A Deep Unsupervised Learning Approach for Airspace Complexity Evaluation

Biyue Li, Wenbo Du, *Member, IEEE,* Yu Zhang, Jun Chen,
Ke Tang, *Senior Member, IEEE*, Xianbin Cao, *Senior Member, IEEE*

***Abstract*—Airspace complexity is a critical metric in current Air Traffic Management systems for indicating the security degree of airspace operations. Airspace complexity can be affected by many coupling factors in a complicated and nonlinear way, making it extremely difficult to be evaluated. In recent years, machine learning has been proved as a promising approach and achieved significant results in evaluating airspace complexity. However, existing machine learning based approaches require a large number of airspace operational data labeled by experts. Due to the high cost in labeling the operational data and the dynamical nature of the airspace operating environment, such data are often limited and may not be suitable for the changing airspace situation. In light of these, we propose a novel unsupervised learning approach for airspace complexity evaluation based on a deep neural network trained by unlabeled samples. We introduce a new loss function to better address the characteristics pertaining to airspace complexity data, including dimension coupling, category imbalance, and overlapped boundaries. Due to these characteristics, the generalization ability of existing unsupervised models is adversely impacted. The proposed approach is validated through extensive experiments based on the real-world data of six sectors in Southwestern China airspace. Experimental results show that our deep unsupervised model outperforms the state-of-the-art methods in terms of airspace complexity evaluation accuracy.**

***Index Terms*—Airspace complexity, data characteristics, deep learning, unsupervised learning.**

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFF0301400; in part by the National Natural Science Foundation of China (NSFC) under Grant 61961146005 and Grant 62088101; and in part by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/N029496/2. *(Corresponding authors: Wenbo Du.)*

B. Li, W. Du and X. Cao are with the the National Engineering Laboratory for Big Data Application Technologies for Comprehensive Traffic, School of Electronic and Information Engineering, Beihang University, Beijing 100191, China.(e-mail:libiyue@buaa.edu.cn;wenbodu@buaa.edu.cn;xbcao@buaa.edu.cn).

Y. Zhang is with the Department of Civil and Environmental Engineering at the University of South Florida 4202 E. Fowler Ave. ENB118 Tampa, FL 33620, USA. (e-mail: yuzhang@usf.edu)

J. Chen is with the School of Engineering and Materials Science, Queen Mary University of London, Mile End Road London E1 4NS, U.K. (e-mail: jun.chen@qmul.ac.uk).

Ke Tang is with the School of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, Guangdong 518055, China (e-mail: tangk3@sustech.edu.cn).

I. INTRODUCTION

WITH increasing globalization, the world's civil aviation industry has been advancing at a fast pace. The number of flights in China has been increased from 7.93 million in 2014 to 11.09 million in 2018. The average growth rate has reached 8.7% in the past five years [1]. This massive air traffic flow has brought high control pressure to air traffic controllers (ATCos). ATCos are in charge of airspace sectors which are the basic control unit of airspace [2]. The control pressure causes fatigue among ATCos, resulting in higher airspace operation risks [3]. A notable example was a returning flight at Wuhan airport in 2014 [4]. Two on-duty ATCos fell asleep due to high workload while the aircraft was approaching. The aircraft had to circle in absence of communication with the control tower.

Accurate evaluation of airspace complexity plays a vital role in adjusting the sector control pressure. When airspace complexity exceeds the control capability of an ATCo, controllers' operational errors are likely to increase [5]. Airspace complexity is affected by a combination of subjective, objective, dynamic and static factors, such as air routes and sector entering and exiting points, etc. Therefore, evaluation of airspace complexity is a non-trivial problem, attracting many investigations from scholars and field practitioners [6]-[16], [22]-[40]. Currently, the state-of-the-art methods for airspace complexity evaluation fall into two main categories: (1) Airspace complexity is described through a single indicator [6]-[14]; (2) Airspace complexity is evaluated based on a multi-indicator system [15], [16], [22]-[31].

Some researchers adopt a single indicator in an ad hoc way to represent the complexity of airspace. Lee et al. proposed an input-output approach by defining complexity as "how difficult" it is to resolve the potential conflicts in a circular airspace [6]-[8]. The approach proposed by Prandini et al. characterizes the mid-term traffic complexity at a certain point in airspace based on conflict risk estimation [9], [10]. Delahaye et al. proposed a complexity indicator which is defined by Lyapunov exponent based on traffic trajectories [11], [12]. While the complexity based on a single indicator can be directly calculated and is easy to use, it is usually not sufficient for comprehensively characterizing airspace complexity.

Other approaches aim to comprehensively evaluate airspace complexity via multiple-complexity factors. Many researchers have investigated the mapping relationship between these complexity factors and airspace complexity. In 1998, NASA Ames Research Center defined the dynamic density as a linear combination of 9 complexity factors [15]; this has become a standard and been implemented in the real system to evaluate airspace complexity until now [16]. The recent advent of machine learning technologies [17]-[21] encourages scholars to explore the nonlinear mapping between varying complexity factors and airspace complexity [22]-[31]. A pioneering work by Chatterji used the artificial neural network (ANN) to establish the nonlinear correlation for airspace complexity evaluation [22]. Gianazza trained a Back Propagation Neural Network (BPNN) based on samples from French sectors [23]. After dimension reduction with principal component analysis, the approach led to a great evaluation result [24]. Andraši et al. developed configuration-optimized ANNs to determine air traffic complexity [25]. Their results show the accuracy of the proposed model is comparable to the linear methods. Along this line, Xiao et al. employed a genetic algorithm to select critical factors in order to build an adaptive boosting model to evaluate airspace complexity [26]. However, the factor selection process through the genetic algorithm is extremely time consuming. Although the parallel genetic algorithm framework in [27] may speed up the selection process, this factor selection approach is still not able to evaluate airspace complexity in real-time. Furthermore, all of the above machine learning methods rely on a large number of labeled airspace samples in order to improve the accuracy of evaluation. Aiming at reducing the high cost in obtaining labeled data, Zhu et al. conducted a series of studies based on small samples [28]-[30] using integrated learning, semi-supervised learning, and transfer learning respectively, and obtained good evaluation results on six airspace sectors in Southwestern China. Similarly, the air traffic controller's tasks are incorporated into training data in order to mitigate the issue of small samples [31]. However, extraction of controller's tasks from traffic situation data is a non-trivial task.

In practice, the airspace operation environment changes dynamically every several months (e.g., airspace structure, operation rules, etc.) [30]. However, the obtained labeled data for evaluating airspace complexity is in general not sufficient to capture such changes. Therefore, the evaluation models obtained through supervised learning must be retrained using newly labeled samples, which entail a high cost for labeling. In light of the above, it is necessary to develop an unsupervised model that does not depend on labeled airspace samples. However, due to the characteristics of practical airspace operational data, such as coupled dimensions, imbalanced categories, and overlapped boundaries, the existing unsupervised models do not generalize well on the problem of airspace complexity evaluation.

In order to tackle the above challenges, we propose a novel ***D****eep* ***U****nsupervised learning approach for* ***A****irspace* ***C****omplexity* ***E****valuation* (DUACE). The main contributions of this paper are summarized as follows: (1) We develop a data-oriented deep unsupervised learning model to solve the airspace complexity evaluation problem which can further reduce the cost due to labeling. To the best of our knowledge, it is the first time that a deep unsupervised learning approach has been used to evaluate airspace complexity. (2) We introduce a new loss function, consisting of reconstruction loss, Kullback-Leibler divergence loss and probabilistic cluster loss, to better describe the characteristics pertaining to practical airspace complexity data. (3) We find that one of the hyperparameters in our proposed model is closely related to geographical regions, indicating a potential way to further remove this parameter in future. (4) Extensive experiments are carried out using real-world airspace complexity datasets for Southwestern China region. Experimental results indicate that our model achieves the best evaluation performances compared to those of the existing baselines.

The remainder of this paper is organized as follows. Section II provides an analysis of the problem, an overview of the existing unsupervised learning methods, and the proposed DUACE. Section III presents an experimental investigation based on the dataset of six airspace sectors in Southwestern China. In Section IV, we conclude this study and provide some further discussion.

II. Methodology

In this section, we first introduce the problem of airspace complexity evaluation. Furthermore, we review and analyze the characteristics and limitations of the existing unsupervised models in solving similar problems. Inspired by these methods, we propose a deep unsupervised learning approach.

A. Problem Description and Review of Existing Methods

The complexity of airspace is determined by synthesizing numerous complexity factors. The most widely used complexity factors, including 28 dimensions (factors), are put forward by Gianazza and Guittet [23]. A detailed explanation of the proposed factors is included in TABLE I, in which similar airspace complexity factors are grouped together. Furthermore, airspace complexity can be divided into several levels, including High, Normal and Low. Correspondingly, the original data can be classified into three clusters based on the complexity level. Therefore, airspace complexity evaluation can be formulated as a clustering problem. The samples belonging to the same centroid form a cluster whose corresponding complexity level can be determined by referring to the levels of the centroids or several samples in the same cluster.

Distance-based clustering methods [41], such as K-means, Gaussian Mixture Models, Spectral Clustering, and Density-Based Spatial Clustering as well as some recent improvements [42], [43] determine the centroids by minimizing the distance of the samples and centroids. However, the above methods may be ineffective when the input data are of high dimensionality and complex coupling, also known as the curse of dimensionality. To address this problem, it is essential to implement dimension reduction to map the original samples into a lower dimensional space which is more suitable for clustering. Some dimension reduction techniques, e.g. Principal Component Analysis (PCA) [44] and manifold learning [45], have been adopted to extract latent representations of the original data. Note that, the dimension reduction methods mentioned above are based on the hypothesis that the original

TABLE I
COMPLEXITY FACTORS (SIMILAR FACTORS ARE GROUPED TOGETHER.)

Factor Number	Annotation
1 ~ 4	Total number of aircraft, square of total number of aircraft, number of descending, number of climbing aircraft.
5 ~ 8	Future incoming flow in horizons of 5 min, 15 min, 30 min, 60 min.
9 ~ 12	Density of aircraft, horizontal proximity between aircraft, two kinds of vertical proximity between aircraft.
13 ~ 15	Variance of aircraft ground speeds, ratio of standard deviation of aircraft ground speeds to aircraft average ground speed, average of absolute values of aircraft vertical speeds.
16, 17	Number of potential crossings of aircraft trajectories, measures the mixing degree of aircraft at different flight states (descending /level /climbing).
18, 19	Variability in aircraft headings, variability in aircraft speeds[23].
20, 21	Rate of divergences between aircraft pairs (*Div*), rate of convergences between aircraft pairs (*Conv*).
22 ~ 25	Sensitivity of distance change between diverging/converging aircraft with speed ($sensi\_d$, $sensi\_c$) and heading modifications applied to them ($\frac{Div^2}{sensi\_d}$, $\frac{Conv^2}{sensi\_c}$).
26, 27	Conflict perception of "good pairs", conflict perception of "bad pairs".
28	Geometric volume of a sector.

data have a linear or manifold structure. In reality, factors in evaluating airspace complexity are interacting nonlinearly. Therefore, the structure of data should not be oversimplified. A non-linear dimension reduction method that can address the complex data structure is needed.

As a mature framework in deep learning, the Autoencoder based algorithms use deep neural network (DNN) architectures to extract inherent features in complex data, leading to effective nonlinear dimension reduction [46]-[48]. The Autoencoder is composed of the encoder and decoder formulated as two neural networks. The Autoencoder is trained based on a reconstruction loss function. The encoder network transforms the input data into the latent representations that are of lower dimension, while the decoder network reconstructs the output from such representations. The data-driven characteristics and flexibility of the Autoencoder make it applicable to many complex tasks. Many deep clustering methods employing the Autoencoder have shown impressive results in clustering [49]-[51]. Such methods generate suitable-for-clustering representations from the original data by adding the designed cluster loss on the basis of the reconstruction loss during the network training process. In [49], Deep Clustering Network (DCN) first jointly learns clustering-friendly representations, clustering centroids, and cluster assignments from Autoencoder with K-means clustering loss. The robustness of the model can be further improved through ensemble Autoencoder learning [52]. However, the learned representations may not be well separable. Therefore, transformed subspace clustering is proposed to relieve the issue [53]. Although these methods can provide outstanding results in discriminating images and texts, they cannot be directly applied to evaluate airspace complexity data for the following reasons.

1) The proportion of the airspace operation period with low/normal/high complexity is naturally imbalanced, resulting in category imbalance in the airspace complexity dataset. Due to this imbalance, the learning model performs poorly in identifying minority categories.
2) For images or texts, a deterministic assignment for clustering is effective because data belonging to different classes are significantly apart from each other spatially after dimension reduction. However, for airspace complexity datasets, data belonging to different complexity levels are still overlapped near the category boundaries even after dimension reduction. This is due to the high-dimensional coupling characteristics of complexity factors. This so-called overlapped boundary phenomenon increases the difficulty for learning models in discriminating the data located near the category boundary. The above characteristics are not unique to the airspace complexity data, representing a great challenge for conventional clustering algorithms. It is worth pointing out that the proposed algorithm can be used in handling other clustering problems having similar data characteristics.

B. Proposed Deep Unsupervised Model

1) Overview of the Model

The network structure of our proposed deep unsupervised model is shown in Fig. 1. The original airspace complexity data will be input into the Synthetic Minority Over-Sampling Technique (SMOTE) algorithm for interpolation in order to generate a new dataset [54]. The newly generated dataset will be the input into the Autoencoder. The encoder network outputs the latent features of the airspace complexity data after dimensionality reduction, which are the input to the decoder network to obtain the reconstructed data.

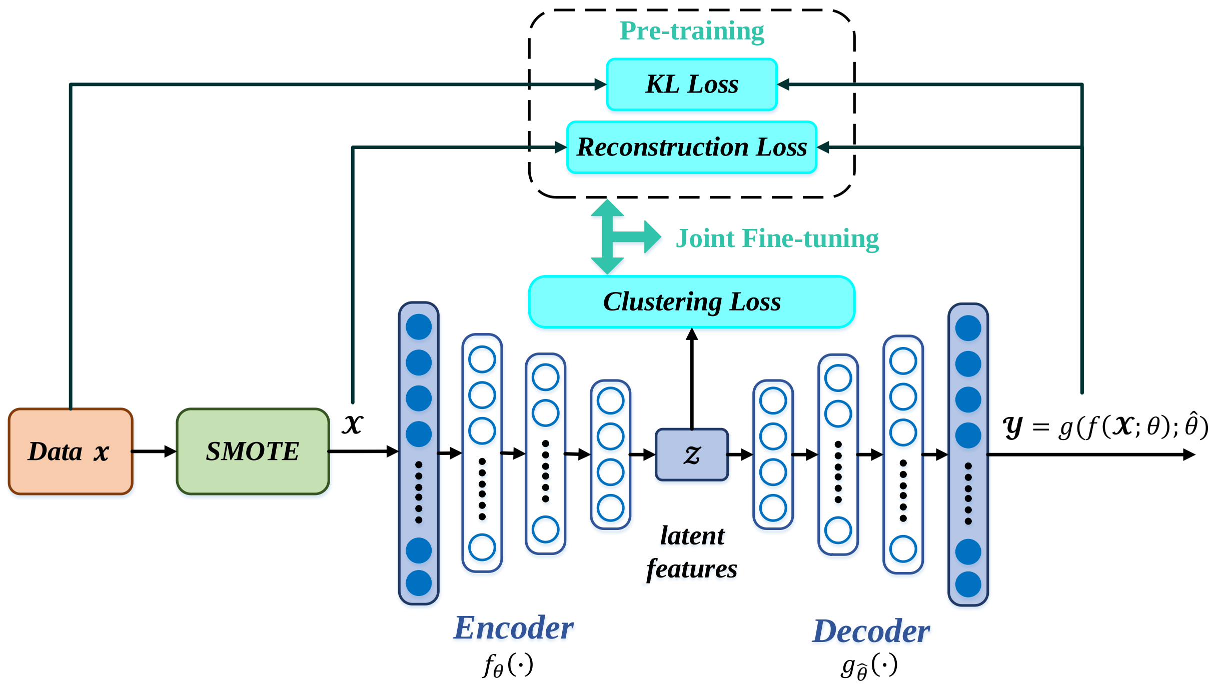

Fig. 1. Network structure of the proposed deep unsupervised model.

As depicted in Fig. 1, the DUACE model is developed based on the loss function of the Autoencoder framework which is deliberately designed based on the characteristics of airspace complexity data.

The Kullback-Leibler Loss (KL Loss) and SMOTE are designed to address the impact of category imbalance. We first follow SMOTE [54] to generate more samples through linear interpolation in the original datasets. The Kullback-Leibler divergence (KL divergence) is calculated between the original datasets and the reconstructed data after the Autoencoder so as to align their distributions to be similar. The difference between the data after SMOTE and the reconstructed data after the Autoencoder is measured by Reconstruction Loss in order to ensure the latent representations are meaningful [46]. Details of KL divergence and SMOTE are described in Section B-2.

As for overlapped boundaries, the Clustering Loss with probabilistic assignment is proposed. Different from a deterministic centroid in K-Means, we assign the samples to a probabilistic centroid which is calculated by a weighted average of all centroids. The Clustering Loss aims to update neural network parameters and centroids by minimizing the distance between the samples and the probabilistic centroids. This process will produce a gradient component for those misclassified samples near the category boundaries to further update network parameters and centroids in the right direction. Details of probabilistic assignment are described in Section B-3.

The optimization of the proposed deep unsupervised model contains two procedures. During the pre-training process, the Reconstruction Loss and KL Loss are firstly computed to initialize the neural network parameters. After a number of predefined epochs, the Clustering Loss is included in the total loss function to further tune network parameters. This tuning process is termed as the joint fine-tuning process. Details of the optimization procedure are discussed in Section C.

2) SMOTE by KL Divergence

SMOTE by KL Divergence is designed to address the category imbalance problem of airspace complexity data. SMOTE generates new data, while KL Divergence guarantees the quality of the interpolated data by calculating the similarity between the original data distribution and the reconstructed data distribution after the autoencoder. Stochastic Neighbor Embedding (SNE) is used to calculate the data distribution. Due to the dimensions mismatch between the original data and the reconstructed data, PCA is used to perform dimension reduction of the original data and reconstructed data so that the KL divergence can be calculated between the original data and reconstructed data.

The inherent category imbalance of airspace data has a negative impact on learning the classification strategy that assigns each sample to its corresponding cluster and complexity level. One solution is to take advantage of data enhancement techniques such as over-sampling and augmentation to generate new data in the minority class [54]. The former is not suitable because over-sampling will duplicate many existing original data. This may lead to the over-fitting problem. Data augmentation generates new data by randomly adding white noise to the original data, which may aggravate the boundary overlapping problem of airspace complexity data. SMOTE constantly generates new data through interpolation between two data points belonging to the same cluster. Therefore, SMOTE has a relatively higher probability of generating new data belonging to the same cluster and is more suitable for the proposed approach.

SMOTE is exploited to generate more data and boost the less represented category. In order to ensure that new data is generated as uniformly as possible, we firstly perform K-means clustering on the original datasets. After that, the generation process by SOMTE can be defined as: $x_g = x_i + \delta(x_j - x_i)$ with $0 < \delta \leq 1$, where x_g represents the newly generated data point, $x_i, x_j \in \boldsymbol{x}, \boldsymbol{x} \in \mathbb{R}^{N\times D}$ represents the original data and x_j is another sample in the same cluster of x_i. The number of samples in the original data is N, and the dimension of each sample is D. The procedure is repeated for several times to get a new dataset: $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{\hat{N}\times D}$, where $x_i, x_j, x_g \in \boldsymbol{\mathcal{X}}$ and the ratio of the interpolation is set to 30%. The number of samples in the new dataset is $\hat{N}$.

The reconstructed data after the Autoencoder (AE) is represented as $\boldsymbol{\mathcal{Y}} = g(f(\boldsymbol{\mathcal{X}};\theta);\hat{\theta})$, $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{\hat{N}\times D}$. $f_\theta(\cdot)$ denotes the encoder of the Autoencoder while $g_{\hat{\theta}}(\cdot)$ denotes the decoder. θ is the parameter of the encoder network and $\hat{\theta}$ is axisymmetric to θ [46]. The Reconstruction Loss (L_{AE}) in the Autoencoder measures similarity between the data after SMOTE and the reconstructed data after the Autoencoder, ensuring the latent representations are meaningful.

$$L_{AE} = min_{\theta,\hat{\theta}} \left\| \boldsymbol{\mathcal{X}} - g(f(\boldsymbol{\mathcal{X}};\theta);\hat{\theta}) \right\|_2^2 \quad (1)$$

Since the airspace complexity data obeys a more complex distribution, the data distribution after SMOTE may be different from the original datasets. In order to impose restrictions on the newly created data, the Kullback-Leibler (KL) divergence is used to calculate the similarity between the original and reconstructed data distributions. SNE [55] is utilized to compute the data distribution, which converts the distance relationship into a two-dimensional probability matrix:

$$p_{ij} = \frac{\exp(-\|x_i - x_j\|_2^2/2\sigma_i^2)}{\sum_{k\neq i} \exp(-\|x_i - x_k\|_2^2/2\sigma_i^2)} \quad (2)$$

$$q_{ij} = \frac{\exp(-\|y_i - y_j\|_2^2)}{\sum_{k\neq i} \exp(-\|y_i - y_k\|_2^2)} \quad (3)$$

where $x_i, x_j, x_k \in \boldsymbol{x}, \boldsymbol{x} \in \mathbb{R}^{N\times D}$ and $y_i, y_j, y_k \in \boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{Y}} \in \mathbb{R}^{\hat{N}\times D}$. σ_i is the variance of the Gaussian that is centered on the features of data point x_i, x_j. P and Q are matrices composed of elements p_{ij} and q_{ij}. $P = (p_{ij})_{N\times N}$ and $Q = (q_{ij})_{\hat{N}\times\hat{N}}$, represent the distribution of original data $\boldsymbol{x}$ and the reconstructed data $\boldsymbol{\mathcal{Y}}$, respectively. The KL divergence between P and Q is minimized to ensure that those newly generated data are sampled from the distribution of the original data instead of randomly by linear interpolation. However, as dimensions of P and Q are different, a transformation of P and Q should be carried out before calculating the KL divergence.

PCA is used to perform dimension reduction on P and Q to ensure the KL divergence can be calculated between the original and reconstructed data. The column vectors in P and Q are respectively represented by $p_j \in \mathbb{R}^{N\times 1}$ and $q_j \in \mathbb{R}^{\hat{N}\times 1}$. PCA is applied to p_j and q_j to reduce their dimensions by multiplying the transpose of projection matrices W_p and W_q:

$$\hat{p}_j = {W_p}^T p_j \quad (4)$$

$$\hat{q}_j = {W_q}^T q_j \quad (5)$$

where $\hat{p}_j \in \mathbb{R}^{m\times 1}$, $\hat{q}_j \in \mathbb{R}^{m\times 1}$, ${W_p}^T \in \mathbb{R}^{m\times N}$, ${W_q}^T \in \mathbb{R}^{m\times\hat{N}}$, and $m < N < \hat{N}$. W_p and W_q are computed as below:

$$PP^T W_p = \lambda_p W_p \quad (6)$$

$$QQ^T W_q = \lambda_q W_q \quad (7)$$

where λ_p and λ_q are composed of the highest m eigenvalues of the diagonal matrices PP^T and QQ^T, respectively. The resulting matrices $\hat{P}_j = (\hat{p}_1, \hat{p}_2, \dots, \hat{p}_j, \dots \hat{p}_N) \in \mathbb{R}^{m\times N}$ and $\hat{Q}_j = (\hat{q}_1, \hat{q}_2, \dots, \hat{q}_j, \dots \hat{q}_{\hat{N}}) \in \mathbb{R}^{m\times\hat{N}}$ are of the same row dimensionality by implementing PCA along the columns of P and Q. In the same way, PCA is applied along the rows of $\hat{P}_j$ and $\hat{Q}_j$. The final resulting matrices $U = (u_{ij})_{m\times m}$ and $V = (v_{ij})_{m\times m}$ are of the same dimensionality. The matrices U and V are the results of P and Q after PCA dimension reduction of the row and column vectors.

Finally, we calculate the KL divergence between U_{ij} and V_{ij}:

$$L_{KL} = min_{\theta,\hat{\theta}} KL(U \parallel V) = min_{\theta,\hat{\theta}} \sum_i \sum_j u_{ij} log \frac{u_{ij}}{v_{ij}} \quad (8)$$

Combining L_{AE} and L_{KL}, not only does it increase the number of samples in the less represented categories, but also ensures that the newly generated data after SMOTE follows the distributions of the original datasets.

$$L_1 = L_{AE} + \gamma L_{KL}$$
$$= min_{\theta,\hat{\theta}} \left(\left\| \boldsymbol{\mathcal{X}} - g(f(\boldsymbol{\mathcal{X}};\theta);\hat{\theta}) \right\|_2^2 + \gamma \sum_i \sum_j u_{ij} log \frac{u_{ij}}{v_{ij}} \right) \quad (9)$$

γ is a constant between [0, 1] to balance the impact of L_{AE} and L_{KL}. L_1 is differentiable with respect to $(\theta, \hat{\theta})$ so that a gradient descent method can be implemented to minimize L_1.

3) *Probabilistic Assignment*

A more suitable latent representative space for clustering is obtained by applying an additional cluster loss in the loss function: $L_{clu} = \sum_{i=1}^{\hat{N}} \|z_i - SR^T\|_2^2$, where R denotes the centroids and S is the assignment vector. $z_i = f(x_i;\theta) \in \boldsymbol{\mathcal{Z}}$ is the latent representation mapped from $x_i \in \boldsymbol{\mathcal{X}}$ through the dimensionality reduction of AE, where $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{\hat{N}\times D'}$, $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{\hat{N}\times D}$, and D' is the reduced dimension. When the assignment is deterministic, S is a one-hot vector and its ith element is 1, indicating the sample belongs to the ith cluster.

Airspace complexity data may still overlap near the category boundaries even after dimensionality reduction. Therefore, at the beginning of neural network training, it is difficult to find the true centroid for those samples around boundaries through the deterministic assignment. Instead of deterministically assigning a sample to a specific centroid, the impact of all centroids on a sample should be considered. The weight w_{ij} is hence introduced and is dependent on the Euclidean distance between latent representation z_i of sample x_i and centroid r_j. w_{ij} represents the effect of different centroids on the ith sample.

$$w_{ij} = \left\| z_i - r_j \right\|_2, \ \text{s.t.}\, i \in \hat{N}, j \in k \quad (10)$$

We use the probability to represent the pulling of a sample by different centroids. The SoftMax function can transform the one-hot matrix S into a probability matrix G. It has also been proven to be effective for multi-categorical samples. [56].

$$G_{ij}(z_i,\alpha,r_j)=\frac{\exp(-\alpha w_{ij})}{\sum_k \exp(-\alpha w_{ik})} \quad (11)$$

The element of probability matrix G_{ij}, represents the probability that z_i is assigned to the centroid r_j. As w_{ij} increases, G_{ij} becomes smaller. α is a user defined hyperparameter, $\alpha \in \mathbb{Z}_0^+$. The larger α is, the more sensitive G_{ij} is with respect to w_{ij}. The setting of α will gradually increase with the epochs of deep neural network training until G approaches a one-hot matrix [56]. Therefore, during the early stage of deep neural network training, samples around cluster boundaries have chances to be corrected in the right direction.

In summary, the Clustering Loss in this paper is defined as:

$$L_{clu}=min_\theta \sum_{i=1}^{\hat{N}} \ell(f(x_i;\theta),G_iR^T) \quad (12)$$
$$\text{s.t.}\, R=(r_1,r_2,\dots,r_k)\in\mathbb{R}^{D'\times K},$$
$$G_i=(G_{i1},G_{i2},\dots,G_{ik})\in\mathbb{R}^{1\times K}.$$

where $\ell(\cdot)=K(x,x')=\exp\left(\frac{-\|x-x\prime\|_2^2}{2\sigma^2}\right)$ is the Gaussian Kernel function. $\ell(\cdot)$ is more suitable for handling high-dimensional data [57]. The probability matrix $G=(G_1,G_2,\dots,G_i,\dots G_{\hat{N}})^T\in\mathbb{R}^{\hat{N}\times K}$.

We formulate the loss function of the whole model as follows.

$$L=L_1+L_{clu}=L_{AE}+\gamma L_{KL}+L_{clu} \quad (13)$$
$$L=min_{\theta,\hat{\theta}}\ \left(\left\|\boldsymbol{\mathcal{X}}-g\left(f(\boldsymbol{\mathcal{X}};\theta);\hat{\theta}\right)\right\|_2^2 \quad (14)$$
$$+\gamma\sum_i\sum_j u_{ij}log\frac{u_{ij}}{v_{ij}}+\sum_{i=1}^{\hat{N}}\ell(f(x_i;\theta),G_iR^T)\right)$$

C. Optimization Procedure

Optimizing L contains two procedures. During the pre-training process, we initialize the parameters of the Autoencoder and the centroids of latent representations. After a number of predefined epochs, the update of network parameters and centroids will be alternately performed during the fine-tuning process.

1) Parameter Initialization

As the Autoencoder possesses a large number of parameters, a random initialization for θ will lead the network to be trapped in local optima. We use the layer-wise pre-training method as in [46] for training the Autoencoder, which means that we use the output of each layer to train the next layer. $L_1(\theta,\hat{\theta})$ is fully differentiable with respect to $(\theta,\hat{\theta})$, so that $L_1(\theta,\hat{\theta})$ can be optimized by the following update formula:

$$(\theta,\hat{\theta})\leftarrow(\theta,\hat{\theta})-\eta\nabla_{(\theta,\hat{\theta})}L_1(\theta,\hat{\theta}) \quad (15)$$

where η denotes the learning rate. After layer-wise pre-training, the result is a multilayer deep Autoencoder with a bottleneck coding layer in the middle. To initialize the centroids of latent representations $\boldsymbol{\mathcal{Z}}\in\mathbb{R}^{\hat{N}\times D'}$, we perform K-means to the outputs of the bottleneck layer to obtain initial values of R and G.

2) Updating Network Parameters

For fixed G and R, the Clustering Loss is included in the total loss function to jointly fine-tune the network parameters. L is differentiable with respect to all parameters.

$$\nabla_{(\theta,\hat{\theta})}L(\theta,\hat{\theta})=\nabla_{(\theta,\hat{\theta})}L_1(\theta,\hat{\theta})+\nabla_{(\theta)}L_{clu}(\theta) \quad (16)$$

$\nabla_{(\theta,\hat{\theta})}L(\theta,\hat{\theta})$ will be used to update the parameters in the network through back-propagation [58].

$$(\theta,\hat{\theta})\leftarrow(\theta,\hat{\theta})-\eta\nabla_{(\theta,\hat{\theta})}L(\theta,\hat{\theta}) \quad (17)$$

3) Updating Centroids

For fixed $\boldsymbol{\mathcal{Z}}$ and network parameters $(\theta,\hat{\theta})$, it is necessary to assign each sample in the latent representative space to its corresponding cluster before updating centroids. We calculate the distance from each $z_i=f(x_i;\theta)$ to all k initial centroids. $d_{j,i}$ indicates whether $f(x_i;\theta)$ belongs to the jth centroid, i.e., $f(x_i;\theta)$ will be assigned to the nearest centroid r_k.

$$d_{j,i}=\begin{cases}1 & if\ j=arg\min_k\|f(x_i;\theta)-r_k\|_2^2\\ 0 & otherwise\end{cases} \quad (18)$$

Following the Deep Clustering Network (DCN) [49] and the Deep Embedded Clustering (DEC) [50], the updating law of centroids is as follows:

$$r_k\leftarrow r_k+\sum_i\left(\frac{1}{C_k^i}\right)(f(x_i;\theta)-r_k)d_{j,i} \quad (19)$$

where C_k^i denotes the total number of the samples assigned to the kth cluster. The gradient step size $1/C_k^i$ controls the learning rate. $\sum_i(f(x_i;\theta)-r_k)d_{j,i}$ indicates a vector formed by subtracting all $f(x_i;\theta)$ in the kth cluster from centroid r_k.

Algorithm 1. A Deep Unsupervised Learning Approach for Airspace Complexity Evaluation (DUACE)

```
Input: Original Airspace sector datasets
1: Data scaling and normalization. Get Data.
2: Expand 30% on Data by SMOTE. Get $\boldsymbol{\mathcal{X}}$.
3: DNN Pre-training:
    for each pre-training epoch:
        for each n batch step:
            3.1: Compute $L_1$ by equation (9)
            3.2: Update network by equation (15)
        end for
    end for
4: Initialize K centroids for $f(\boldsymbol{\mathcal{X}};\theta)$ by k-means.
Get $R_k=(r_1,\ r_2,\ r_3)$
5: DNN Fine-tuning:
    for $\alpha$: $0\rightarrow\alpha_0$:
        for each fine-tuning epoch:
            for each n batch step:
                5.1: Compute $G_{ij}(f(x_i;\theta),\alpha,r_j)$ by
                equations (10) (11)
                5.2: Compute cluster loss by equation (12)
                5.3: Compute total loss by equation (14)
                5.4: Update network by equations (16) (17)
            end for
            5.5: Update the centroids by equation (19)
        end for
    end for
Output: $\theta,\hat{\theta},\boldsymbol{\mathcal{Z}},R$.
```

D. The DUACE Algorithm

The proposed deep unsupervised model is summarized in Algorithm 1. Note that an epoch, whether ***pre-training epoch*** or ***fine-tuning epoch***, corresponds to a pass of all data samples through the network. ***n batch step*** is automatically calculated by the program, indicating how many times the mini-batch needs to be fetched within an epoch so that the network can pass all samples. The algorithmic procedure and the code demo of the DUACE are available at https://github.com/LiBiyue/demo-ITS.

III. Experiment Studies

In the following experiments, the proposed DUACE model was respectively trained using one of the six sectors dataset. To verify our model's performance, we calculate the complexity evaluation accuracy given by our model using the complexity rated by ATM experts.

A. Datasets

The experimental data were collected from the six airspace sectors located in Southwestern China, including "Chengdu01" (CD01), "Chengdu02" (CD02), "Chengdu04" (CD04), "Guiyang01" (GY01), "Guiyang02" (GY02) and "Kunming03" (KM03) (Fig. 2). The datasets cover the air traffic operation of these six sectors from 8:00 to 24:00 on July 28, 2010 [30]. Each sample corresponds to a one-minute air traffic scenario of one sector. Each sample is composed of 28 complexity factors and a corresponding complexity level (Low/Normal/High) assigned by ATM experts. There are 5760 (960 for each sector) samples in total. The number of samples in different categories of each sector is shown in TABLE II.

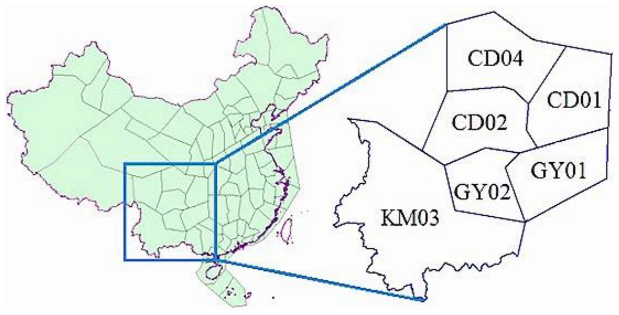

Fig. 2. Sectors used in the experiments.

TABLE II
Basic Information of Sector Datasets

Sector	CD01	CD02	CD04	GY01	GY02	KM03	Total
Low	60	582	77	52	274	279	1328
Normal	360	297	498	411	391	458	2411
High	540	81	385	497	295	223	2021
Total	960	960	960	960	960	960	5760

B. Baseline methods and evaluation metrics

We compared the performances of the proposed DUACE model with two existing representative airspace complexity evaluation methods (BPNN_PCA, SOCKT), four well-known deep unsupervised models (AE_K-means, DEC, DCN, ASPC-DA), and one promising tree boosting approach (XGBoost). Among them, DEC and DCN share the same centroid updating laws as our algorithm, but they do not use SMOTE by KL divergence and probabilistic assignment. Except for not considering the above two aspects, AE_K-means does not incorporate clustering into the deep learning model either. As a relatively new deep unsupervised learning algorithm, ASPC-DA adopts a data augmentation approach and performs well in some image and text clustering tasks. BPNN_PCA, SOCKT, and XGBoost apply the classical supervised model to realize the evaluation of airspace complexity without deep neural networks and the proposed loss function in our model. Additionally, the above models (including the proposed DUACE model and all benchmark models) are independently trained for different sectors.

1) *BPNN_PCA*
   The complexity factor reduction was implemented based on the principal component analysis and Bayesian information criterion. A backpropagation neural network (BPNN) is applied to classify the complexity level [23], [24].
2) *SOCKT*
   Sector operation complexity evaluation framework based on knowledge transfer is proposed to measure the sector's traffic complexity under the condition of small samples. Zhu et al. employed the transfer learning method to classify the complexity level using labeled samples from not only target sector, but also other non-target sectors [30].
3) *AE_K-means*
   This approach extracts the latent features through a stacked Autoencoder (AE) [46]. After dimensionality reduction, the extracted latent features are clustered by K-means.
4) *DEC*
   The Deep Embedded Clustering (DEC) approach performs joint dimensionality reduction and clustering, using the encoder as the network architecture and the KL divergence between the original data and embedded representations as the loss [50].
5) *DCN*
   Deep Clustering Network (DCN) combines the k-means algorithm with an Autoencoder network. DCN is trained by reconstruction loss and k-means loss jointly[49].
6) *XGBoost*
   XGBoost is a scalable end-to-end tree boosting system [59].
7) *ASPC-DA*
   The Adaptive Self-Paced Deep Clustering with Data Augmentation (ASPC-DA) approach is a two-stage deep clustering algorithm by incorporating data augmentation and self-paced learning [60].

To conduct fair comparisons, we adopt the standard unsupervised evaluation metrics for all unsupervised methods. We set the number of clusters to the number of ground-truth categories, that is, "High", "Normal" and "Low", and evaluate performance with the unsupervised clustering accuracy (ACC) [50].

$$ACC = \max_{map} \frac{\sum_{i=1}^{N} \delta(l_i, map(c_i))}{N}. \tag{20}$$

where l_i is the ground-truth label of each sample x_i; c_i is the cluster assignment of x_i produced by the algorithm; and N is the total number of the samples. $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise. $map(\cdot)$ ranges over all possible one-to-one mappings between clusters and labels. ACC is the result under the best mapping, which is represented as $best\_map(\cdot)$. The Hungarian algorithm can efficiently compute the best mapping [61]. Based on the best mapping, we finally determined the label of each cluster.

ACCH, ACCN, ACCL of clusters "High", "Normal" and "Low" can be calculated respectively. The number of the samples in each cluster is $N_s, s \in \{H, N, L\}$.

$$ACCH/ACCN/ACCL = \frac{\sum_{i=1}^{N_s} \delta(l_i, best\_map(c_i))}{N_s} \tag{21}$$

For the supervised learning methods, ACC is the ratio of correctly classified samples to the total number of the samples. ACCH (N/L) is defined as the percentage of correctly classified samples in the high (normal/low) complexity category.

C. Implementation

For a fair comparison with the other Autoencoder based deep unsupervised methods (AE_K-means, DEC, DCN, ASPC-DA), the settings of the Autoencoder are the same. The settings of the user-defined parameters in the model are determined through the grid-search method, which is an exhaustive searching through a manually specified subset of the hyperparameter space of a learning algorithm. The optimal combination of hyperparameters is as follows: the encoder network is set as a fully connected multilayer perceptron (MLP) with dimensions D-24-20-16-12-8 for all experiments, where D is the dimension of the input data (features). The decoder network is a mirror of the encoder, i.e. an MLP with dimensions 8-12-16-20-24-D, where D is the dimension of the reconstructed data (features). All the output layers compute the data from the former layers using the ReLU activation function [48]. The network training is based on the Adam optimizer [48] with a learning rate η=0.001. The training mini-batch is 128 and γ is 0.8. The original data are preprocessed and scaled through standard normalization. The interpolation ratio of SMOTE in the proposed DUACE is set to 30%.

During the layer-wise pre-training of DUACE, we initialize the weights with random numbers drawn from a zero-mean Gaussian distribution with a standard deviation of 0.01. In order to avoid the over-fitting issue, we determine the architecture of the model in a simple-to-complex manner. In the initial process, the model is set to be a shallow network with fewer layers and hidden units. Its performance is evaluated and the model is gradually added with more layers and hidden units continually until its performance starts to degrade. Each layer is pre-trained for 50 epochs without considering any complementary training or regulation strategy such as dropout and batch normalization. The centroids are initialized by K-means.

During the fine-tuning of DUACE, experiments show that the probabilistic matrix G will turn into a similar one-hot matrix as $\alpha = 20$. Therefore, we set α to increase from 0 to 20. The reason for this setting is that in the early stage of the network training, we hope the model assigns a relatively random probability to the samples. With continuous training of the model, it can be gradually deterministic for the probabilistic assignment matrix. For each fixed α, 50 fine-tuning epochs will be implemented because the training error tends to be stable after 50 epochs.

AE_K-means only has the pre-training process with 50 epochs. For DEC, DCN and ASPC-DA, the pre-training and fine-tuning epochs are both set to 50, which is experimentally proved to be sufficient.

For the supervised learning methods BPNN_PCA, SOCKT and XGBoost, their hyperparameters are tuned empirically to get the best results. The neural network of BPNN_PCA has two hidden layers with 100 units, where the activation function is ReLU and the loss function is the cross-entropy-loss. For XGBoost, the depth of the tree is set to 4, and the number of the estimators is 45. For the above two methods, 70% of the labeled dataset of each sector (672 samples) are used for training while the remaining 30% (288 samples) are used for testing. As for SOCKT, for a fair comparison, in each experiment group, the sizes of the target and non-target training sectors are set to 112 (hence the total number of the training samples is $6 * 112$=672). Furthermore, we randomly select 288 data as the test samples of the target sector, which do not overlap with the training data. In this way, the number of training data and test data of the supervised methods are consistent.

D. Results

In order to present the results in a statistically significant way, we carry out 60 seeded runs and use statistical metrics, mean and variance of the accuracy, to evaluate the performance of various models. The experiments are carried out on a ThinkPad P51 laptop, with the version of Python 3.7.3 and Tensorflow 1.13.1.

1) Performance Comparison

The average accuracy and variance of 60 runs for each method are shown in TABLE III. We report the best result for each (data, metric) pair which is highlighted in bold face in this table. It can be seen from the results that the evaluation accuracy of DUACE outperforms those of all other models in most datasets (CD01, CD02, GY01, GY02). It is noteworthy that it performs better than the other models in three sectors on the ACCH indicator (CD01, CD02, GY02). In summary, the DUACE model can assist ATCos to comprehend whether the airspace operation status is complicated and provide an objective evaluation when ATCos are experiencing high workload and stress.

It is also worth noting that although a higher mean of accuracy (ACC, ACCH, ACCN, ACCL) is achieved by our model, the variance is also relatively high. This high variance is not only because the gradient descent optimization of the neural network is a local search, but also the initialization of network parameters and cluster centroids is stochastic. In addition, the structure of deep neural networks is more complicated. There exists a trade-off between the mean and variance of a model [62]. From this perspective, the variance is compromised in exchange for the higher accuracy.

TABLE III
PERFORMANCE COMPARISON WITH EXISTING METHODS (THE DATA IN THE TABLE IS REPRESENTED AS "AVERAGE ACCURACY (VARIANCE)".)

Sector	Metrics (%)	DUACE	BPNN_PCA	SOCKT	XGBoost	AE_K-means	DEC	DCN	ASPC-DA
CD01	ACC	**77.18(7.36)**	75.75(4.21)	65.41(2.85)	70.83(3.88)	64.89(6.92)	67.98(16.98)	69.14(19.66)	71.53(15.29)
	ACCH	**77.60(12.87)**	76.94(5.66)	74.82(5.59)	71.99(4.72)	66.74(6.01)	68.04(18.05)	73.11(14.76)	72.75(13.56)
	ACCN	**77.81(9.98)**	74.01(7.79)	52.23(8.09)	71.74(6.45)	65.09(11.88)	69.19(19.67)	69.02(13.77)	70.63(11.87)
	ACCL	**69.67(13.99)**	63.33(19.10)	59.57(17.23)	68.94(11.32)	54.31(17.95)	60.91(24.33)	62.83(17.59)	62.87(13.66)
CD02	ACC	**82.43(6.43)**	80.08(2.43)	76.44(2.74)	82.42(3.56)	66.78(7.02)	71.49(10.74)	73.68(13.79)	76.81(19.01)
	ACCH	**77.65(14.51)**	74.81(10.45)	53.03(11.61)	71.43(12.78)	64.98(11.21)	67.87(11.28)	73.30(16.89)	71.64(18.57)
	ACCN	**85.86(10.04)**	82.61(5.10)	61.99(7.90)	81.33(6.19)	65.33(13.67)	72.80(14.02)	69.98(14.45)	75.39(12.06)
	ACCL	81.36(14.04)	84.93(3.24)	**86.89(4.5)**	85.43(4.88)	67.83(9.87)	70.09(17.75)	70.88(15.99)	78.25(14.40)
CD04	ACC	78.70(10.38)	80.75(5.35)	62.25(2.99)	**81.67(4.92)**	65.59(10.07)	69.78(14.84)	70.55(18.30)	72.75(12.82)
	ACCH	76.77(15.86)	**82.97(6.46)**	65.17(5.56)	75.51(5.32)	63.74(14.41)	67.83(19.68)	68.01(21.77)	73.68(10.94)
	ACCN	80.81(15.07)	79.92(7.54)	57.88(6.84)	**85.48(8.09)**	66.09(12.22)	70.31(17.09)	72.65(20.82)	73.25(12.88)
	ACCL	**74.77(15.66)**	62.32(14.91)	74.59(10.79)	70.89(10.44)	65.786(14.99)	68.71(18.59)	69.21(22.97)	68.37(18.26)
GY01	ACC	**80.05(7.58)**	77.31(5.78)	73.88(2.44)	79.75(4.12)	66.87(7.98)	68.46(15.88)	73.41(16.70)	75.26(16.67)
	ACCH	79.88(10.19)	**82.77(8.74)**	82.16(4.29)	82.03(7.46)	67.88(11.43)	69.90(13.23)	75.58(15.76)	76.86(13.80)
	ACCN	**80.80(9.44)**	67.74(10.76)	65.26(6.09)	78.91(8.36)	66.31(14.56)	67.79(16.71)	69.56(16.88)	74.28(16.96)
	ACCL	**75.81(10.06)**	58.93(13.34)	61.38(12.93)	57.14(7.93)	60.41(13.39)	63.08(22.35)	65.50(20.09)	69.58(19.54)
GY02	ACC	**78.29(13.45)**	77.33(4.20)	74.47(3.31)	78.23(5.80)	63.21(11.21)	65.61(23.76)	68.92(22.47)	68.58(15.38)
	ACCH	**78.12(18.25)**	73.33(10.75)	77.57(5.61)	75.01(9.63)	55.63(19,01)	59.98(26.43)	61.73(24.62)	67.63(12.67)
	ACCN	**83.07(14.67)**	81.52 (8.70)	70.23(7.49)	82.86(9.65)	60.98(14.92)	61.71(19.11)	65.66(17.89)	69.19(15.33)
	ACCL	73.41(14.72)	73.48(9.35)	77.19(6.04)	**83.58(7.57)**	64.80(10.97)	70.03(15.72)	72.83(18.38)	71.81(12.50)
KM03	ACC	82.87(16.80)	84.67(4.24)	**91.85(1.84)**	87.58(5.16)	67.93(14.51)	68.91(22.73)	73.87(18.30)	76.90(13.38)
	ACCH	84.63(15.04)	80.53(11.37)	**90.77(5.40)**	81.03(8.32)	66.71(15.67)	69.22(21.03)	68.80(24.17)	73.67(18.20)
	ACCN	80.99(14.71)	86.14(12.36)	91.54(2.61)	**92.12(9.60)**	69.41(12.31)	70.53(19.87)	75.51(19.62)	78.07(18.52)
	ACCL	84.58(16.40)	83.46(8.33)	**93.37(3.90)**	87.98(5.79)	59.07(19.71)	62.02(23.52)	64.71(17.98)	79.93(16.72)

Our model outperforms the existing representative airspace complexity evaluation models (BPNN_PCA, SOCKET) and XGBoost in 14 out of 24 indicators, without using any expert labels for training. Although the above supervised models have the advantage of learning expert knowledge, our model achieves a relatively high evaluation accuracy. This higher accuracy is achieved through the power of non-linear mapping realized by deep neural networks and the deliberately designed loss function to address the characteristics of airspace complexity data.

In comparison to other deep unsupervised methods (AE_K-means, DEC, DCN, ASPC-DA), it is evident that our model emerges as the best across all airspace datasets. AE_K-means provides the lowest accuracy among these unsupervised methods. Instead of performing AE and K-means separately [63], DUACE, DEC, DCN, and ASPC-DA jointly optimize dimensionality reduction and clustering in a deep neural network, leading to superior results than those AE_K-means. Although the accuracy of DCN and DEC is slightly higher compared to AE_K-means, it is still lower than the DUACE model. The reason is that in our model the SMOTE and KL divergence loss is developed to address the category imbalance problem. Moreover, DCN and DEC only use a simple K-means loss function rather than the Clustering Loss with probabilistic assignment specifically designed for overlapped airspace complexity data. ASPC-DA utilizes an adaptive self-paced learning mechanism to improve the classification accuracy of the examples near cluster boundaries. However, ASPC-DA still adopts a deterministic cluster assignment approach.

For those imbalanced datasets (CD01, CD02, CD04, GY01), the accuracy of using other deep unsupervised methods to classify the category with fewer samples is always lower, while the variance is higher. The results give clear evidence that those methods cannot extract underlying patterns of the skewed datasets effectively. In contrast, DUACE performs better on these imbalanced datasets and significantly improves the accuracy on the imbalanced categories (CD01- ACCL, CD02- ACCH, CD04- ACCL, GY01- ACCL). This improvement is mainly attributed to the integration of the KL divergence loss. The KL divergence loss ensures the distribution of the newly generated data after SMOTE to be similar to the original data.

It is worth pointing out that all methods perform better on KM03 than they do on other datasets. An intuitive explanation is KM03's geographical location. KM03 is on the southwestern border of China (Fig. 2) and its route topology is different from those of other airspace sectors.

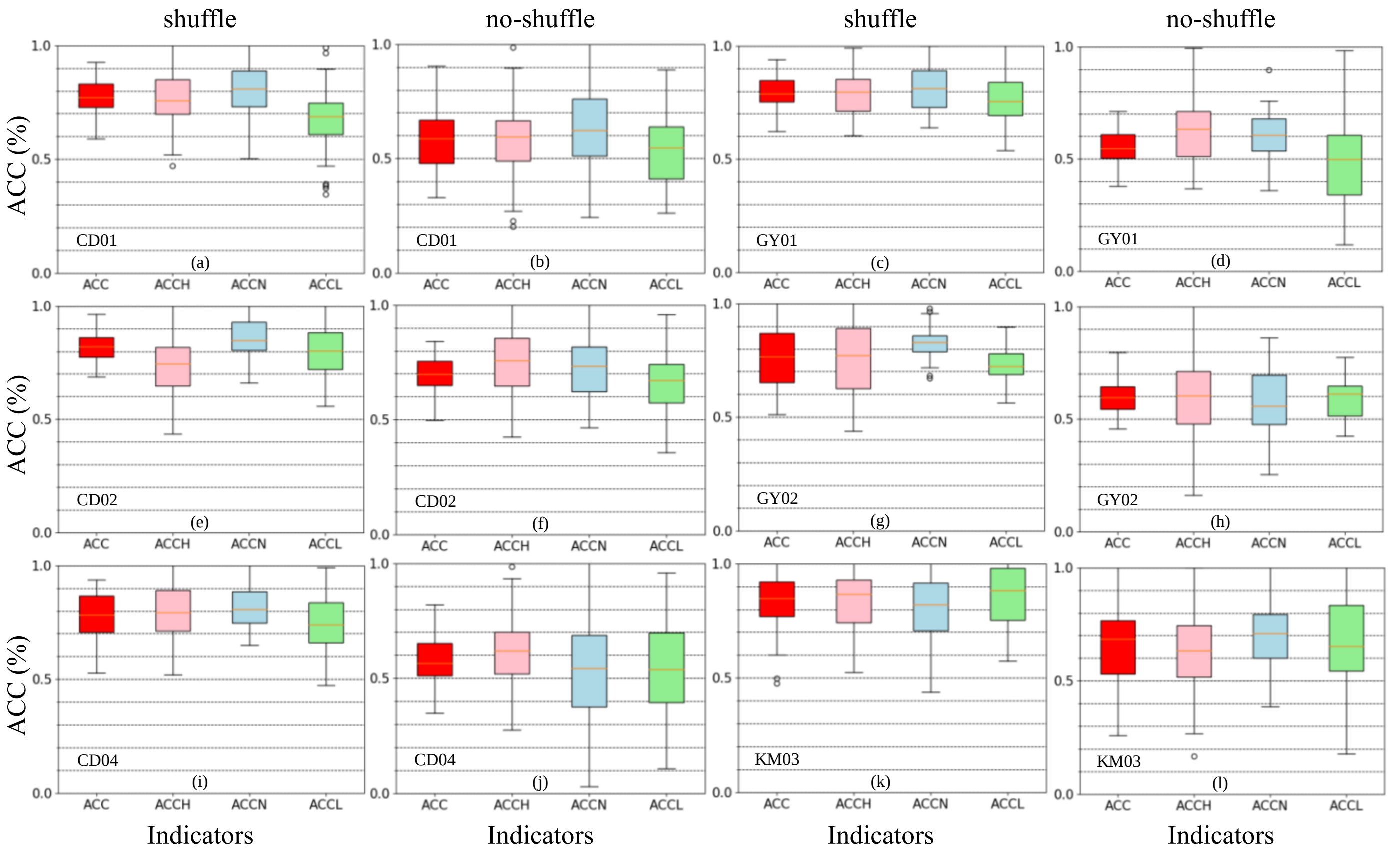

Fig. 3. Results between "shuffle" model and "no-shuffle" model.

2) The Role of SMOTE by KL Divergence

To further demonstrate the critical role of SMOTE and L_{KL}, experiments with the variable control method are implemented. According to Section II-B, SMOTE by KL divergence plays a crucial role in handling the problem of data imbalance. Therefore, we carry out a more specific evaluation and analysis of DUACE on airspace complexity datasets of each sector to demonstrate the effectiveness of SMOTE and KL divergence. Fig. 3 provides a comparison of the "shuffle" (with SMOTE and L_{KL}) and "no-shuffle" (without SMOTE and L_{KL}). The box-plot of accuracy through 60 runs on each airspace sector dataset is shown in Fig. 3.

As can be seen from Fig. 3, the model with SMOTE and L_{KL} contributes to improving the overall evaluation accuracy of the airspace complexity datasets. In addition, since SMOTE by KL Divergence is designed to address the category imbalance problem of airspace complexity data, the proposed model with "shuffle" enhances the accuracy on those categories of fewer airspace complexity samples (CD01-ACCL, CD02-ACCH, CD04-ACCL, GY01-ACCL, GY02-ACCL, KM03-ACCH). The design of SMOTE by KL Divergence greatly improves the evaluation accuracy of the DUACE model.

3) The Role of Probabilistic Assignment

In order to determine the role of probabilistic assignment, we observe how the hyperparameter α in the probability matrix G of DUACE affects the evaluation accuracy and variance on each sector. Using the same experimental settings in Section C, we record all the accuracy as α changes from 0 to 20. For each fixed α, the mean and the variance of accuracy for all 60 runs are computed. We then plot the error-bar for each airspace sector dataset as shown in Fig. 4.

Instead of deterministically assigning a sample to a specific centroid, the probabilistic assignment considers the impact of all centroids on a sample. Therefore, the airspace complexity data with overlapped boundaries can still be clustered accurately. The experimental results in Fig. 4 confirm that the proposed probabilistic assignment improves the evaluation accuracy of the model.

The results show that the accuracy varies with α. For some datasets (GY01, GY02, KM03), the accuracy is an incremental function of α, while for others (CD01, CD02, CD04), it is a parabolic function. This means that there exists an optimal choice for α for DUACE to reach the best performance. Specifically, for GY01, GY02 and KM03, the accuracy is increased while the variance is decreased with the increase of α from 0 to 20. However, for CD01, CD02 and CD03, the most robust model with a satisfactory accuracy is obtained around $\alpha = 6$. In summary, it is recommended to set α to 18 for GY01, GY02 and KM03 and 6 for CD01, CD02 and CD04.

A more in-depth look into the datasets reveals that α is closely related to the geographical regions (the hidden structure) of airspace sectors (see Fig. 2). For the sectors in the South, including GY01, GY02 and KM03, α is bigger. For the sectors in the North, including CD01, CD02 and CD04, α is smaller.

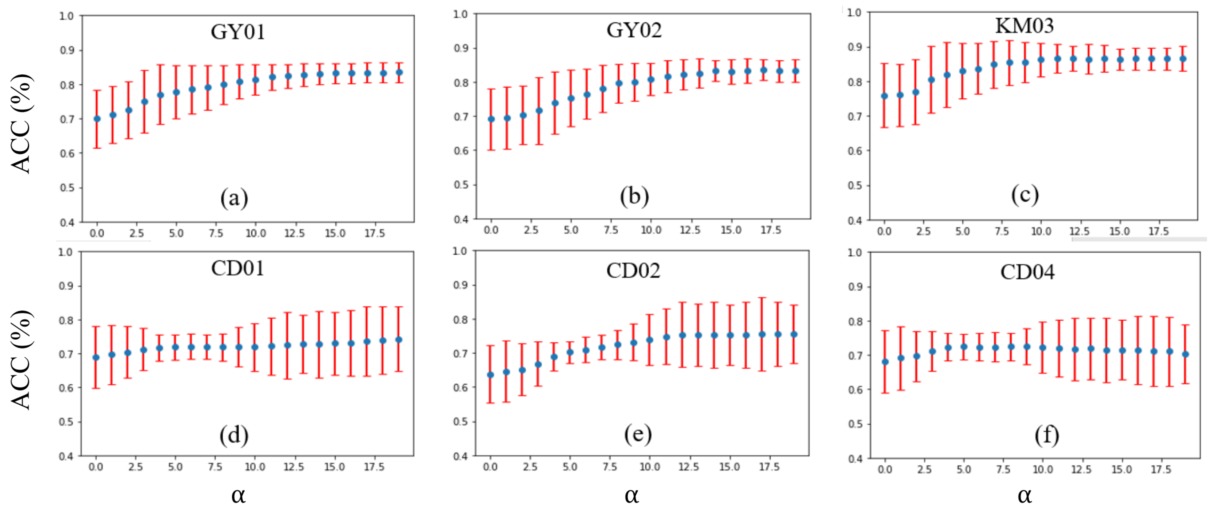

Fig. 4. The effect of α on accuracy and robustness.

Therefore, the setting of the hyperparameter α in DUACE can be automated. The value of α can also provide more information for airspace decision making.

IV. Conclusion

We have proposed a deep unsupervised learning approach for airspace complexity evaluation (DUACE). The approach takes full advantage of the non-linear mapping power provided by DNN and a specifically designed latent representative space which is tailored for clustering. The characteristics of the airspace datasets, including category imbalance and boundaries overlapping, pose significant challenges for machine learning in general and unsupervised learning in particular. In order to address the former problem, we used SMOTE to generate more data that is constrained by KL divergence. The aim is to balance the amount of data in different categories. For the latter, we proposed a probabilistic assignment loss function in the process of training to improve the clustering performance. Furthermore, we optimized the loss function through the pre-training and the joint fine-tuning processes.

The proposed DUACE has been validated through a range of rigorously designed experiments. The results demonstrate consistent improvements in accuracy on different airspace datasets compared to two existing representative airspace complexity evaluation methods (BPNN_PCA, SOCKT), one promising tree boosting approach (XGBoost) and four well-known deep unsupervised models (AE_K-means, DEC, DCN, ASPC-DA). Furthermore, we investigate the roles of the two newly proposed loss functions L_{KL} and L_{clu}, and demonstrate their contributions towards the improved performance. More interestingly, we discover that one of the hyperparameters of our proposed model is closely related to the geographical regions (the hidden structure) of airspace sectors, prompting a need for further research.

In real practice, the DUACE model can assist ATCos to comprehend whether the airspace operation status is complicated and provide an objective evaluation when ATCos are experiencing high workload and stress. The output of the model is three data clusters and the corresponding centroids. Air traffic controllers can refer to the output centroids and few samples in the same cluster to determine the corresponding airspace complexity levels. In this way, the manpower, workload and material costs of the airspace complexity evaluation work will be greatly reduced.

Acknowledgement

This paper is supported by National Key Research and Development Program of China under Grant 2019YFF0301400, National Natural Science Foundation of China under Grant 62088101, 61961146005 and the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/N029496/2.

References

[1] Report on national civil aviation flight operation efficiency in 2017, Center, CAAC Operation Monitoring (Center C. O. M.), Beijing, China, 2018.
[2] M. Bloem, and P. Gupta, "Configuring airspace sectors with approximate dynamic programming," in *27th Cong. Int. Counc. Aeronaut. Sci.*, vol. 5, Nice, France, Sept. 2010, pp. 4085-4097.
[3] M. Hansen and Y. Zhang, "The link between operational performance and operational errors in the national airspace system," in *Proc. 6th USA/Eur. Air Traffic Manag. R&D Semin.*, Baltimore, MD, Jun. 2005.
[4] F. Poli, Air traffic controllers falling asleep while on the job, 2015, [online] Available: http://www.linkedin.com/pulse/ait-traffic-controllers-falling-asleep-while-job-fabrizio-poli.
[5] A. Lecchini Visintini, W. Glover, J. Lygeros, and J. Maciejowski, "Monte Carlo Optimization for Conflict Resolution in Air Traffic Control," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 470-482, Dec. 2006.
[6] K. Lee, E. Feron, and A. Pritchett, "Air Traffic Complexity: An Input-Output Approach," in *Proc. Amer. Control Conf.*, New York, NY, 2007, pp. 474-479.
[7] K. Lee, E. Feron, and A. Pritchett, "Describing airspace complexity: Airspace response to disturbances," *J. Guid. Control Dyn.*, vol. 32, no. 1, pp. 210–222, Jan./Feb. 2009.
[8] Y. Hong, Y. Kim, and K. Lee, "Conflict management in air traffic control using complexity map," *J. Aircr.*, vol. 52, no. 5, pp. 1524-1534, 2015.
[9] M. Prandini and J. Hu, "A probabilistic approach to air traffic complexity evaluation," in *Proc. 48th IEEE Conf. Decision Control*, Shanghai, China, Dec. 2009, pp. 5207–5212.

[10] M. Prandini, V. Putta, and J. Hu, “A probabilistic measure of air traffic complexity in three-dimensional airspace,” *Int. J. Adapt. Control Signal Process.*, vol. 24, no. 10, pp. 813–829, 2010.

[11] S. Puechmorel and D. Delahaye, “New trends in air traffic complexity,” in *Proc. EIWAC*, Tokyo, Japan, Mar. 2009.

[12] D. Delahaye, S. Puechmorel, R. Hansman, and J. Histon, “Air traffic complexity based on nonlinear dynamical systems,” in *Proc. 5th USA/Eur. Air Traffic Manag. R&D Semin.*, Budapest, Hungary, Jun. 2003.

[13] S. Alam, C. Lokan, and H. Abbass, “What can make an airspace unsafe? characterizing collision risk using multi-objective optimization,” in *Proc. IEEE Congr. Evol. Comput. (IEEE CEC)*, Brisbane, QLD, Australia, Jun. 2012, pp. 1–8.

[14] M. Nguyen and S. Alam, "Airspace Collision Risk Hot-Spot Identification using Clustering Models," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 48-57, Jan. 2018.

[15] B. Sridhar, K. Sheth, and S. Grabbe, “Airspace complexity and its application in air traffic management,” in *Proc. 2nd USA/Eur. Air Traffic Manag. R&D Semin.*, Orlando, FL, Dec. 1998.

[16] P. Kopardekar and S. Magyarits, “Measurement and prediction of dynamic density,” in *Proc. 5th USA/Eur. Air Traffic Manag. R&D Semin.*, Budapest, Hungary, Jun. 2003, pp. 1–9.

[17] D. Wu, Z. Jiang, X. Xie, X. Wei, W. Yu, and R. Li, “LSTM Learning With Bayesian and Gaussian Processing for Anomaly Detection in Industrial IoT,” *IEEE Trans. Ind. Inform.*, vol. 16, no. 8, p. 10, 2020.

[18] E. Principi, D. Rossetti, S. Squartini, and F. Piazza, “Unsupervised electric motor fault detection by using deep autoencoders,” *IEEECAA J. Autom. Sin.*, vol. 6, no. 2, pp. 441–451, Mar. 2019.

[19] P. Ping, W. Qin, Y. Xu, C. Miyajima, and K. Takeda, “Impact of Driver Behavior on Fuel Consumption: Classification, Evaluation and Prediction Using Machine Learning,” *IEEE Access*, vol. 7, pp. 78515–78532, 2019.

[20] C. Ieracitano, A. Paviglianiti, M. Campolo, A. Hussain, E. Pasero, and F. C. Morabito, “A novel automatic classification system based on hybrid unsupervised and supervised machine learning for electrospun nanofibers,” *IEEECAA J. Autom. Sin.*, vol. 8, no. 1, pp. 64–76, Jan. 2021.

[21] G. Cai, Y. Wang, L. He, and M. Zhou, “Unsupervised Domain Adaptation With Adversarial Residual Transform Networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 3073–3086, Aug. 2020.

[22] G. Chatterji and B. Sridhar, "Measures for air traffic controller workload prediction," in *Proc. 1st AIAA, Aircraft, Technol. Intgn., Oper. Forum*, Los Angeles, CA, Oct. 2001.

[23] D. Gianazza and K. Guittet, “Selection and evaluation of air traffic complexity metrics,” in *Proc. 25th DASC*, 2006, pp. 1–12.

[24] D. Gianazza, "Forecasting Workload and Airspace Configuration with Neural Networks and Tree Search Methods", *Artif. Intell.*, vol. 174, no. 7-8, pp. 530-549, May 2010.

[25] P. Andraši, T. Radišić, D. Novak, and B. Juričić, “Subjective Air Traffic Complexity Estimation Using Artificial Neural Networks,” *PROMET - TrafficTransportation*, vol. 31, no. 4, pp. 377–386, Aug. 2019.

[26] M. Xiao, J. Zhang, K. Cai, and X. Cao, "ATCEM: A Synthetic Model for Evaluating Air Traffic Complexity", *J. Adv. Transp.*, vol. 50, no. 3, pp. 315-325, Apr. 2016.

[27] D. I. Arkhipov, D. Wu, T. Wu, and A. C. Regan, “A Parallel Genetic Algorithm Framework for Transportation Planning and Logistics Management,” *IEEE Access*, vol. 8, pp. 106506–106515, 2020.

[28] X. Zhu, X. Cao, and K. Cai, "Measuring air traffic complexity based on small samples", *Chin. J. Aeronaut.*, vol. 30, no. 4, pp. 1493-1505, Aug. 2017.

[29] X. Zhu, K. Cai, and X. Cao, "A semi-supervised learning method for air traffic complexity evaluation," in *Proc. 17th ICNS*, Herndon, VA, Apr. 2017, pp. 1A3-1-1A3-11.

[30] X. Cao, X. Zhu, Z. Tian, J. Chen, D. Wu, and W. Du, "A knowledge-transfer-based learning framework for airspace operation complexity evaluation", *Transp. Res. C Emerg. Technol.*, vol. 95, pp. 61-81, Oct. 2018.

[31] B. Antulov-Fantulin, B. Juričić, T. Radišić, and C. Çetek, “Determining Air Traffic Complexity – Challenges and Future Development,” *Promet - TrafficTransportation*, vol. 32, no. 4, pp. 475–485, Jul. 2020.

[32] S. Alam, H. A. Abbass, and M. Barlow, "ATOMS: Air Traffic Operations and Management Simulator," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 2, pp. 209-225, Jun. 2008.

[33] S. Alam et al., "Real time prediction of worst case air traffic sector collision risk using evolutionary optimization," in *Proc. IEEE Symp. Ser. Comput. Intell.*, Singapore, Apr. 2013, pp. 72-79.

[34] M. Prandini, L. Piroddi, S. Puechmorel, and S. L. Brazdilova, "Toward Air Traffic Complexity Assessment in New Generation Air Traffic Management Systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 809-818, Sept. 2011.

[35] A.M. Churchill, and D. J. Lovell, “Assessing the impact of stochastic capacity variation on coordinated air traffic flow management,” *Transp. Res. Rec.*, vol. 2114, no. 1, pp.111-116, Dec. 2011.

[36] K. Vlachou, and D. J. Lovell, “Mechanisms for equitable resource allocation when airspace capacity is reduced,” *Transp. Res. Rec.*, vol. 2325, no. 1, pp.97-102, 2013.

[37] T. Radišić, D. Novak, and B. Juričić, “Reduction of Air Traffic Complexity Using Trajectory-Based Operations and Validation of Novel Complexity Indicators,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3038-3048, Nov. 2017.

[38] T. Koca, M. A. Piera, and M. Radanovic, “A Methodology to Perform Air Traffic Complexity Analysis Based on Spatio-Temporal Regions Constructed Around Aircraft Conflicts,” *IEEE Access*, vol. 7, pp. 104528-104541, Jul. 2019.

[39] K. Treleaven and Z. Mao, “Conflict Resolution and Traffic Complexity of Multiple Intersecting Flows of Aircraft,” *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 4, pp. 633-643, Dec. 2008.

[40] C. S. Y. Wong, S. Sundaram, and N. Sundararajan, “CDAS: A Cognitive Decision-Making Architecture for Dynamic Airspace Sectorization for Efficient Operations,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1659-1668, May 2019.

[41] R. Bonetti and A. Guglielmetti, “Cluster radioactivity: an overview after twenty years,” *Rom. Rep. Phys.*, vol. 59, no. 2, pp. 301-310, 2007.

[42] X. Xu, J. Li, M. Zhou, J. Xu, and J. Cao, “Accelerated Two-Stage Particle Swarm Optimization for Clustering Not-Well-Separated Data,” *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 50, no. 11, pp. 4212–4223, Nov. 2020.

[43] M. Ghahramani, M. Zhou, and C. T. Hon, “Extracting Significant Mobile Phone Interaction Patterns Based on Community Structures,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 1031–1041, Mar. 2019.

[44] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometr. Intell. Lab. Syst.*, vol. 2, no. 1-3, pp. 37-52, Aug. 1987.

[45] S. Roweis and L.K. Saul, “Nonlinear Dimensionality Reduction by Locally Linear Embedding,” *Science*, vol. 290, no. 5500, pp. 2323-2326, Dec. 2000.

[46] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion,” *J. Mach. Learn. Res.*, vol. 11, pp. 3371-3408, 2010.

[47] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, W. W. Cohen, A. McCallum, and S. T. Roweis, Eds., “Extracting and composing robust features with denoising autoencoders,” in *Proc. 25th Int. Conf. Mach. Learn.*, Helsinki, Finland, Jul. 2008, pp. 1096–1103.

[48] G. Hinton and R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” *Science*, vol. 313, no. 5786, pp. 504-507, 2006.

[49] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, “Towards k-means-friendly spaces: Simultaneous deep learning and clustering,” in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 8, Sydney, NSW, Australia, Aug. 2017, pp. 5888-5901.

[50] J. Xie, R. Girshick, and A. Farhadi, “Unsupervised deep embedding for clustering analysis,” in *Proc. 33rd Int. Conf. Mach. Learn.*, vol. 1, New York, NY, Jun. 2016, pp. 478-487.

[51] M. M. Fard, T. Thonet, and E. Gaussier, “Deep k-means: Jointly clustering with k-means and learning representations,” 2018, [online] Available: https://arxiv.org/abs/1806.10069.

[52] S. Affeldt, L. Labiod, and M. Nadif, “Spectral clustering via ensemble deep autoencoder learning (SC-EDAE),” *Pattern Recognit.*, vol. 108, p. 107522, Dec. 2020.

[53] J. Maggu, A. Majumdar, and E. Chouzenoux, “Transformed Subspace Clustering,” *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2020.

[54] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[55] G. Hinton, S. Roweis, “Stochastic neighbor embedding,” in *Adv. neural inf. proces. Syst.*, Cambridge, MA, 2003, pp. 833-840.

[56] E. Jang, S. X. Gu, and B. Poole, “Categorical reparameterization with gumbel-softmax,” in *Proc.5th Int. Conf. Learn. Represent.*, Toulon, France, Apr. 2017.

[57] N. D. Lawrence, "Probabilistic Non-Linear Principal Component Analysis with Gaussian Process Latent Variable Models," *J. Mach. Learn. Res.*, vol. 6, pp. 1783-1816, Nov. 2005.
[58] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back propagating errors," *Nature*, vol. 323, no. 6088, pp. 533-536, 1986.
[59] T. Chen, C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc.22nd Int. Conf. Knowl. Discov. Data Min*., San Francisco, CA, Aug. 2016.
[60] X. Guo et al., "Adaptive Self-paced Deep Clustering with Data Augmentation," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 9, pp. 1680-1693, Sept. 2020
[61] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logist.*, vol. 52, no. 1, pp. 7-21, Feb. 2005.
[62] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.
[63] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798-1828, Aug. 2013.

Biyue Li received the B.S. degree from the College of Information and Electrical Engineering, China Agricultural University, Beijing, China, in 2018. She is currently pursuing the Ph.D. degree in traffic information engineering and control at Beihang University, Beijing, China. Her current research interests include airspace complexity analysis and machine learning.

Wenbo Du (M'17) received the B.S. and Ph.D. degrees from the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively.

He is a Professor with the School of Electronic and Information Engineering, Beihang University, Beijing, China. His current research interests include data science and intelligent transportation.

Yu Zhang received the B.S. degree in transportation engineering from Southeast University, Nanjing, China, the M.S. and Ph.D. degrees in civil and environmental engineering from the University of California Berkeley, Berkeley, CA, USA.

She is currently an Associate Professor with the Department of Civil and Environmental Engineering at the University of South Florida, Tampa, FL, USA. Her current research interests are: Transportation system modeling, analysis, and simulation; Resilient system design and operations; Air transportation and global airline industry; Multimodal transportation planning and sustainable transportation.

Jun Chen received the B.Sc. degree in electrical engineering and automation from the Nanjing University of Science and Technology, Nanjing, China, and the M.Sc. degree in software engineering from Tongji University, Shanghai, China. He received the second M.Sc. (with distinction) and Ph.D. degrees in systems engineering and control from The University of Sheffield, Sheffield, U.K.

He is currently a Senior Lecturer in Engineering Science at Queen Mary University of London, London, U.K. He has published more than 60 scientific papers in areas of multi-objective optimization, interpretable fuzzy systems, data-driven modelling, and intelligent transportation systems. From 2020, he serves as a full member of the EPSRC Peer Review College. He is also a Turing Fellow at Alan Turing Institute.

Ke Tang (S'05-M'07-SM'13) received the B.Eng. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2002, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2007. From 2007 to 2017, he was with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China, first as an Associate Professor from 2007 to 2011 and later a Professor from 2011 to 2017.

He is currently a Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. He has published more than 70 journal papers and more than 80 conference papers. According to Google Scholar, his publications have received more than 8000 citations and the H-index is 42. His major research interests include evolutionary computation, machine learning, and their applications.

Xianbin Cao (M'08-SM'10) received the B.Eng and M.Eng degrees in computer applications and information science from Anhui University, Hefei, China, in 1990 and 1993, respectively, and the Ph.D. degree in information science from the University of Science and Technology of China, Hefei, in 1996.

He is currently a Professor with the School of Electronic and Information Engineering, Beihang University, Beijing, China. His current research interests include intelligent transportation systems, air traffic management, and intelligent computation.