Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8081177, 14 pages
https://doi.org/10.1155/2022/8081177

Research Article

A Generative Clustering Ensemble Model and Its Application in IoT Data Analysis

Hangyuan Du ,[1] Wenjian Wang,[1] Liang Bai,[2] and Jinsong Feng[3]

[1]*School of Computer and Information Technology, Shanxi University, Taiyuan, 030006 Shanxi, China*
[2]*Institute of Intelligent Information Processing, Shanxi University, Taiyuan, 030006 Shanxi, China*
[3]*Taiyuan Urban and Rural Administration Bureau, Taiyuan, 030002 Shanxi, China*

Correspondence should be addressed to Hangyuan Du; duhangyuan@sxu.edu.cn

Received 13 April 2022; Revised 27 May 2022; Accepted 6 June 2022; Published 29 June 2022

Academic Editor: Wentao Li

Data analysis is the foundation of Internet of Things (IoT) based applications, and clustering is an effective technology of data analysis. Clustering ensemble integrates multiple base clustering results to obtain a consensus result and thus improves the clustering performance in stability and robustness. However, it is difficult for existing clustering ensemble algorithms to achieve a satisfying ensemble result, when the base clustering results are unreliable. Concerning this problem, we develop a new clustering ensemble model in this paper, which has several advantages compared with traditional algorithms: (i) structure information about the data is effectively extracted from the base clusterings; (ii) data characteristics and structure information are integrated in an elegant fashion, in the production of the consensus clustering result; and (iii) our model has the generative ability that makes the model achieve outstanding performance when training samples are insufficient. In our model, the structural information is extracted by explicating the coupling relationships between base clusterings and between samples in clustering members. Then, data characteristics and structure information are combined in a generative graph representation learning framework. And the objectives of representation learning and consensus clustering are integrated into a unified optimization model, in which the prior distribution of the data is approximated by a Gaussian mixture model (GMM). Extensive experiments are conducted on multiple IoT datasets; the results prove that our model not only performs better than the conventional clustering ensemble algorithms but also outperforms the state-of-the-art deep clustering methods.

1. Introduction

With the rapid development and widespread use of IoT technology, many types of data are produced constantly with unprecedented speed. The effective analysis and mining around these huge amounts of data have gradually become an important requirement to enhance the value of IoT data [1]. As a typical data mining technology, clustering analysis plays an important role in many IoT data analysis scenarios, such as network energy saving [2, 3], privacy protection [4], attack detection [5, 6], service computing [7], and pattern discovery [8, 9]. The goal of clustering analysis is to divide the unknown data into a set of clusters based on a certain similarity measurement between data samples, so that samples in the same cluster are close to each other, and those in different clusters are different from each other. With the help of clustering analysis, the distribution pattern hidden in the unknown data can be easily identified. Typical clustering strategies mainly consist of partitioning methods, hierarchical methods, grid-based methods, and graph methods. In the past decades, a great deal of research works have been conducted to improve the clustering performance from multiple directions, such as similarity measurement, cluster number recognition, cluster structure optimization, atypical data processing, and performance evaluation [10, 11]. The performance and adaptability of clustering analysis in various data analysis scenarios have been improved significantly. However, as an unsupervised learning method, clustering analysis has the following limitations: (i) due to the lack of supervision information, the design of the clustering algorithm depends on human's subjective hypothesis. As a result, different clustering algorithms may get distinct partitioning results on the same data. (ii) Searching the optimal clustering result is often a nonconvex optimization problem; thus, clustering result always depends on the input parameters and initializations to a great extent. (iii) Real-world data, such as the IoT data, is often multidimensional or multisource. Therefore, one cluster may have diverse distributions or structures from different perspectives. And it is difficult for any clustering method to identify all cluster patterns completely.

To solve these problems, clustering ensemble is proposed driven by the idea of ensemble learning. Clustering ensemble can obtain a final partition result which describes the inner cluster structure of the data more effectively, by combining multiple clustering results on the data. Compared with any single clustering algorithms, the result of clustering ensemble has several significant advantages in terms of reliability, robustness, interpretability, and scalability. Besides, clustering ensemble is friendly to parallel computation and distributed deployment. There are two main phases in the clustering ensemble [12]: (i) generating a set of cluster partitions for the data, which are called base clusterings, and (ii) designing an efficient consensus function to integrate base clusterings into a final partition result. It is shown that the validity of ensemble result is closely related to the diversity of base clusterings. To this end, several strategies are used to generate disparate base clusterings [13], such as using different clustering algorithms, setting different parameters or initializations for a clustering algorithm, extracting different subsets of data, and projecting the data to different feature spaces. To produce a consensus clustering result by combining base clusterings, many works focused on designing various consensus functions for the clustering ensemble model. Each consensus function abstracts the base clustering results to a specific form of ensemble-information matrix. Based on three general types of such matrix (i.e. the label-assignment matrix, the pairwise similarity matrix, and the binary cluster-association matrix), different consensus functions found in these works can be categorized to four major families: (i) relabeling strategy [14]. These algorithms find label correspondence and relabel each partition in accordance with a reference partition and produce the final result by use of a combination method such as voting. (ii) Feature-based methods [15, 16]. These techniques predict cluster assignments using the nominal information that is originally obtained from base clusterings, without searching for correspondence among labels or relabelling. (iii) Pairwise similarity-based algorithms [17, 18]. This specific category of clustering ensemble methods is based principally on the pairwise similarity among data samples. (iv) Graph-based approaches [19, 20]. This family of strategies utilizes the graph structure to solve the clustering ensemble problem. They generally construct a weighted graph from the base clusterings and produce the final result by partitioning the graph using certain graph partitioning methods. In recent years, some clustering ensemble algorithms [18, 21] have been used in IoT data analysis and achieved superior performance to a single clustering algorithm. It is worth noting that IoT data always includes a large number of explicit characteristic information, as well as abundant structure information that describes the intrinsic organization of the data. The data characteristics and structure information describe IoT data from different aspects; therefore, both of them can provide valuable guidance on producing the final ensemble result. However, existing clustering ensemble algorithms, according to our knowledge, produce the final partition result by employing base clusterings either in feature space or exploring structure relations. They seldom consider to combine these two types of data information in the design of the consensus function. This limitation raises a problem that the clustering ensemble result may be suffering from the unreliable base clusterings or incomplete data information.

To explore and utilize various types of information implicit in the IoT data comprehensively, we propose a novel clustering ensemble model in this paper, which integrates data characteristics with structure information in producing the consensus clustering result. As will be discussed, our work devotes to solve the following two key problems to achieve the information integration: (i) how can we extract effective structural information hidden in the raw data? In general, structure information of the data can be expressed by certain relationships, such as the pairwise similarity among data samples, the nominal information originally obtained from an ensemble, and the associations between data samples or those among clusters. In fact, the raw data and base clustering results can be viewed as different organization forms for same data samples. Therefore, extracting the structure information by solely focusing on the similarity between data samples or associations among base clusterings is far from sufficient. This is the first key problem we intend to address. (ii) How to integrate data characteristics and structure information into appropriate representations for producing the final ensemble result? Data characteristics and structure information describe the data in different space and interact with each other in the formation of cluster structure. However, they cannot be simultaneously processed in existing ensemble strategies. Therefore, how to combine these two different types of information elegantly and learn their appropriate representations for clustering ensemble is another key problem.

For the first key problem, we consider to capture the coupled clustering and sample similarity from base clusterings to describe the structure information of data. On the one hand, all the base clusterings are produced on the same data, and there must be some relationships among those ensemble members. On the other hand, samples from the data are more or less associated in terms of certain coupling relationships rather than independent. Based on these knowledge, we plan to extract structural information by explicating and integrating the coupling relationships between base clusterings and between data samples.

To address the second key problem, we employ a variational graph autoencoder (VGAE) [22] module to learn the specific representations from both data characteristics and structure information, which are suitable for clustering objective. And by assuming the prior statistic of the latent representations to be a Gaussian mixture distribution, we derive a joint optimization model which combines representation learning and cluster partitioning into a unified framework.

In this work, we propose a novel clustering ensemble model with the motivation that integrates data characteristics with structure information in aggregation of the clustering results by employing the powerful representation ability of deep learning. In fact, our work can be viewed as an improvement for clustering ensemble approach with structure constraints to handle data with complex distribution. Alternatively, our work can be also viewed as an enhancement of deep clustering method by imposing a global model explicitly in latent space. Our main contributions can be summarized as follows:

(i) We discuss how to capture effective structure information for the data by exploiting the base clustering results comprehensively

(ii) We design an encoder-decoder specific network to transform the integration of data characteristics and structure information into a graph representation learning problem by treating the data as a graph organized by global structure relationships. We employ a mixture of Gaussian to approximate the prior distribution of the latent representation, which is a tractable parametric model for clustering tasks by nature

(iii) We construct a unified optimization model with aggregation of representation learning and consensus cluster partitioning and show how to train the network by maximizing the evidence lower bound (ELBO) using the stochastic gradient variational Bayes (SGVB) estimator and the reparameterization trick

(iv) Extensive experiments on several IoT datasets demonstrate the superiority of our approach in comparison with several clustering ensemble algorithms and deep clustering methods

The remainder of this paper is organized as follows. In Section 2, some related works are introduced, respectively. In Section 3, our generative clustering ensemble model is proposed, and each component in the model is illustrated in detail. To evaluate the performance of the proposed model, a series of experiments are conducted and analyzed in Section 4. Finally, the conclusions and discussions of future work are given in Section 5.

2. Related Work

In this section, we introduce the most related works: autoencoder (AE) and variational autoencoder (VAE), representation learning for clustering, and graph representation learning.

2.1. Autoencoder and Variational Autoencoder. AE can be regarded as a nonlinear generalization of PCA to reduce data dimensionality, in which high-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer. AE is composed of three basic elements: an adaptive, multilayer encoder network that transforms the high-dimensional data into a low-dimensional code; a similar decoder network that recovers the data from the code; and a loss function that evaluates the lost information in dimensionality reduction. There are several important characters of autoencoder: (i) the module is learned automatically from data samples, (ii) the reconstructed data is degenerated compared with the original data, and (iii) the module is data-specific.

Using a similar encoder-decoder structure, the idea of VAE actually has relatively little to do with classical AE models but is deeply rooted in the variational Bayesian methods [23, 24]. Instead of mapping the input into a fixed vector, the VAE maps it into a distribution in the latent space. And by sampling from the latent distribution, the decoder network can be viewed as a generative module that creates some new samples similar to, but not identical to, the training data. The assumptions of this model are relatively weak, and its training process is fast via backpropagation. VAE does make an approximation, but the error introduced by this approximation is arguably small. These characteristics significantly make VAE to be a popular generative model.

2.2. Representation Learning for Clustering. Recently, some works focus on exploiting the powerful representation ability of deep learning model to learn a better data representation for clustering task, in order to improve the clustering performance. In [25], a two-stage deep clustering framework is constructed, in which deep learning is used to acquire feature representations in subspace, and then, these features are used to predict the cluster assignments. To guarantee that the learned features are suitable for clustering task, many latter works attempt to incorporate clustering objective into the deep learning framework. Specifically, the deep embedded clustering (DEC) algorithm [26] learns low-dimensional data representations using an autoencoder construction without the decoder and proposes an assistant objective distribution based on soft cluster assignment produced on the learned representations. In DCE, the clustering optimization and training of encoder parameters are simultaneously implemented in a self-learning form. To overcome the misguidance issue in feature mapping, Guo et al. [27] utilized a complete autoencoder to improve the DEC, in which the clustering loss not only accomplishes cluster partitioning but also guarantees that the learned data representations maintain the original local structure. Similarly, DECE [28] uses a convolutional autoencoder and a single layer classifier to learn the data representation and the cluster distributions, respectively, in which the DNN is optimized by minimizing the reconstruction error and the relative entropy between the cluster distributions and their priori. In [29], the authors proposed a joint dimensionality reduction (DR) and K-means clustering algorithm, to obtain the "clustering-friendly" latent representations and better clustering result simultaneously. Its optimization criterion is composed of three parts: dimensionality reduction which is realized by a SAE framework, data reconstruction, and cluster structure-promoting regularization. Bo et al. [30] developed a structural deep clustering network (SDCN) to integrate the structural information into deep clustering, in which a delivery operator is designed to transfer the representations learned by autoencoder to the corresponding GCN layer, and a dual self-supervised mechanism is constructed to unify these two DNN architectures and guide the update of the whole model.

In the above algorithms, AE learns effective representations for input data and then reconstructs samples with one-to-one correspondence between original data. Unfortunately, the deep clustering models may suffer from the overfitting problem caused by AE networks, due to its powerful learning capacity and the insufficiency of training samples. To overcome this drawback, some works attempt to replace AE with VAE as the network construction for deep clustering. In VAE, the encoder turns to find the mapping relation of data distribution. The latent variables sampled from the learned distribution can effectively capture statistic characteristics of the data. And then, the decoder which is also named generator is capable of generating new data samples for any cluster distribution. This mechanism will help the clustering model acquire more abundant information about the inherent cluster structure of the data. For example, Jiang et al. [31] design an unsupervised generative deep clustering algorithm variational deep embedding (VaDE), in which the cluster distributions are modeled by GMM, and the latent representations of data are learned by DNN. The solution of VaDE is realized by variational inference; specifically, its ELBO is optimized using the SGVB estimator and the reparameterization trick. Choong et al. [32] considered the community structure discovering as a graph clustering problem and proposed a generative model, namely, variational graph autoencoder for community detection (VGAECD). Unlike traditional approaches, the VGAECD does not require a predefined community structure, and it is capable of exploiting feature-rich information of a network. Hwang et al. [33] addressed the issue of clustering complex and high-dimensional wafer maps in semiconductor manufacturing, by proposing a variational deep clustering algorithm namely one-step VAE+DPGMM. In this algorithm, a GMM is implemented to a VAE framework to extract more suitable features for the clustering task, and a Dirichlet process is further applied in the variational autoencoder mixture framework for automated one-step clustering. Compared with conventional two-step clustering methods, the model can considerably increase the chance to distinguish small differences of wafer map patterns.

All the algorithms above construct deep clustering framework based on deep networks, and they provide effective data representations for clustering task by introducing the powerful representational ability of the deep network. However, these algorithms all focus on the general clustering problem, rather than clustering ensemble. And they predict the cluster assignments by use of data characteristics but neglect the structure information implicit in the data. This fact motivates us to consider how to take full advantage of deep networks to learn appropriate representations for multiple data information, in order to enhance the performance of clustering ensemble model.

2.3. Graph Representation Learning. The structure information reveals the intrinsic relationships among data samples, which can provide an important guidance on learning the data representation for clustering ensemble task. As a ubiquitous data organizational form, graph is of many intuitive advantages on describing structure information of the data. It can capture interactions between data samples and make the structure information be efficiently recorded and accessed. In order to incorporate the graph-formed structure information into a machine learning model, graph representation learning can be employed to encode the high-dimensional, non-Euclidean graph information into a feature vector. Graph representation learning is aimed at converting graph data into a low-dimensional, compact, and continuous feature space and preserves the topological structure, vertex content, and other side information in graph as complete as possible. From the encoder-decoder perspective, various graph representation learning methods can be abstracted to a framework consisting two key mapping functions [34]: an encoder, which maps each vertex to a low-dimensional vector or embedding, and a decoder, which reconstructs the graph data from the learned embeddings. Generally, the objective of the encoder-decoder graph representation learning model is optimized by minimizing the reconstruction error or loss of the pairwise vertex similarities between the input graph and the reconstructed graph. Most graph representation learning methods fall into two broad categories: (i) shallow representation approaches, which are largely inspired by classic matrix factorization techniques [35] or random walks [36, 37] using an embedding lookup encoder function, and (ii) generalized encoder-decoder architectures, which use a more complex encoder, often based on DNNs [38, 39] and dependent on the topological structure and vertex attributes [40] of the graph more generally. Among the latter categories, VAE is a common DNN construction. Some works introduce VAE by adding a prior constraint to compress information about a node's local neighborhood. For example, the algorithm in [22] learns representations of an attribute network under the VAE framework by employing a graph convolutional network (GCN) encoder and an inner product decoder. To address the incomplete filtering issue encountered in traditional GCN-based graph autoencoders, [41] proposed graph convolutional autoencoders with colearning of graph structure and node attributes (GASN) based on VAE. The GASN encodes and decodes the node attributes and graph structure comprehensively by use of a completely low-pass graph encoder and a high-pass graph decoder.

3. The Proposed Model

Given a set of N IoT data samples $X = \{x_i\}_{i=1}^{N}$ in D-dimensional space, each sample x_i is represented by a vector of D attribute values. Also, let $\Pi = \{\pi_t\}_{t=1}^{T}$ be a set of T base clusterings and $\pi_t = \{C_\gamma^t\}_{\gamma=1}^{k_t}$ be the t^{th} base clustering, such that $\bigcup_{\gamma=1}^{k_t} C_\gamma^t = X$, where k_t denotes the number of clusters in π_t and C_γ^t is the γ^{th} cluster. For each x_i, λ_i^t denotes the cluster

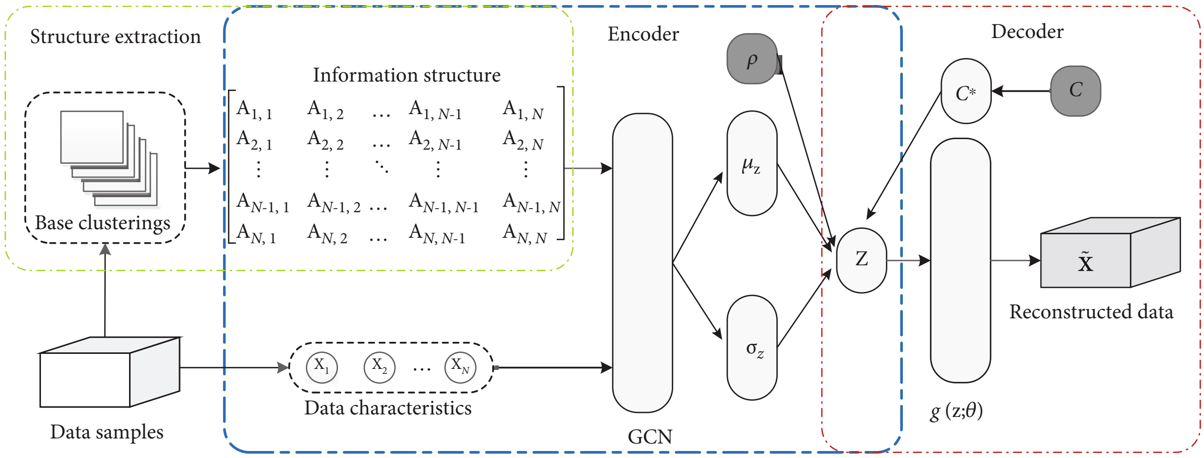

Figure 1: Overall framework of the proposed GCE model. The proposed model consists of three components. (i) In the structure extraction component, structure information among data samples is extracted from a series of base clusterings at first. (ii) In the encoder component, data characteristics and structure information are encoded integrated by a GCN module, which learns parameters for the distribution of latent representation. (iii) In the decoder component, $\tilde{X}$ is reconstructed by a generator $g(z;\theta)$, and the representation z sampled from the learned latent distribution is assumed to lie in a cluster characterized by a Gaussian component of the GMM. The objective of representation learning and clustering ensemble are jointly optimized by maximizing the ELBO of GCE, which is calculated and backpropagated to the latent representation.

label in the t^{th} base clustering to which x_i belongs. The label set containing all the different labels in base clustering π_t is denoted by Λ_t, and the set of samples whose cluster labels is λ_i^t in π_t is specified as $s_t(\lambda_i^t)$. The problem of clustering ensemble is to find a new partition $\pi^* = \{C_k^*\}_{k=1}^K$, where K denotes the number of clusters in the final result.

3.1. Overview of the Model. In this section, we will illustrate our proposed generative clustering ensemble (GCE) model, where the overall framework is shown in Figure 1. In this work, the data characteristics mainly refer to the data features, which are used to express properties of individual data object. And the structure information represents the relationships between data objects, which are used to describe connections between different objects intrinsic in the data. We first discuss how to effectively extract structure information from an ensemble of base clusterings. Then, we construct a variational network that combines representation learning and clustering ensemble objective into a joint optimization process. In the representation learning phase, we treat data samples as the vertexes of a relationship graph and integrate data characteristics with structure information in a VGAE module. While in the production of the consensus clustering result, we utilize a GMM to approximate the prior distribution of the latent representation. At last, we optimize the encoder-decoder module and the GMM jointly in the form of stochastic inference.

3.2. Structure Information Extracting. In our proposed GCE model, it is important to construct an appropriate affinity matrix to describe structure information of the data. In many graph-based clustering ensemble methods, the affinity matrix is usually constructed by finding several nearest neighbors for a given sample and evaluating their similarities through a certain mapping function. The similarity defined in this way is limited in revealing the local relationships between data samples rather than global relationships among the whole data. In clustering ensemble context, we can actually acquire richer information about data structure from base clusterings. Different from typical methods, we believe that the global relationships among the data can be determined by two elements [42]: coupling of base clusterings and coupling of data samples.

3.2.1. Coupling of Base Clusterings. The coupling of base clusterings is defined to represent relationship among different ensemble members, which are composed of two components: intracoupling that reflects the involvement of cluster label occurrence frequency within a base clustering and intercoupling that indicates the interaction between two base clusterings. Specifically, we define the coupled clustering similarity for clusters (CCSC) between two cluster labels in a certain base clustering to describe the coupling relationship of base clusterings, which can be calculated as

$$\mathrm{CCSC}\left(\lambda_i^t, \lambda_j^t \mid \{\Lambda_o\}_{o=1}^T\right) = \mathrm{IaCSC}\left(\lambda_i^t, \lambda_j^t\right) \mathrm{IeRSC}\left(\lambda_i^t, \lambda_j^t \mid \{\Lambda_o\}_{o\neq t}\right). \tag{1}$$

$\mathrm{CCSC}(\lambda_i^t, \lambda_j^t \mid \{\Lambda_o\}_{o=1}^T)$ is the CCSC between λ_i^t and λ_j^t, $\mathrm{IaCSC}(\lambda_i^t, \lambda_j^t)$ is the intracoupled clustering similarity for clusters (IaCSC) between λ_i^t and λ_j^t, $\mathrm{IeRSC}(\lambda_i^t, \lambda_j^t \mid \{\Lambda_o\}_{o\neq t})$ is the intercoupled relative similarity for clusters (IcRSC) between λ_i^t and λ_j^t based on another base clustering π_o, and Λ_o is the label set of π_o. The IaCSC captures the base clustering frequency distribution by calculating the frequency of cluster labels within a base clustering, and it is defined as

$$\mathrm{IaCSC}\left(\lambda_i^t, \lambda_j^t\right) = \frac{\left|s_t(\lambda_i^t)\right|\left|s_t\left(\lambda_j^t\right)\right|}{\left|s_t(\lambda_i^t)\right| + \left|s_t\left(\lambda_j^t\right)\right| + \left|s_t(\lambda_i^t)\right|\left|s_t\left(\lambda_j^t\right)\right|}. \tag{2}$$

The IcRSC characterizes the base clustering dependency aggregation by comparing cooccurrence of the cluster labels among different base clusterings. It is defined as

$$\mathrm{IeRSC}\left(\lambda_i^t, \lambda_j^t \mid \{\Lambda_o\}_{o \neq t}\right) = \sum_{o=1, o \neq t}^{T} \omega_o \mathrm{Sim}_{t|o}\left(\lambda_i^t, \lambda_j^t \mid \Lambda_o\right), \tag{3}$$

where $\omega_o \in [0, 1]$ is the weight of base clustering π_t, $\sum_{o=1,o\neq t}^{T} \omega_o = 1$, and $\mathrm{Sim}_{t|o}(\lambda_i^t, \lambda_j^t \mid \Lambda_o)$ is calculated as

$$\mathrm{Sim}_{t|o}\left(\lambda_i^t, \lambda_j^t \mid \Lambda_o\right) = \sum_{\lambda^o \in \Omega} \min\left\{\frac{\left|s_o(\lambda^o) \cap s_t(\lambda_i^t)\right|}{\left|s_t(\lambda_i^t)\right|}, \frac{\left|s_o(\lambda^o) \cap s_t\left(\lambda_j^t\right)\right|}{\left|s_t\left(\lambda_j^t\right)\right|}\right\}. \tag{4}$$

In equation (4), Ω is a set that represents $L_o(s_t(\lambda_i^t)) \cap L_o(s_t(\lambda_j^t))$, and $L_o(s_t(\lambda_i^t))$ is the subset of cluster labels in base clustering π_o for the corresponding samples $s_t(\lambda_i^t)$.

3.2.2. Coupling of Data Samples. Similarly, the coupling relationships among data samples can be also discussed from intraperspective and interperspective, respectively. In terms of the intraperspective, the similarity between x_i and x_j is represented by intracoupled sample similarity (IaSS), which is defined as

$$\mathrm{IaSS}\left(x_i, x_j\right) = \frac{1}{T}\sum_{t=1}^{T} \mathrm{IeRSC}\left(\lambda_i^t, \lambda_j^t \mid \{\Lambda_o\}_{o \neq t}\right). \tag{5}$$

The IaSS refers to the average sum of the CCSC between the associated cluster labels ranging over all the base clusterings. From the interperspective, we can describe the interaction between different samples by mining the correlation among their neighbors. Accordingly, we define the intercoupled sample similarity (IeSS) between two samples using their common neighbors:

$$\mathrm{IeSS}\left(x_i, x_j\right) = \frac{1}{N}\left|\left\{x_n \in X \mid x_n \in N_{x_i} \cap N_{x_j}\right\}\right|, \tag{6}$$

where $\mathrm{IeSS}(x_i, x_j)$ is the IeSS between samples x_i and x_j in terms of other samples in X. N_{x_i} is the neighbor set of x_i, and it is defined as

$$N_{x_i} = \{x_n \mid \kappa(x_i, x_n) \geq \theta\}, \tag{7}$$

where $\kappa(\cdot, \cdot)$ is the kernel function, and $\theta \in [0, 1]$ is a threshold of κ. For instance, with the Gaussian kernel, the $\kappa(x_i, x_n)$ is defined as

$$\kappa(x_i, x_n) = \frac{1}{\alpha_i} \exp\left(-\frac{\varphi(x_i, x_n)^2}{2\vartheta^2}\right), \tag{8}$$

where α_i is a normalizer to make $\sum_n \kappa(x_i, x_n) = 1$, $\vartheta > 0$ is the width of Gaussian kernel, and $\varphi(\cdot)$ denotes a certain similarity measure for samples, such as Euclidean dissimilarity for numeric charicteristics or Jaccard coefficient for categorical attributes.

3.2.3. Construction of Structure Information. Obviously, the position of a data sample in a clustering depends on which cluster it belongs to. Thus, the clustering coupling and sample coupling can be integrated through the corresponding clusters. Specifically, we employ IaSS as the similarity measure in equation (8) to define coupled clustering and sample similarity (CCSS) between samples, and we have

$$\mathrm{CCSS}\left(x_i, x_j\right) = \frac{1}{N}\left|\left\{x_n \in X \mid x_n \in N_{x_i}^{\mathrm{IaSS}} \cap N_{x_j}^{\mathrm{IaSS}}\right\}\right|, \tag{9}$$

where $\mathrm{CCSS}(x_i, x_j)$ is the CCSS between x_i and x_j and the neighbor sets are defined as $N_{x_i}^{\mathrm{IaSS}} = \{x_n \mid \kappa^{\mathrm{IaSS}}(x_i, x_n) \geq \theta\}$ and $N_{x_j}^{\mathrm{IaSS}} = \{x_n \mid \kappa^{\mathrm{IaSS}}(x_j, x_n) \geq \theta\}$, respectively. The kernel function in equation (8) can be rewritten as

$$\kappa(x_i, x_n)^{\mathrm{IaSS}} = \frac{1}{\alpha_i} \exp\left(-\frac{\mathrm{IaSS}(x_i, x_n)^2}{2\vartheta^2}\right). \tag{10}$$

In this way, the CCSS not only takes into account both the intracoupled and intercoupled interactions between base clusterings but also incorporates both the intracoupled and intercoupled relationships between samples. Given a series of clustering members, we can define an affinity matrix $A_{N\times N}$ that stores the structure information about the data, and each entry $A_{i,j}$ of the matrix denotes the global similarity between samples x_i and x_j. Specially, we set the value of $A_{i,j}$ as

$$A_{i,j} = \mathrm{CCSS}\left(x_i, x_j\right), \tag{11}$$

where $\mathrm{CCSS}(x_i, x_j)$ is defined in equation (9).

3.3. The Generative Clustering Ensemble Model. Based on the affinity matrix, the data can be viewed as a sample similarity graph implying both data characteristics and structure information. To combine these two types of data description, we design a joint clustering ensemble model within the framework of VGAE, in which the priori of the latent representation is approximated as a mixture of Gaussian distributions.

3.3.1. Inference Model. Given a dataset X and its structure information formed by an affinity matrix A, the inference model is parameterized by a two-layer GCN:

$$q(Z \mid X, A) = \sum_{i=1}^{N} q(z_i \mid X, A). \tag{12}$$

Here, $Z = \{z_i\}_{i=1}^{N}$ denotes latent representation of X. For a certain data sample x and its structure information vector a, the corresponding latent vector in the low-dimentional space can be obtained by

$$q(z \mid x, a) = N\left(z \mid \mu_z, \operatorname{diag}\left(\sigma_z^2\right)\right), \tag{13}$$

where $\mu_{z,i}$ and $\sigma_{z,i}^2$ are mean and variance of the i^{th} latent vector z_i. Each latent vector is sampled from a distribution obtained by a GCN, i.e., $\mu_z = \text{GCN}_\mu(X, A)$, and $\log \sigma_z = \text{GCN}_\sigma(X, A)$. The GCN structure is defined as

$$\text{GCN}(X, A) = \text{Gconv}(\text{ReLU}(\text{Gconv}(A, X ; W_0)) ; W_1), \tag{14}$$

where the $\text{Gconv}(\cdot)$ function is a graph convolutional layer and W_0 and W_1 are learnable weight matrices for the first layer and second layer, respectively. And W_0 is shared between $\text{GCN}_\mu(X, A)$ and $\text{GCN}_\sigma(X, A)$.

3.3.2. Generative Model. In our model, we assume the training data is generated from a Gaussian mixture distribution; i.e., the clustering ensemble result $\{C_k^*\}_{k=1}^{K}$ can be approximated by a GMM, and each sampled latent vector should lie in a cluster modeled by one Gaussian component with a certain probability. For each training data sample x, we learn a latent representation z and introduce a K-dimensional vector c satisfying $c_k > 0$ and $\sum_{k=1}^{K} c_k = 1$ to indicate the prior cluster distributions of the data. The generative process can be modeled as follows:

From the consistent clustering partition, sample a cluster $C_k^* \sim \text{Cat}(c)$, where $\text{Cat}(c)$ is the categorical distribution parameterized by c.

(i) From the picked cluster, sample a vector $z \sim N(z \mid \mu_{c,k}, \operatorname{diag}(\sigma_{c,k}^2))$, where $\mu_{c,k}$ and $\sigma_{c,k}^2$ denote the mean and variance of the k^{th} Gaussian component, respectively

(ii) From the reconstructed [id = V2]datadataset $\tilde{X} = \{\tilde{x}_i\}_{i=1}^{N}$, sample a vector a. For binary data, choose $a \sim \text{Ber}(\mu_{\tilde{x}})$, where $\text{Ber}(\mu_{\tilde{x}})$ is a multivariate Bernoulli distribution, and $\mu_{\tilde{x}}$ is computed by $\mu_{\tilde{x}} = g(z ; \phi)$. For real-value data, choose $a \sim N(\mu_{\tilde{x}}, \operatorname{diag}(\sigma_{\tilde{x}}^2))$, where $N(\mu_{\tilde{x}}, \operatorname{diag}(\sigma_{\tilde{x}}^2))$ is a multivariate Gaussian distribution, and $\mu_{\tilde{x}}, \sigma_x^2$ are learned by $[\mu_{\tilde{x}} ; \log \sigma_{\tilde{x}}^2] = g(z ; \phi)$. The function $g(z ; \phi)$ is a nonlinear function parameterized by ϕ, and in our model, the inner product decoder is used

$$g(z ; \phi) = \sigma\left(z_i^T z_j\right). \tag{15}$$

According to the above generative process, we can factorize the joint probability $p(a, z, C_k^*)$ as

$$p(a, z, C_k^*) = p(a \mid z) p(z \mid C_k^*) p(C_k^*). \tag{16}$$

Since a and C_k^* are independently conditioned on z, we have

$$p(a \mid z) = \text{Ber}(\mu_{\tilde{x}}) or N\left(\mu_{\tilde{x}}, \operatorname{diag}\left(\sigma_{\tilde{x}}^2\right)\right), \tag{17}$$

$$p(z \mid C_k^*) = N\left(z \mid \mu_{c,k}, \operatorname{diag}\left(\sigma_{c,k}^2\right)\right), \tag{18}$$

$$p(C_k^*) = Cat(C_k^* \mid c). \tag{19}$$

3.4. Learning Algorithm. Our GCE model can be tuned by maximizing the log-likelihood of the given data samples as

$$\max_{W, \phi, C^*} \sum_x \log p_\phi(x) = \max_{W, \phi, C^*} \sum_x \log \int_z \sum_{C^*} p_\phi(x, z, C_k^*). \tag{20}$$

By using Jensen's inequality, we have

$$\log p_\phi(x) > L_{\text{ELBO}}(x) = E_{q(z, C^* \mid x, a)}\left[\log \frac{p(a, z, C_k^*)}{q(z, C_k^* \mid x, a)}\right], \tag{21}$$

where $L_{\text{ELBO}}(x)$ denotes the evidence lower bound (ELBO) of x and $q(z, C_k^* \mid x, a)$ is the variational posteriori approximating the true posterior $p(z, C_k^* \mid x, a)$. By assuming $q(z, C_k^* \mid x, a)$ to be a mean field distribution, we can factorize it as

$$q(z, C_k^* \mid x, a) = q(z \mid x, a) q(C_k^* \mid x, a). \tag{22}$$

According to equations (16) and (22), the $L_{\text{ELBO}}(x)$ can be rewritten as equation (23). By submitting the inference model $q(z \mid x, a)$ defined by equations (13), (17), (18), and (19) and using the Monte Carlo SGVB estimator, the $L_{\text{ELBO}}(x)$ can be further rewritten as equation (24).

$$\begin{aligned} L_{\text{ELBO}}(x) &= E_{q\left(z, C_k^* \mid x, a\right)}\left[\log \frac{p(a \mid z) p(z \mid C_k^*) p(C_k^*)}{q(z \mid x, a) q(C_k^* \mid x, a)}\right] \\ &= E_{q\left(z, C_k^* \mid x, a\right)}[\log p(a \mid z) + \log p(z \mid C_k^*) + \log p(C_k^*) \\ &\quad - \log q(z \mid x, a) - \log q(C_k^* \mid x, a)]. \end{aligned} \tag{23}$$

$$
\begin{aligned}
L_{\mathrm{ELBO}}(x) = {} & \frac{1}{M}\sum_{m=1}^{M}\sum_{d=1}^{D} x_d \log \mu_{\tilde{x}}^{(m)}\Big|_d \\
& + (1-x_d)\log\left(1-\mu_{\tilde{x}}^{(m)}\Big|_d\right) - \frac{1}{2}\sum_{k=1}^{K} q(C_k^* \mid x, a)\sum_{r=1}^{R} \\
& \left(\log \sigma_c^2\Big|_r + \frac{\sigma_z^2\big|_r}{\sigma_{c,k}\big|_r} + \frac{(\mu_z|r-\mu_c|r)^2}{\sigma_{c,k}^2\big|_r}\right) \\
& + \sum_{k=1}^{K} q(C_k^* \mid x, a)\log\frac{p(C_k^*)}{q(C_k^* \mid x, a)} \\
& + \frac{1}{2}\sum_{r=1}^{R}\left(1+\log\sigma_z^2\Big|_r\right),
\end{aligned}
\tag{24}
$$

where M is the total number of samples in the SGVB estimator, D and R are the dimensionalities of training data and latent vector, respectively, x_d is the d^{th} element of x, $\bullet|_i$ and $\bullet|_r$ denote the i^{th} and r^{th} element of vector $\bullet$, respectively. $\mu_{\tilde{x}}^{(m)}$ is calculated by $\mu_{\tilde{x}}^{(m)} = g(z^{(m)}; \phi)$, in which $z^{(m)}$ is the m^{th} Monte Carlo sample picked from $q(C_k^* \mid x, a)$ defined in equation (13). In order to employ gradient backpropagation on the stochastic layer, the reparameterization trick is used here, and $z^{(m)}$ can be calculated as

$$
z^{(m)} = \mu_z + \sigma_z * \rho^{(m)},
\tag{25}
$$

where the learning rate $\rho^{(m)}$ is sampled from $N(0, I)$, $*$ is the element-wise multiplication operator, and μ_z and σ_z are learned by the GCN network formulated by equation (14).

In our variational clustering ensemble framework, the solution of consistent clustering result is to find the posterior distribution $q(C_k^* \mid x, a)$ that maximizes the ELBO. By regrouping the like terms in equation (23), the $L_{\mathrm{ELBO}}(x)$ can be further rewritten as

$$
\begin{aligned}
L_{\mathrm{ELBO}}(x) &= E_{q\left(z, C_k^* \mid x, a\right)}\left[\log\frac{p(a, z, C_k^*)}{q(z, C_k^* \mid x, a)}\right] \\
&= \int_z \sum_{C^*} q(z \mid x, a) q(C_k^* \mid x, a)\left[\log\frac{p(x, a \mid z)p(z)}{q(z \mid x, a)} + \log\frac{p(C_k^* \mid z)}{q(C_k^* \mid x, a)}\right] dz \\
&= \int_z q(z \mid x, a)\log\frac{p(x, a \mid z)p(z)}{q(z \mid x, a)} dz \\
&\quad - \int_z q(z \mid x, a)\mathrm{KL}[p(C_k^* \mid z) \| q(C_k^* \mid x, a)] dz,
\end{aligned}
\tag{26}
$$

where $\mathrm{KL}(\cdot)$ is the Kullback-Leibler divergence function that measures the distance between two distributions and $p(z) = N(z \mid 0, I)$ is a Gaussian prior distribution for latent vector. The first term in equation (26) is independent of C_k^*, and the second term is nonnegative due to the definition of KL divergence. Thus, the $L_{\mathrm{ELBO}}(x)$ achieves the maximum value when $\mathrm{KL}[p(C_k^* \mid z) \| q(C_k^* \mid x, a)] \equiv 0$ is satisfied. Consequently, the optimal distribution $q(C_k^* \mid x, a)$ can be approximated by

$$
q(C_k^* \mid x, a) = p(C_k^* \mid z) = \frac{p(C_k^*)p(z \mid C_k^*)}{\sum_{k'=1}^{K} p(C_{k'}^*)p(z \mid C_{k'}^*)}.
\tag{27}
$$

Since the representation learning and the cluster partitioning are incorporated in an integrated framework, the latent vector z is guaranteed to be an appropriate representation of (x, a) for clustering ensemble, and we use $p(C_k^* \mid z)$ as an approximation to $q(C_k^* \mid x, a)$. Meanwhile, the information loss introduced by the mean field assumption in equation (19) can be mitigated by the relationship between C_k^* and z captured in $p(C_k^* \mid z)$.

To further explore how our optimal model could work on producing a consensus clustering result by incorporating data characteristics and structure information, we rewrite the ELBO in equation (21) as

$$
\begin{aligned}
L_{\mathrm{ELBO}}(x) &= E_{q\left(z, C_k^* \mid x, a\right)}\left[\log\frac{p(a, z, C_k^*)}{q(z, C_k^* \mid x, a)}\right] \\
&= E_{q\left(z, C_k^* \mid x, a\right)}[\log p(a, z, C_k^*) - \log q(z, C_k^* \mid x, a)] \\
&= E_{q\left(z, C_k^* \mid x, a\right)}\left[\log\frac{p(a, z, C_k^*)}{p(z, C_k^*)} + \log p(z, C_k^*) - \log q(z, C_k^* \mid x, a)\right] \\
&= E_{q\left(z, C_k^* \mid x, a\right)}\left[\log p(a \mid z, C_k^*) - \log\frac{q(z, C_k^* \mid x, a)}{p(z, C_k^*)}\right] \\
&= E_{q\left(z, C_k^* \mid x, a\right)}[\log p(a \mid z, C_k^*)] - \mathrm{KL}[q(z, C_k^* \mid x, a) \| p(z, C_k^*)].
\end{aligned}
\tag{28}
$$

It is obvious that the first term in equation (28) is a reconstruction component, which promotes our framework employing latent embedding and clustering ensemble result to explain the relationships among data samples effectively. And the second term is the KL divergence between the variational posterior $q(z, C_k^* \mid x, a)$ and the prior distribution $p(z, C_k^*)$ modeled by a Gaussian mixture distribution. This KL divergence can be considered a regularization term in our optimal objective that guarantees the learned representation z to lie on a Gaussian mixture manifold. As a result, two advantages can be clearly recognized. (i) From an overall perspective, our framework jointly optimizes VGAE and GMM to obtain effective data representation and an appropriate cluster partitioning. (ii) Particularly, in the representation learning section, the data characteristics and the structure information are integrated elegantly in a generative framework.

3.5. Overall Implementation. By integrating the above derivation steps and optimization solution, the implementation of the proposed GCE model is summarized in Algorithm 1.

4. Results and Discussion

In this section, a number of experiments are conducted on several IoT datasets to evaluate the validity and superiority of the proposed model.

```
Input: Data samples $\mathbf{X}$, learning rate $\rho$, number
       of Monte Carlo samples in SGVB
       estimator $M$, epochs $L$.
Output: Consistent clustering result
       $q(C_k^* \mid \boldsymbol{x}_i, \boldsymbol{a}_i)$.
Produce an ensemble of base clusterings for $\mathbf{X}$;
From Equation (11), construct the affinity matrix
    $\mathbf{A}$ to represent structure information for $\mathbf{X}$;
Choose $\sim U(0, 1)$;
for $l = 1, \cdots, L$ do
    for $i = 1, \cdots, N$ do
        $\mu_{z,i} = GCN_\mu(\boldsymbol{x}_i, \boldsymbol{a}_i)$;
        $\log\ \sigma_{z,i} = GCN_\sigma(\boldsymbol{x}_i, \boldsymbol{a}_i)$;
        Sample $C_k^* \sim Cat\ (C_k^* \mid c)$;
        Sample $z_i \sim N(\boldsymbol{z} \mid \mu_{c,k}, \mathrm{diag}\ (\sigma_c^2, k))$;
        Generate reconstructed $\tilde{\boldsymbol{a}}_i = \sigma(\boldsymbol{z}_i^T \boldsymbol{z}_j)$;
        From Equation (24), compute
            $LELBO\ (\boldsymbol{x}_i)$;
        Backpropogate gradients
    end
end
From Equation (27), obtain the category
    assignment $q(C_k^* \mid \boldsymbol{x}_i, \boldsymbol{a}_i)$;
return$q\ (C_k^* \mid \boldsymbol{x}_i, \boldsymbol{a}_i)$
```

Algorithm 1: The implementation of the GCE model.

4.1. Datasets and Evaluation Metrics. Several widely known real datasets are employed for testing, namely, KDD'99, NSL-KDD, AWID, and UCI-IoT. The KDD'99 is a comprehensive network flow dataset, which is usually used as a benchmark in intrusion detection tasks. The NSL-KDD is a dataset suggested to solve some of the inherent problems of the KDD'99, which removes redundant and duplicate records and is more suitable for comparing different intrusion detection methods. AWID is another commonly used network security dataset, which consists of both normal and intrusive network traffic records collected from real 802.11 wireless networks. UCI-IoT is a real traffic dataset collected by lots of commercial devices, whose goal is to recognize 11 different types of traffic situations: one normal operations and 10 malicious attacks. All these datasets are very large and complex; it is difficult to conduct experiments on the whole dataset. Thus, preprocessing is required to construct appropriate datasets for our experiments. For the above four datasets, we randomly draw 20,000 samples from each of the whole dataset. In these preprocessed datasets, symbolic features are mapped to a series of numeric values by one-hot encoding for ease of handling. Besides, we also employ the Network Simulator 2 (NS2) to generate a wireless sensor network (WSN) security dataset, in which 50 sensor nodes are used to simulate attacker records. The sensor nodes transport protocol messages and data messages according to Ad hoc On-Demand Distance Vector (AODV) protocol with a constrained bit rate. These synthetic records contain normal messages and 7 types of security attack cases, which account for 10% scale of the whole dataset. The details of all the datasets used here are summarized in Table 1.

Table 1: Details of datasets: number of data samples (N), number of dimensions (D), and number of clusters (K).

Dataset	N	D	K
KDD'99	20,000	41	5
NSL-KDD	20,000	41	5
AWID	20,000	154	16
UCI-IoT	20,000	115	11
Synthetic dataset	20000	15	8

In the experiments, we use clustering accuracy rate (CAR), adjusted rand index (ARI), and normalized mutual information (NMI), which are widely utilized in clustering task, to evaluate performances of different algorithms. For a set of N data samples X, $\pi = \{C_1, C_2, \cdots C_k\}$ and $P = \{p_1, p_2, \cdots, p_{k'}\}$ are used to denote the clustering result and the true category assignment, respectively. The number of data samples in C_i and p_j are represented as N_i^C and N_j^p. And the number of common samples of C_i and p_j is recored as N_{ij}. Then, the three clustering performance indexes can be defined as follows.

$$\begin{aligned}
\mathrm{CAR} &= \frac{\sum_{i=1}^{k} \max_{j=1}^{k'} N_{ij}}{N}, \\
\mathrm{ARI} &= \frac{\binom{N}{2} \sum_{i=1}^{k} \sum_{j=1}^{k'} \binom{N_{ij}}{2} - \left[\sum_{i=1}^{k} \binom{N_i^C}{2} \sum_{j=1}^{k'} \binom{N_j^p}{2} \right]}{(1/2) \binom{N}{2} \left[\sum_{i=1}^{k} \binom{N_i^C}{2} + \sum_{j=1}^{k'} \binom{N_j^p}{2} \right] - \left[\sum_{i=1}^{k} \binom{N_i^C}{2} \sum_{j=1}^{k'} \binom{N_j^p}{2} \right]}, \\
\mathrm{NMI} &= \frac{\sum_i^k \sum_j^{k'} N_{ij} \log \left(N_{ij} N / N_i^C N_j^p \right)}{\sqrt{\sum_{i=1}^{k} N_i^C \cdot \log \left(N_i^C / N \right) \cdot \sum_j^{k'} N_j^p \cdot \log \left(N_j^p / N \right)}}.
\end{aligned} \tag{29}$$

4.2. Contrastive Algorithms. In the experiments, we compare the proposed model with several representative clustering ensemble methods, which consist of four categories:

(i) Two relabeling-based approaches, including the selective voting (SV) and selective weighted voting (SWV) ensemble algorithms [14]

(ii) Two feature-based approaches, including the expectation maximization (EM) algorithm [15] and the coclustering ensemble (CoCE) approach [16]

(iii) Two pairwise similarity approaches, including the Weighted Connected-Triple (WCT) algorithm proposed by Iam-On et al. [17] and the Hierarchical Flexi Ensemble Clustering (HFEC) model [18]

(iv) Two graph-based approaches [19], including the cluster-based similarity partitioning algorithm (CSPA) and the ultrascalable spectral clustering (U-SPEC) algorithm [20].

From another perspective, the proposed model can be viewed as an improved deep clustering method that

Table 2: CAR metrics of different approaches on IoT datasets.

Datasets	Approaches												
	K-means	SV	SWV	EM	CoCE	WCT	HFEC	CSPA	U-SPEC	DEC	IDEC	VaDE	GCE
KDD'99	0.6784	0.7286	0.7585	0.8083	0.7864	0.8138	0.7966	0.8291	0.8336	0.8314	0.8303	0.8624	**0.9062**
NSL-KDD	0.7228	0.7550	0.7863	0.8465	0.8127	0.8359	0.8260	0.8332	0.8869	0.8103	0.8522	0.8539	**0.9377**
AWID	0.7680	0.8032	0.8266	0.8477	0.8543	0.8307	0.8644	0.8279	0.9093	0.7968	0.8377	0.8541	**0.9362**
UCI-IoT	0.6593	0.7362	0.7633	0.8005	0.7634	0.7441	0.7864	0.7533	0.8351	0.7608	0.8022	0.7879	**0.9073**
Synthetic dataset	0.7253	0.7498	0.7758	0.7531	0.8716	0.8242	0.7826	0.8331	0.8864	0.8174	0.8206	0.8376	**0.9542**

Table 3: ARI metrics of different approaches on IoT datasets.

Datasets	Approaches												
	K-means	SV	SWV	EM	CoCE	WCT	HFEC	CSPA	U-SPEC	DEC	IDEC	VaDE	GCE
KDD'99	0.0473	0.0814	0.1255	0.3762	0.4263	0.5088	0.4207	0.2206	0.4527	0.4018	0.4290	0.4652	**0.5161**
NSL-KDD	0.0988	0.1339	0.1524	0.4036	0.4835	0.5231	0.4275	0.3318	0.3814	0.4364	0.4541	0.4662	**0.5538**
AWID	0.1548	0.1907	0.2583	0.4077	0.3590	0.3302	0.4651	0.3788	0.5007	0.4321	0.4349	0.4792	**0.6145**
UCI-IoT	0.1904	0.2244	0.2419	0.2163	0.2506	0.1338	0.2853	0.2527	0.3645	0.3162	0.3177	0.3203	**0.4264**
Synthetic dataset	0.2279	0.3775	0.4366	0.0577	0.3741	0.4559	0.6013	0.4832	0.6643	0.5506	0.5530	0.5866	**0.7561**

incorporates the structure information extracted from base clusterings. Thus, in our experiments, we also compare it with three deep clustering methods including DEC [26], IDEC [27], and VaDE [31].

4.3. Experimental Setup. To evaluate all algorithms under the same condition, the following experimental settings are adopted in experiments.

(i) For all algorithms and all datasets, the numbers of clusters are set to be the true numbers of categories. And parameters of all the reference algorithms are set according to their authors' suggestions

(ii) The K-means is conducted 200 times independently with random and different initializations to produce 200 base clustering results. And these results are equally divided into 10 subsets, where each subset consists of 20 base results. Then, each ensemble clustering algorithm in the experiments is run on these subsets and produce 10 ensemble results. The average values of these ensemble results are reported as the final outcomes for comparisons

(iii) For deep clustering algorithms, they employ the network architectures adopted in the GCE model, for a fair comparison. All the layers in the encoder-decoder framework are fully connected, and ReLU is selected as the activate function. The network construction of encoder and decoder are mirrored set as $D-500-500-2000-5$ and $5-2000-500-500-D$, respectively, where D is the dimensionality of the input data. To improve the computational efficiency, the Adam optimizer is used, and the mini-batch size is 100. The learning rate is initialized to be 0.02 and decreases every 10 epochs with a decay factor of 0.9. To prevent the models from trapping into local optima or saddle point at the beginning of training, the pretraining method in DEC is adopted in these deep clustering algorithms. In the testing experiments, all the deep clustering algorithms are executed 50 times on each dataset, respectively, and the average results are used for comparison

(iv) For the VaDE and the proposed model, a stacked autoencoder is used to pretrain the encoder and decoder. And in the proposed model, the Gaussian kernel is employed to extract the structure relationships between data samples. The kernel width ϑ is set in the interval $[0.1,2]$ for different datasets, with a step size 0.1. Besides, the parameter θ in the proposed model is set to be 0.5

4.4. Experimental Results

4.4.1. Performance Analysis. Tables 2–4 illustrate the CAR, ARI, and NMI results of GCE and other compared algorithms on all datasets, respectively. The best results of different algorithms are marked in boldface. From these results, some notable points can be found:

(i) Compared with K-means, both clustering ensemble algorithms and deep clustering algorithms achieve better results in aspects of CAR, ARI, and NMI. It can be concluded that the cluster assignments produced by K-means are quit unreliable, as each dataset consists of several linearly inseparable categories. Clustering ensemble algorithms can enhance the clustering performance effectively by integrating multiple weak base results. Deep clustering algorithms learn appropriate representations in latent

Table 4: NMI metrics of different approaches on IoT datasets.

Datasets	Approaches												
	K-means	SV	SWV	EM	CoCE	WCT	HFEC	CSPA	U-SPEC	DEC	IDEC	VaDE	GCE
KDD'99	0.1457	0.1968	0.2073	0.1964	0.3750	0.4273	0.4038	0.1638	0.3647	0.3204	0.3944	0.4164	**0.4742**
NSL-KDD	0.1686	0.2147	0.2566	0.2773	0.3905	0.4765	0.4164	0.2664	0.4367	0.4151	0.4633	0.4846	**0.5402**
AWID	0.2280	0.2661	0.2764	0.5051	0.3348	0.3927	0.5229	0.3562	0.6003	0.4471	0.4533	0.4462	**0.6482**
UCI-IoT	0.1517	0.1843	0.1856	0.2236	0.3146	0.2504	0.3359	0.2742	0.3057	0.3316	0.3513	0.3345	**0.4217**
Synthetic dataset	0.2509	0.3162	0.3865	0.3374	0.4285	0.4196	0.4957	0.4152	0.6960	0.4618	0.5176	0.5504	**0.7160**

Table 5: Std of running results for the proposed algorithm on different datasets.

Indices	Datasets				
	KDD'99	NSL-KDD	AWID	UCI-IoT	Synthetic dataset
CAR (std)	0.0179	0.0155	0.0362	0.0552	0.0328
ARI (std)	0.0208	0.0184	0.0510	0.0758	0.0559
NMI (std)	0.0126	0.0161	0.0563	0.0673	0.0624

space for clustering objective; as a result, they recognize cluster patterns of these datasets more reliably than K-means which divides clusters on raw datasets directly

(ii) Compared to reference clustering ensemble algorithms, which produce final clustering result solely by means of unreliable base clusterings, our GCE model outperforms them considerably on all datasets. Different from these contrastive clustering ensemble algorithms, our model not only explores base clusterings to extract comprehensive structure information but also integrates data characteristics in feature space with extracted structure information to generate effective reorganization for the data. In this way, the unreliable partitioning in base clusterings can be rectified, to a certain extent. That is why our model can easily outperform other clustering ensemble algorithms

(iii) From the experiment results, we can also find that the GCE model achieves better clustering performances in terms of CAR, ARI, and NMI compared with reference state-of-the-art deep clustering algorithms. The results strongly demonstrate the effectiveness and superiority of our joint representation learning strategy that integrates structure information and data characteristics elegantly for prediction of cluster assignments. It is worth noting that, on most datasets, the VaDE achieves the second best results after our model. This is mainly because these two algorithms can recognize cluster distributions more precisely from the random-sampled subsets owing to the capability of generating samples. And VaDE and GCE both utilize the GMM model to be their classifiers, which can approximate arbitrary distribution smoothly. Unlike the VaDE, our model learns joint data representations that provide a positive guidance to depict the formation of clusters not only depending on data characteristics but also incorporating the structure information captured from base clusterings. That is to say, the representations learned in the GCE contain richer information about intrinsic pattern of the data. Therefore, the GCE model outperforms VaDE

To illustrate the impact of randomness on the GCE model, the standard deviation (std) of its running results (CAR, ARI, and NMI) based on 10 base clustering subsets are listed in Table 5. It can be found that the std value is really small on each dataset. Thus, we can draw a conclusion that the GCE model is of strong robustness to randomness.

4.4.2. Parameter Analysis. In this part, we analyze the relationship between the parameter θ and the ensemble result of the GCE model. θ is used to control the size of neighbor set in structure information extraction, which may make some impacts on the construction of the affinity matrix for data reorganization. A smaller θ will bring more samples into the neighbor set of a given sample, and more neighbors will involve more sufficient information giving assistance for the cluster partitioning. However, the other side of the coin is that inconsistent neighbors will also be more likely included, which may introduce misguidance for the cluster partitioning as well as additional computations. In the experiments, we conduct the GCE model 50 times on 5 datasets with different values of θ. And the average results on each dataset are shown in Figure 2. According to these results, some notes can be found as follows. (i) The proposed model achieves the best result 5when the parameter θ takes a certain value, and its performance will degrade as the value of θ gets too large or too small. That is to say, too large or too small neighbor set may lead to a degradation of the

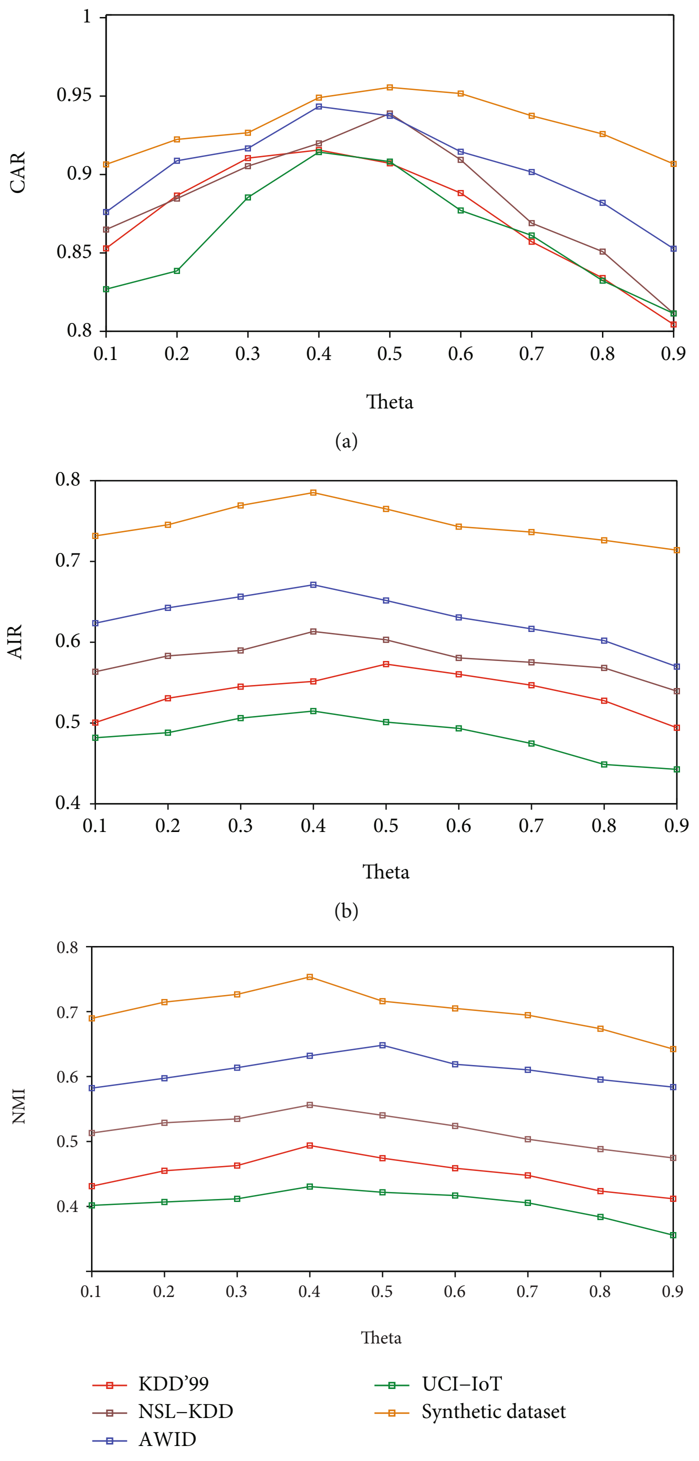

Figure 2: Clustering results of the GCE algorithm with different values of parameter θ: (a) CAR results; (b) ARI results; (c) NMI results.

clustering performance. (ii) No matter what value the parameter θ is, the GCE model reflects more or less advantages over the reference deep clustering algorithms. It demonstrates that introducing structure information into representation learning can and do enhance the ability of capturing cluster distributions.

5. Conclusion

A novel clustering ensemble model is developed in this paper, in order to effectively integrate data characteristics and structure information. Different from conventional clustering ensemble algorithms which generate the final clustering result solely relying on potentially unreliable base clusterings, our model produces the consensus cluster assignment depending on both base clusterings and raw dataset. It first exploits structure information about the data from a set of base clusterings and then transformed the integration of data characteristics and structure information into a graph representation learning problem by reconstructing the data as a sample similarity graph. The learned data representations can not only capture description of the data from both of feature space and structure space but also be suitable for clustering objective. Thus, the final consensus clustering result can be responsibly acquired from it. We conduct experiments by comparing the model with several state-of-the-art clustering ensemble and deep clustering algorithms on several IOT datasets. The experimental results demonstrate the effectiveness and superiority of the proposed model in contrast to the reference algorithms.

In future work, we will extend our model for more complex data types and application tasks. And we also plan to improve it to deal with super-large-scale datasets by optimizing its execution mechanism.

Data Availability

The datasets used in the experiments can be acquired from the following websites: KDD'99—http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html; NSL-KDD—https://www.unb.ca/cic/datasets/nsl.html; AWID—https://icsdweb.aegean.gr/awid/; and UCI-IoT—http://archive.ics.uci.edu/ml/index.php.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

We wish to thank the authors of the compared algorithms for sharing their codes. This work is supported by the National Natural Science Foundation of China (Nos. 61902227, 62076154, 62022052, and U21A20513).

References

[1] R. Lohiya and A. Thakkar, “Application domains, evaluation data sets, and research challenges of IoT: a systematic review,” *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8774–8798, 2021.

[2] V. Vimal, K. U. Singh, A. Kumar et al., “Clustering isolated nodes to enhance network's life time of WSNs for IoT applications,” *IEEE Systems Journal*, vol. 15, no. 4, pp. 5654–5663, 2021.

[3] S. Liu, X. Ma, Y. Jia, and Y. Liu, “An energy-saving task scheduling model via greedy strategy under cloud environment,” *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 8769674, 13 pages, 2022.

[4] X. Guo, H. Lin, Y. Wu, and M. Peng, “A new data clustering strategy for enhancing mutual privacy in healthcare IoT systems,” *Future Generation Computer Systems*, vol. 113, pp. 407–417, 2020.

[5] N. Ravi and S. M. Shalinie, “Semisupervised-learning-based security to detect and mitigate intrusions in IoT network,” *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 11041–11052, 2020.

[6] J. Haseeb, M. Mansoori, Y. Hirose, H. Al-Sahaf, and I. Welch, “Autoencoder-based feature construction for IoT attacks clustering,” *Future Generation Computer Systems*, vol. 127, pp. 487–502, 2022.

[7] S. Zhao, L. Yu, B. Cheng, and J. L. Chen, “IoT service clustering for dynamic service matchmaking,” *Sensors*, vol. 17, no. 8, p. 1727, 2017.

[8] A. K. Tripathi, K. Sharma, M. Bala, A. Kumar, V. G. Menon, and A. K. Bashir, “A parallel military-dog-based algorithm for clustering big data in cognitive industrial Internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2134–2142, 2021.

[9] J. Shuja, M. A. Humayun, W. Alasmary, H. Sinky, E. Alanazi, and M. K. Khan, “Resource efficient geo-textual hierarchical clustering framework for social IoT applications,” *IEEE Sensors Journal*, vol. 21, no. 22, pp. 25114–25122, 2021.

[10] C. Tang, Z. Li, J. Wang, X. Liu, W. Zhang, and E. Zhu, “Unified one-step multi-view spectral clustering,” *IEEE Transactions on Knowledge and Data Engineering*, p. 1, 2022.

[11] A. E. Ezugwu, A. M. Ikotun, O. O. Oyelade et al., “A comprehensive survey of clustering algorithms: state-of-the-art machine learning applications, taxonomy, challenges, and future research prospects,” *Engineering Applications of Artificial Intelligence*, vol. 110, article 104743, 2022.

[12] N. C. Sandes and A. L. Coelho, “Clustering ensembles: a hedonic game theoretical approach,” *Pattern Recognition*, vol. 81, pp. 95–111, 2018.

[13] K. Golalipour, E. Akbari, S. S. Hamidi, M. Lee, and R. Enayatifar, “From clustering to clustering ensemble selection: a review,” *Engineering Applications of Artificial Intelligence*, vol. 104, article 104388, 2021.

[14] Z. Zhou and W. Tang, “Clusterer ensemble,” *Knowledge-Based Systems*, vol. 19, no. 1, pp. 77–83, 2006.

[15] A. Topchy, A. Jain, and W. Punch, “Clustering ensembles: models of consensus and weak partitions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1866–1881, 2005.

[16] X. Yu, G. Yu, J. Wang, and C. Domeniconi, “Co-clustering ensembles based on multiple relevance measures,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1389–1400, 2021.

[17] N. Iam-On, T. Boongoen, S. Garrett, and C. Price, “A link-based approach to the cluster ensemble problem,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2396–2409, 2011.

[18] P. Priyanga and A. R. N. N. B. Kamal, “Mobile app usage pattern prediction using hierarchical flexi-ensemble clustering (HFEC) for mobile service rating,” *Wireless Personal Communications*, vol. 122, no. 4, pp. 3247–3268, 2022.

[19] A. Strehl and J. Ghosh, “Cluster ensembles: a knowledge reuse framework for combining multiple partitions,” *Journal of Machine Learning Research*, vol. 3, no. 3, pp. 583–617, 2003.

[20] D. Huang, C. Wang, J. Wu, J. Lai, and C. K. Kwoh, “Ultra-scalable spectral clustering and ensemble clustering,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1212–1226, 2020.

[21] T. Chakraborty, F. Pierazzi, and V. S. Subrahmanian, "EC2: ensemble clustering and classification for predicting android malware families," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 262–277, 2020.

[22] T. N. Kipf and M. Welling, "Variational graph auto-encoders," in *Proceedings of the 30th Conference on Neural Information Processing Systems*, Barcelona, Spain, 2016.

[23] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proceedings of the 2nd International Conference on Learning Representations*, Ithaca, NY, 2014.

[24] E. M. Bårli, A. Yazidi, E. H. Viedma, and H. Haugerud, "DoS and DDoS mitigation using variational autoencoders," *Computer Networks*, vol. 199, no. 9, article 108399, 2021.

[25] X. Peng, S. Xiao, J. Feng, W. Yau, and Z. Yi, "Deep subspace clustering with sparsity prior," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, New York New York USA, 2016.

[26] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proceedings of the 33rd International Conference on Machine Learning*, New York, USA, 2016.

[27] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 2017.

[28] X. Guo, X. Liu, E. Zhu, and J. Yin, "Deep clustering with convolutional autoencoders," in *Proceedings of the 25th International Conference on Neural Information Processing*, Guangzhou, China, 2017.

[29] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards K-means-friendly spaces: simultaneous deep learning and clustering," in *Proceedings of the 33rd International Conference on Machine Learning*, Sydney NSW Australia, 2017.

[30] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, and P. Cui, "Structural deep clustering network," in *Proceedings of the 29th Web Conference*, Taipei Taiwan, 2020.

[31] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: an unsupervised and generative approach to clustering," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 2017.

[32] J. J. Choong, X. Liu, and T. Murata, "Learning community structure with variational autoencoder," in *Proceedings of the IEEE International Conference on Data Mining*, Singapore, 2018.

[33] J. Hwang and H. Kim, "Variational deep clustering of wafer map patterns," *IEEE Transactions on Semiconductor Manufacturing*, vol. 33, no. 3, pp. 466–475, 2020.

[34] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: methods and applications," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, vol. 40, no. 3, pp. 52–74, 2018.

[35] S. Cao, W. Lu, and Q. Xu, "Grarep: learning graph representations with global structural information," in *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Melbourne Australia, 2015.

[36] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," in *Proceedings of the 23rd International Conference on World Wide Web*, New York New York USA, 2014.

[37] I. Chaturvedi, K. Thapa, S. Cavallari, E. Cambria, and R. E. Welsch, "Predicting video engagement using heterogeneous DeepWalk," *Neurocomputing*, vol. 465, pp. 228–237, 2021.

[38] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, Phoenix, USA, 2016.

[39] S. Fan, X. Wang, C. Shi, K. Kuang, N. Liu, and B. Wang, "Debiased graph neural networks with agnostic label selection bias," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2022.

[40] D. Jin, Z. Liu, W. Li, D. He, and W. Zhang, "Graph convolutional networks meet Markov random fields: semi-supervised community detection in attribute networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 152–159, 2019.

[41] J. Wang, J. Liang, K. Yao, J. Liang, and D. Wang, "Graph convolutional autoencoders with co-learning of graph structure and node attributes," *Pattern Recognition*, vol. 121, article 108215, 2022.

[42] H. Du and W. Wang, "A clustering ensemble framework with integration of data characteristics and structure information: a graph neural networks approach," *Mathematics*, vol. 10, no. 11, p. 1834, 2022.