate the effectiveness of our approach in Section IV. The conclusion is drawn in Section 5.

II. RELATED WORK

The present work is mainly based on a wide range of literature. In this section, we outline the most related works including multi-view spectral clustering and federated learning respectively.

A. MULTI-VIEW SPECTRAL CLUSTERING

Spectral clustering [16], [17] is a popular clustering technique that has shown good performance on arbitrary shaped data. It is extensively studied as a significant framework by many literatures on multi-view datasets. Reference [18] proposed a random walk based solution to derive a global graph cut over the single graphs in other views. In [7], the authors applied the idea of co-training and proposed a Co-Training Multi-view Spectral Clustering (CTMSC), where the new graph similarity of one view is limited by solving the eigenvectors of the Laplacian of other views. Reference [8] added a co-regularization term to enforce the eigenvectors of diverse views having high pairwise similarities, and proposed a method called Co-Regularized Multi-view Spectral Clustering (CRMSC). Reference [10] presented a novel method named by MVSC-CEV, to computing the common eigenvectors of the Laplacian matrices derived from the similarity matrices of the input views. To address the possible dependencies among views, [11] adopts the brainstorming process to compensate the biases caused by information sharing between multiple views with dependent opinions, and finally a compromise opinion is merged.

Recently, deep learning based methods [9], [19] have been dedicated to incorporating the local invariance within every single view and the consistency across different views. However, all the aforementioned approaches assume the multi-view data can be collected on a central server, which will expose their sensitive data to other parties. In this article, we consider a secure and privacy preserving framework to perform the multi-view spectral clustering.

B. FEDERATED LEARNING

With the prevalence of complex machine learning technique, most newly proposed models are data-hungry. There is a contradiction among many data owners or parties. On one hand, the accessible data for each party may be very limited, making the model un-trustable when learning only on its own data. As a result, an increasing number of data parties have desired to collaboratively learn a model together. On the other hand, they are not willing to share their data directly to other parties since the data may be sensitive and private. Some organizations or governments have formulated regulations and laws on privacy protection especially in the big data era (e.g., GDPR [14]).
To address this contradiction between the demand of big data for learning and the needs to protect the data privacy and security, the idea of federated learning (FL) was first proposed by Google [20] to learn a centered model among android users without revealing user’s raw data. Research efforts then shed light on this area to apply FL to many scenarios and tasks, including, but not limited to, wireless networks [21], edge computing [22], urban environment sensing [23], topic modeling [24], latent dirichlet allocation [25], matrix factorization [26], neuro network [27], to name a few. To give a typical assumption of federated learning [15], we assume all the parties (including the central server and all view parties) are honest-but-curious, which means they will be honest in operations and faithfully obey the collaborative learning protocol, but be curious to find out the sensitive data from others parties.

To solve Eq.(2) in a distributed way, we can split the objective into two sub-objectives and alternatively update the solutions of sub-objectives by each party. The first sub-objective is to update view specific eigenvector matrix \( \mathbf{U}^{(v)} \) when fixing \( \mathbf{U}^{*} \). Optimizing Eq.(2) equals to:

\[
\arg \max_{\mathbf{U}^{(v)}} tr(\mathbf{U}^{(v)^T} \mathbf{L}^{(v)} \mathbf{U}^{(v)})
\]

subject to: \( \mathbf{U}^{(v)^T} \mathbf{U}^{(v)} = \mathbf{I} \), \( v = 1, \cdots, m \). (3)

Since \( tr(\mathbf{U}^{(v)^T} \mathbf{U}^{(v)} \mathbf{U}^{*}) = tr(\mathbf{U}^{(v)^T} \mathbf{U}^{*} \mathbf{U}^{(v)}) \), the second sub-objective is to update \( \mathbf{U}^{*} \) by fixing eigenvector matrix \( \mathbf{U}^{(v)} \), \( v = 1, \cdots, m \). Eq.(2) can be simplified as:

\[
\arg \max_{\mathbf{U}^{*}} tr(\mathbf{U}^{*^T}(\sum_{v=1}^{m} \lambda^{(v)} \mathbf{U}^{(v)^T} \mathbf{U}^{(v)}) \mathbf{U}^{*})
\]

subject to: \( \mathbf{U}^{*^T} \mathbf{U}^{*} = \mathbf{I} \). (4)

By alternatively updating \( \mathbf{U}^{(v)} \) and \( \mathbf{U}^{*} \) according to Eq.(3) and Eq.(4), we can derive the final solution of Eq.(2).

Note that to derive the common centroid eigenvector matrix \( \mathbf{U}^{*} \), a party needs to obtain \( \sum_{v=1}^{m} \lambda^{(v)} \mathbf{U}^{(v)^T} \mathbf{U}^{(v)} \) denoted by \( \mathbf{S} \) according to Eq.(4). We design a secure architecture to compute this term via a central server and run the rest optimizations at each party independently. Let \( \mathbf{S}^{(v)} = \sum_{v=1}^{m} \lambda^{(v)} \mathbf{U}^{(v)^T} \mathbf{U}^{(v)} \) denote the degree matrix,laplacian matrix and eigenvector matrix for the view \( v \) respectively. To encourage the similarities of cluster structure on different views, a regularization term [8] is introduced to force each view-specific eigenvector matrix \( \mathbf{U}^{(v)} \) towards a common centroid eigenvector matrix \( \mathbf{U}^{*} \). Then the objective of MSC can be defined as follows:

\[
\arg \max_{\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \cdots, \mathbf{U}^{(m)}, \mathbf{U}^{*}} \sum_{v=1}^{m} tr(\mathbf{U}^{(v)^T} \mathbf{L}^{(v)} \mathbf{U}^{(v)}) + \sum_{v=1}^{m} \lambda^{(v)} tr(\mathbf{U}^{(v)^T} \mathbf{U}^{*})
\]

subject to: \( \mathbf{U}^{(v)^T} \mathbf{U}^{(v)} = \mathbf{I} \), \( \mathbf{U}^{*^T} \mathbf{U}^{*} = \mathbf{I} \), \( v = 1, \cdots, m \). (2)

where \( \lambda^{(v)} > 0 \) denotes the normalized weight for the \( v \)-th view. If we believe all views are equally contributed to the objective function, we can set \( \lambda^{(v)} = \frac{1}{m} \) for \( v = 1, \cdots, m \). With the optimal solution \( \mathbf{U}^{*} \), we can conduct k-means algorithm on it to derive the clustering assignment. More detail can be seen in [7], [8].

B. FMSC ARCHITECTURE

Obtaining the solution of Eq.(2) centralized scenarios is straightforward since the central server owns data of all views. Now we consider a distributed scenario that each party has a sensitive view specific data \( X^{(v)} \). Following the typical assumption of federated learning [15], we assume all the parties (including the central server and all view parties) are honest-but-curious, which means they will be honest in operations and faithfully obey the collaborative learning protocol, but be curious to find out the sensitive data from others parties.

In this section, we first introduce the preliminaries of centralized MSC approach. Next we present the proposed Federated Multi-view Spectral Clustering framework, namely, FMSC for short. We give the details of FMSC including the architecture, learning algorithm and finally efficiency analysis.

A. MSC PRELIMINARIES

Given a training dataset \( X = \{x_i | i = 1, \cdots, n, x_i \in \mathbb{R}^d\} \), the clustering task aims to partition \( X \) into \( k \) clusters. Spectral clustering (SC) [16] is one of the most widely used techniques in unsupervised learning and often outperforms the traditional approaches such as k-means. The first step of spectral clustering is to define a similarity matrix \( \mathbf{W} \in \mathbb{R}^{n \times n} \), indicating a measure of the similarity between data instances. The objective of standard SC is defined by:

\[
\arg \max_{\mathbf{U}} tr(\mathbf{U}^T \mathbf{L} \mathbf{U})
\]

subject to: \( \mathbf{U}^T \mathbf{U} = \mathbf{I} \), (1)

where \( \mathbf{L} = \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2} \) denotes the normalized laplacian matrix, \( \mathbf{D} \) denotes the degree matrix defined as a diagonal matrix with degrees \( \mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij} \), and \( tr(\cdot) \) denotes the trace of a matrix.

The solution matrix \( \mathbf{U} \in \mathbb{R}^{n \times k} \) in Eq.(1) can be derived by computing the top \( k \) eigenvectors of \( \mathbf{L} \) and integrating them as a \( \mathbb{R}^{n \times k} \) matrix. Then it employs a traditional clustering method (e.g., k-means) on the row vectors of normalized matrix \( \mathbf{U} \).

For the multi-view scenarios, we assume there are \( m(m \geq 2) \) views, each having diverse features of data. Let \( X^{(v)} = \{x^{(v)}_i | i = 1, \cdots, n, x^{(v)}_i \in \mathbb{R}^{d^{(v)}}\} \) denotes the data instances in view \( v \). \( \mathbf{W}^{(v)}, \mathbf{D}^{(v)}, \mathbf{L}^{(v)} \) and \( \mathbf{U}^{(v)} \) denote similarity matrix, degree matrix, laplacian matrix and eigenvector matrix for the
denote the information that party $v$ needs to share. The key idea of keeping $S^{(v)}$ private to central server is to encrypt it via Additive Homomorphic Encryption (AHE) technique. Then central server can aggregate all shared information, since the additive homomorphic property enables the aggregation over every encrypted entry of matrices. After deriving the encrypted aggregation $S$, the central server transmits it to all view parties. The architecture is illustrated in Fig.1.

From Fig.1, two types of participants are involved in FMSC, the central server and the view parties. Taking view party $v$ as example, there are three components: a cryptor, an optimizer, and a perturbation module. The optimizer includes two modules: a CEM Optimization (Opt in Fig.1 for short) module and a VEM Optimization module. The CEM Optimization module can solve the common eigenvector matrix $U^*$ by Eq.(4), and the VEM Optimization module can solve the view specific eigenvector matrix $U^{(v)}$ by Eq.(3).

The most important part of FMSC is the encryption, decryption modules in view parties, and aggregation module in the central server. The encryption module encrypts and sends the sensitive parameters, namely, $S^{(v)}$, to the central server. The aggregation module receives all the encrypted parameters, and computes the aggregation $Enc(S)$ via AHE. The decryption module decrypts the aggregation and derives $S$ to feed into the CEM Opt module for the next iteration. In this article, we use Paillier Homomorphic Cryptosystem to endow central server the ability of aggregation on the cypher text, see subsection IV-A.

The role of perturbation module is to prevent differential inference attack to the information of eigenvector matrix $U^{(v)}$ as well as similarity matrix $S^{(v)}$, especially when there are few parties. For example, we suppose the number of view parties are two (denoted by index 1 and 2). When receiving the aggregation $S$, party 1 can derive the similarity matrix $S^{(2)}$ of party 2 using $S^{(1)} = S - S^{(2)}$, which violates the second regulation in section I. We will introduce the details of this module in subsection IV-B.

\[
\lambda^{(v)}U^{(v)}U^{(v)^T}
\]

\[
S^{(1)} = \lambda^{(1)}U^{(1)}U^{(1)^T} = (S - S^{(2)})
\]

\[
S^{(2)} = \lambda^{(2)}U^{(2)}U^{(2)^T} = S - (S - S^{(2)})
\]

Algorithm 1 FMSC Protocol

**Input:** Training datasets $X^{(1)}, X^{(2)}, \ldots, X^{(m)}$, parameters $\lambda, \epsilon$

**Output:** Cluster assignments $Y$

1. **Initialize** similarity matrices $W^{(1)}, W^{(2)}, \ldots, W^{(m)}$ for each view party
2. **for** view party $v = 1, \ldots, m$ **do**
   3. initialize the laplacian matrix $L^{(v)}$.
   4. solve the initial eigenvector matrix $U^{(v)}$ by Eq.(1).
   5. perturb $U^{(v)}$ and derive $U^{(v)^*}$.
   6. encrypt $S^{(v)}$ and send it to central server.
3. **end for**
4. **while** not converged in each iteration **do**
5. **Central Server:**
   6. compute the aggregation $S$ on the cipher text.
   7. perturb $S$ and derive a noisy $S$.
   8. send $S$ back to all parties.
7. **end for**
8. **View Party:**
   9. **for** view party $v = 1, \ldots, m$ **do**
      10. decrypt $Dec(S)$ and derive $S$.
      11. solve $U^{(v)}$ by Eq.(4).
      12. solve $U^{(v)^*}$ by Eq.(3).
      13. encrypt $S^{(v)}$ and send it to central server.
      14. **end for**
8. **end while**
15. **all view parties perform k-means algorithm on $U^{(v)^*}$ and**
16. **return** cluster assignments $Y$

The above modules will be executed iteratively until the eigenvector matrix satisfies the convergence criterion. After obtaining the final common eigenvector matrix $U^{(s)}$, all view parties can perform traditional clustering algorithm (e.g., k-means) on $U^{(s)}$ to derive the final cluster assignments. We now present the clustering procedure of FMSC in Alg.(1) as a protocol that all parties should obey.
IV. SECURITY MECHANISMS

In this section, we will introduce the details of two security mechanisms. The first one is how to implement the secure aggregation, and the second one is to elucidate the perturbation scheme. Finally, we will give some privacy and efficiency analysis to the proposed FMSC approach.

A. SECURE AGGREGATION

To overcome the possible information leakage to central server, the shared weighted similarity matrices $S^{(v)}$ should be encrypted such that the server cannot inverse. Additively Homomorphic encryption (AHE) is an efficient scheme that allows a third party to aggregate on the encrypted data without decrypting it in advance. In this article, we use Paillier Homomorphic Cryptosystem [30], an AHE cryptosystem which supports homomorphic addition operation on the cipher texts as follows:

$$Enc_{pk}(m_1) + Enc_{pk}(m_2) = Enc_{pk}(m_1 + m_2),$$

where $m_1, m_2$ are the cipher text of the messages, $Enc(\cdot)$ is an encryption algorithm, and $pk$ denotes the public key used by the encryption process.

Note that a view party need to encrypt matrix $S^{(v)} = \lambda^{(v)}U^{(v)}U^{(v)T}$ before send to central server. Thus, $S^{(v)}$ should be encrypted on each entry:

$$Enc(S^{(v)}) = (Enc(S^{(v)}_{ij}))_{n \times n},$$

where $S^{(v)}_{ij}$ denotes the $i$-th row and $j$-th column of $S^{(v)}$, and $n$ is the number of data instances. Here we omit $pk$ for simplicity.

When central server has received all shared view-specific similarity matrices, it can implement the add operation on those cypher text and derive the encrypted aggregation:

$$Enc(S) = Enc(\sum_{i=1}^{m} S^{(v)}_{ij}) = \sum_{i=1}^{m} Enc(S^{(v)}).$$

Next the central server sends $Enc(S)$ back to all parties. The view party $v$ decrypts $Enc(S)$ using the private key $sk$:

$$S = Dec_{sk}(Enc(S)).$$

Key Generation: Before running the FMSC protocol, the public key and secret key should be generated in advance and distributed to all parties. The key generation process can be carried out on an independent Key Distributor (KD), or one of the view parties. After the keys are generated, the public key can be distributed through an untrustworthy channel to all the parties including the central server, but the distribution of secret key should be shared by establishing different TLS/SSL secure channels among view parties.

B. PERTURBATION MECHANISM

In the proposed FMSC architecture, the perturbation module add some noises before sending the aggregated similarity matrix $S$ back to view parties. Since we assume each view party is honest-but-curious, the aim of perturbation module is to prevent the possible leakage of sensitive information from other parties. If not perturbed, a view party $v$ can easily derive the aggregations of other parties by subtracting $S^{(v)}$ from $S$. In addition, one party can directly obtain another parties’ sensitive $S^{(v)}$ if the number of parties is small (e.g., 2).

To overcome this issue, we introduce differential privacy (DP) as a perturbation technique to add small noise on $S$. Differential privacy is one of the state-of-the-art privacy models against differential attack, which is first introduced in [28]. DP guarantees that the adversary couldn’t infer sensitive information about any specific instance with high confidence (parameterized by a privacy budget $\epsilon$) from the queries(e.g., querying aggregations in this article), even if all the remaining instances of the data are known by the adversary. Similar to this concept, we want to guarantee that any view party $v$ couldn’t differentiate $S - S^{(v)}$ with high confidence from $S$, even if the similarity matrix $S^{(v)}$ is known in advance. Note that a smaller privacy budget value $\epsilon$ will induce a stronger privacy guarantee but a higher noise.

We firstly consider preserving the privacy of one element $S_{ij}$ in the $i$-th row and $j$-th column of $S$. Let $x = [S^{(1)}_{ij}, \ldots, S^{(m)}_{ij}]$ denotes the database of all received data, and $x'$ denotes a neighboring database that differs in at most one entry, e.g., $x' = [S^{(1)}_{ij}, \ldots, S^{(m-1)}_{ij}]$. One of the commonly used DP models is defined as follows:

**Definition 1 ($\epsilon$-Differential Privacy [31]):** A randomized mechanism $M : X \rightarrow T$ is $\epsilon$-differentially private, if for any neighboring databases $x, x' \in X$, and for any possible output $O \in T$, we have

$$Pr[M(x) = O] \leq e^\epsilon Pr[M(x') = O],$$

where $Pr[\cdot]$ denotes the probability.

To satisfy the definition of DP, traditional approach is to employ the laplace mechanism and define the randomized mechanism $M$ as:

$$M(x) = f(x) + Y,$$

where $f(\cdot)$ denotes the aggregation function, and $Y$ denotes the noise variable following a laplacian distribution. According to the theorem 3.6 in [32], we have the following results:

**Lemma 2:** the laplace mechanism $M$ preserves $\epsilon$-differential privacy if we define $Y \sim L(0, \frac{\Delta f}{\epsilon})$, where $\Delta L$ denotes the laplacian distribution, and $\Delta f = \max_{x, x'} \|f(x) - f(x')\|_1$ denotes the $l_1$-sensitivity of function $f$.

While this is acceptable when applying all elements in $S$ satisfying differential privacy, it is too restrictive in practice because the added noises are so large that the clustering algorithm couldn’t easily converge. To address this issue we propose a method called Random Laplacian Response (RLR) mechanism, which composes a random response and a laplacian mechanism. We formulate the RLR perturbation mechanism as an algorithm in Alg.(2), where some parameters should be initialized as inputs. We denote $\eta$ as parameter of a bernoulli distribution to control how many elements are
Algorithm 2 RLR Perturbation Mechanism

Input: $Enc(S)$, $\Delta f$, $\epsilon$, $\eta$

Output: Perturbed matrix $Enc(S)$

1: for each element $Enc(S_{ij})$ in $Enc(S)$ do
2: \hspace{1em} sample $b \sim Bernoulli(\eta)$
3: \hspace{1em} if $b == 0$ then
4: \hspace{2em} $Y = 0$
5: \hspace{1em} else
6: \hspace{2em} sample $Y \sim L(0, \frac{\Delta f}{\epsilon})$
7: \hspace{1em} end if
8: \hspace{1em} $Enc(S_{ij}) \leftarrow Enc(S_{ij}) + Enc(Y)$
9: end for
10: return $Enc(S)$

selected to be perturbed. If $b$ sampled from the bernoulli distribution equals to zero, then the noise would be set to zero (line 4). Otherwise we produce a noise sampled from a laplacian distribution and add it to the current elements. Note that the addition is implemented on the cyphertext via paillier AHE system (line 8).

In deed, parameter $\Delta f$ controls the $l_1$-sensitivity of function $f$, but suffers from a major limitation: it couldn’t directly be computed in central server because $\{S^{(1)}, \cdots, S^{(m)}\}$ are encoded in cyphertext. Since $\Delta f$ is bounded by the maximum of $\{S^{(1)}, \cdots, S^{(m)}\}$, we enlarge $\Delta f$ to fit all elements:

$$\max_{x,x'} \|f(x) - f(x')\|_1 = \max\{S^{(1)}_{ij}, \cdots, S^{(m)}_{ij}\} \leq \{S^{(1)}_{\text{max}}, \cdots, S^{(m)}_{\text{max}}\} := \Delta f,$$  \hspace{1em} (11)

where $S^{(i)}_{\text{max}}$ denotes the maximum value in matrix $S^{(i)}$. To this end, the central server can collect all the maximum values in plaintext for every iteration, without revealing the sensitive $S^{(i)}$. Although the relaxed $\Delta f$ would introduce larger noise, we can control the amount of overall noise by adjusting parameters $\epsilon$ and $p$.

C. PRIVACY ANALYSIS

In this subsection we give analysis about the privacy of our proposed FMSC approach.

1) SECURITY AGAINST CENTRAL SERVER

As demonstrated in Fig.1, only ciphertext from view parties are sent to the central server in FMSC. So no bit of information will be leaked to the central server because the paillier homomorphic encryption cryptosystem guarantees ciphertext indistinguishability against any possible plaintext attacks [33].

2) PRIVACY AGAINST VIEW PARTIES

From Alg.(2) we can see that the random laplacian response perturbation mechanism $M$ is composed by two sub mechanisms: a random response $M_1$ and a laplacian mechanism $M_2$. $M_1$ flips a coin and the result determines whether $M_2$ will be activated. Define the new privacy budget as:

$$\epsilon' = \max \left(\epsilon, \epsilon + \ln \frac{1 - \eta}{\eta}, \ln \frac{\eta}{1 - \eta}\right).$$  \hspace{1em} (12)

Then we give the following theorem:

**Theorem 3:** The Random Laplacian Response (RLR) perturbation mechanism preserves $\epsilon'$-differential privacy.

The proof of Theorem 3 is demonstrated in Appendix.

V. EXPERIMENTS

In this section, we conduct experiments on both synthetic and real-world multi-view datasets. First, we introduce the experimental settings as well as baseline methods. Then we evaluate the effectiveness and performance of the proposed FMSC approach against the baseline methods. Finally, we analyze different parameters $\eta$ and $\epsilon$ to demonstrate how perturbation effect the results.

A. EXPERIMENT SETTINGS

To evaluate the performance of the proposed FMSC approach, we adopt five multi-view datasets including both synthetic and real-world datasets.

Synthetic data is generated from two views to four views with two clusters. Similar to the settings in [7] and [8], we randomly sample 1000 instances ($n = 1000$) for each view via 2-dimentional gaussian mixture model. The means and covariances of all clusters in different views are defined in Table 1.

We also adopt five popular real-world datasets: UCI digits, Reuters, BBC, BBCSports and Yale datasets. The datasets are introduced as follows:

- UCI digits. This dataset is created from handwritten numerals from 0 to 9, and each instance forms an image with $15 \times 16$ pixels. It is available at the UCI repository.
- Reuters. Reuters multilingual corpus is a set of news articles written in five languages: English, French, German, Italian and Spanish. We sample 10,000 instances into our experiments from the original corpus that contains 18,758 articles. For each view, we use a word dictionary as features and compute the term frequencies as feature values.
- BBC and BBCSports datasets. Both of the two datasets are multi-view news articles from the BBC. BBC contains five possible topics (business, entertainment, politics, sport and technology), and BBCSport focuses on...
TABLE 2. Description of the five real-world datasets.

| Dataset    | #Sample | #Feature | #View | #Class |
|------------|---------|----------|-------|--------|
| UCI digits | 2000    | 649      | 6     | 10     |
| Reuters    | 10000   | 2500     | 5     | 6      |
| BBC        | 544     | 6386     | 2     | 5      |
| BBC        | 685     | 18491    | 4     | 5      |
| Yale       | 165     | 14150    | 3     | 15     |

five sports topics (athletics, cricket, football, rugby and tennis). We also adopt term frequencies as feature values.

- Yale. Yale is an image dataset that consists of 165 face images of 15 classes in pixel. Each class has 11 images.

We summarize some characters of the above datasets in Table 2. Note that the numbers in column ‘#Feature’ are the sum of features in all views.

To validate the effectiveness of our FMSC approach, we run our approach on the abovementioned datasets and compare the results against three baseline clustering methods: Concatenation, Co-training, and Co-reg. Concatenation is a straightforward approach that features of each view are concatenated to perform a standard spectral clustering. Co-training [7] alternatively performs spectral clustering on each view by incorporating other view’s eigenvector matrices as constraints. Co-reg refers to Co-Regularized multi-view spectral clustering approach proposed by [8]. Our work is based on Co-reg and extends it to the federated scenario.

We measure the quality of clustering results on several metrics: Normalized Mutual Information(NMI), Adjusted mutual information(AMI), Adjusted Rand index(ARI), Completeness(COM) and Homogeneity(HOM). NMI and AMI are two different normalized versions of Mutual Information (MI), which quantifies how much the estimated clustering is informative about the true clustering [34]. The ARI metric is similar to the clustering accuracy and measures the degree of agreement between the estimated clustering and the true clustering. COM and HOM are two important conditional entropy based metrics [35]. COM measures the ratio of the member of a given class that is assigned to the same cluster, while HOM measures the ratio of instances of a single class pertaining to a single cluster.

In order to evaluate the execution time of the proposed approach, we also introduce Running Time in seconds as a metric. For all the above metrics, a higher value means the corresponding method has a better performance. We repeat 10 times and report the mean values for each experiment. All approaches are run on an Intel i9-9900K 3.6 GHz with 64GB of RAM, using single-threaded processes.

B. RESULTS

Firstly, we test our FMSC approach without adding any noises. The aim is to verify whether it can receive as considerable clustering performance as centralized approaches. We omit the Perturbation module in Fig.1 and report the results as follows.

Table 3 demonstrates the clustering results on synthetic datasets. We can easily see that the overall performance of four-views dataset is better than that of three-views, and the overall performance of three-views dataset is better than that of two-views. The results suggest that more views can bring improvements on the clustering performance. For different approaches, the performance of Co-reg is superior than Co-training and Concatenation. Our FMSC approach has almost as the same performance as Co-reg since FMSC is a distributed and secure variant of Co-reg.

Table 4 shows the spectral clustering results on five real-world datasets. For UCI digits and Reuters, our FMSC approach has the best performance than other baselines. Our performance is slightly worse than Co-training or Co-reg on BBCSports and BBC datasets. But the declined performance of FMSC is acceptable as it protects the privacy of sensitive data. For Yale dataset, the performance of our approach is as almost the same as Co-training and Co-reg methods on all metrics. So, we can safely say that FMSC is a secure distributed spectral clustering approach with small performance drawbacks.

Finally, Table 5 shows the average running time over 10 executions of each approaches. Note that we omit the
communication time over parties for our FMSC since it is trivial compared with computation time. As expected, all three centralized baselines run faster than FMSC. Specifically, it only needs less than one second for Concatenation on all datasets. Our FMSC costs larger execution time because on each iteration it needs lots of encryption and decryption operations which are very time consuming. We can also see that the running times of FMSC are different for different datasets. The reason is these datasets have different number of views with different number of features. More views and features will cost more running times.

C. PARAMETER ANALYSIS

In this subsection, we report the effect of two parameters used in FMSC approach. The first parameter $\eta$ controls how many elements in aggregated matrix $S$ are sampled to add noises. The second parameter $\epsilon$ determines the original privacy budget for each element.

We first investigate the influence of parameter $\eta$ by fixing $\lambda = 0.1$ and $\epsilon = 0.001$. We varying $\eta$ from 0 to 1 with step 0.1. Fig.2 shows the performance of FMSC on five real-word datasets. We can find two interesting points: (1) for UCI digits and Reuters datasets, our approach has one clear minimum value on all five metrics. (2) But for BBCSports and BBC datasets, there are two obvious minimum values. Three minimum values could be find out for Yale dataset. In summary, if we get the privacy budget $\epsilon$ in advance, $\eta$ can be adjusted easily to get a better clustering performance.

Then we conduct experiments on two representative datasets to observe the effect of different parameters $\epsilon$. We fix $\lambda = 0.1$ and $\eta = 0.5$. Thus from Eq.(12) we derive the system privacy budget $\epsilon' = \epsilon$. From Fig.3, we can see that when $\epsilon$ is small, the measurement values on all metrics are small, and the values tend to increase with the increase of $\epsilon$. With $\epsilon \geq 1$, the values are stable and don’t increase any more. This result conforms our idea that $\epsilon$ controls the privacy budget. Given a very large privacy budget, the clustering performance will not be influenced by the added negligible noises.

![FIGURE 2. The effect of $\eta$ on different datasets.](image-url)

![TABLE 5. Running times on real-word datasets.](table-url)
Fig. 4 shows similar results of how $\epsilon$ effect clustering performance on yale dataset. However, the curves is flatter for all metrics. If $\epsilon$ becomes small, the clustering performance will slightly decrease. When $\epsilon \geq 10$, the clustering performance will not change. This reflects that yale is not sensitive to noise and we can distribute more privacy budgets for it.

VI. CONCLUSION

In this article, we propose a novel approach named Federated Multi-view Spectral Clustering (FMSC) to address the possible privacy leakage problem in centralized Multi-view Spectral Clustering (MVSC) approaches. Specifically, we design an architecture for all view parties to collaboratively learning the underlying cluster structure, without revealing their own data to other participants. On one hand, we borrow Homomorphic Encryption (HE) to compute the aggregations on cypher text of sensitive parameters to protect privacy against central server. On the other hand, we also introduce a Differential Privacy (DP) mechanism to prevent the information leakage between view parties when there are few parties involved. We theoretically prove that our approach can achieve secure computation and differential privacy on sensitive shared parameters. Our quantitative experiments show that FMSC has considerable good performance compared with centralized approaches.

In the future work, we try to improve the efficiency of FMSC from the following direction: drop the time costing encryption-decryption mechanism and instead adopt a locally differential-privacy mechanism to get a more efficient distributed model. However, the performance of this kind of model may reduce compared with the proposed FMSC approach. In another way, there exists newly published approaches that may have good performance on multi-view data clustering task, e.g., low-rank approximation [36], subspace matrix factorization [37], probabilistic model [38] and deep learning [9]. However, they are central based methods and it is not trivial to transfer them on the distributed and federated scenarios. In the future, we will shed lights on the deep methods that transform them into more secure and private approaches.
3) If \( \epsilon \neq f(x) \) and \( \epsilon = f(x') \), we have:

\[
p_x(\epsilon) = \frac{\eta \exp \left(\frac{-|\epsilon' - f(x')|}{\Delta f}\right)}{1 - \eta} \leq \frac{\eta \exp \left(\frac{-|\epsilon' - f(x')|}{\Delta f}\right)}{1 - \eta}.
\]

By defining \( \epsilon' \) in Eq. (12), we can finally have: \( p_x(\epsilon) \leq \exp(\epsilon') p_x(\epsilon') \), which proves the result.

REFERENCES

[1] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, “Binary multi-view clustering,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 7, pp. 1774–1782, Jul. 2019.

[2] C. Xu, Z. Guan, W. Zhao, H. Yu, Niu, and B. Ling, “Adversarial incomplete multi-view clustering,” in Proc. Int. Joint Conf. Artif. Intell., Macao, China, Aug. 2019, pp. 3933–3939.

[3] J. Konen, H. Brendan McMahan, F. X. Yu, P. Richtárík, A. Theertha Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” 2017, arXiv:1610.05492. [Online]. Available: http://arxiv.org/abs/1610.05492

[4] IEEE Access
ANG LI was born in 2000. He is currently pursuing the B.S. degree in computer science with North China Electric Power University, Baoding, China. His main research interests include distributed learning, secure machine learning, and natural language processing.

BOLIN SHEN is currently pursuing the B.S. degree in computer science with North China Electric Power University, Baoding, China. His research interests include machine learning and natural language processing.

YUYAN SUN was born in 1982. He received the B.S. degree in computer science from Peking University, Beijing, China, in 2004, and the M.S. and Ph.D. degrees in computer science from the University of Chinese Academy of Sciences, Beijing, in 2007 and 2016, respectively.

From 2007 to 2012, he was a Research Assistant with the Institute of Software, Chinese Academy of Sciences. Since 2016, he has been an Assistant Professor with the Beijing Key Laboratory of IoT Information Security, Institute of Information Engineering, Chinese Academy of Sciences, and the School of Cyber Security, University of Chinese Academy of Sciences. His main research interests include security of industrial control systems and the IoT.

HONGMEI WANG received the B.S. degree in mathematics from Northeast Forestry University, Harbin, China, in 2003, and the master’s degree in applied mathematics from the Harbin Institute of Technology, Harbin, China, in 2006. Since 2006, she has been a Lecturer with the School of Science and Technology, North China Electric Power University. Her current research interests include multi-view learning, anomaly detection, and private data mining.