Review Article

Multi-View Clustering Goes Federated: A Survey

Kristina P. Sinaga[1]

1. Independent researcher

The advent of contemporary data evolution has precipitated an unparalleled escalation in data generation, thereby necessitating the development of innovative methodologies to harness its latent potential. Federated Learning (FL) has emerged as a transformative paradigm, enabling collaborative model training across decentralized data sources while preserving privacy. This review explores the intersection of FL and Multi-View Clustering (MVC), investigating how these methodologies synergize to address the challenges of unsupervised learning in distributed environments. The present study examines the theoretical foundations, algorithmic advancements, and practical applications of MVC within the FL framework. The study under discussion highlights key contributions and identifies future research directions. Synthesizing insights from recent studies, this review aims to provide a comprehensive understanding of how decentralized learning can be effectively leveraged for MVC tasks, ultimately advancing the field of unsupervised learning in federated settings. Moreover, the ramifications of this integration for privacy-preserving Machine Learning (ML) and distributed systems are deliberated, with particular emphasis placed on the potential for innovative solutions to emerge from this convergence of methodologies.

Corresponding author: Kristina P. Sinaga, kristinasinaga41@gmail.com

1. Introduction

The deployment of agents in various domains is becoming increasingly prevalent. These domains include, but are not limited to, healthcare [1][2][3], finance [4][5][6][7], and social media [8][9][10][11]. The purpose of this deployment is to analyze and extract insights from large volumes of data. This phenomenon also followed the advent of quantum computing [12], a development with the potential to bring about a paradigm shift in data processing and analysis. Nonetheless, the conventional centralized

strategy for ML, whereby data is amassed and processed in a central location, presents considerable challenges with regard to privacy, security and scalability. In order to address the aforementioned issues, FL has emerged as a promising paradigm. This paradigm allows multiple clients to collaboratively train ML models without sharing their raw data [13]. The FL system has been developed to facilitate decentralized learning while preserving data privacy, thus rendering it an attractive solution for a variety of applications.

Clustering is an unsupervised learning technique that groups similar data points together based on their features. The concept of data clustering can be traced back to the early 20th century, with the advent of hierarchical clustering methodologies pioneered by Ward in 1963 [14]. Subsequent to this development, a variety of clustering algorithms have been formulated, including k-means [15], Fuzzy c-means (FCM) [16], Possibilistic c-means (PCM) [17], Density-Based Spatial Clustering of Application with Noise (DBSCAN) [18], and spectral clustering [19]. The application of these algorithms has been extensive in a variety of contexts, including image segmentation [20][21][22], customer segmentation [23][24], and anomaly detection [25][26][27]. However, conventional clustering algorithms frequently encounter difficulties when processing data from multiple perspectives or sources, resulting in suboptimal clustering outcomes. In order to address the issue under discussion, MVC techniques have been developed. The objective of these techniques is to integrate information from multiple perspectives, thereby enhancing the efficacy and reliability of clustering algorithms.

In 1984, Bezdek *et al.* [16] introduced the FCM algorithm, a pioneering development in data classification that facilitates the identification of multiple clusters, characterized by varying degrees of membership. This approach has been expanded to encompass MVC, a process that integrates multiple perspectives of the data with a view to enhancing the efficacy of clustering algorithms. Following a period of twenty years, the system underwent expansion with the objective of accommodating a greater quantity of resource data. For instance, Kumar *et al.* [28] proposed a co-regularized MV spectral clustering method that encourages consistency between different views. In a similar manner, Bickel *et al.* [29] presented an MVC algorithm founded upon the Expectation-Maximization (EM) framework. This algorithm undertakes iterative updates to both cluster assignments and model parameters across a multitude of views. From 2004 to 2026, the MVC approach was extensively adopted in a centralized setting, utilizing a range of methodologies. The range of methods employed in this study encompassed both fundamental k-means approaches and advanced hybrid methods incorporating Deep Learning (DL) techniques [30][31]

[32][33][34] and complex matrix reconstruction of tensors [35][36][37][38][39][40]. The employment of MVC techniques has yielded encouraging outcomes in a range of applications, including image clustering [32][41], document clustering [42], and social network analysis [43][44][45].

The concept of FL in MVC was initially proposed by Chen *et al.* (2023) [46], and was subsequently elaborated upon by Hu *et al.* (2023) [47] and Yang *et al.* (2024) [48]. From 2023 to 2026, a series of extensions were implemented, encompassing both rudimentary versions and tensor versions of MVC in FL [49][50][51][52][53]. However, a period of three years is inadequate for the completion of this research topic. The integration of MVC with FL is a nascent field of research, which is characterized by numerous challenges that must be addressed. The present review has been conceived with the aim of providing a comprehensive overview of the current state of research in MVC within the context of FL. The following paper will provide architectures illustrating the transition of MVC from a centralized to a decentralized model. In addition to emphasizing salient contributions, the review will also identify future research directions.

The rest of this review is organized as follows: Section 1 provides an overview of the theoretical foundations of FL and MVC. Section 2 discusses various methodologies for integrating MVC with FL. Section 3 highlights practical applications of MVC in FL settings. Finally, Section 4 identifies future research directions and concludes the review.

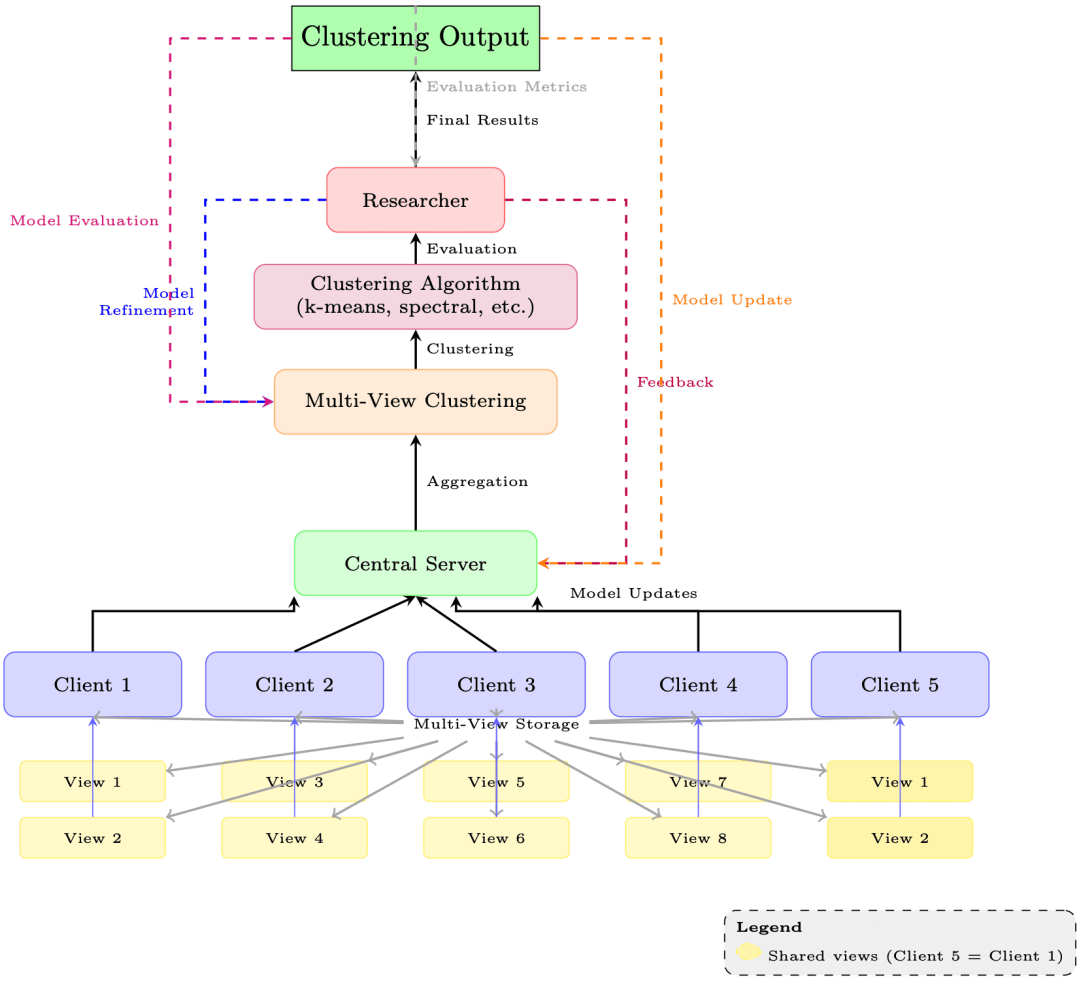

Figure 1. A unified framework for decentralized MVC in FL. It is evident that a number of clients engage in collaborative endeavors with a central server, thereby facilitating feedback loops that are instrumental in the evaluation and refinement of the model. It is demonstrable that the clustering process is iterative in nature, thereby enabling continuous enhancement of both the model and the clustering performance. Subsequently, researchers are able to evaluate the clustering output and provide feedback for the purpose of refining the model further, thus creating a dynamic and adaptive clustering framework in an FL setting.

2. A Study of MVC in FL

Before delving into the methodologies and applications of MVCs in FL, it is essential to establish a solid understanding of the theoretical foundations of both FL and MVC. Figure 2 illustrates the conceptual framework of MVC in FL, where multiple clients collaboratively train a clustering model while keeping their data decentralized. While decentralized MVC learning can be globally illustrated as a single server-

client architecture that include updating cluster centers and cluster assignments as well as sharing the global model, the actual implementation can be more complex and may involve various communication protocols, synchronization strategies, and privacy-preserving techniques to ensure effective collaboration among clients while maintaining data privacy.

As shown in Figure 2, the MVC process in an FL setting involves multiple clients collaborating with a central server to train a clustering model on decentralized data, while the non-federated environment allows for centralized data access and direct integration of multiple views for clustering, as illustrated in Figure 3. The iterative nature of the clustering process, along with the communication between clients and the central server, allows for continuous improvement of the clustering performance over time as the model learns from the data and adapts accordingly. The integration of MVC with FL presents unique challenges and opportunities for advancing unsupervised learning in distributed environments.

The centralized conceptual framework of MVC in a non-federated environment, as illustrated in Figure 3, allows for direct access to all views of the data for clustering without any communication constraints. This framework enables the clustering model to integrate information from multiple views more effectively, which can lead to improved clustering performance. However, it also raises privacy concerns, as all data is centralized and accessible to the clustering model. In contrast, the FL framework allows for decentralized learning while preserving data privacy, but it also introduces challenges in terms of communication and synchronization among clients and the central server. The comparison between these two frameworks highlights the trade-offs between centralized and decentralized approaches to MVC. Figure 4 provides a visual comparison of the two frameworks, highlighting the key differences in data flow, model training, and clustering processes between the FL setting and the non-federated environment.

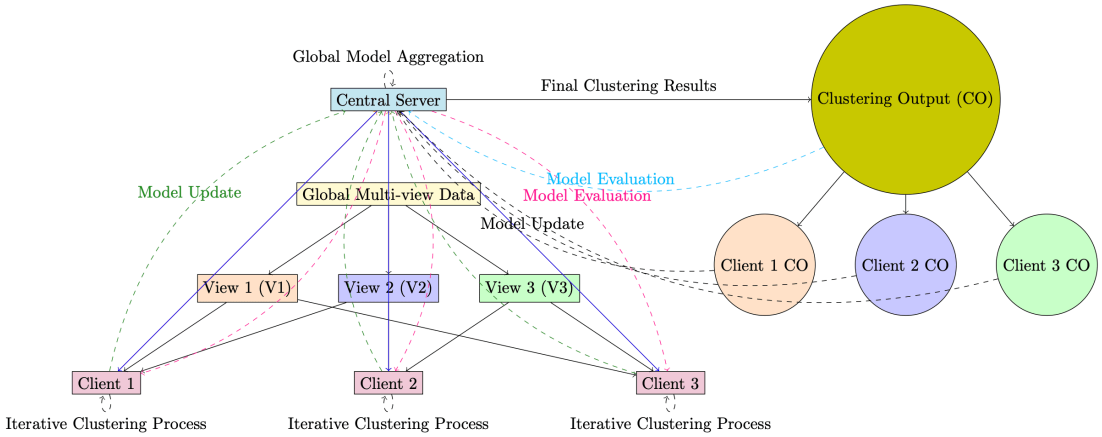

Figure 2. Conceptual framework of multi-view clustering in federated learning. Multiple clients (Client 1, Client 2, Client 3) collaboratively train a clustering model on their local data while communicating with a central server to aggregate model updates and share the global model. Each client has access to different views of the data, which are integrated to improve clustering performance.

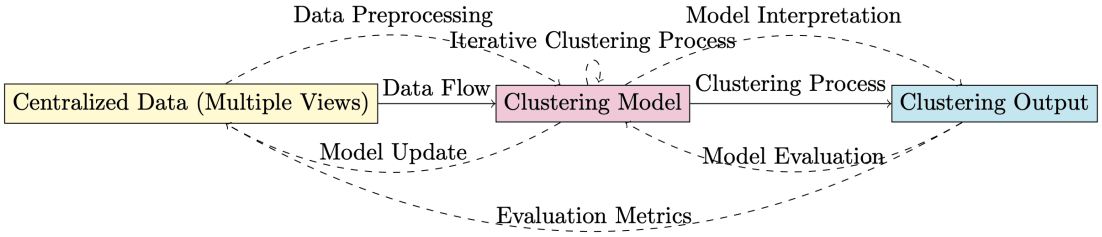

Figure 3. Conceptual framework of multi-view clustering in a non-federated environment. In this setting, all data is centralized, and the clustering model has access to all views of the data without any privacy constraints. The model can directly integrate information from multiple views to improve clustering performance.

As displayed in Figure 4, unlike the FL setting, the non-federated environment allows for direct access to all views of the data for clustering without any communication constraints. This framework enables the clustering model to integrate information from multiple views more effectively, which can lead to improved clustering performance. However, it also raises privacy concerns, as all data is centralized and accessible to the clustering model. In contrast, the FL framework allows for decentralized learning while preserving data privacy, but it also introduces challenges in terms of communication and

synchronization among clients and the central server. The comparison between these two frameworks highlights the trade-offs between centralized and decentralized approaches to MVC, which will be further discussed in the subsequent sections of this review.

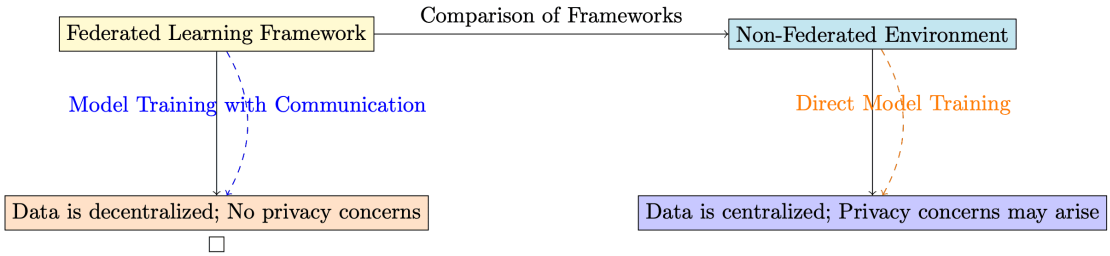

Figure 4. Comparison of multi-view clustering in federated learning (Figure 2) and non-federated environment (Figure 3). The federated learning setting involves multiple clients collaborating with a central server while keeping data decentralized, whereas the non-federated environment allows for centralized data access and direct integration of multiple views for clustering.

2.1. Mathematical Symbol: Conceptual Entities

This section will discuss two aspects of the interaction between the formation of conceptual entities and their role in helping readers build increasingly complex mathematical concepts of MVC in a FL setting. The construction of an objective function as a conceptual entity exemplifies the entitication process, which is a set of predefined regularizations that aims to define a set of clustering tasks systematically and establish an effective algorithm based on the initial derivation that employed the Lagrange and differential approaches. The mathematical symbols utilized in this paper are enumerated in Table 1. While a comprehensive comparison of conceptual entities between standard k-means and FCM are presented in Table 2.

Category	Symbol	Description
Dataset Parameters	n	Number of data points
	s	Number of views
	d_h	Number of dimensions in the h-th view
	L	The number of participated clients
	x_{ij}^h	Value of the j-th feature of the i-th data point in the h-th view
Clustering Parameters	c	Number of clusters
	m	Fuzzification parameter (used in traditional FCM)
	t	Current iteration number
	μ_{ik}	Membership degree of the i-th data point to the k-th cluster
	a_{kj}^h	Center of the k-th cluster for the j-th feature in the h-th view
Weighting Parameters	v_h	Weight of the h-th view
	w_j^h	Weight of the j-th feature in the h-th view
	δ_j^h	Regularization parameter for the j-th feature in the h-th view
	β	Balancing parameter controlling the distribution of view weights v_h
	η	Balancing parameter controlling the distribution of feature weights w_j^h

Table 1. Mathematical notation and symbols used in this study

2.1.1. The Standard Hard Clustering

In standard hard clustering, also known as k-means [15], the process of updating the associated entities culminates in a final clustering result by modifying a singular condition. For instance, one can manipulate binary membership rules to enforce one-to-zero-or-one relationships within relational databases. This stipulation ensures that each member occupies only one role. From a mathematical perspective, the constraint of $\sum_{k=1}^{c} \mu_{ik} = 1, \mu_{ik} \in \{0, 1\}$ in k-means clustering is designed to partition n observations into c clusters, wherein observations are assigned to the cluster that is closest to the mean distance between data points or observations x_i and cluster centers a_k in feature space d. The k-means

constraint has been demonstrated to be a viable solution for addressing network security challenges, particularly in the context of intrusion detection systems (IDS) [54][55]. This constraint facilitates the differentiation between normal and abnormal network traffic without the requirement for pre-labeled data, thereby enhancing the efficacy of network security measures. The objective function of standard k-means can be expressed as below.

$$J_{\text{k-means}} = \sum_{i=1}^{n}\sum_{k=1}^{c}\mu_{ik}\sum_{j=1}^{d}\left(x_{ij}-a_{kj}\right)^2 \tag{1}$$

$$\text{s.t.}\sum_{k=1}^{c}\mu_{ik}=1, \mu_{ik}\in\{0,1\}$$

2.1.2. The Standard Soft Clustering

The standard FCM clustering algorithm, as outlined by Bezdek [16], shares notable similarities with the well-known k-means clustering algorithm. A key distinction, however, resides in the integration of a membership vector, denoted as μ_i, which quantifies the proportion of a particular observation $i = 1, \cdots, n$ belonging to each designated cluster $k = 1, \cdots, c$. In essence, FCM facilitates the allocation of observations n across multiple clusters. Therefore, it can be concluded that an observation may be associated with two distinct cluster centers, each exhibiting varying degrees of membership μ_{ik}. This phenomenon is determined by a fuzzification parameter m. From a mathematical perspective, the constraint imposed by the standard k-means algorithm, expressed as $\sum_{k=1}^{c}\mu_{ik}=1, \mu_{ik}\in[0,1]$ in FCM, fails to capture the overlapping and multidimensional behaviors characteristic of the k-means algorithm.

FCM is not only a methodological preference, but also a good option for a more accurate, dependable, and adaptable in complex, real-world applications. This is because FCM is not only a recognized method for discovering latent structures in complex datasets, but it also a more compeling choice than traditional like k-means when dealing with nuanced, non-linear data. Although some may contend that k-means is simpler and more computationally efficient, this efficiency is achieved at the expense of accuracy in contexts where clusters overlap or boundaries are inherently ambiguous. These superior approaches in FCM are useful in domain such as pattern recognition, bioinformatics, and business intelligence, where real-world data rarely conforms to crisp separations. In medical image analysis, FCM effectively segments brain tumors by assigning varying degrees of membership to pixels, thereby managing uncertainty, noise, and intensity inhomogeneity in magnetic resonance imaging (MRI) data [56][57][58].

This enhances segmentation robustness compared to hard clustering, making FCM an ideal method for precise anatomical delineation. Te objective function of FCM can be expressed as follows.

$$J_{\mathrm{FCM}} = \sum_{i=1}^{n} \sum_{k=1}^{c} \mu_{ik}^{m} \sum_{j=1}^{d} \left(x_{ij} - a_{kj}\right)^2 \tag{2}$$

$$\text{s.t.} \sum_{k=1}^{c} \mu_{ik} = 1, \mu_{ik} \in [0,1]$$

Aspect	k-means	FCM
Clustering type	Hard clustering: assigns each data point to a single cluster	Soft clustering: assigns degrees of membership to data points
Handling of overlapping clusters	Poor; struggles when clusters overlap or boundaries are uncertain	Effective; supports graded memberships and non-crisp boundaries
Applications	General-purpose clustering; works best with well-separated and spherical clusters	Pattern recognition, bioinformatics, business intelligence, and medical image analysis (e.g., brain tumor segmentation in MRI)
Performance in noisy/uncertain data	Sensitive to noise and outliers; may mislabel uncertain points	Robust; manages uncertainty, noise, and intensity inhomogeneity
Cluster shape assumption	Assumes clusters are spherical, convex, and of comparable size and density	Flexible; can adapt to non-linear or irregular clusters
Suitability for non-crisp boundaries	Low; cannot represent partial memberships	High; ideal for data with overlapping or fuzzy class structures
Handling of categorical variables	Cannot handle categorical variables directly; requires encoding that increases dimensionality	Not inherently designed for categorical variables, but can be adapted
Sensitivity to initialization	Highly sensitive; poor initialization can lead to sub-optimal local minima	Less sensitive due to membership distribution
Impact of cluster transformation	Transforming non-spherical clusters to spherical may degrade intrinsic feature relationships	Preserves intrinsic data characteristics, as no shape transformation is required
Robustness in complex data	Limited; struggles with irregular, elongated, or heterogeneous clusters with significant variability in density or size	High; suitable for irregular, heterogeneous, and noisy data

Table 2. Comparison of standard clusterings in Non-Federated Settings

2.1.3. The Challenging of Standard Clustering

Conventional clustering methods are increasingly inadequate in the face of rapidly expanding data volumes and the complexity of modern data environments. Equations 1 and 2 are designated for a single resource in non-federated setting. The two traditional approaches face a significant disadvantage in processing information distributed across multiple sources and locations. This limitation restricts their effectiveness in practical applications. This limitation has become more evident as contemporary trends emphasize bringing machine learning to where the data resides. In response to these challenges, there is a critical need to design more advanced and complex objective functions, particularly within unsupervised learning frameworks, in order to address the demands of heterogeneous and distributed datasets.

2.2. Centralized MVC

Bickel et al. [29] firstly proposed a multi-view clustering method based on the principle of co-training, which assumes that different views of the data can provide complementary information for clustering. The method iteratively updates cluster assignments and cluster centers based on the agreement between different views, leading to improved clustering performance. This approach has been extended and adapted in various ways in subsequent research, including the integration of deep learning techniques for feature extraction and representation learning in multi-view clustering. Since then, extensive expansion of multi-view clustering methods has been developed, including spectral clustering, subspace clustering, and deep learning-based approaches, which have been applied to various domains such as image clustering, text clustering, and bioinformatics. The integration of multi-view clustering with federated learning presents unique challenges and opportunities for advancing unsupervised learning in distributed environments, which will be further explored in the subsequent sections of this review.

In Figure 3, the clustering process allowing users to compute the cluster centers and cluster assignments iteratively until convergence. The mathematical formulation of multi-view clustering can be expressed as follows:

$$\min_{U,A} \sum_{h=1}^{s} \sum_{i=1}^{n} \sum_{k=1}^{c} \mu_{ik}^{h} \|x_i^h - a_k^h\|^2 \tag{3}$$

where $A = \{a_k^h\}$ represents the cluster centers for each view h, $U = \{\mu_{ik}^h\}$ represents the cluster assignments for each data point i to cluster k in view h, and x_i^h represents the data point i in view h. The objective function aims to minimize the within-cluster variance across all views, which can be achieved

by iteratively updating the cluster centers and cluster assignments until convergence. The optimization process can be performed using various algorithms, such as k-means, spectral clustering, or fuzzy c-means, depending on the specific requirements of the multi-view clustering task and the characteristics of the data. Table 3 outline the evolution of MV learning (MVL) category of hard clustering objective functions in centralized environment from 2011 to 2024. While Table 4 presents the evolution of MVL category of centralized MVC soft clustering or “fuzzy” objective functions.

As demonstrated in Table 3, the evolution commenced with the entropy function for feature-view weights and transitioned to a Euclidean distance in Gaussian space. The simplest objective function is referred to as simultaneous weighting of views and features (SWVF), a concept proposed by Jiang *et al.* (2016) [59]. The function under consideration consists of a single-term optimization function. In contrast to Sinaga’s methodology [60], SWVF employs the fundamental Euclidean distance between data points and cluster centers in view h to assess dispersion within a specified dataset. While the parameters α and β are user-defined, achieving optimal clustering outcomes can be costly due to the fact that tuning these parameters may require a trial-and-error process to find the best combination. Furthermore, there is an absence of established guidelines concerning the composition of input datasets. The efficacy of different types of multi-view (MV) data may vary, necessitating distinct combinations to achieve the desired clustering outcomes. In contrast, TW-Co-k-means [61] incorporates an additional stage referred to as ”collaborative learning,” irrespective of its activation updating procedures for view and feature weights. Collaborative learning of MVC in a centralized environment differs from a decentralized environment involving a number of participating clients. In the context of TW-Co-k-means, the dissemination of information across feature components is facilitated by collaborative learning, thereby enabling the generation of distinct membership values for each view. For a more detailed summary of how this algorithm works, readers are referred to Table 7. The following table facilitates a more profound examination of the environment, key ideas, and main parameters for addressing MV data for each algorithm.

As illustrated in Table 4, the objective function of soft clustering for MVL in a centralized environment was examined from 2009 to 2021. The configuration of the system is subject to variation, encompassing both non-collaborative and collaborative settings, with a consideration given to the allocation of view and feature weights. The evolution of soft clustering in MV is analogous to the evolution of hard clustering in MV, albeit with distinct variations. The approach was initiated with a fundamental collaborative-based fusion strategy, which regarded all data views and feature-view components as

equally significant. After a period of five years, the system underwent an evolution that led to the consideration of view importances. In 2017, a procedure that was uncomplicated yet efficient and precise in its maintenance outcomes was introduced: Minmax-FCM. However, the model's sensitivity to the exponent parameter, which governs the distribution of view weights, has been demonstrated to impact the overall outcome. The combination of the exponent-based view weight parameter and the fuzzification parameter is contingent upon the data input. The approach of treating all feature-view components equally is not without its drawbacks; as such, this technique necessitates development. Subsequent to this, FMVC and Co-FW-MVFCM resolved the issue with or without a collaborative stage, respectively. For a more detailed summary of how these algorithms function, readers are referred to Table 6. The following table offers a comprehensive overview of the key concepts, principles, and parameters associated with each algorithm, particularly in the context of handling MV data.

Algorithm	Objective	FunctionConstraint(s)
Category: Feature-View Weighted		
TW-k-means (2011) [62]	$J = \sum_{h=1}^{s} v_h \sum_{i=1}^{n} \sum_{k=1}^{c} \mu_{ik} \sum_{j=1}^{d_h} w_j^h (x_{ij}^h - a_{kj}^h)^2 + \alpha \sum_{h=1}^{s} v_h \log v_h + \beta \sum_{j=1}^{d_h} w_j^h \log w_j^h$	$\sum_{k=1}^{c} \mu_{ik} = 1, \mu_{ik} \in \{0,1\}$, $\sum_{h=1}^{s} v_h = 1, v_h \in [0,1]$ $\sum_{j=1}^{d_h} w_j^h = 1, w_j^h \in [0,1]$
SWVF (2016) [59]	$J = \sum_{h=1}^{s} v_h^{\alpha} \sum_{i=1}^{n} \sum_{k=1}^{c} \mu_{ik} \sum_{j=1}^{d_h} \left(w_j^h\right)^{\beta} \left(x_{ij}^h - a_{kj}^h\right)^2$	Same as TW-k-means
WMCFS (2016) [63]	$J = \sum_{h=1}^{s} v_h^{\alpha} \sum_{i=1}^{n} \sum_{k=1}^{c} \mu_{ik} \left\|\mathrm{diag}(w^h)(x_i^h - a_k^h)\right\|^2 + \beta \sum_{h=1}^{s} \left\|w^h\right\|^2$	Same as TW-k-means
TW-Co-k-means (2018) [61]	$J = \sum_{h=1}^{s} v_h \left(\sum_{i=1}^{n} \sum_{k=1}^{c} \mu_{ik}^h d_{ik,j}^h \right) + \frac{\alpha}{s-1} \Delta + \eta \sum_{h=1}^{s} \sum_{j=1}^{d_h} w_j^h \text{~log~} w_j^h + \beta \sum_{h=1}^{s} v_h \text{~log~} v_h$	$\sum_{k=1}^{c} \mu_{ik}^h = 1, \mu_{ik}^h \in \{0,1\}$ $\sum_{h=1}^{s} v_h = 1, v_h \in [0,1]$ $\sum_{j=1}^{d_h} w_j^h = 1, w_j^h \in [0,1]$
FRMVK (2019) [64]	$J = \sum_{h=1}^{s} v_h^{\alpha} \sum_{i=1}^{n} \sum_{k=1}^{c} \mu_{ik} \sum_{j=1}^{d_h} w_j^h \delta_j^h \left(x_{ij}^h - a_{kj}^h\right)^2 + \frac{n}{d_h} \sum_{h=1}^{s} \sum_{j=1}^{d_h} w_j^h \text{-log-} w_j^h$	Same as TW-k-means
MVKM-ED (2024) [60]	$J = \sum_{h=1}^{s} v_h^{\alpha} \sum_{i=1}^{n} \sum_{k=1}^{c} \mu_{ik} \left\{ 1 - \exp\left(-\beta^h \sum_{j=1}^{d_h} (x_{ij}^h - a_{kj}^h)^2 \right) \right\}$	$\sum_{k=1}^{c} \mu_{ik} = 1, \mu_{ik} \in \{0,1\}$ $\sum_{h=1}^{s} v_h = 1, v_h \in [0,1]$

doi.org/10.32388/PC8X49

Algorithm	Objective	FunctionConstraint(s)
GKMVKM (2024) [60]	$J = \sum_{h=1}^{s} v_h^{\alpha} \sum_{i=1}^{n} \sum_{k=1}^{c} \mu_{ik} \left\{ 1 - \left[\exp\left(-\beta^h \sum_{j=1}^{d_h} (x_{ij}^h - a_{kj}^h)^2 \right) \right]^p \right\}$	Same as MVKM-ED
Category: Non-Negative Matrix Factorizations (NMF)		
RMKMC (2013) [65]	$J = \sum_{h=1}^{s} v_h^{\alpha} \left\| (X^h)^T - U^* (A^h)^T \right\|_{2,1}$	$\sum_{k=1}^{c} \mu_{ik}^* = 1, \mu_{ik}^* \in \{0,1\}$ $\sum_{h=1}^{s} v_h = 1, v_h \in [0,1]$
MultiNMF (2013) [66]	$J = \sum_{h=1}^{s} \left\| (X^h)^T - U^h (A^h)^T \right\|_F^2 + \sum_{h=1}^{s} \eta_h \| U^h - U^* \|_F^2$	$\sum_{k=1}^{c} \mu_{ik}^* = 1, \mu_{ik}^* \in \{0,1\}$

Table 3. Hard clustering (Non-Fuzzy) objective functions

In essence, the manifestation of distinct perspectives gives rise to disparate membership values. These memberships are then integrated to establish a global membership. Therefore, it can be posited that divergent perspectives may result in the classification of the same object in different ways. To illustrate, one perspective may encompass features derived from audio, while another may comprise features derived from images. The integration of audio and visual elements through collaborative learning enables the acquisition of contemporary information from both perspectives, fostering a comprehensive understanding. To illustrate, the audio view may ascertain that the object possesses two clusters, while the image view may determine that the object exhibits five views. However, the fusion memberships

suggest that combining the audio and image views in a collaborative manner leads the algorithm to conclude that the object has three views.

Algorithm	Objective Function	Constraint(s)
Co-FKM (2009) [67]	$J = \sum_{h=1}^{s}\sum_{i=1}^{n}\sum_{k=1}^{c}(1-\alpha)(\mu_{ik}^{h})^{m} + \frac{\alpha}{s-1}\sum_{h'\neq h}(\mu_{ik}^{h'})^{m}\|x_i^h - a_k^h\|^2$	$\sum_{k=1}^{c}\mu_{ik}^{h} = 1, \mu_{ik}^{h} \in [0,1]$
WV-Co-FCM (2014) [68]	$J = \sum_{h=1}^{s} v_h \left(\sum_{i=1}^{n}\sum_{k=1}^{c}(\mu_{ik}^{h})^{m}\|x_i^h - a_k^h\|^2 + \Delta_h\right) + \eta\sum_{h=1}^{s} v_h \log v_h$ $\Delta_h = \sum_{i=1}^{n}\alpha_{ik}^{h}\sum_{k=1}^{c}\mu_{ik}^{h}\left(1-(\mu_{ik}^{h})^{m-1}\right) - \sum_{i=1}^{n}\beta_{ik}^{h}\sum_{k=1}^{c}\mu_{ik}^{h}\left(1-(\mu_{ik}^{h})^{m-1}\right)$	$\sum_{k=1}^{c}\mu_{ik}^{h} = 1, \mu_{ik}^{h} \in [0,1]$ $\sum_{h=1}^{s} v_h = 1, v_h \in [0,1]$
MinMax-FCM (2017) [69]	$J = \sum_{h=1}^{s} v_h^{\alpha}\sum_{i=1}^{n}\sum_{k=1}^{c}(\mu_{ik}^{*})^{m}\|x_i^h - a_k^h\|^2$	$\sum_{k=1}^{c}\mu_{ik}^{*} = 1, \mu_{ik}^{*} \in [0,1]$ $\sum_{h=1}^{s} v_h = 1, v_h \in [0,1]$
FMVC (2018) [70]	$J = \sum_{h=1}^{s} v_h^{\alpha}\sum_{i=1}^{n}\sum_{k=1}^{c}\mu_{ik}\left\|\mathrm{diag}(w^h)(x_i^h - a_k^h)\right\|^2 + \beta\sum_{h=1}^{s}\left\|w^h\right\|^2$	$\sum_{k=1}^{c}\mu_{ik} = 1, \mu_{ik} \in [0,1]$ $\sum_{h=1}^{s} v_h = 1, v_h \in [0,1]$ $\sum_{j=1}^{d_h} w_j^h = 1, w_j^h \in [0,1]$
Co-FW-MVFCM (2021) [71]	$J = \sum_{h=1}^{s} v_h^{\alpha}\left(\sum_{i=1}^{n}\sum_{k=1}^{c}(\mu_{ik}^{h})^{2}(d_{ik,w}^{h})^{2} + \beta\sum_{h'\neq h}\sum_{i=1}^{n}\sum_{k=1}^{c}(\mu_{ik}^{h} - \mu_{ik}^{h'})^{2}(d_{ik,w}^{h})^{2}\right)$	$\sum_{k=1}^{c}\mu_{ik}^{h} = 1, \mu_{ik}^{h} \in [0,1];$ $\sum_{h=1}^{s} v_h = 1, v_h \in [0,1]$ $\sum_{j=1}^{d_h} w_j^h = 1, w_j^h \in [0,1]$

Table 4. Soft clustering (Fuzzy) objective functions

Algorithm	Category	Objective Function	Constraint(s)
Fed-MVFCM (2023) [47]	Soft clustering	$J = \sum_{l=1}^{L} \sum_{h=1}^{s} \left(v_{[l]}^{h}\right)^{2} \sum_{i=1}^{n(l)} \sum_{k=1}^{c} \left(\mu_{[l]ik}\right)^{m} \left\|x_{[l]i}^{h} - a_{[l]k}^{h}\right\|^{2}$	$\sum_{k=1}^{c} \mu_{[l]ik} = 1, \mu_{[l]ik} \in [0,1]$ $\sum_{h=1}^{s} v_{[l]h} = 1, v_{[l]h} \in [0,1]$
Fed-MVFPC (2023) [47]	NMF	$J = \sum_{l=1}^{L} \sum_{h=1}^{s} \left(v_{[l]}^{h}\right)^{2} \sum_{i=1}^{n(l)} \sum_{k=1}^{c} \left(\mu_{[l]ik}\right)^{m} \left\|x_{[l]i}^{h} - W^{h} p_{k}\right\|_{2}^{2}$	$\sum_{k=1}^{c} \mu_{[l]ik} = 1, \mu_{[l]ik} \in [0,1]$ $\sum_{h=1}^{s} v_{[l]h} = 1, v_{[l]h} \in [0,1]$ $W^{h^{\top}} W^{h} = I, P^{\top} P = I$
Fed-MVKM (2024) [48]	Hard clustering	$J = \sum_{l=1}^{L} \sum_{h=1}^{s} v_{[l]h}^{\alpha} \sum_{i=1}^{n(l)} \sum_{k=1}^{c} \mu_{[l]ik} \left\{1 - \exp\left(-\beta_{[l]}^{h} \left\|x_{[l]i}^{h} - a_{[l]k}^{h}\right\|^{2}\right)\right\}$	$\sum_{k=1}^{c} \mu_{[l]ik} = 1, \mu_{[l]ik} \in \{0,1\}$ $\sum_{h=1}^{s} v_{[l]h} = 1, v_{[l]h} \in [0,1]$

Table 5. Federated learning objective functions

Algorithm Name	Environment	Basic Concept(s)	Parameter(s)
Soft Clustering			
Co-FKM (2009) [67]	Collaborative centralized learning; shared-view setup; no weighting	Centralized collaborative learning without joint feature-view weighting	Views s; clusters c; fuzzification m; balancing α
WV-Co-FCM (2014) [68]	Collaborative centralized learning; view pruning; no feature selection	Centralized collaborative MV learning with view weighting only	Views s; clusters c; fuzzification m; balancing η; terms $\alpha_{ik}^h, \beta_{ik}^h$
MinMax-FCM (2017) [69]	Non-collaborative centralized learning; view pruning; no feature selection	Centralized setting with adaptive view weighting, but without irrelevant-feature filtering	Views s; clusters c; fuzzification m; exponent α
Co-FW-MVFCM (2021) [71]	Collaborative centralized learning; joint feature and view weighting; automatic irrelevant-feature suppression	Centralized collaborative MV setting with joint feature-view weighting and automatic feature reduction	Views s; clusters c
Distributed Clustering with Federated learning			
Fed-MVFCM (2023) [47]	Privacy-preserving decentralized learning; client-specific weighting; local adaptation	Privacy-preserving federated fuzzy c-means across clients	Clients L; views s; clusters c; fuzzification m; server learning rate

Table 6. Categorized summary of MVC algorithms – Soft and NMF-based clustering

Algorithm	Environment	Key idea	Main parameters
Non-collaborative centralized learning			
TW-k-means (2011) [62]	Non-collaborative centralized; joint feature-view weighting; no privacy concern	Uses entropy-based view/feature weighting; global membership from early iterations	Views s; clusters c; balancing α, β
WMCFS (2016) [63]	Same as TW-k-means	Selects informative views/features for sparse data; applies global membership early	Views s; clusters c; exponent α; balancing β
SWVF (2016) [59]	Same as TW-k-means	Bi-level feature and view weighting; global membership from early iterations	Views s; clusters c; exponents α, β
FRMVK (2019) [64]	Adaptive feature-view weighting; automatic irrelevant-feature suppression; no privacy concern	Filters redundant features by assigning low weights; improves robustness	Views s; clusters c; exponent α
MVKM-ED (2024) [60]	View weighting; no privacy concern	Uses Gaussian-kernel weighting to reflect view importance	Views s; clusters c; exponent α; regularization β_h
Category: Hard Clustering-Based Non-Negative Matrix Factorizations			
RMKMC (2013) [65]	Non-collaborative centralized learning; includes view weighting	Applies U-orthogonal NMF with $\ell_{2,1}$ norm	Views s; clusters c; exponent α
MultiNMF (2013) [66]	Collaborative centralized learning; no view weighting	Forms a consensus matrix from view-wise low-rank memberships	Views s; clusters c; non-negativity; balancing η_h
Collaborative centralized learning			
TW-Co-k-means (2018) [61]	Collaborative centralized; joint feature-view weighting; no privacy concern	Handles view disagreement via weighted membership fusion	Views s; clusters c; balancing α, η, β

Algorithm	Environment	Key idea	Main parameters
Distributed Clustering with Federated learning			
Fed-MVKM (2024) [48]	Local adaptive feature-view weighting; gradient-based federated aggregation; privacy-preserving/decentralized	Uses kernel-based weights across distributed clients	Views s; clusters c; clients L; exponent α; regularized $\beta_{[l]}^{h}$

Table 7. Categorized summary of MVC algorithms – Hard clustering

Figure 2 illustrates the multi-view clustering process in a federated learning setting, where multiple clients collaborate with a central server to train a clustering model on decentralized data. Each client has access to different views of the data, which are integrated to improve clustering performance. The central server is responsible for aggregating model updates from the clients and sharing the global model back to the clients. The clustering process involves updating cluster centers and cluster assignments iteratively until convergence, while also incorporating feedback loops for model evaluation and refinement based on the clustering results. This framework allows for effective multi-view clustering in a federated learning environment while preserving data privacy and enabling collaboration among clients. The iterative nature of the clustering process, along with the communication between clients and the central server, allows for continuous improvement of the clustering performance over time as the model learns from the data and adapts accordingly. The integration of multi-view clustering with federated learning presents unique challenges and opportunities for advancing unsupervised learning in distributed environments, which will be further explored in the subsequent sections of this review.

2.3. Decentralized MVC

As FL involves numbers of participated clients, the clustering process can be more complex and may involve various communication protocols, synchronization strategies, and privacy-preserving techniques to ensure effective collaboration among clients while maintaining data privacy. The iterative clustering process from the clients to the central server indicates that the clients are continuously updating their local models based on the clustering results and sharing these updates with the central server. The global model aggregation at the central server indicates that the server is continuously aggregating the model updates from the clients to create a global model that can be shared back with the clients for further

improvement in clustering performance. The model evaluation from the central server to the clients indicates that the central server can evaluate the performance of the global model using various metrics and provide feedback to the clients for further improvements in their local models and clustering performance. This feedback loop can help the clients to refine their local models and improve the overall clustering performance in subsequent iterations. The unified cluster refinement from the clients to reveal the final clustering results indicates that the clustering results from the clients can be used to refine the global model at the central server, which can further improve the clustering performance in subsequent iterations. This feedback loop can help the central server to continuously improve the global model based on the clustering results from the clients, leading to better clustering performance over time.

The mathematical formulation of multi-view clustering in a federated learning setting can be expressed as follows:

$$\min_{U,A} \sum_{l=1}^{L} \sum_{h=1}^{s(l)} \sum_{i=1}^{n(l)} \sum_{k=1}^{c(l)} \mu_{[l]ik}^{h} \| x_{[l]i}^{h} - a_{[l]k}^{h} \|^2 \tag{4}$$

where $A = \{a_{[l]k}^{h}\}$ represents the cluster centers of h view held by client l, $U = \{\mu_{[l]ik}^{h}\}$ represents the cluster assignments for each data point i to cluster k in view h held by client l, $x_{[l]i}^{h}$ represents the data point i in view h held by client l, $n(l)$ is the number of data points held by client l, $c(l)$ is the number of clusters held by client l, and $s(l)$ is the number of views located at client l. The objective function aims to minimize the within-cluster variance across all views and clients, which can be achieved by iteratively updating the cluster centers and cluster assignments until convergence while accounting for the decentralized nature of the data and communication constraints in federated learning. The optimization process can be performed using various algorithms, such as federated k-means, federated spectral clustering, or federated fuzzy c-means, depending on the specific requirements of the multi-view clustering task and the characteristics of the data in the federated learning setting. The integration of multi-view clustering with federated learning presents unique challenges and opportunities for advancing unsupervised learning in distributed environments, which will be further explored in the subsequent sections of this review.

In the context of FL, the optimization process can be adapted to account for the decentralized nature of the data and communication constraints between clients and the central server. This adaptation may entail the implementation of additional steps for aggregating model updates and disseminating the global model across clients while maintaining data privacy. Table 5 presents the objective functions of MVL for both hard and soft clustering categories in a decentralized setting. As indicated in Table 5, the

extension of MVC in a federated setting commenced in late 2023. The number of participating clients is considered to unify the clustering outcomes from multi-view data distributed across different locations. The number of participating clients during the clustering process is denoted as L, and the number of data points stored in client l is denoted as $n(l)$. Fed-MVFCM, Fed-MVFPC, and Fed-MVKM are classified as horizontal federated learning (HFL), and they adopt three variables utilized in centralized settings, as outlined in Tables 3 and ??. The aforementioned variables are designated as the memberships μ_{ik}^{h}, cluster centers a_{kj}^{h}, and view weights v_h. In this context, it is assumed that all participating clients are accounts, and each account has the same number of views, despite the variability in the number of features in each view. Consequently, the number of clusters treated identically for all participating clients' initialization procedures. The distribution of view weights may exhibit local variations. It is possible that a single client may assign equal weights to all views, or conversely, may assign unequal weights to all views.

2.4. MVC in FL: How to Integrate MVC with FL ?

The integration of multi-view clustering with federated learning involves several key steps and considerations to ensure effective collaboration among clients while preserving data privacy. First, each client needs to perform local clustering on their respective views of the data, which can be achieved using various clustering algorithms such as k-means, spectral clustering, or fuzzy c-means. The local clustering results can then be shared with the central server in the form of model updates, such as cluster centers and cluster assignments, without sharing raw data. The central server can then aggregate the model updates from all clients to create a global model that can be shared back with the clients for further refinement. This iterative process allows for continuous improvement of the clustering performance over time as the model learns from the data and adapts accordingly. Additionally, the central server can evaluate the performance of the global model using various metrics and provide feedback to the clients for further improvements in their local models and clustering performance. This feedback loop can help the clients to refine their local models and improve the overall clustering performance in subsequent iterations. The integration of multi-view clustering with federated learning also requires careful consideration of communication protocols, synchronization strategies, and privacy-preserving techniques to ensure effective collaboration among clients while maintaining data privacy. Overall, the integration of multi-view clustering with federated learning presents unique challenges and opportunities for advancing unsupervised learning in distributed environments, which will be further explored in the subsequent sections of this review.

For example, [48] proposed a federated multi-view k-means (Fed-MVKM) algorithm that allows clients to perform local k-means clustering on their respective views of the data and share model updates with a central server for aggregation. The central server then averages the cluster centers from all clients to create a global model, which is shared back with the clients for further refinement. The algorithm also incorporates a feedback loop for model evaluation and refinement based on the clustering results, allowing for continuous improvement of the clustering performance over time. This approach demonstrates how multi-view clustering can be effectively integrated with federated learning to enable unsupervised learning in distributed environments while preserving data privacy. Another example is [47] proposed federated multiview fuzzy C-means consensus prototypes clustering (FedMVFPC) algorithm that allows clients to perform local fuzzy C-means clustering on their respective views of the data and share model updates with a central server for aggregation. The central server then combines the cluster centers from all clients to create a global model, which is shared back with the clients for further refinement. The algorithm also incorporates a feedback loop for model evaluation and refinement based on the clustering results, allowing for continuous improvement of the clustering performance over time. Table 8 provides a summary of recent developments in multi-view clustering with federated learning, highlighting the key features and contributions of each approach. These examples illustrate how multi-view clustering can be effectively integrated with federated learning to enable unsupervised learning in distributed environments while preserving data privacy, and they also highlight the unique challenges and opportunities associated with this integration.

References (year)	Algorithm	Key Features	FL Settings	Complexity Level
[46] (2023)	FedDMVC	Autoencoder-based fuzzy clustering	HFL	Medium
[47] (2023)	FedMVFPC	Local fuzzy C-means clustering	HFL	Medium
[48] (2024)	Fed-MVKM	Local k-means clustering	HFL	Medium
[49] (2024)	FMVFCMSP	Local FCM with schatten p-norm regularization	HFL	High
[72] (2024)	FMVC-IMK	Factorized local k-means clustering	HFL	High
[73] (2025)	Vertical MVFC	Local FCM vertical clustering	VFL	High
[74] (2026)	V-HDKM	Local k-means with upper bound of the loss function	VFL	High

Table 8. Summary of recent developments in multi-view clustering with federated learning. This table highlights the key features and contributions of each approach, demonstrating how multi-view clustering can be effectively integrated with federated learning to enable unsupervised learning in distributed environments while preserving data privacy.

In conclusion, the integration of MVC with FL is the reconstruction of the objective function of MVC in its baseline formulation to fit the FL setting, which requiring the consideration of the decentralized (non-iid) data distribution across clients, the communication constraints between clients and the central server, and the privacy-preserving techniques to ensure effective collaboration among clients while maintaining data privacy. The optimization process can be performed using various algorithms, such as federated k-means, federated spectral clustering, or federated fuzzy c-means, depending on the specific requirements of the multi-view clustering task and the characteristics of the data in the federated learning setting.

2.5. Challenges and Opportunities in MVC with FL

In the domain of traditional engineering optimization tasks and distributed unsupervised learning scenarios, algorithms continue to grapple with persistent challenges related to population or observation quality, search stability, and global optimization capability. The pervasive reliance on random initialization in the majority of population-based methods often leads to uneven population distributions, resulting in incomplete coverage of the search space, particularly in high-dimensional data views. This limitation becomes particularly pronounced in real-world applications where access to distributed data is inherently constrained by privacy concerns.

To address the issue of restricted data availability, sub-space methodologies have been introduced to partition MVC datasets across client nodes. However, the generation of distributed datasets on each client is highly sensitive to specific configuration parameters, which introduces significant variability in the resulting sub-groups. Consequently, repeated executions may yield different parttitions in both the feature space and the number of observations. It is well established that the performance of machine learning models is heavily contingent upon the quality of input data. As a result, some clients may inadvertently host low-quality MVC datasets due to internal system deficiencies, whereas others benefit from robust data collection pipelines that facilitate higher-quality subsets. This heterogeneity in data quality can severely impair the overall efficacy of the distributed optimization process.

Another fundamental challenge lies in balancing the trade-off between exploration and exploitation. Excessive exploration tends to decelerate convergence, while overemphasis on exploitation risks premature convergence and entrapment in local optima. The situation is further complicated by the use of static or weakly adaptive parameters, which struggle to accomodate the varying demands of different search phases. This mismatch can ultimately undermine both convergence accuracy and algorithmic robustness.

2.5.1. Illustration: LLM-based Intelligence

In the context of the challenges posed by the MVC architecture on clients, it is imperative to analyze the prevailing trends in LLM-based approaches and agents. These approaches have been demonstrated to possess the remarkable ability to process an extensive volume of information and address a spectrum of problems, ranging from rudimentary to intricate. Presently, LLM-based approaches are being deployed across diverse nations and continents. It is noteworthy that the United States possesses a number of advanced technologies in this domain. Switzerland has recently developed a technology known as

SwissGPT, Japan has its own localized versions, Mainland China possesses some of this intelligence technology, and numerous other countries are also making significant strides in this area.

When confronted with the task of solving a mathematical problem, each of these agents will generate a distinct solution that may be identical. The outcomes produced by each LLM-based agent are significantly influenced by the data or information that their system has access to or has been provided with. In this context, the classification of these agents into tiers of superiority, medium, and low based on their aptitude for completing a task is contingent upon the quality of the data features to which they have access. These data features may include multimedia datasets, videos, audio, images, journal or book repositories, and more, in not only one language but multiple languages. A considerable number of researchers have endeavored to categorize these agents according to various factors and criteria, with the objective of addressing a singular issue. The classification of superior intelligence is contingent upon the types of information to which the subjects have access, the advanced and updated internal systems supported by the most sophisticated hardware and software technologies, and other relevant factors. The aforementioned factors have been shown to expedite the performance of one agent in addressing a given problem, characterized by substantial convergence and optimal iterations. The degree to which a system or method is consistent is of the utmost importance to its success. The degree of consistency can be determined through the personal exploration and exploitation of the system by its users. In essence, agents with superior quality will consistently emerge as the ones who possess access to an abundance of high-quality information.

Collectively, these considerations underscore the complexity of achieving reliable and scalable optimization in distributed MVC environments. Subsequent research endeavors must concentrate on mechanisms that augment population diversity, harmonize exploration-exploitation strategies, and implement adaptive parameter control to enhance search performance under heterogeneous and privacy-constrained conditions.

2.5.2. Conceptual: Privacy Preservation

Each client may have different data distributions, which can lead to non-iid (independent and identically distributed) data across clients. This heterogeneity can affect the convergence and performance of the global model, as the model updates from different clients may not be directly comparable or may have varying levels of contribution to the global model. To address this challenge, various techniques can be employed, such as incorporating proximal terms in the optimization process (as in FedProx) to handle

heterogeneity among clients, or using adaptive learning rates (as in FedOpt) to optimize the aggregation process based on the variability in model updates from clients. Additionally, privacy-preserving techniques, such as differential privacy or secure multi-party computation, can be integrated with multi-view clustering in federated learning to ensure data privacy while still enabling effective collaboration among clients. The integration of multi-view clustering with federated learning also presents opportunities for advancing unsupervised learning in distributed environments, as it allows for leveraging the complementary information from multiple views of the data while preserving data privacy and enabling collaboration among clients. This can lead to improved clustering performance and new insights into the underlying structure of the data across different views and clients. However, it also requires careful consideration of the communication protocols, synchronization strategies, and privacy-preserving techniques to ensure effective collaboration among clients while maintaining data privacy. Overall, the integration of multi-view clustering with federated learning presents unique challenges and opportunities for advancing unsupervised learning in distributed environments, which will be further explored in the subsequent sections of this review.

Figure 4 illustrates the comparison between multi-view clustering in a federated learning setting (Figure 2) and a non-federated environment (Figure 3). In the federated learning setting, data is decentralized across multiple clients, and the clustering process involves communication between clients and a central server to update the global model iteratively. In contrast, in the non-federated environment, data is centralized, and the clustering model can directly access all views of the data without any communication constraints. This comparison highlights the unique challenges and opportunities associated with multi-view clustering in federated learning compared to traditional clustering approaches in a non-federated setting. The decentralized nature of data and communication in federated learning requires careful consideration of aggregation strategies, synchronization protocols, and privacy-preserving techniques to ensure effective collaboration among clients while maintaining data privacy. On the other hand, the centralized data access in a non-federated environment allows for direct integration of multiple views for clustering but may raise privacy concerns. Overall, this comparison provides insights into the differences in data flow, model training, and clustering processes between the two frameworks, which can inform future research directions in multi-view clustering with federated learning.

2.5.3. Comparison of MVC in FL and Non-Federated Environment

Non-federated multi-view clustering is a traditional approach where all data is centralized, and the clustering model has access to all views of the data without any privacy constraints. In this setting, the clustering process can directly integrate information from multiple views to improve clustering performance. However, this approach may raise privacy concerns, as it requires sharing raw data across different views and clients. In contrast, multi-view clustering in a federated learning setting allows for effective unsupervised learning in distributed environments while preserving data privacy. In this setting, clients can perform local clustering on their respective views of the data and share model updates with a central server for aggregation, without sharing raw data. The concept is to enable collaboration among clients while maintaining data privacy, which can lead to improved clustering performance by leveraging the complementary information from multiple views of the data. However, it also presents unique challenges, such as handling heterogeneity among clients, designing effective aggregation strategies, and ensuring privacy-preserving techniques are in place. Overall, the comparison between multi-view clustering in federated learning and non-federated environments highlights the trade-offs between data privacy and clustering performance, and underscores the importance of developing novel approaches to effectively integrate multi-view clustering with federated learning for advancing unsupervised learning in distributed environments.

Table 9 provides a summary of the key differences between multi-view clustering in a federated learning setting and a non-federated environment. The table highlights the differences in data access, communication, privacy concerns, and clustering performance between the two frameworks. In the federated learning setting, data is decentralized across multiple clients, and communication is required between clients and a central server for model updates and aggregation. Privacy concerns are addressed through techniques such as differential privacy or secure multi-party computation. Clustering performance can be improved by leveraging complementary information from multiple views while preserving data privacy. In contrast, in a non-federated environment, data is centralized, and there is no need for communication between clients and a central server. However, privacy concerns may arise due to the need to share raw data across different views and clients. Clustering performance can be improved by directly integrating information from multiple views without any communication constraints. Overall, this comparison provides insights into the trade-offs between data privacy and clustering performance in multi-view clustering with federated learning compared to traditional clustering approaches in a non-federated setting.

Aspect	Federated Learning	Non-Federated Environment
Data access	Decentralized; data remains on clients	Centralized; all views available in one location
Complexity	Higher (communication, coordination, privacy)	Lower (direct access to data)
Convergence	May be slower due to communication overhead and client heterogeneity	Typically faster due to direct access to all data
Memory usage	Lower on clients (local processing)	Higher (centralized storage and processing)
Communication	Required for model updates and aggregation	Not required between clients and a central server
Privacy concerns	Addressed via differential privacy, secure MPC, etc.	Higher risk (raw data sharing across views/clients)
Clustering performance	Can improve by leveraging complementary views while preserving privacy	Can improve by integrating all views without communication constraints
Effectiveness	Depends on aggregation, synchronization, and privacy mechanisms	Typically effective given full data access
Challenges	Handling non-IID data, dynamic client participation, and heterogeneous feature spaces	Privacy risks and potential overfitting from centralized data

Table 9. Comparison of Multi-View Clustering in Federated Learning and Non-Federated Environment

Yang *et al.* [75] outlined three architectural frameworks in the area of federated learning systems: These models comprise horizontal federated learning (HFL), vertical federated learning (VFL), and federated transfer learning (FTL), associated with a blockchain concern. In HFL, ”horizontal” originates from ”horizontal partitioning,” often associated with the traditional tabular format typical of database architectures [76]. For example, the rows of a table are split horizontally into various sections, with each section containing all the required data attributes. VFL presents a model where data is divided by feature space, retaining the same observations while integrating different features [77]. In comparison, FTL is a

hybrid approach that merges federated learning with transfer learning to tackle scenarios where the participating entities lack adequate shared features or samples [78]. Unlike traditional federated learning models, like HFL and VFL, that assume a common feature or sample space, FTL enables collaboration in situations where these assumptions do not hold true.

The concept of FL in MVC also highly depends on the number of participated clients and the heterogeneity of data across clients, which can lead to non-iid (independent and identically distributed) data distributions and affect the convergence and performance of the global model. So far, the MVC in FL is assigned a fixed number of clients during the clustering process, which may not reflect the real-world scenarios where the number of clients can vary over time. In addition to that, most MVC in FL using a horizontal FL seeting, where clients have the same feature space but different data samples. However, in real-world scenarios, clients may have different feature spaces (vertical FL) or a combination of both (federated transfer learning), which can further complicate the clustering process and require more sophisticated approaches to effectively integrate multi-view clustering with federated learning. Therefore, future research in this area should consider the dynamic nature of client participation and the heterogeneity of data across clients to develop more robust and scalable multi-view clustering algorithms for federated learning environments. Table 10 provides a summary of the different FL settings (horizontal FL, vertical FL, and federated transfer learning) and their implications for multi-view clustering in federated learning.

FL Setting	Description	Implications for Multi-View Clustering
Horizontal FL	Clients share the same feature space but hold different data samples.	Algorithms must be robust to non-IID data distributions across clients; aggregation should account for sample heterogeneity.
Vertical FL	Clients hold different feature spaces for (often) the same set of samples.	Methods need to align and integrate complementary feature spaces while preserving privacy (e.g., secure aggregation, split learning).
Federated Transfer Learning	Mix of horizontal and vertical settings or limited feature/sample overlap across clients.	Requires hybrid approaches that handle both non-IID samples and heterogeneous feature spaces; transfer and representation learning techniques are useful.

Table 10. Summary of FL Settings and Implications for Multi-View Clustering in Federated Learning

As reported in Table 10, the different FL settings (horizontal FL, vertical FL, and federated transfer learning) have different implications for multi-view clustering in federated learning. In horizontal FL, where clients have the same feature space but different data samples, algorithms need to be designed to handle non-iid data distributions across clients, which can affect the convergence and performance of the global model. In vertical FL, where clients have different feature spaces but the same data samples, algorithms need to be developed to effectively integrate information from different feature spaces, which can be challenging due to the heterogeneity of data across clients. In federated transfer learning, which is a combination of horizontal and vertical FL, algorithms need to be designed to handle both non-iid data distributions and different feature spaces across clients, which can further complicate the clustering process and require more sophisticated approaches. Therefore, future research in multi-view clustering with federated learning should consider these different FL settings and their implications for developing robust and scalable algorithms that can effectively integrate multi-view clustering with federated learning in real-world scenarios. These implications highlight the need for developing novel approaches that can handle the challenges associated with different FL settings and effectively leverage the complementary information from multiple views of the data while preserving data privacy and enabling collaboration among clients in federated learning environments.

The aggregation strategies for these three FL settings may also differ, as the communication protocols, synchronization strategies, and privacy-preserving techniques may need to be tailored to the specific characteristics of each FL setting. For example, in horizontal FL, aggregation strategies may need to account for the non-iid data distributions across clients and design algorithms that can effectively combine model updates from clients with different data samples. In vertical FL, aggregation strategies may need to focus on integrating information from different feature spaces and designing algorithms that can effectively combine model updates from clients with different feature spaces. In federated transfer learning, aggregation strategies may need to handle both non-iid data distributions and different feature spaces across clients, which can require more sophisticated approaches to effectively integrate model updates from clients in a way that preserves data privacy and enables collaboration among clients. Therefore, future research in multi-view clustering with federated learning should also consider the implications of different FL settings for designing effective aggregation strategies that can enhance clustering performance while preserving data privacy in federated learning environments.

2.5.4. Aggregation Strategies for Model Updates in FL

In FL environments, the aggregation of model updates from clients to the central server is a critical step that can significantly impact the performance of the global model. Various aggregation strategies have been proposed to effectively combine model updates while preserving privacy and improving clustering performance. Figure 5 illustrates different aggregation strategies, including federated averaging (FedAvg) [13], federated proximal (FedProx) [79], and federated optimization (FedOpt) [80], which are commonly used in federated learning for multi-view clustering. FedAvg averages model updates from clients, FedProx incorporates proximal terms to handle heterogeneity among clients, and FedOpt optimizes the aggregation process using adaptive learning rates. The mathematical formulations for each aggregation strategy are also provided to illustrate the underlying principles and mechanisms of these strategies in the context of multi-view clustering in federated learning. Additionally, the figure includes a standard deviation aggregation method to illustrate the variability in model updates from clients and how it can inform the aggregation process. The clustering aggregation method is also highlighted to show how the clustering results from different clients can be combined to improve overall clustering performance in the federated learning setting. These aggregation strategies play a crucial role in enabling effective multi-view clustering while preserving data privacy and facilitating collaboration among clients in federated learning environments.

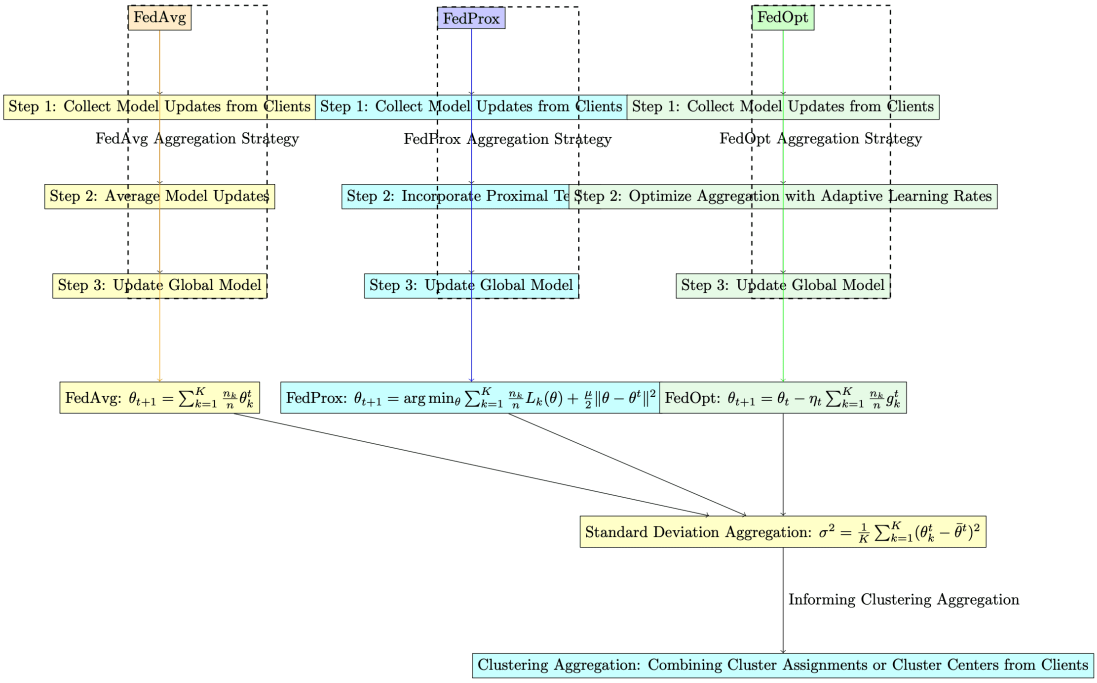

Figure 5. Illustration of different aggregation strategies for model updates from clients to central server in federated learning. The scripts can be used to demonstrate how different aggregation methods, such as FedAvg, FedProx, and FedOpt, can be employed to combine model updates from multiple clients while preserving privacy and improving clustering performance.

2.6. Implementation of FedAvg, FedProx, and FedOpt in MVC

Fuzzy c-means (FCM) and k-means are two popular clustering algorithms that can be adapted for this purpose. In the context of multi-view clustering in federated learning, these algorithms can be implemented in a distributed manner, where each client performs local clustering on their respective views of the data and shares model updates with the central server for aggregation.

For example, in the case of FedAvg, each client can perform local clustering using FCM or k-means on their respective views of the data and then share the cluster centers and cluster assignments with the central server. The central server can then average the cluster centers and cluster assignments from all clients to create a global model that can be shared back with the clients for further refinement. In the case of FedProx, each client can perform local clustering while incorporating proximal terms to handle heterogeneity among clients, and then share the model updates with the central server for aggregation. The central server can then optimize the aggregation process using adaptive learning rates in the case of

FedOpt to improve the performance of the global model. These implementations allow for effective multi-view clustering in a federated learning setting while preserving data privacy and enabling collaboration among clients. What we need is the gradient descent optimization for the clustering process, which can be achieved by updating the cluster centers and cluster assignments iteratively based on the model updates from the clients and the global model aggregation at the central server. The specific implementation details may vary depending on the characteristics of the data, the number of clients, and the communication constraints in the federated learning environment.

The gradient descent optimization for the clustering process can be expressed as follows:

$$c^v_{[m]j} = c^v_{[m]j} - \eta \sum_{i=1}^{N(m)} a^v_{[m]ij} (x^v_{[m]i} - c^v_{[m]j}) \tag{5}$$

Eq. 5 represents the update of the cluster centers based on the model updates from the clients and the global model aggregation at the central server. The learning rate η controls the step size of the update, and the summation term represents the contribution of each data point to the update of the cluster center based on its cluster assignment. This iterative update process continues until convergence, where the cluster centers and cluster assignments stabilize, indicating that the clustering process has reached a local optimum. The specific implementation details may vary depending on the characteristics of the data, the number of clients, and the communication constraints in the federated learning environment. The gradient descent optimization can be performed in a distributed manner, where each client updates their local cluster centers and cluster assignments based on their respective views of the data, and then shares the model updates with the central server for aggregation and further refinement of the global model. This iterative process allows for effective multi-view clustering in a federated learning setting while preserving data privacy and enabling collaboration among clients.

2.7. Challenges and Considerations in Aggregation Strategies

While FedAvg, FedProx, and FedOpt are effective aggregation strategies for model updates in federated learning, there are several challenges and considerations that need to be addressed when implementing these strategies for multi-view clustering. One of the main challenges is the heterogeneity of data across clients, which can lead to non-iid (independent and identically distributed) data distributions and affect the convergence and performance of the global model. FedProx addresses this challenge by incorporating proximal terms to handle heterogeneity among clients, but it may still require careful tuning of hyperparameters to achieve optimal performance. Additionally, the communication overhead and

privacy concerns in federated learning can also impact the effectiveness of these aggregation strategies, as clients may be reluctant to share model updates that contain sensitive information. Therefore, privacy-preserving techniques, such as differential privacy or secure multi-party computation, may need to be integrated with these aggregation strategies to ensure data privacy while still enabling effective collaboration among clients. Furthermore, the choice of clustering algorithm and the specific implementation details of the clustering process can also influence the performance of the global model, and may require additional considerations when designing the aggregation strategies for multi-view clustering in federated learning. Overall, while FedAvg, FedProx, and FedOpt are powerful aggregation strategies for model updates in federated learning, careful consideration of the challenges and constraints in the federated learning environment is essential for achieving effective multi-view clustering performance while preserving data privacy and enabling collaboration among clients.

3. Applications: The Hypothetical Case Studies

In this section, I will discuss the various applications of multi-view clustering in federated learning across different domains, such as healthcare, finance, and social media. I will highlight how the integration of multi-view clustering with federated learning can enable effective data analysis and insights while preserving data privacy and enabling collaboration among clients in these domains. I will also provide examples of specific use cases and scenarios where multi-view clustering in federated learning has been successfully applied to address real-world challenges and improve decision-making processes. Additionally, I will discuss the potential benefits and limitations of using multi-view clustering in federated learning for different applications, and provide insights into future research directions for further advancing this field.

3.1. Healthcare

In the medical domain, Yuan *et al.* [81] developed an end-to-end model capable of collectively learning multi-view features from supervised seizure detection using spectrogram representation and unsupervised, multi-channel electroencephalogram (EEG) reconstruction. Yang *et al.* [82] proposed a multi-view, multi-scale convolutional neural network for ECG classification, leveraging diverse views of ECG data at various sizes to capture both local and global cardiac patterns. To identify cancer subtypes and comprehend the heterogeneity of cancer based on various omics data, Liu *et al.* [83] developed a Multi-view Spectral Clustering Based on Multi-smooth Representation Fusion (MRF-MSC). In order to

enhance the precision of disease diagnosis, Puyol-Antn *et al.* [84] employed a multi-view learning strategy to identify dilated cardiomyopathy by integrating cardiac imaging data from multiple sources. To anticipate drug-induced QT prolongation, Zhang *et al.* [85] developed a multi-view learning strategy using L1-norm co-regularization. In order to diagnose and analyze instances of the novel coronavirus known as SARS-CoV-2, Dornaika *et al.* [86] introduced a methodology termed one-step multi-view clustering by estimating constrained cluster assignment matrix (OMVC). In conclusion, a comprehensive analysis of medical data can offer valuable insights into various aspects of a patient's health or the management of a particular illness. The objective of multi-view learning strategies is to effectively incorporate these diverse perspectives to facilitate a comprehensive understanding and enhance the medical systems' capacity for diagnosis and prediction.

In the healthcare domain, multi-view clustering in federated learning can be used for patient stratification, disease subtyping, and personalized treatment recommendations. For example, different hospitals or medical institutions may have their own datasets with different views of patient data, such as electronic health records, medical imaging, and genomic data. By using multi-view clustering in a federated learning setting, these institutions can collaborate to identify meaningful clusters of patients based on their multi-view data while preserving patient privacy. This can lead to improved understanding of disease heterogeneity, identification of novel disease subtypes, and development of personalized treatment strategies based on the clustering results. The scenario of multi-view clustering in federated learning can enable healthcare providers to leverage the collective knowledge from multiple institutions while ensuring that sensitive patient data remains secure and private.

Figure 6 illustrates the application of multi-view clustering in federated learning for healthcare. In this scenario, multiple hospitals collaborate with a central server to perform multi-view clustering on their respective patient datasets, which may include electronic health records, medical imaging, and genomic data. The central server aggregates the model updates from the hospitals and shares the global model back to the hospitals for further refinement. The clustering results can be used for patient stratification, disease subtyping, and personalized treatment recommendations, leading to improved healthcare outcomes while preserving patient privacy.

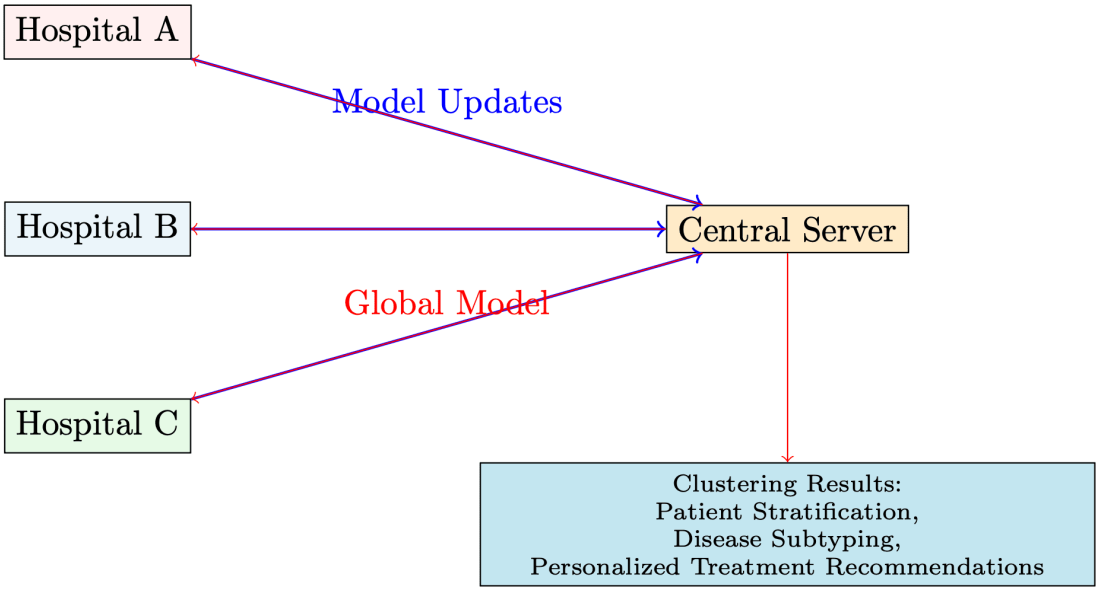

Figure 6. Application of multi-view clustering in federated learning for healthcare. Multiple hospitals collaborate with a central server to perform multi-view clustering on their respective patient datasets, which may include electronic health records, medical imaging, and genomic data. The clustering results can be used for patient stratification, disease subtyping, and personalized treatment recommendations while preserving patient privacy.

3.2. Finance

In the financial context, MVC has formulated a set of visual interface designs that allow users to enrich data-driven clustering results with knowledge-driven clustering exploration when identifying dynamic financial clusters. Zheng *et al.* [87] presented FAIR-MVC, an MVC that is aware of fairness and utilizes contrastive and non-contrastive learning to manage heterogeneous data in complex scenarios with missing or noisy data. The model addresses heterogeneous data, which includes not only total debt and yearly income, but also text and images, such as financial statements and invoices, which may lack label information (e.g., fraud transactions) for building predictive models. In 2025, Kam-Kwai *et al.* [88] proposed Prismatic, a module consisting of three procedures: computing yearly correlation networks, mining business relational knowledge, and graph-based MVC (GMC) with a multi-layer network for business relational knowledge. The experimental phase entailed the exclusive utilization of quantitative data, encompassing the analysis of time-sensitive market dynamics through the medium of expert interviews.

To illustrate HFL, consider the case of two local banks. These banking institutions may possess distinct user demographics within their respective geographic areas, resulting in minimal overlap between their user bases. However, a thorough examination reveals that their business models are strikingly similar. Consequently, the characteristics of their datasets are identical. Within the domain of finance in VFL, two financial institutionsfor example, a bank and an insurance companymay provide services to a considerable number of the same customers. This phenomenon results in a significant degree of overlap between their respective user bases. However, each institution collects different types of information about these customers based on its business operations. For instance, financial institutions maintain account balances and transaction histories, while insurance companies store policy details and claims records. Consequently, while the user sets largely overlap, the feature spaces differ. TFL is a combination of HFL and VFL. Within the domain of finance, two financial organizations may operate in disparate markets and offer a variety of services, resulting in minimal overlap in their customer bases and the features they provide. To illustrate, a domestic bank and an overseas lending platform may have only a few common customers while maintaining distinct data attributes. Despite these discrepancies, knowledge acquired from one institution can be transferred to enhance the learning performance of the other without the need for an exchange of raw data. Consequently, both the user overlap and the feature overlap are limited, rendering transfer learning essential. For summary, Table 11 is detailing the differences of these three architectural systems from different aspects.

FL Type	User Overlap	Feature Overlap	Typical Financial Example
HFL	Low	High	Two regional banks are in the business of offering similar services, yet they have different clienteles.
VFL	High	Low	A banking institution and an insurance company that provide services to the same clientele possess disparate sets of data.
FTL	Low	Low	A domestic bank and a foreign lending platform are distinguished by their distinct customer bases and features.

Table 11. A Conceptual Analysis of the Implementation of FL Architectures inSystem Design

doi.org/10.32388/PC8X49

3.3. Social Media

This paper will provide social media analogical approaches to illustrate the application of MVC in FL. These approaches demonstrate the incorporation of multiple perspectives (MP) on the same user and multiple resources (i.e., different platforms or organizations) within three FL paradigms: HFL, VFL, and FTL.

3.3.1. MVC via HFL (MVC-HFL) : Social Media Across Different Platforms

It is hypothesized that three regional social media platforms function independently: AsiaNet boasts a sizable and heterogeneous user base in Asia; USLink has users with high levels of engagement in the United States; and EuroConnect has users across Europe. The user bases of these platforms are largely distinct, as are their regional users IDs and the number of accounts that can be used across different platforms. These platforms collect a wide range of data, including user profiles, posts, likes, comments, friendships/connections, and engagement statistics (horizontal setting). However, it should be noted that each user's data naturally exists in multiple complementary views. These complementary perspectives encompass a textual perspective, visual perspective, network perspective, and a behavioral perspective. Table 12 offers a comprehensive overview of these cross-platform views from AsiaNet, UsLink, and EuroConnect. Fig. 7 illustrates the application of the MVC framework via the HFL approach for cross-platform target communities, which encompasses prominent social media such as AsiaNet, UsLink, and EuroConnect.

No	View's Name	Feature Component
1	Textual	Content from posts, comments, and bios (topics, sentiment, keywords)
2	Visual	Images, videos, likes on media, and visual engagement patterns
3	Network	Friendship graphs, community structures, and interaction networks
4	Behavioral	Engagement metrics (time spent, frequency, interaction types, temporal patterns)

Table 12. Summary of the artifical features of AsiaNet, UsLink, and EuroConnect in MV scenarios

During the local MVC phase on each platform stage, each platform performs local MVC on its own users without sharing raw data. The identification of user communities by each platform is achieved through the implementation of techniques such as MV k-means, MV FCM, MV spectral clustering, co-training, and deep MV autoencoders. These techniques work in concert to leverage consensus and complementarity across the four view, thereby facilitating the accurate and effective identification of user communities. The target communities (which may differ slightly per region due to cultural nuances) include sports fans, gamers, travelers/adventure seekers, food lovers/culinary enthusiasts, technology and innovation enthusiasts, and environmental activists. For each community, the platform learns cluster centers or representative embeddings per view, view-specific importance weights (e.g., the visual view may dominate for activists), and intra-cluster similarity patterns and cross-view agreement metrics.

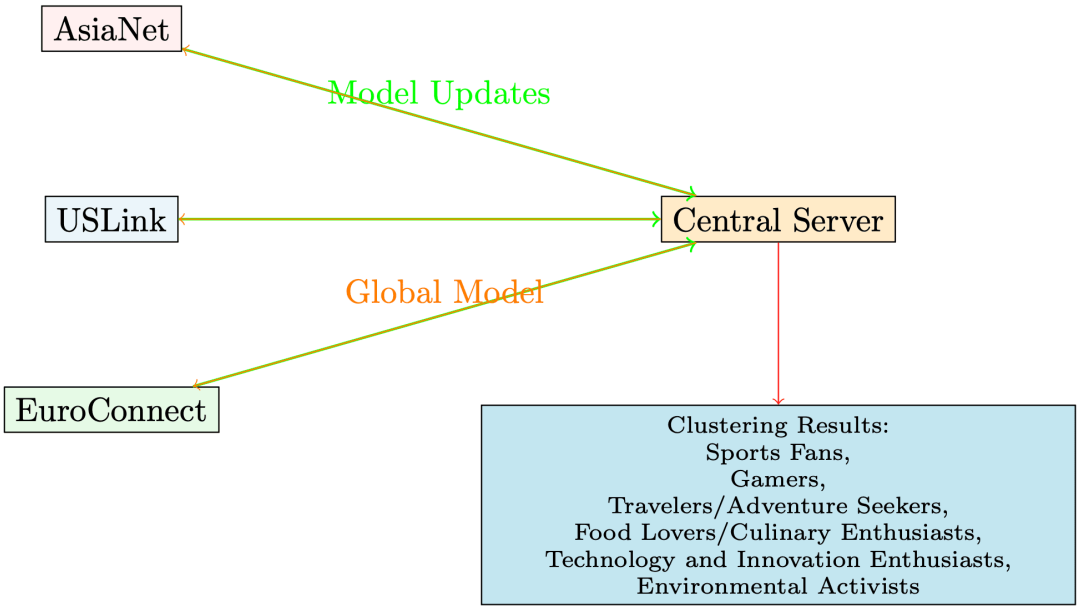

Figure 7. Application of multi-view clustering in HFL for Social Media.

In the phase of federated server for aggregation, the neutral federated server or coordinator orchestrates four steps in the process. The initial step entails the training of a local MVC model by each platform on its proprietary data. Subsequently, model updates are extracted, including aggregated cluster centers, view-consensus embeddings, and parameters of a shared clustering centers. It is imperative to note that raw user data or individual profiles are never involved in this process. Subsequently, these updates are

transmitted to the federated server via secure aggregation (e.g., FedAvg variants adapted for clustering, or federated MV fusion methods). At the third step, the server aggregates the knowledge to build global consensus clusters that generalize across regions. It facilitates the formation of communities of similar interests, such as “sports fans” from Asia, the United States, and Europe, while respecting the diversity of viewpoints among these groups. The refined global model and parameters are ultimately transmitted to each platform, thereby enhancing their local clustering. For instance, the AsiaNet system is able to more accurately identify emerging global sub-communities, such as “sports enthusiasts,” by leveraging the stronger gaming signals observed in USLink, while maintaining the anonymity of individual users.

3.3.2. MVC via VFL (MVC-VFL) for Cross-Platform User Profiling

In the contemporary digital landscape, users engage in interactions across diverse platforms, including Instagram (visual interface featuring components such as photos, stories, reels, and visual embeddings), Twitter/X (textual interface comprising tweets, hashtags, mentions, and text embeddings), Spotify (audio interface displaying genres, artists, and listening history), and YouTube (video interface encompassing watch history, subscriptions, and viewing time). Each platform holds a distinct vertical slice of features for the same set of users (aligned by user IDs or pseudonymous identifiers), but raw data sharing is prohibited due to privacy regulations (GDPR, CCPA) and competitive concerns. The objective is to conduct unsupervised discovery of latent user populations that may be categorized as entertainment enthusiasts, technology followers, fitness community, and travelers. Entertainment enthusiasts are expected to exhibit high engagement across visual reels, music streaming, and video consumption. Technology followers are anticipated to engage in textual discourse on technology topics, gadget reviews on YouTube, and niche audio content. The fitness community is characterized by the consumption of various forms of media, including workout videos and reels, motivational audio playlists, and related social posts. The estimation of travelers is derived from their consumption of visual content, travel vlogs, destination music, and exploratory content. As illustrated in Figure 8, the application of the MVC framework via the VFL approach is employed for the purpose of cross-platform user profiling, encompassing prominent social media and music streaming platforms such as Instagram, Twitter/X, Spotify, and YouTube.

MVC utilizes complementary information from diverse modalities to generate more robust clusters than single-view methods. By integrating VFL, collaborative model training becomes feasible, where parties (platforms) share the same sample space but distinct feature spaces. Communication between parties is restricted to intermediate representations or gradients, safeguarded by cryptographic measures, thereby

preventing the exposure of raw data. This approach effectively addresses critical challenges such as feature heterogeneity, view incompleteness (as not all users are active on every platform), non-IID data distributions, and stringent privacy requirements.

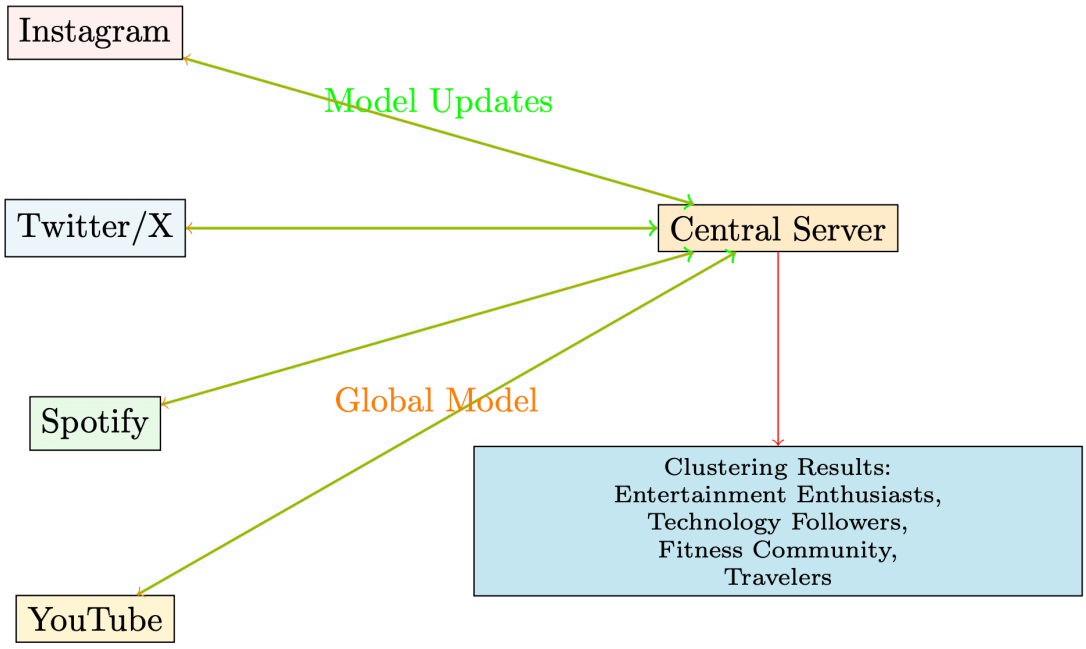

Figure 8. Application of multi-view clustering via VFL for Cross-Platform User Profiling.

To facilitate the implementation of MVC with VFL, a client-server VFL paradigm is designed for system architecture. Potential extensions for decentralized variants are considered to enhance resilience. This VFL-based MVC will utilize three distinct protocols that enable communication between clients (platforms), servers (coordinator), and users. Participating clients, such as Instagram, Twitter/X, Spotify, and YouTube, will retain their local storage of user-aligned data and train modality-specific encoders. The server(coordinator) will manage a neutral aggregator (or trusted execution environment/TEE) that facilitates secure fusion without accessing raw features. User alignment will address secure ID matching (e.g., through privacy-preserving record linkage or hashed pseudonyms). Incomplete views are managed through masking or imputation in federated incomplete MVC (FedIMVC) frameworks.

Platform	Modality	Encoder Models & Techniques	Key Local Adaptations
Instagram	**Visual**	Pre-trained CNNs/ViTs (CLIP), SimCLR, MAE	Fine-tuned on platform-specific images and videos to enhance aesthetic appeal and activity-related features
Twitter/X	**Textual**	Transformers (BERT, RoBERTa, sentence-transformers) + graph features from mentions/hashtags	Enhanced embeddings serve as representations of social interaction graphs
Spotify	**Audio**	VGGish, AudioMAE, CLAP with spectrograms/raw waveforms	Sequential listening patterns are modeled using RNNs and Transformers
YouTube	**Video**	Multimodal video encoders (VideoMAE, CLIP-Video) + metadata (watch time, subscriptions)	User-level embeddings are aggregated through attention mechanisms based on watch history

Table 13. The local representation learning stage across heterogeneous platforms, illustrating the modality-specific encoders, self-supervised or pre-trained backbone choices, and fine-tuning approaches to capture behavioral signals before federated MVC

Local representation learning is the initial protocol of the process, during which platforms such as Instagram, Twitter/X, Spotify, and YouTube participate. Table 13 presents an overview of this process, elucidating the distinct encoder models and techniques untilized by each platform to extract features from their respective local databases. For instance, Instagram employs pre-trained convolutional neural networks (CNNs) or vision transformers (such as CLIP vision encoder) or self-supervised models (SimCLR and MAE) to extract embeddings from images and reels. These embeddings are subsequently fine-tuned locally on platform-specific data to capture behavioral features such as aesthetic preferences and activity frequency. In contrast, Twitter/X employs transformer-based encoders, including BERT variants, RoBERTa, and sentence-transformers, to generate tweet embeddings. These embeddings are further enriched by incorporating graph features derived from mention and hashtag networks. Each client generates a local embedding ($\mathbf{z}_h \in \mathbb{R}^{d_h}$) for view h, or intermediate activations.

In the subsequent phase, the system transitions from isolated local representation learning to secure, collaborative fusion across diverse perspectives. This phase establishes the foundation of the system, enabling the extraction of complementary signals from Instagram users with visual embeddings, Twitter/X users with textual representations, Spotify users with audio features, and YouTube users with video-derived embeddings, all while ensuring privacy safeguards. In contrast to HFL, in which clients share the same feature space but hold different samples (e.g., different users on similar platforms), VFL operates on the same sample space (aligned users) but disjoint feature spaces. This alignment renders VFL particularly well-suited to this multi-platform scenario, yet it introduces distinctive fusion challenges. Direct sharing of complete embeddings or gradients is impractical, necessitating cryptographic intermediaries. Table 14 presents an overview of the fundamental techniques employed for secure interaction and fusion within the MVC-VFL framework. These mechanisms guarantee that no single entity, including the coordinator server, can access another's raw data, complete local embeddings, or even obtain complete intermediate representations.

Technique	Core Mechanism	Privacy Strength	Communication Overhead	Suitability for MV Fusion	Limitations/Challenges
Additive secret sharing (SMPC variant)	Split data into random shares; linear operations on shares	Information-theoretic (perfect secrecy against *t*-threshold colluders)	Moderate (local additions cheap; reconstruction needed)	Excellent for secure summation of embeddings/similarities	Not natively multiplicative; expensive for non-linear operations
Homomorphic encryption (HE, esp. CKKS/Paillier)	Encrypt data; compute directly on ciphertexts	Computational (based on hardness assumptions)	High (especially FHE for arbitrary circuits)	Good for aggregation & basic statistics; partial for deep models	Noise accumulation; slow for high-dimensional embeddings
Hybrid SMPC + DP	Secret sharing + calibrated noise	Layered (information-theoretic + statistical)	High	Strong for noisy contrastive learning or clustering objectives	Trade-off between utility (noise) and privacy (ε)
Secure multi-party computation (General SMPC, e.g., SPDZ/Garbled circuits)	Joint function evaluation via protocols	High (malicious security possible)	Very high	Flexible for complex fusion (attention, CCA)	Scalability with 4+ parties; protocol complexity
Trusted execution environments (TEEs, e.g., Intel SGX)	Hardware-isolated execution	Hardware-rooted	Moderate	Efficient secure aggregation	Hardware trust assumptions; side-channel risks

Table 14. Comparative overview of privacy mechanisms in VFL fusion

3.3.3. Additive Secret Sharing and SMPC

In additive secret sharing, for a scalar value $x_{[l]}$ (for instance, a component of a local embedding $\mathbf{z}_j$ derived from Instagram), each client l generates a series of random shares $(s_{l1}, s_{l2}, \cdots, s_{lS(l)})$ such that the following relation holds:

$$x_{[l]} = \sum_{s=1}^{S(l)} s_{[l]s} \pmod{\mathbb{F}} \tag{6}$$

where $\mathbb{F}$ represents an appropriately selected finite field, and $S(l)$ denotes the number of shares generated by client l. Each share s_{ls} is subsequently distributed to distinct parties or the central server or coordinating server. This mechanism serves as the fundamental principle of privacy-preserving distributed computation, ensuring that no single entity can reconstruct the original value without obtaining all the requisite shares. Consequently, it establishes a mathematically sound foundation for confidential data fusion across diverse perspectives and platforms. Crucially, additive secret sharing enables linear operations, such as summation and averaging, to be executed directly on the shares, bypassing the need for intermediate reconstruction of sensitive values. This property is formally expressed as:

$$\sum_{l=1}^{L} x_l = \sum_{l=1}^{L} \left(\sum_{s=1}^{S(l)} s_{ls} \right) \pmod{\mathbb{F}} \tag{7}$$

In the context of MV user profiling across platforms such as Instagram, Twitter/X, Spotify, and YouTube, Eqs. 6-7 enable each platform to compute its local embedding $\mathbf{z}_{[l]h}$ using modality-specific encoders (e.g., visual CNNs, textual transformers, sequential audio encoders, and video feature extractors). During the fusion phase, these platforms split their intermediate tensors (e.g., projected features aligned via secure ID matching) into additive shares. The server or a decentralized set of peer parties then aggregates these shares to compute secure global representations for downstream tasks such as spectral clustering, cross-modal similarity computation, and comprehensive user-behaviour profiling. VFL addresses the inherent privacy challenges of this approach. Unlike HFL, which primarily focuses on sample-disjoint but feature-aligned data, VFL emphasizes feature-disjoint but sample-aligned scenarios. The multi-platform setting aligns perfectly with the VFL paradigm, where all platforms observe the same user base, albeit from

different modalities. Consequently, any integration of features must adhere to strict confidentiality constraints. Direct gradient or embedding exchange would violete privacy, but secret sharing provides a viable alternative that is both mathematically and operationally feasible.

Despite its advantages, additive secret sharing incurs substantial computational expenses. High-dimensional embeddings, ranging from 512 to 4096 dimensions, significantly impact the overhead. Consequently, there is a linear increase in communication volume directly proportional to the number of shares per feature element. Furthermore, the non-linearity bottleneck presents an additional challenge. While linear operations are efficient under secret sharing, non-linear transformations (such as multiplications or activations functions) necessitate either interaction-intensive protocols or degree-reduction techniques similar to Shamir's secret sharing. These techniques increase the number of communication rounds, which introduces additional complexity due to cross-platform heterogeneity. For instance, Spotify's sequential embeddings may need to be aligned with YouTube's session-level watch-time vectors. This alignment could necessitate secure preprocessing techniques such as Private Set Intersection (PSI) to identify active user subsets. Furthermore, the scalability of secure multi-party protocols must consider varying levels of collusion tolerance, such as 1-2 colluding parties. Ensuring robustness under partial participation or platform dropout requires auxiliary methods like federated imputation or secure propagation of view-missing masks.

In essence, additive secret sharing presents a theoretically sound and practically viable solution for multi-platform, privacy-preserving user profiling. Its application in vertical MVC not only enhances privacy-preserving analytics but also creates an ideal environment for further research on optimizing communication overhead, supporting non-linear operations, and ensuring robustness in dynamic multi-party settings.

3.3.4. Complementary Mechanisms: HE, Hybrids, and Alternatives

Homomorphic encryption (HE) is a cyptographic technique that enables computations to be performed on encrypted data. This can be achieved by utilizing algorithms such as Paillier for addition or CKKS for approximate arithmetic on floating-point numbers. Clients encrypt their input vectors and transmit the resulting ciphertexts to the server. The server subsequently executes computations like summation or averaging without decrypting the encrypted data. Fully HE (FHE) extends this capability to arbitrary circuits, albeit at a substantial cost, particularly for deep clustering iterations. Hybrid approaches, such as utilization of HE for initial aggregation followed by SMPC for refinement, or the integration of additive

sharing with differential privacy (DP), offer a balance between security and efficiency. DP introduces Gaussian or Laplacian noise to the shares or aggregates, thereby limiting the potential leakage of information through a predetermined privacy budget. The fusion protocols of MVC-VFL are depicted in Figure 9.

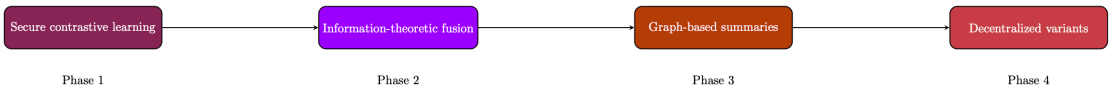

Figure 9. Fusion protocols in MVC-VFL

In the protocol illustrated in Figure 9, secure contrastive learning constitutes the initial phase. During this phase, clients collaboratively generate positive (same-user cross-view) and negative pairs locally. It is imperative to securely compute infoNCE-style losses on shared similarities utilizing SMPC dot-product subroutines. In the subsequent phase, information-theoretic fusion (e.g., ESFMC) maximizes mutual information or minimizes disagreement across views through secure optimization, frequently employing gradient sharing under encryption. Graph-based summaries are employed to reduce overhead, as clients exchange compressed, privacy-preserving graph representations (e.g., differentially private sketches or encrypted adjacency matrices) instead of full embeddings. This approach facilitates the efficient spectral or subspace clustering. Decentralized variants are employed to eliminate the central server by utilizing peer-to-peer SMPC or blockchain-augmented protocols. This further diminishes trust assumptions while simultaneously increasing coordination complexity.

Compatibility challenges arise due to inherent differences in dimensionality, distribution, and semantics between clients. Secure alignment techniques, such as private CCA or federated procrustes analysis, are crucial to address these challenges. Incompatible features, like sparse text versus dense visual data, may require local dimensionality reduction or projection onto a shared secure subspace before sharing, adding computational overhead. Dynamic user activity, such as a user being inactive on Spotify, introduces incompleteness, which can be managed through secure masking or view-specific weighting within the fusion objective. Table 15 presents the challenges, effects, and practical mitigations of these complementary mechanisms from various perspectives.

Aspect	Challenges	Effects	Mitigations
Privacy-utility trade-off	Robust privacy assurances, such as malicious-secure SMPC	Increased latency and diminished clustering quality are attributed to noise or approximation	Empirical tuning through privacy accounting to achieve a harmonious balance between privacy and utility
Attack resistance	Side-channel leaks (timing and communication patterns) and potential model inversion on final clusters	Despite resistance to gradient inversion, membership inference, and colluding-server attacks, there remains a risk of information leakage	Implement constant-time protocols, output perturbation, and additional security measures to mitigate side-channel and inversion vulnerabilities
Overhead mitigation	High computation and communication overhead from encryption, SMPC, and DP	Scalability, responsiveness, and multi-platform coordination are all impacted by this	Reduce overall communication and computational requirements by employing quantization/compression techniques, asynchronous updates, hybrid trusted hardware (TEEs), and graph summaries
Implementation notes	The challenge of integrating privacy-preserving mechanisms while minimizing performance bottlenecks	Potential delays or security vulnerabilities may arise if not implemented with utmost care	Utilize libraries such as PySyft, MP-SPDZ, or Flower, which support SMPC extensions. Prior to production, simulate data using synthetic aligned data.

Table 15. The trade-offs, security analysis, and practical mitigations of comple-mentary mechanisms

3.3.5. Future Research Directions

This privacy-preserving fusion layer of MVC-VFL not only facilitates the formation of nuanced clusters, such as identfying “fitness community” users through correlated workout visuals, motivational audio,

and travel-related text, but also establishes a benchmark for ethical cross-platform analytics. Future advancements include adaptive mechanisms that dynamically adjust security level based on sensitivity and the integration of post-quantum cryptography for long-term resilience.

3.3.6. MVC via FTL (MVC-FTL) for Privacy-Preserving Cross-Platform User-Profiling

Federated transfer learning (FTL) facilitates the transfer of knowledge across platforms that possess partially overlapping user bases and feature sets. This approach is particularly well-suited for heterogeneous ecosystems, such as TikTok, GitHub, and Reddit, where shared users are limited. In contrast to the VFL and HFL models, which employ shared users and different features or shared features and distinct users, respectively, FTL utilizes pre-trained local models to establish a foundation for learning across domains. Privacy is maintained through secure aggregation, knowledge distillation, and cryptographic transfer mechanisms. The primary objective of this research is the development of an MVC framework that prioritizes privacy preservation and facilitates the execution of applications such as personalized recommendations, community detection, talent identification, and behavioral research. This framewok ensures that sensitive user data remains centralized, thereby enhancing user privacy.

Table 16 presents the problem and platform-specific views for heterogeneous ecosystems such as TikTok, GitHub, and Reddit. A significant challenge in addressing the issue of privacy-preserving cross-platform user-profiling is the heterogeneity of the ecosystems involved, as evidenced by the participation of TikTok, GitHub, and Reddit. This challenge is further compounded by the sparsity of user overlap, which indicates that only as a small percentage of TikTok users also maintain active profiles on GitHub or Reddit. In this particular instance, direct data sharing is not viable option. Consequently, FTL facilitates the transfer of cluster priors or representation mappings across these partial intersections. Figure 10 presents the MVC-FTL protocol, which has been developed to address this issue.

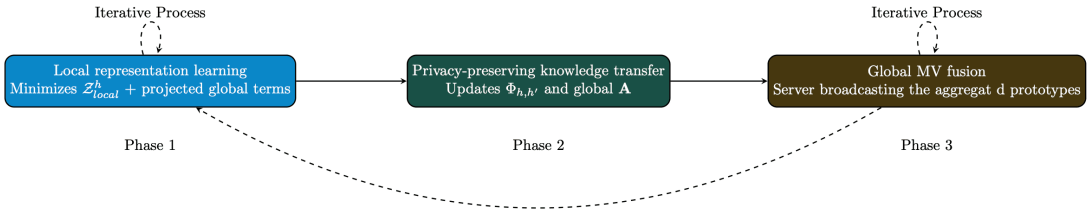

Figure 10. The Flowchart of MVC-FTL

doi.org/10.32388/PC8X49

Platform-Specific Views	Features	Objective of Clustering
TikTok (Short-form video)	Video embeddings, watch time, engagement metrics, creator interactions	Entertainment sub-populations (comedy enthusiasts, dance communities, trend followers)
GitHub (Code/Repository)	Repository metadata, commit history, contribution graphs, programming language embeddings	Professional profiles (backend developers, data scientists, AI researchers)
Reddit (Community/Textual)	Subreddit participation, comment embeddings, voting patterns, post karma	Interest groups (gamers, tech enthusiasts, finance communities)

Table 16. A summary of the problem, along with platform-specific views forheterogeneous ecosystems such as TikTok, GitHub, and Reddit.

As can be seen in Figure 10, the protocol of MVC-FTL is comprised of three phases. The first phase of the framework involves local representation learning. Each participated platform employs distinct methodologies to extract information from each user's features. For instance, TikTok utilizes video transformers (VideoMAE) and sequential engagement models to extract features from video embeddings, watch time, engagement metrics, and creator interactions. These features are subsequently clustered into entertainment sub-populations, encompassing comedy enthusiasts, dance communities, and trend followers. Conversely, GitHub employs graph neural network (GNN) and code transformers (CodeBERT) to extract features from users' repository metadata, commit history, contribution graphs, and programming language embeddings. These features are subsequently clustered into professional profiles of backend developers, data scientists, and AI researchers. Similarly, Reddit employs topic modeling and graph embeddings to extract features from users' subreddit participation, comment embeddings, voting patterns, and post karma. These features are subsequently clustered into interest groups, such as gamers, tech enthusiasts, and finance communities. After all participated platforms extract local features, they independently perform unsupervised clustering (e.g., k-means or spectral clustering). This process generates prototypes or soft labels that serve as transferable knowledge.

In the subsequent phase, user alignment is employed to enable private set intersection (PSI) or secure multi-party computation (SMPC) to detect overlapping users without revealing their identities. Subsequently, representation mapping is employed as a knowledge transfer technique to securely project between modalities, leveraging overlapping users as anchors. In this phase, cluster prototype transfer is employed to facilitate soft assignments or centroid exchanges through the use of homomorphic encryption or secure aggregation. Concurrently, federated distillation substitutes raw gradients with distilled models or pseudo-labels for model exchange. Finally, federated variational autoencoders (VAEs) or graph propagation is employed for handling missing views by imputing non-overlapping users. Secure primitives, including additive secret sharing, differential privacy, and trusted execution environments (TEEs), ensure the confidentiality of user-level data throughout the process.

In the final phase, referred to as global MV fusion, the algorithm optimizes local reconstruction, cross-view alignment, and clustering regularization in a unified manner. It employs dual-head encoders, one for local and the other for global, to achieve this objective. Anchor graphs are utilized for efficient alignment, and they are iteratively refined while maintaining global synchronization. This ensures the formation of cohesive clusters that integrate behaviors across platforms. For instance, an AI researcher, a participant in the technology community, and an educational TikTok consumer would all be grouped together in the same cluster. These phases facilitate multi-domain user profiling and clustering across partially overlapping platforms. They also ensure data privacy through federated transfer mechanisms.

3.3.7. Mathematical Formulation: Concurrent Optimization of MVC via FTL

In the MVC-FTL framework, the primary objective is to seamlessly integrate local representation learning, privacy-preserving knowledge transfer across partial user overlaps, and global MVC. The formulation is meticulously designed to be optimized in a federated, alternating manner, while simultaneously ensuring that privacy constraints are upheld through secure computation of cross terms.

Let $H = \{h_1, h_2, h_3\}$ denote the views corresponding to TikTok (h_1: video), GitHub (h_2: code/repository), and Reddit (h_3: community/textual). For each view h, let $\mathscr{D}_h = \{(\mathbf{x}_i^h, l_i)\}_{i \in \mathscr{L}_h}$ be the local dataset at platform h, where $\mathbf{x}_i^h$ is the feature vector (or raw input) for user l_h, and $\mathscr{L}_h$ is the set of users on that platform. The intersection of the privacy-preserving record linkage (e.g., PSI) and the overlap of the two datasets, denoted as $\mathscr{L}_{overlap}^{h,h'} = \mathscr{L}_h \cap \mathscr{L}_{h'}$, are typically small and can be identified. Each participating platform maintains a local encoder f_{θ_h} producing embeddings $\mathbf{p}_i^h = f_{\theta_h}(\mathbf{x}_i^h) \in \mathbb{R}^{d_h}$. A transfer mapping

$\Phi_{h \to h'}$, such as linear projection or neural adapter, aligns representations across views for overlapping users. The refined joint optimization objective can be expressed as follows:

$$\begin{aligned} \min_{\{\theta_h\},\{\Phi_{h,h'}\},\mathbf{A}} \quad J = & \underbrace{\sum_{h} \mathcal{Z}^h_{local}(\theta_h; \mathcal{D}_h)}_{\text{Local Reconstruction \& Self-Supervision}} + \lambda_h \underbrace{\sum_{(h,h')} \mathcal{Z}^{h,h'}_{transfer}(\Phi_{h,h'}; \mathcal{L}^{h,h'}_{overlap})}_{\text{Privacy-Preserving Transfer}} \\ & + \gamma_h \underbrace{\mathcal{Z}_{cluster}(\{\mathbf{p}^h_i\}, \mathbf{A}; \mathbf{F}_{fused})}_{\text{Multi-view Clustering}} + \Omega(\{\theta_h\}, \{\Phi_{h,h'}\}) \end{aligned} \tag{8}$$

where $\mathbf{A} = \{\mathbf{a}_k\}_{k=1}^{c}$ is defined as the set of c clusters centers (or prototypes) in the fused space, and $\mathbf{F}_{fused}$ is the aligned MV representation matrix constructed via secure methods, such as weighted concatenation or attention, as expressed below.

$$\mathbf{p}_i^{fused} = \text{Fuse}(\{\mathbf{p}_i^h\}, \{\Phi_{h,h'}(\mathbf{p}_i^h)\}) \tag{9}$$

As illustrated in Eq. 8, the joint optimization objective of MVC-FTL is comprised of four distinct terms. The first term is defined as a local loss, denoted by $\mathscr{Z}^h_{local}$, which is optimized independently on each platform using the following expression:

$$\mathscr{Z}^h_{local} = \mathscr{Z}^h_{rec}(\mathbf{x}^h_i, \hat{\mathbf{x}}^h_i) + \alpha \mathscr{Z}^h_{ssl}(\mathbf{p}^h_i) + \beta \mathscr{Z}^h_{local-cluster}(\mathbf{p}^h_i, \mathbf{a}_h) \tag{10}$$

In the realm of deep learning (DL), the reconstruction loss, denoted by $\mathscr{Z}^h_{rec}$, measures the disparity between the reconstructed data and the original data. This loss is typically expressed as the mean squared error (MSE) for autoencoders or the contrastive loss for self-supervised models like SimCLR and VideoMAE. On the other hand, the self-supervised regularization, also known as"Self-Supervised Regularization" or "SSRL" (denoted by $\mathscr{Z}^h_{ssl}$), aims to enhance the model's invariance. It is defined using the Barlow Twins or VICReg. To identify clusters in the data, a preliminary local clustering process $\mathscr{Z}^h_{local-cluster}$ is employed, which can be achieved using the k-means objective or a deep clustering loss such as the Kullback-Leibler divergence (KL-divergence) for soft assignment.

The evaluation of the second term in Eq. 8, denoted as transfer loss $\mathscr{Z}^{h,h'}_{transfer}$, can only be conducted securely when it overlaps. This mathematical formula facilitates such an evaluation.

$$\mathscr{Z}^{h,h'}_{transfer} = \mathbb{E}_{i \in \mathscr{L}^{h,h'}_{overlap}} \left[||\Phi_{h \to h'}(\mathbf{p}^h_i) - \mathbf{p}^{h'}_i||_2^2 + \text{MMD}(\mathscr{P}^h, \mathscr{P}^{h'}) + \mathscr{Z}_{proto-align}(\mathbf{a}_h, \mathbf{a}_{h'}) \right] \tag{11}$$

The square embedding alignment is employed for direct correspondence. The maximum mean discrepancy (MMD) is a statistical technique that is employed for distribution alignment in non-overlapping regimes, such as kernel-based methods and secure aggregation of moments or random

features. The prototype alignment, denoted as $\mathscr{L}_{proto-align}$, facilitates the transfer of cluster knowledge, such as the cosine similarity or the optimal transport between local centroids. The term is computed via SMPC on additive platforms. The injection of differential privacy noise can be utilized to ensure the fulfillment of guarantees of the form (ϵ, δ).

The third term, the clustering loss $\mathscr{L}_{cluster}$, is employed for global evaluation of fused representatios. This mathematical formula facilitates such an evaluation.

$$\mathscr{L}_{cluster} = \sum_i \sum_k t_{ik} \log \frac{t_{ik}}{\mu_{ik}} + \eta \mathrm{Tr}(\mathbf{F}_{fused}^T \mathbf{G} \mathbf{F}_{fused}) \tag{12}$$

The initial term in Eq. 12 signifies the KL-divergence for soft assignment consistency, drawing inspiration from deep embedded clustering. The model encompasses target distributions, denoted as t_{ik}, and soft assignments, denoted as μ_{ik}. The soft assignment can be modeled using distributions such as the Student's t-distribution. The second term introduces spectral regularization using the securely aggregated affinity graph. The construction of this graph entails the utilization of cosine similarities or k-nearest neighbors (k-NN) on the fused feature matrix, denoted by $\mathbf{F}_{fused}$. In essence, the fusion function in the second term has the capacity to incorporate cross-view attention or canonical correlation analysis (CCA) with an adaptive balancing parameter η. These evaluations are conducted in accordance with secure protocols.

Finally, the fourth term in Eq. 8 represents the regularization Ω, which encompasses the estimation of weight decay, orthogonality constraints on projections Φ, and view-weight balancing (e.g., adaptive λ_h based on local variance or overlap size.

The convergence of the MVC-FTL in Figure 10 is substantiated by theoretical analyses presented in the federated MVC (FedMVC) literature including bounded variance under partial participation and secure aggregation. The privacy integration here is that all cross-platform terms are instantiated using cryptographic primitives, so only aggregates (sums, means, kernels), or noisy statistics are revealed. Communication overhead is controlled by exchanging low-dimensional prototypes or compressed gradients. This refined formulation is modular, extensible (e.g., to temporal or hierarchical clustering), and directly implementable. It strikes a balance between fidelity to local data distributions and effective knowledge transfer, resulting in superior clustering performance on sparse-overlap multi-platform data compared to isolated or non-transfer baselines. Researchers or experts can further specify the losses based on modality (e.g., video contrastive vs. graph contrastive) and validate them by performing ablation on overlap ratios.

3.3.8. Mathematical Formulation: Decentralized MVC-FTL

To accommodate decentralized, MV data distributed across L clients, where each client $l = 1, \cdots, L$ may correspond to a platform slice, we need to ensure that the data is accessible and can be viewed from multiple perspectives. Alternatively, a generalized model can be established by integrating a user cohort, a decentralized device, or a centralized device. The user cohort approach will combine multiple user cohorts to form a generalized model. The decentralized system approach will utilize a decentralized system to consolidate views from multiple sources. In contrast, the centralized device approach utilizes centralized devices that maintain a consolidated view of the data, but it encompasses multiple perspectives. In this context, data is entirely decentralized, thereby eliminating the requirement for a stringent central server. Nevertheless, a lightweight coordinator can be utilized to facilitate secure routing.

Each client l holds a subset of views ($h_{[l]} \subseteq H$) for a local user set ($\mathrm{src}M_{[l]}$). There are partial overlaps ($\mathscr{M}_{[l,l']} = \mathscr{M}_{[l]\to[l']} = \mathscr{M}_{[l]} \cap \mathscr{M}_{[l']}$) between clients. Here, $l' \neq l$, and $l' = l^r$, where r is the cardinality of l. This reflects real-world scenarios where data from platforms like TikTok, GitHub, and Reddit (or their sub-cohorts) is partitioned across multiple nodes to enhance privacy and scalability. The global optimization problem of a distributed consensus problem can be formulated as follows.

$$\begin{aligned} \min_{\{\theta^h_{[l]}\}^h_{[l]}, \{\Phi^{h,h'}_{[l]\to[l']}\}, \{\mathbf{A}_{[l]}\}} \quad J_{[l]} = & \sum_{l=1}^{L} \left[\sum_{h} \mathcal{Z}^h_{[l]local}(\theta^h_{[l]}; \mathcal{D}^h_{[l]}) + \lambda^h_{[l]} \sum_{(h,h')} \mathcal{Z}^{h,h'}_{[l]transfer}(\Phi^{h,h'}_{[l]\to[l']}; \mathcal{L}^{h,h'}_{[l]\to[l']overlap}) \right] \\ & + \sum_{l=1}^{L} \gamma^h_{[l]} \mathcal{Z}_{[l]cluster}(\{\mathbf{p}^h_{[l]i}\}, \mathbf{A}_{[l]}; \mathbf{F}_{[l]fused}) + \underbrace{\mathcal{R}_{consensus}(\{\mathbf{A}_{[l]}\}, \{\mathbf{F}_{[l]fused}\})}_{\text{Decentralized Consensus}} \end{aligned} \tag{13}$$

Subject to privacy constraints, all inter-client communications are conducted through secure channels, utilizing secret-shared values or encrypted tensors. This ensures that client l discloses only necessary aggregates.

In Eq. 13, $\mathscr{D}^h_{[l]}$ is defined as the local MV (or partial-view) data at client l, can be estimated by the following equation.

$$\mathscr{D}^h_{[l]} = \bigcup_{h_{[l]} \in H} \{(\mathbf{x}^h_{[l]i}, \mu_{[l]i}) \mid \mu_{[l]i} \in \mathscr{L}_{[l]}\} \tag{14}$$

The local encoder parameter for view h at client l in Eq. 13 is denoted as $\Phi^{h,h'}_{[l]\to[l']}$ defined as the transfer mapping from client l's view h to client l''s view h', active primarily on overlaps $\mathscr{L}_{[l,l']}$. $\mathbf{F}_{[l]fused}$ is defined

as the fused representation for users in $\mathscr{L}_{[l]}$, incorporating local views and transferred knowledge from peers; While $\mathbf{A}_{[l]}$ is defined as the local cluster centers or prototypes at client l where the global consensus yields a shared $\mathbf{A}$. The detailed loss components of decentralized adaptation in Eq. 13 is summarized in Table 17.

Loss Components	**Defenition**	**Objectives**
$\mathscr{Z}^{h}_{[l]local}$	Local loss at client l	$\sum_{h\in H_{[l]}}\left[\mathscr{Z}^{h}_{[l]rec}(\mathbf{x}^{h}_{i},\hat{\mathbf{x}}^{h}_{i};\theta^{h}_{[l]})+\alpha\mathscr{Z}^{h}_{[l]ssl}(\mathbf{z}^{h}_{[l]i})+\beta\mathscr{Z}^{h}_{[l]local-cluster}\left(\mathbf{z}^{h}_{[l]i},\mathbf{a}_{[}l^{h}\right)\right]$
$\mathscr{Z}^{h,h'}_{[l]transfer}$	Decentralized transfer loss at client l	$\sum_{[l']\neq[l]}\mathbb{E}_{i\in\mathscr{L}[l,l']}\left[D(\Phi^{h,h'}_{[l]\to[l']}(\mathbf{p}^{h}_{[l]i}),\mathbf{p}^{h'}_{[l']i})+\mathrm{MMD}_{c}(\mathscr{P}^{h}_{[l]},\mathscr{P}^{h'}_{[l']})+\mathscr{Z}_{[l,l']proto}(\mathbf{a}^{h}_{[l]},\mathbf{a}^{h'}_{[l']})\right]$
$\mathscr{Z}_{[l]cluster}$	Local Clustering Loss at client l	$\sum_{i\in\mathscr{C}_{[l]}}\sum_{k=1}^{c}\mu_{[l]ik}\log\frac{\mu_{[l]ik}}{t_{[l]ik}(\mathbf{P}_{[l]fused})}+\eta,\mathrm{Tr}\left((\mathbf{P}_{[l]fused})^{T}\mathbf{Z}_{[l]}\mathbf{P}_{[l]fused}\right)$
$\mathscr{R}_{consensus}$	Decentralized Consensus Regularizer	$\frac{1}{2L}\sum_{l=1}^{L}\sum_{l'=1,l'\neq l}^{L}\lvert\mathbf{c}_{[l]}-\mathbf{c}_{[l']}\rvert_{F}^{2}+\mu\sum_{l}\mathrm{Var}(\mathbf{P}_{[l]fused})$

Table 17. The loss components of decentralized adaptation MVC-FTL

In real-world scenarios, data from platforms such as TikTok, GitHub, and Reddit (or their sub-cohorts) is partitioned across multiple nodes to enhance the privacy and scalability of partial participation, as outlined in the aforementioned formulations. Decentralized MVC-FTL permits clients to join asynchronously and provides robustness via view-missing masks or imputation modules. The system's scalability is further enhanced by low-rank approximations of Φ and compressed communication techniques, such as quantized shares. In the course of the iterative process, platform mapping is utilized to allocate clients to platform-specific groups (e.g., multiple TikTok shards as separate l), thereby facilitating fine-grained decentralization. The extended formulation in Eq. 13 meticulously encapsulates the decentralized nature of multi-view data across L clients while upholding stringent privacy

doi.org/10.32388/PC8X49

constraints and facilitating efficacious knowledge transfer for clustering operations. This formulation provides a solid foundation for implementation in frameworks that support decentralized FL, such as Flower with P2P extensions or custom SMPC libraries, and for empirical validation in cross-platform academic studies.

4. Future Research Directions

This paper reviews the multi-view clustering (MVC) algorithm via federated learning (FL) approaches with the objective of learning multi-view data for integrative clustering analysis. A salient feature of MVC via an FL is its capacity to diseminate local models and adopt a globally aggregated model, which quantifies information from each view across participating clients while preserving privacy. Consequently, the multi-view data of clients remains centralized, and models such as cluster centers or prototypes are shared with a central or coordinator server. This approach ensures the robustness of the clustering process, even in the presence of potentially heterogeneous multi-view data structures that are not present in a particular client. This capacity endows MV with the capability to manage heterogeneous and noisy data via horizontal FL (HFL), vertical FL (VFL), and transfer FL (TFL). The integration of this review scheme with other methodologies, such as model-based approaches, has the potential to enhance the efficiency and performance of centralized MVC in various applications, including healthcare, finance, and social media. Consequently, it is possible to integrate the advantages of model-based MVC methodologies with diverse FL approaches to examine the theoretical characteristics and practical applications of MVC for a range of real-world problems. This analysis should encompass strategies to minimize overhead, optimize communication, and ensure privacy and precision. In essence, this review offers reliable and knowledgeable information about MVC in federated environments and supports the concept of MVC in centralized systems. Additionally, it elucidates the process by which existing centralized MVC systems can be transitioned to decentralized systems in the future.

Statements and Declarations

Funding

This research received no external funding.

Potential Competing Interests

The author declare no conflict of interest.

Acknowledgments

The author has no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

1. ^*Zhao L, Liu S, Xin T, Tan J, Wang X, Li Y, Bian Z, Chen Y, Kong F, Bian J, others (2026). "AI Agent in Healthcar e: Applications, Evaluations, and Future Directions." npj Artif Intell. **2**(1):31.*
2. ^*Njei B, Al-Ajlouni Y, Sidney Kanmounye U, Boateng S, Loic Nguefang G, Njei N, Hamouri S, Al-Ajlouni A (20 26). "Artificial Intelligence Agents in Healthcare Research: A Scoping Review." PLoS One. **21**(2):e0342182.*
3. ^*Xu G, Li X, Chen Y, Duan Y, Wu S, Yu H, Chiu C, Ni J, Tang N, Li T, others (2026). "A Comprehensive Survey of AI Agents in Healthcare." J Biomed Inform. :105045.*
4. ^*Kumar S (2025). "Multi-Agent AI Systems in Finance: Models, Applications, and Challenges." Int J Adv Res Comput Sci Technol. **8**(1):11555–11573.*
5. ^*Ante L (2026). "Autonomous AI Agents in Decentralized Finance: Market Dynamics, Application Areas, an d Theoretical Implications." Technol Forecast Soc Change. **228**:124669.*
6. ^*Rizinski M, Trajanov D (2026). "AI Agents in Finance and Fintech: A Scientific Review of Agent-Based Syst ems, Applications, and Future Horizons." Comput Mater Contin. **86**(1):1.*
7. ^*Aldridge I, An J, Burke R, Cao M, Chien C, Deng K, Deng R, Gao Y, Guo O, He S, others (2026). "Agentic Artific ial Intelligence in Finance: A Comprehensive Survey." arXiv preprint arXiv:2604.21672.*
8. ^*Ferraro A, Galli A, La Gatta V, Postiglione M, Orlando G, Russo D, Riccio G, Romano A, Moscato V (2024). "A gent-Based Modelling Meets Generative AI in Social Network Simulations."*
9. ^*Orlando G, La Gatta V, Russo D, Moscato V (2025). "Can Generative Agent-Based Modeling Replicate the Fr iendship Paradox in Social Media Simulations?."*
10. ^*Gausen A, Guo C, Luk W (2025). "An Approach to Sociotechnical Transparency of Social Media Algorithms Using Agent-Based Modelling." AI Ethics. **5**(2):1827–1845.*
11. ^*Sarr L, Barthe-Delanoë A, Bork D, Ayite K, Macé-Ramète G, Bénaben F (2026). "From Chat to Process: A Co nversational Agent Framework for Social Business Process Management." Inf Process Manage. **63**(8):10494*

0.

12. ^Whig P, Sharma P, Elngar A, Silva N (2026). Quantum Learning: Bridging Artificial Intelligence, Quantum Computing, and Data Science in Education. CRC Press.
13. a, bMcMahan B, Moore E, Ramage D, Hampson S, y Arcas B (2017). "Communication-Efficient Learning of Deep Networks from Decentralized Data."
14. ^Ward Jr J (1963). "Hierarchical Grouping to Optimize an Objective Function." J Am Stat Assoc. **58**(301):236–244.
15. a, bMcQueen J (1967). "Some Methods of Classification and Analysis of Multivariate Observations."
16. a, b, cBezdek J, Ehrlich R, Full W (1984). "FCM: The Fuzzy C-Means Clustering Algorithm." Comput Geosci. **10**(2-3):191–203.
17. ^(1982). "Least Squares Quantization in PCM." IEEE Trans Inf Theory. **28**(2):129–137.
18. ^Ester M, Kriegel H, Sander J, Xu X, others (1996). "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise."
19. ^Ng A, Jordan M, Weiss Y (2001). "On Spectral Clustering: Analysis and an Algorithm." Adv Neural Inf Proces Syst. **14**.
20. ^Liew A, Yan H (2003). "An Adaptive Spatial Fuzzy Clustering Algorithm for 3-D MR Image Segmentation." IEEE Trans Med Imaging. **22**(9):1063–1075.
21. ^Chen S, Zhang D (2004). "Robust Image Segmentation Using FCM with Spatial Constraints Based on New Kernel-Induced Distance Measure." IEEE Trans Syst Man Cybern B Cybern. **34**(4):1907–1916.
22. ^Cinque L, Foresti G, Lombardi L (2004). "A Clustering Fuzzy Approach for Image Segmentation." Pattern Recognit. **37**(9):1797–1807.
23. ^Shin H, Sohn S (2004). "Segmentation of Stock Trading Customers According to Potential Value." Expert Syst Appl. **27**(1):27–33.
24. ^Kleinberg J, Papadimitriou C, Raghavan P (2004). "Segmentation Problems." J ACM. **51**(2):263–280.
25. ^Shah H, Undercoffer J, Joshi A (2003). "Fuzzy Clustering for Intrusion Detection."
26. ^Leon E, Nasraoui O, Gomez J (2004). "Anomaly Detection Based on Unsupervised Niche Clustering with Application to Network Intrusion Detection."
27. ^Fan W, Miller M, Stolfo S, Lee W, Chan P (2004). "Using Artificial Anomalies to Detect Unknown and Known Network Intrusions." Knowl Inf Syst. **6**(5):507–527.

28. ^Kumar A, Rai P, Daume H (2011). "Co-Regularized Multi-View Spectral Clustering." Adv Neural Inf Process Syst. **24**.

29. a, bBickel S, Scheffer T, others (2004). "Multi-View Clustering."

30. ^Zhao H, Ding Z, Fu Y (2017). "Multi-View Clustering via Deep Matrix Factorization."

31. ^Li Z, Wang Q, Tao Z, Gao Q, Yang Z, others (2019). "Deep Adversarial Multi-View Clustering Network."

32. a, bXie Y, Lin B, Qu Y, Li C, Zhang W, Ma L, Wen Y, Tao D (2020). "Joint Deep Multi-View Learning for Image C lustering." IEEE Trans Knowl Data Eng. **33**(11):3594–3606.

33. ^Wen J, Zhang Z, Zhang Z, Wu Z, Fei L, Xu Y, Zhang B (2020). "Dimc-Net: Deep Incomplete Multi-View Cluste ring Network."

34. ^Wang D, Li T, Deng P, Liu J, Huang W, Zhang F (2022). "A Generalized Deep Learning Algorithm Based on N MF for Multi-View Clustering." IEEE Trans Big Data. **9**(1):328–340.

35. ^Xia W, Gao Q, Wang Q, Gao X, Ding C, Tao D (2022). "Tensorized Bipartite Graph Learning for Multi-View C lustering." IEEE Trans Pattern Anal Mach Intell. **45**(4):5187–5202.

36. ^Cai B, Lu G, Li H, Song W (2024). "Tensorized Scaled Simplex Representation for Multi-View Clustering." IE EE Trans Multimedia. **26**:6621–6631.

37. ^Ji J, Feng S, Li Y (2024). "Tensorized Unaligned Multi-View Clustering with Multi-Scale Representation Lea rning."

38. ^Wang R, Gao Q, Yang M, Wang Q (2025). "Tensorized Tri-Factor Decomposition for Multi-View Clustering." IEEE Trans Circuits Syst Video Technol. **35**(6):5355–5366.

39. ^Zhong Z, Gu Z, Yang P, Guo R, others (2026). "Gaussian Regression-Driven Tensorized Incomplete Multi-Vie w Clustering with Dual Manifold Regularization." Adv Neural Inf Process Syst. **38**:72102–72131.

40. ^Wang H, Chen Y, Yao M, Liu Q, Jiang G, Fu X (2026). "Tensorized Parameter-Free Multi-View Spectral Clust ering Based on Fair Representation Learning." IEEE Trans Multimedia.

41. ^Nie F, Cai G, Li J, Li X (2017). "Auto-Weighted Multi-View Learning for Image Clustering and Semi-Supervis ed Classification." IEEE Trans Image Process. **27**(3):1501–1511.

42. ^Dhillon P, Foster D, Ungar L (2011). "Multi-View Learning of Word Embeddings via CCA." Adv Neural Inf Pr ocess Syst. **24**.

43. ^Greene D, Cunningham P (2009). "Multi-View Clustering for Mining Heterogeneous Social Network Data." Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. :3 37–346.

44. ^Mekthanavanh V, Li T, Meng H, Yang Y, Hu J (2019). "Social Web Video Clustering Based on Multi-View Clustering via Nonnegative Matrix Factorization." Int J Mach Learn Cybern. **10**(10):2779–2790.

45. ^Wang H, Yang Y, Liu B (2019). "GMC: Graph-Based Multi-View Clustering." IEEE Trans Knowl Data Eng. **32**(6):1116–1129.

46. a, bChen X, Xu J, Ren Y, Pu X, Zhu C, Zhu X, Hao Z, He L (2023). "Federated Deep Multi-View Clustering with Global Self-Supervision."

47. a, b, c, d, e, fHu X, Qin J, Shen Y, Pedrycz W, Liu X, Liu J (2023). "An Efficient Federated Multiview Fuzzy C-Means Clustering Method." IEEE Trans Fuzzy Syst. **32**(4):1886–1899.

48. a, b, c, d, eYang M, Sinaga K (2024). "Federated Multi-View K-Means Clustering." IEEE Trans Pattern Anal Mach Intell. **47**(4):2446–2459.

49. a, bFeng W, Wu Z, Wang Q, Dong B, Tao Z, Gao Q (2024). "Federated Multi-View Clustering via Tensor Factorization."

50. ^Feng W, Liu D, Wang Q, Liang W, Yan Z (2025). "Scalable Federated One-Step Multi-View Clustering with Tensorized Regularization."

51. ^Sinaga K (2025). "Personalized Federated Learning with Heat-Kernel Enhanced Tensorized Multi-View Clustering." arXiv preprint arXiv:2509.16101.

52. ^Li Y, Hu X, Liu J, Liu Z (2025). "Federated Incomplete Multi-View Clustering with Individual Structure Preservation and Central Representation Tensorization."

53. ^Feng W, Liu D, Wang Q, Jiang M, Liu B (2026). "Federated Incomplete Multi-View Clustering with Tensorized Low-Rank Constraint."

54. ^Jianliang M, Haikun S, Ling B (2009). "The Application on Intrusion Detection Based on K-Means Cluster Algorithm."

55. ^Tian L, Jianwen W (2009). "Research on Network Intrusion Detection System Based on Improved K-Means Clustering Algorithm."

56. ^Mohamed N, Ahmed M, Farag A (1998). "Modified Fuzzy C-Mean in Medical Image Segmentation."

57. ^Zhang D, Chen S (2004). "A Novel Kernelized Fuzzy C-Means Algorithm with Application in Medical Image Segmentation." Artif Intell Med. **32**(1):37–50.

58. ^Wang J, Kong J, Lu Y, Qi M, Zhang B (2008). "A Modified FCM Algorithm for MRI Brain Image Segmentation Using Both Local and Non-Local Spatial Constraints." Comput Med Imaging Graph. **32**(8):685–698.

59. a, b, cJiang B, Qiu F, Wang L (2016). "Multi-View Clustering via Simultaneous Weighting on Views and Features." Appl Soft Comput. **47**:304–315.

60. a, b, c, dSinaga K (2024). "Rectified Gaussian Kernel Multi-View K-Means Clustering." arXiv preprint arXiv:2405.05619.

61. a, b, cZhang G, Wang C, Huang D, Zheng W, Zhou Y (2018). "TW-Co-K-Means: Two-Level Weighted Collaborative K-Means for Multi-View Clustering." Knowledge-Based Syst. **150**:127–138.

62. a, bChen X, Xu X, Huang J, Ye Y (2011). "TW-K-Means: Automated Two-Level Variable Weighting Clustering Algorithm for Multiview Data." IEEE Trans Knowl Data Eng. **25**(4):932–944.

63. a, bXu Y, Wang C, Lai J (2016). "Weighted Multi-View Clustering with Feature Selection." Pattern Recognit. **53**:25–35.

64. a, bYang M, Sinaga K (2019). "A Feature-Reduction Multi-View K-Means Clustering Algorithm." IEEE Access. **7**:114472–114486.

65. a, bCai X, Nie F, Huang H (2013). "Multi-View K-Means Clustering on Big Data."

66. a, bLiu J, Wang C, Gao J, Han J (2013). "Multi-View Clustering via Joint Nonnegative Matrix Factorization."

67. a, bCleuziou G, Exbrayat M, Martin L, Sublemontier J (2009). "CoFKM: A Centralized Method for Multiple-View Clustering."

68. a, bJiang Y, Chung F, Wang S, Deng Z, Wang J, Qian P (2014). "Collaborative Fuzzy Clustering from Multiple Weighted Views." IEEE Trans Cybern. **45**(4):688–701.

69. a, bWang Y, Chen L (2017). "Multi-View Fuzzy Clustering with Minimax Optimization for Effective Clustering of Data from Multiple Sources." Expert Syst Appl. **72**:457–466.

70. ^Gothania N, Kumar S (2018). "Multi-View Fuzzy Clustering with Weighted Attributes and Views."

71. a, bYang M, Sinaga K (2021). "Collaborative Feature-Weighted Multi-View Fuzzy C-Means Clustering." Pattern Recognit. **119**:108064.

72. ^Feng W, Wu Z, Wang Q, Dong B, Tao Z, Gao Q (2024). "Efficient Federated Multi-View Clustering with Integrated Matrix Factorization and K-Means."

73. ^Li Y, Hu X, Yu S, Ding W, Pedrycz W, Kiat Y, Liu Z (2025). "A Vertical Federated Multiview Fuzzy Clustering Method for Incomplete Data." IEEE Trans Fuzzy Syst. **33**(5):1510–1524.

74. ^Li F, Jiang J, Wang J, Du L, Qian Y (2026). "Vertical Federated K-Means for Multi-View Data Guided by a K-Means Cost Bound after Projection."

75. ^Yang Q, Liu Y, Chen T, Tong Y (2019). "Federated Machine Learning: Concept and Applications." ACM Trans Intell Syst Technol. **10**(2):1–19.

76. ^Yang Q, Liu Y, Cheng Y, Kang Y, Chen T, Yu H (2022). Horizontal Federated Learning. Springer.

77. ^Liu Y, Kang Y, Zou T, Pu Y, He Y, Ye X, Ouyang Y, Zhang Y, Yang Q (2024). "Vertical Federated Learning: Concepts, Advances, and Challenges." IEEE Trans Knowl Data Eng. **36**(7):3615–3634.

78. ^Liu Y, Kang Y, Xing C, Chen T, Yang Q (2020). "A Secure Federated Transfer Learning Framework." IEEE Intell Syst. **35**(4):70–82.

79. ^Li T, Sahu A, Zaheer M, Sanjabi M, Talwalkar A, Smith V (2020). "Federated Optimization in Heterogeneous Networks." Proceedings of Machine learning and systems. **2**:429–450.

80. ^Konečnỳ J, McMahan H, Ramage D, Richtárik P (2016). "Federated Optimization: Distributed Machine Learning for On-Device Intelligence." arXiv preprint arXiv:1610.02527.

81. ^Yuan Y, Xun G, Jia K, Zhang A (2018). "A Multi-View Deep Learning Framework for EEG Seizure Detection." IEEE J Biomed Health Inform. **23**(1):83–94.

82. ^Yang S, Lian C, Zeng Z, Xu B, Zang J, Zhang Z (2023). "A Multi-View Multi-Scale Neural Network for Multi-Label ECG Classification." IEEE Trans Emerging Top Comput Intell. **7**(3):648–660.

83. ^Liu J, Ge S, Cheng Y, Wang X (2021). "Multi-View Spectral Clustering Based on Multi-Smooth Representation Fusion for Cancer Subtype Prediction." Front Genet. **12**:718915.

84. ^Puyol-Antón E, Ruijsink B, Gerber B, Amzulescu M, Langet H, De Craene M, Schnabel J, Piro P, King A (2018). "Regional Multi-View Learning for Cardiac Motion Analysis: Application to Identification of Dilated Cardiomyopathy Patients." IEEE Trans Biomed Eng. **66**(4):956–966.

85. ^Zhang J, Huan J (2013). "Predicting Drug-Induced QT Prolongation Effects Using Multi-View Learning." IEEE Trans Nanobioscience. **12**(3):206–213.

86. ^Dornaika F, El Hajjar S, Charafeddine J (2024). "Towards Unsupervised Radiograph Clustering for Covid-19: The Use of Graph-Based Multi-View Clustering." Eng Appl Artif Intell. **133**:108336.

87. ^Zheng L, Zhu Y, He J (2023). "Fairness-Aware Multi-View Clustering."

88. ^Kam-Kwai W, Luo Y, Yue X, Chen W, Qu H (2025). "Prismatic: Interactive Multi-View Cluster Analysis of Concept Stocks." IEEE Trans Vis Comput Graph.

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.