aerospace

MDPI

Article

Congestion Recognition of the Air Traffic Control Sector Based on Deep Active Learning

Xianghua Tan [1,2,*], Yushi Sun [2,3], Weili Zeng [2,3] and Zhibin Quan [4]

[1] Public Foundational Courses Department, Nanjing Vocational University of Industry Technology, Nanjing 210023, China
[2] College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; yushisun@nuaa.edu.cn (Y.S.); zwlnuaa@nuaa.edu.cn (W.Z.)
[3] National Key Laboratory of Air Traffic Flow Management, No. 29 General Road, Nanjing 211106, China
[4] School of Automation, Southeast University, Nanjing 210096, China; sigma_quan@seu.edu.cn
* Correspondence: tanxh@niit.edu.cn

Abstract: The air traffic control sector (ATCS) is the basic unit of the airspace system. If we can identify the congestion of an ATCS, it will help provide decision support for planning and daily operations. However, current methods mainly characterize congestion from the static structure and the dynamic operational features, resulting in poor generalization and operability. To this end, we propose a deep learning method from the perspective of complex networks. It takes aircraft as nodes to construct an aircraft network and utilizes the complexity indices to characterize it. So, the problem of identifying congestion becomes the complexity of the aircraft network. Inspired by active learning methods, we construct a deep active learning (DAL) model for congestion recognition. It adopts an iterative semi-supervised approach to reduce the number of labeled samples while ensuring recognition performance. To make full use of a large number of unlabeled samples, the sparse autoencoder is employed to characterize all labeled samples and unlabeled samples. The hidden layer of the deep neural network is constructed by stacking. In the process of active learning iteration, minimum confidence, marginal sampling, and information entropy are introduced as measures to select samples from the unlabeled sample set with significantly different features from the labeled sample set. The model is applied to three representative sectors in China's airspace as cases. Results suggest that DAL can reduce the labeled sample set's redundancy and achieve the desired performance with the smallest number of samples. Additionally, DAL is superior to the existing mainstream methods in the four objective evaluation indices.

Keywords: air traffic control sector (ATCS); congestion recognition; deep active learning (DAL); sparse self-coding; sample labeling strategy

MSC: 90B20

check for updates

Citation: Tan, X.; Sun, Y.; Zeng, W.; Quan, Z. Congestion Recognition of the Air Traffic Control Sector Based on Deep Active Learning. *Aerospace* **2022**, *9*, 302. https://doi.org/10.3390/aerospace9060302

Academic Editor: Alexei Sharpanskykh

Received: 8 April 2022
Accepted: 30 May 2022
Published: 2 June 2022

1. Introduction

With the rapid development of the civil aviation industry, air transport demand continues to grow rapidly. However, limited airspace resources and the slow progress of air traffic control technologies cannot meet the rapidly increasing demand for transportation, resulting in increasingly crowded airspace. Airspace congestion can lead to a series of problems, such as flow control, flight delays (increased potential flight conflicts, reduced flight safety), and longer flight times (increase pollutant emissions). In the face of airspace congestion, identifying air traffic congestion is the premise and foundation of air traffic management [1].

Currently, the air traffic management mode adopts an operation mode based on the air traffic control sector (ATCS). The airspace divides into multiple sectors with relatively fixed boundaries. Air traffic controller management is responsible for the orderly transfer of aircraft into and out of the sector and guides the aircraft to fly safely within the controlled sector [2,3]. This paper aims to identify the degree of ATCS congestion, which has strategic and tactical significance. Strategically, recognizing the historical congestion of a sector can provide a basis for decision-making on sector consolidation and addition. Tactically, real-time recognition of each sector's congestion allows controllers to dynamically grasp the congestion status of the sector and neighboring sectors, which allows them to more effectively guide the aircraft through this sector and reasonably hand it over to adjacent sectors.

ATCS congestion recognition is also known as complexity assessment [4,5]. Many researchers have studied sector congestion recognition and improved methods using two aspects: evaluation indicators and evaluation methods. In terms of the construction of congestion evaluation indicators, current research mainly considers the reasons for congestion, the workload of controllers, and the characteristics of congestion. To construct indicators from the perspective of congestion causes, we mainly consider the imbalance between traffic demand and traffic capacity. Wanke et al. [6] quantified the uncertainty of predicted flow level and airspace capacity, and then solved the probabilistic model of traffic demand prediction. At the same time, they obtained a monitor for predicting the probability of a congestion alarm. Subsequently, Mulgund et al. [7] applied a genetic algorithm to airspace structure management in 2006, easing the degree of air traffic congestion through two solutions: reprogramming of flight tracks and flight delays on the ground. Sood et al. [8] went a step further. They developed adaptive genetic algorithms based on weather factors and traffic predictions to balance multiple metrics, which was different from traditional viewing genetic algorithms, and simulated its results.

Since the controller manages each sector, the controller's workload directly reflects the sector's congestion. Some studies have constructed congestion metrics from the controller's workload perspective. Djokic et al. [9] proposed the conjecture that there might be a connection between the communication load and the controller's load according to the execution process of the controller's work. They proved the conjecture through principal component analysis of many parameters related to aircraft, followed by step-by-step multiple regression analysis. Van Paassen et al. [10] established an airspace complexity indicator system and evaluated the complexity based on the characteristic quantity in the process of aircraft conflict relief. On the other hand, input-output flow is also an important indicator to reflect the degree of congestion. Lee et al. [11] constructed an input-output system framework to analyze the airspace and expressed the airspace complexity in a given traffic situation using a complex graph. Then, on the original basis, the disturbance to the closed system was considered, that is, the change of airspace complexity when the system enters the aircraft, and how the air traffic controllers respond to it. Additionally, Laudeman et al. [12] proposed the concept of spatial dynamic density. Based on the existing assessment methods of traffic density, combined with traffic complexity, the indicator of traffic complexity was determined, and a dynamic density equation was constructed to measure the workload of traffic controllers. Kopardekar et al. [13], based on their predecessors, researched the city of Cleveland and constructed a measurement matrix of dynamic density, they then used code reduction and analysis tools to simulate the aircraft in the sector in real-time, thereby simulating the load of the controller. In addition, some researchers have combined the selection of evaluation indicators with machine learning. Chatterji and Sridhar [14] divided the constructed evaluation indicators of air traffic complexity into ten sets for neural network training to select the best set for evaluating controller workload performance.

In addition to the characterization of congestion behaviors from the causes and controller's workload, researchers have also constructed congestion indicators from the characteristics of congestion. Delahaye and Puechmorel [15] divided the air traffic congestion indicator into two parts based on the complexity indicator constructed by predecessors. One is the aircraft geometry attribute; the other uses the expression of air traffic flow as a dynamic system to obtain the complexity measurement index through topological entropy. Further, Delahaye et al. [16] carried out the nonlinear expansion of the linear dynamic system, calculated its related Lyapunov index, and obtained the complexity, which can be used to identify the high (or low) complexity regions on the map. Lee et al. [17], based on the nonlinear dynamic system, incorporated the uncertainty of flight trajectory into the spatial dynamical system's construction process and improved the original system.

Another challenging issue is assessing the level of congestion. The present research can be divided into the threshold discrimination method, comprehensive weighting method, cluster analysis method, and machine learning method. Early studies were mainly based on threshold discrimination, comprehensive weighting, and cluster analysis. Bilimoria and Lee [18] proposed an automatic cluster recognition method that uses the cluster quality indicator obtained from the threshold distance of aircraft to identify the crowding situation in the high-dynamic region. Surakitbanharn et al. [19] proposed a linear weighting method to integrate congestion evaluation indicators. Because of the nonlinear relationship between the congestion indicators, the model precision was reduced by linear weighting. Brinton and Pledgie [20] used the collection characteristics of aircraft action trajectories for clustering, constructed complexity indicators from the perspective of dynamic density, calculated airspace complexity, and partitioned its airspace. Nguyen [21] segmented the aircraft trajectory, used a deterministic annealing algorithm for linear clustering, classified it according to time, and finally output the clustering result. Based on the complex network theory, Wang et al. [22] divided the aviation network into 2D and 3D, established a two-layer and multi-level dynamic network model, and divided the complexity into three categories by using a clustering algorithm to discuss the change frequency of traffic conditions.

In recent years, machine learning methods have attracted researchers' attention. Gianazza [23] used principal component analysis to extract six key factors from 28 complexity indicators and used BPNN (Back Propagation neural network) to evaluate the crowding level of sectors. Xi et al. [24] proposed an integrated learning model based on the complexity of air traffic in the region, considering that the cost of sample labeling is too high, which leads to the problem that the labeled samples are insufficient for training. Jiang et al. [4] put forward a method for identifying sector congestion based on the independent component analysis. A sector congestion indicator system was established based on complex network theory. The independent component analysis method was used to analyze the main components of all discriminative indicators and extracted by detection. The main element of the indicator is the abnormal value of the actual sector operation to determine whether the sector's operating status is abnormal.

The machine learning method mentioned above can achieve excellent results, but the prerequisite is based on having enough labeled samples. However, sample marking must be done by experienced controllers, requiring a lot of time and energy. Even if we obtain a large number of labeled samples, mainstream recognition methods, such as support vector machines (SVM) or shallow neural network models, cannot effectively capture the nonlinear relationship between overall congestion level and individual congestion evaluation indicators. Additionally, the traditional congestion indicators, such as the distance, density, and conflict times, are the results of the regulation of controllers. They cannot truly reflect the self-organizing behavior of aircraft. Due to the difference in the structure for each ATCS, the congestion characteristics vary from ATCS to ATCS. Therefore, it is necessary to construct indicators to measure congestion behavior from the mechanism of formation, thereby guaranteeing that the same indicator behaves consistently at the same congestion level in various ATCSs and is independent of spatial structure. So, the congestion recognition model is more general and suitable for the current mainstream ATCSs with fixed routes and the free route airspace partly used in Europe.

However, at present, most methods mainly use static structure and dynamic operation characteristics to characterize congestion, which leads to poor generalization and operability. In essence, ATCS is a complex network system composed of aircraft. The nodes in the network are aircraft, and the connections in the network are the interrelationships between aircraft. Thus, this paper transforms the problem of ATCS congestion recognition into the complexity of the aircraft state network system from the perspective of complexity theory. We propose the deep active learning (DAL) method for ATCS congestion recognition, which can achieve a relatively high recognition rate with the smallest labeled samples. The main contributions of this paper are as follows: (a) An aircraft state network is built from the perspective of complexity theory. Some complexity indicators are selected to characterize congestion according to the relationship between the aircraft status network and the actual operation of the control; (b) This paper adopts an iterative semi-supervised deep learning architecture, learns features from a small number of labeled samples and a large number of unlabeled samples through a sparse autoencoder, and builds a deep neural network model by stacking to improve the congestion recognition rate; (c) Least confidence, marginal sampling, and information entropy are introduced as metrics of sample selection. This method can choose samples that are significantly different from the features of the labeled sample set to reduce data redundancy and the cost of manually labeling.

The organization of this paper is as follows: In Section 2, the problem of ATCS recognition is elaborated; Section 3 proposes a deep active learning method to identify the degree of ATCS congestion, including how to use a sparse autoencoder to train hidden layer features, how to select representative samples from unlabeled samples, and the description of implementation steps. In Section 4, the effectiveness and superiority of the model are verified through experiments. Finally, Section 5 concludes this paper.

2. Problem Formulation and Method Overview

ATCS congestion recognition aims to classify the overall congestion of a sector according to multiple congestion characteristics. Suppose there are n evaluation indicators for sector congestion. Classify sector congestion level into c category, and the category label set is $Y = \{y_1, y_2, \cdots, y_c\}$, where y_i is the label for the i-th class of congestion. Let $X(t) = \{x_1(t), x_2(t), \cdots, x_n(t)\}$ and $y(t)$ denote a feature vector composed of n indicator values and the congestion level at time t. Identifying the sector congestion level can be regarded as a classification problem, that is, by training the classification model Ψ, the feature vector $X(t)$ is mapped to the corresponding congestion level $y(t)$:

$$\Psi(X(t)) \to y(t), \tag{1}$$

As can be seen from (1), the problem of sector congestion recognition involves the construction of a suitable feature vector X and a classification model Ψ. In response to this problem, there are three main challenges:

(a) Congestion evaluation indicators should be able to indirectly reflect the workload of the controller and have universal applicability and operability. Since sector congestion is ultimately reflected in the controller's workload, if the congestion evaluation index cannot indirectly reflect the controller's workload, the constructed index will not help improve the ability to identify congestion. To improve the generalization and practicability of the subsequent recognition model, the crowding index must have excellent crowding feature characterization capabilities and be easy to obtain from the monitoring information system.

(b) The congestion recognition model has the ability to characterize a large amount of training data. In general, the more training data, the better the model's performance. However, shallow learning models (such as BPNN and SVM) have limited ability to capture features from data, and it is easy to saturate in performance. Entering more data does not improve model performance [23,24]. In contrast, for deep learning models, the more data involved in training, the more they can achieve their potential.

(c) The sample labeling strategy can reduce the redundancy of the labeled sample set. Usually, training a high-precision congestion recognition classifier requires a large number of labeled samples. The sector congestion level needs to be manually marked by an experienced controller, which is expensive. For example, every 10 s, sector status information will update. If one month of data is needed to train the classification model, the total number of samples is 259,200, which is labeled by a controller, six samples per minute, and 8 h of work a day, then it will take three months to mark them. In addition, if the data is not distinguished, some classes will be redundant, and some will be insufficiently labeled.

In response to the above three challenges, we transform sector congestion recognition into the complexity evaluation of aircraft state networks. This paper proposes a new sector congestion recognition architecture, as shown in Figure 1. We first build a network with aircraft as nodes based on the aircraft's dynamic trajectory and select appropriate complexity indicators to characterize sector congestion according to the aircraft's self-organizing characteristics. The congestion indicators built in this way are not limited by the sector structure and are more universal. It is worth noting that the controller's primary responsibility is to ensure the smooth passage of the aircraft without conflict. Therefore, the relationship between aircraft networks is determined according to the distance between aircraft and potential conflicts, which can ensure that the complexity index of aircraft networks can indirectly reflect the controller's workload.

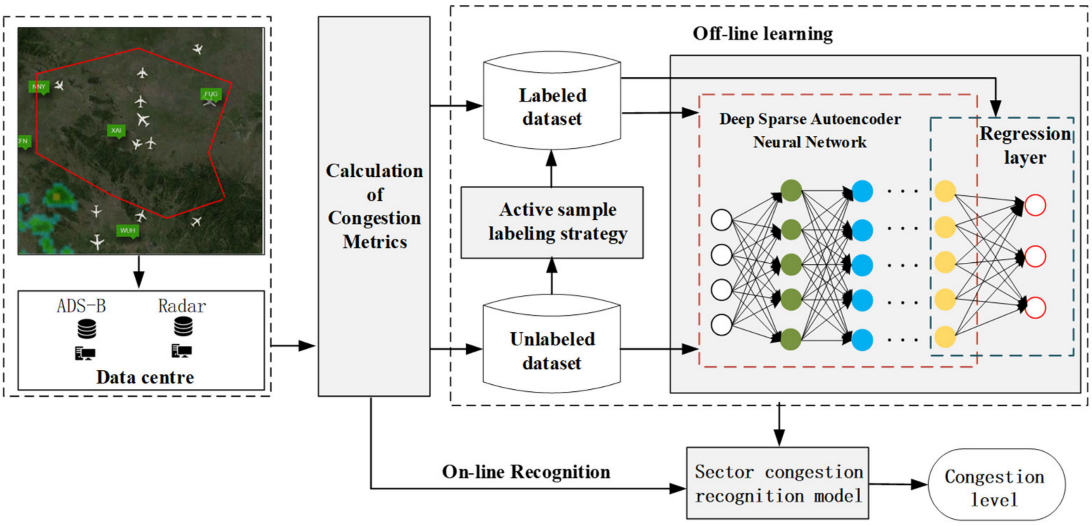

ADS-B
Radar
Data centre
Calculation of Congestion Metrics
Off-line learning
Labeled dataset
Active sample labeling strategy
Unlabeled dataset
Deep Sparse Autoencoder Neural Network
Regression layer
On-line Recognition
Sector congestion recognition model
Congestion level

Figure 1. The framework of the Proposed DAL for ATCS Congestion Recognition.

Based on the construction of indicators, we will establish a deep learning model to identify congestion. The model includes two modules, a classification model based on deep sparse auto-encoding and a sample labeling strategy. The model is trained offline using the idea of active learning. The classification model based on deep sparse auto-encoding is composed of hidden-layer sparse auto-encoding and output layer logistic regression. First, all data samples (labeled and unlabeled) are used to train the hidden layer parameters, and then the labeled sample set is used to fine-tune the model parameters. The model in this paper belongs to a type of deep neural network model, which can better characterize congestion features from a large amount of training data and improve the recognition rate. Simultaneously, according to certain selection principles, samples are selected from unlabeled samples for labeling to achieve the expected recognition rate with the least sample labeling. During the iterative process, the classification module and labeling strategy work together. Starting from the initial labeled training set, the model began to train the classification model based on deep sparse self-coding. The sample query strategy is adopted to select several samples from the unlabeled sample set. The classification model will be trained again after the newly labeled samples marked by controllers are added to the labeled sample set. The process is repeated until the model reaches the required performance.

3. Methodology

In this section, we first construct the sector congestion metric from the perspective of the aircraft state network; then we describe the deep active learning model proposed in this article, including the sector congestion recognition model and sample labeling strategy based on deep sparse autoencoding; and finally, introduce the specific implementation steps of the method in this article.

3.1. Congestion Metrics

Complex network theory is widely used in aviation networks [4,25–27]. This article will use the existing complex network theory to build a network constructed between aircraft within a sector from the perspective of a complex network system. In an aircraft network system, aircraft are nodes, and the interrelationships between aircraft constitute the edges of the network. So, what relationship should be used to build the network?

As we know, the congestion of an ATCS is ultimately reflected in the workload of the controller. The controller's primary responsibility is to ensure that the aircraft passes through without conflicts and completes the handover with neighboring sectors. When a potential conflict occurs among aircraft, the controller will resolve the conflict. The more potentially conflicting aircraft, the higher the workload of the controller. Aircraft conflict refers to the distance between aircraft or between aircraft and obstacles during the flight of aircraft that is less than the specified minimum interval. This article focuses on high-altitude ATCS. Aircraft conflict can be simplified as the distance between aircraft and aircraft is less than the minimum separation. According to the minimum safety separation stipulated by the Civil Aviation Administration, the distance between front and rear is 20 km, the horizontal distance is about 10 km, and the vertical safety interval is 300 m. According to the aircraft's minimum safety separation, an ellipsoidal protection area centered on the aircraft is formed (as shown in Figure 2). When the elliptical protection areas between two aircraft overlap, there may be potential conflicts between the aircraft as a basis for the connection between them. Furthermore, the closer the distance between aircraft, the higher the probability of collision between them.

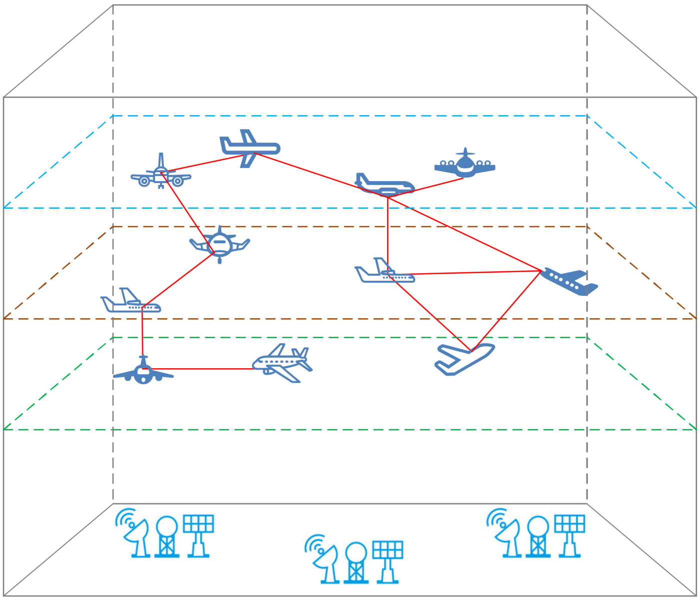


Figure 2. Aircraft network diagram.

This paper constructs a weighted aircraft network based on the overlap of elliptical protection areas between aircraft and the distance between them. If there is an overlap between the protected areas of two aircraft, we will connect them and calculate the 3D distance between them, using the reciprocal of the distance as the weight of the connection. At any time t, assuming there are n aircraft in the ATCS, then the aircraft is defined by a weighted adjacency matrix $A = \{a_{ij}\}_{n \times n}$ of size $n \times n$. If there is no connection between nodes i and j, then $a_{ij} = 0$, otherwise $a_{ij} = 1/d_{ij} > 0$, where d_{ij} represents the distance between nodes i and j. After the network is built, the network topology can be used to define the complexity of the system [28]. In terms of the actual situation of operation, seven network complexity indicators, including the number of nodes, the number of edges, the average node degree, the number of loops, the average clustering coefficient, the average path length, and the average node strength, are utilized as the evaluation indicators of sector congestion. Their corresponding meanings are shown in Table 1.

Table 1. Sector congestion evaluation indicators.

Evaluation Indicator	Description
Number of nodes	The number of nodes in the network reflects the total number of aircraft in the sector. For a fixed-size sector, the larger the number of aircraft, the more crowded the airspace.
Number of Edges	With edge-connected aircraft pairs, potential conflicts may occur over time. The more edges there are, the higher the number of aircraft pairs that may conflict in the future.
Average node degree	A coefficient reflects the average number of aircraft associated with each aircraft in the sector. The higher the value, the higher the degree of association between nodes.
Number of Loops	The number of Loops Reflects the connectivity of the entire network. The higher the value, the greater the degree of mutual interference between aircraft.
Average clustering coefficient	A coefficient describes the degree of clustering between aircraft. An enormous value indicates a higher concentration between aircraft.
Average path length	An essential indicator of the degree of aircraft dispersion in a sector, reflecting the proximity of aircraft in the aircraft network.
Average node strength	A coefficient takes into account the weighted structure of the network graph. The higher the node strength, the more aggregated the network structure is.

3.2. Deep Sparse Autoencoder for Sector Congestion Recognition

3.2.1. Sparse Autoencoder

The autoencoder belongs to a class of unsupervised learning methods, which use a neural network for feature learning. It is widely used in disease prediction [29], traffic flow prediction [30], and other fields. The autoencoder is a three-layer neural network, consisting of three parts: an input layer, a hidden layer, and an output layer. When the number of hidden layer nodes is less than the number of input layer nodes, it is called a compressive autoencoder. Otherwise, it is called a sparse autoencoder. In general, compression encoders are suitable for cases where the input data have relatively high dimensions. In contrast, sparse encoders are intended for situations where the input data has a relatively small dimension. In Section 3.1, we constructed seven indicators to represent sector congestion from different perspectives, with lower data dimensions. Therefore, this paper uses sparse autoencoders to learn features from a large number of unlabeled samples and expects to learn better feature descriptions than the original data.

Suppose m training samples $\left\{X^{(1)}, X^{(2)}, \cdots, X^{(q)}\right\}$ are known, where $X^{(i)} \in \mathbb{R}^n$. In the encoding process, the input layer $X^{(i)}$ is mapped to the hidden layer by the mapping function f to obtain the hidden layer feature vector $H = [h_1, h_2, \cdots, h_m] \in \mathbb{R}^m$, where $m \geq n$ represents the number of neurons in the hidden layer, and the calculation formula is

$$H = f(W_e X + b_e), \tag{2}$$

where W_e and b_e represent the encoding weight matrix and the encoding bias vector. In the decoding process, the hidden layer feature vector H is mapped to the output layer $\overline{X} \in \mathbb{R}^n$

by a mapping function g, where $\overline{X}$ is the reconstructed feature vector corresponding to the original input feature vector, and the calculation formula is

$$\overline{X} = g(W_d H + b_d), \tag{3}$$

where W_d and b_d represent a decoding weight matrix and a decoding bias vector. In this paper, the sigmoid function is used as the decoding mapping functions f and g.

To solve the equation to get the parameters $\theta = \{W_e, W_d, b_e, b_d\}$, the autoencoding neural network minimizes the reconstructed feature vector and the original feature vector error as the model training cost function.

$$E(W,b) = \frac{1}{2}\sum_{i=1}^{q}\left\|\overline{X^{(i)}} - X^{(i)}\right\|^2, \tag{4}$$

where $\|\cdot\|$ is the 2-norm.

Unlike a compression encoder, the number of nodes in the sparse coding hidden layer is higher than that in the input layer. If Equation (4) uses as the training target, the trained model will approach the identity mapping. This problem can then be resolved by adding a sparsity regularization term. Suppose $a_j(X)$ represents the activation degree of the hidden layer neurons j of the autoencoding neural network given the input vector X. The average neuron activity is

$$\overline{\lambda_j} = \frac{1}{q}\sum\nolimits_{i=1}^{q} a_j(X^{(i)}), \tag{5}$$

The sparsity constraint can be understood to minimize the average activation of neurons. The goal is to approximate the sparse parameter λ to $\overline{\lambda}_j$, where λ is a constant close to zero. This article defines the sparsity positive term based on Kullback–Leibler (KL) divergence:

$$\Psi(W,b) = \sum_{j=1}^{m}\left[\lambda \log\frac{\lambda}{\hat{\lambda}_j} + (1-\lambda)\log\frac{1-\lambda}{1-\hat{\lambda}_j}\right], \tag{6}$$

According to Equations (3) and (5), the total cost function can be expressed as

$$E_{sparse}(W,b) = \frac{1}{2}\sum_{i=1}^{q}\left\|\overline{X^{(i)}} - X^{(i)}\right\|^2 + \beta\sum_{j=1}^{m}\left[\lambda \log\frac{\lambda}{\hat{\lambda}_j} + (1-\lambda)\log\frac{1-\lambda}{1-\hat{\lambda}_j}\right], \tag{7}$$

where β is the parameter of the balance error term and the sparse regular term. After the optimization objective function is determined, the parameters of the sparsely encoded neural network are solved by a gradient descent algorithm.

3.2.2. Deep Sparse Autoencoder Neural Network

A deep sparse autoencoder neural network is composed of multiple sparse autoencoders, which belong to a class of semi-supervised learning neural networks. Figure 3 is a schematic structural diagram of a congestion recognition model based on a deep sparse autoencoder. The sparse encoder is used to pre-train the hidden layer of the deep neural network layer-by-layer, where the output of the previous layer encoder is used as the input of the latter layer encoder, and the output of the last sparse autoencoder is used as the input of the regression layer. Both labeled and unlabeled sample sets participate in the training of hidden layer parameters. It should be noted that, in addition to the pre-training of hidden layer parameters, the labeled sample set also includes training of the parameters of the regression layer and fine-tuning of the parameters of the hidden layer.

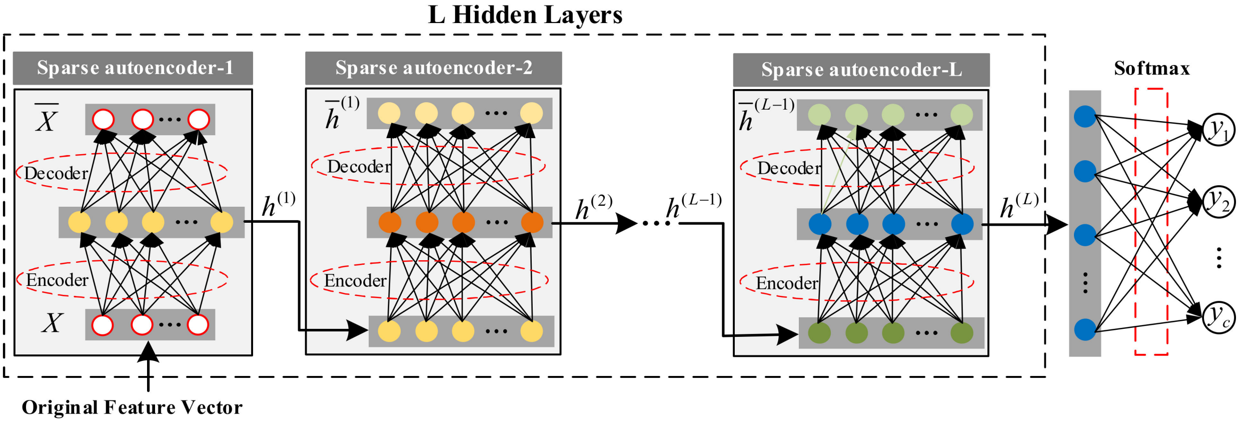


Figure 3. The architecture of a Deep Sparse Autoencoder Neural Network.

Suppose a deep neural network has L hidden layers. Let $h^{(l)}$ denote the output of the l-th encoder, $W_e^{(l)}$ and $W_d^{(l)}$ represent the encoding and decoding weight matrices of the l-th encoder, $b_e^{(l)}$ and $b_d^{(l)}$ denote the encoding and decoding bias. The feedforward transfer of encoder and decoder is performed according to the following formula:

$$h^{(l+1)} = f(W_e^{(l+1)} h^{(l)} + b_e^{(l+1)}), \tag{8}$$

$$\overline{h}^{(l+1)} = g(W_e^{(l+1)} h^{(l+1)} + b_e^{(l+1)}), \tag{9}$$

where $\overline{h}^{(l)}$ is the decoded reconstruction output of the $(l+1)$-th layer.

To realize sector congestion recognition, a regression layer needs to be added at the outermost layer. In this paper, we use the softmax layer as the output layer to achieve congestion recognition. For the outermost regression layer, take the hidden layer $h^{(L)}$ of the $L-th$ encoder as input, and pass the softmax function to get the congestion probability output of each level:

$$p(Y|\ X) = p(Y|\ h^{(L)}) = \text{softmax}\left(W^O h^{(L)}\right) = \frac{\exp(W^O h^{(L)})}{\sum_{i=1}^{m^{(L)}} \exp((W^O h^{(L)})_i)}, \tag{10}$$

where $m^{(L)}$ represents the number of neurons in the hidden layer L, W^O represents the weight matrix of the regression layer, and $(W^O h^{(L)})_i$ represents the i-th component of the vector.

From the above discussion, it can be seen that the deep autoencoder adopts unsupervised learning, while the output softmax layer is trained by supervised learning. The deep sparse autoencoder model combines the advantages of unsupervised and supervised approaches. This can improve the recognition rate of the sector's congestion level.

3.3. Active Sample Labeling Strategy

In Section 3.2, we will introduce sparse autoencoders to pre-train the hidden layers of deep neural networks from unlabeled samples and describe how to use labeled samples to train the outermost regression classification layer and adjust the hidden layers. Judging from the model pre-training process, the proposed congestion recognition model based on deep sparse autoencoders belongs to a class of semi-supervised learning techniques. For a semi-supervised deep neural network model, a large number of labeled samples are needed to obtain a relatively high recognition rate. In the problem of congestion recognition, we need air traffic controllers to help us mark samples. If samples are labeled indiscriminately, there may be sample redundancy, and a labeled sample set obtained with much workforce may not necessarily achieve the desired recognition performance.

To make the model achieve the best recognition performance with the least labeled samples, this paper introduces the idea of active learning to train a congestion recognition model [31]. The model training process is shown in Figure 1. Suppose b is a set of all data, including labeled samples and unlabeled data. Denote A and U to represent the set D consisting of labeled data and unlabeled data in the data set. First, the model is initially trained on a sample set A. A particular labeling strategy is adopted to select a representative sample from a set for labeling, add it to the labeled sample set, and then train the model's parameters through a supervised learning mechanism. This process continues until the expected performance of the model is reached. The most critical step in the entire iterative supervised learning process is the selection of a certain number of samples from the remaining unlabeled sample set. If we randomly select samples from the unmarked sample set, it will lead to data redundancy. For active learning methods, samples are usually selected based on uncertainty and difference criteria. The uncertainty criterion is to find samples with high uncertainty because they contain rich information, which helps improve the model's recognition rate. The difference criterion is that each selected sample has a big difference in characteristics from the labeled sample. In the iterative process, the labeled set's information is made more comprehensive while reducing data redundancy. According to these two criteria, this paper uses the least confidence (LC), marginal sampling (MS), and information entropy (IE) to select samples with uncertainty and substantial differences [32].

Least confidence: Select the smallest sample in the most massive activation, as shown below:

$$X_{LC}^{(i)} = \underset{X^{(i)}}{\operatorname{argmin}} \ \max_j (p(y_j|X^{(i)})), \tag{11}$$

where X_i is the i-th sample in the unlabeled sample set, and h_j^N is the activation of a unit j in top N layers.

The process is executed sample by a sample in a greedy manner to select multiple samples. The basic idea is that if the probability of the most likely label of the sample is low, then the sample classification is uncertain, so it does not consider the label probability distribution of other types of labels.

Marginal sampling: select the sample with the smallest predicted separation in the first two layers:

$$X_{MS}^{(i)} = \underset{X_i}{\operatorname{argmin}}(p(y^{(1)}|\ X^{(i)}) - p(y^{(2)}|\ X^{(i)})), \tag{12}$$

where $y^{(1)}$ and $y^{(2)}$ denote the first and second probability values identified by the deep learning model. Intuitively, if the probability of classifying the sample into its most probable class is too close to the second most probable class, it is uncertain about the sample.

Information entropy: The larger the information entropy, the higher the uncertainty of the sample, and the more information it contains. The information entropy of any sample $X^{(i)}$ is:

$$X_{Entroy}^{(i)} = -\sum_j p(y_j \Big| X^{(i)}) \log p(y_j \Big| X^{(i)}), \tag{13}$$

According to Equation (13), we can calculate the information entropy of all samples in the unlabeled sample set, and then select the first K samples with the largest information entropy for labeling.

3.4. Model Implementation

The specific implementation of the sector congestion recognition method based on deep active learning divides into an offline training phase and online recognition. In the offline training phase, the model is trained using active learning. In the online recognition phase, the flight information of all aircraft in the sector uses as the model input trained in the offline phase, and the sector congestion level is output. The algorithm flow of this method is shown in Algorithm 1.

Algorithm 1. Implementation of offline training and online recognition of the proposed method.

```
Inputs:
  • Labeled sample set $A = \{(X_A^{(1)}, y^{(1)}), (X_A^{(2)}, y^{(2)}), \cdots\}$;
  • Unlabeled sample set $U = \{X_U^{(1)}, X_U^{(2)}, \cdots\}$.
Parameters:
  • Number of hidden layers $L$;
  • Number of neurons in each hidden layer $\{N_1, N_2, \cdots, N_L\}$, where $N_l$ is the number of neurons in the $l$-th hidden layer;
  • Sparse regular parameter $\beta$;
  • The number of samples $k$ choosing from the unlabeled sample set $U$ during the iteration;
  • The maximum number of active learning iterations $T$.
Outputs:
  A sector congestion recognition model.
Off-line training:
  Step 1: The labeled sample set $A$ divides into a training set $A_1$ and validation set $A_2$ according to a certain proportion;
  Step 2: Using all samples in $A_1$ and $U$ as the input of the first sparse autoencoder, by minimizing the objective function (7), the encoding parameter $\{W_e^{(1)}, b_e^{(1)}\}$ is obtained as the network parameter between the input layer and the first hidden layer initial value. Repeat this process to get the initial values of all hidden layer network parameters;
  Step 3: Randomly initialize the regression layer network parameters $\{W_e^{(L+1)}, b_e^{(L+1)}\}$;
  Step 4: A labeled training set $A_1$ and stochastic gradient descent (SGD) are used to modify network parameters;
  Step 5: Apply the sample labeling strategy $U$ to select $k$ samples from the unlabeled sample set for labeling;
  Step 6: Add the latest labeled $k$ samples to $A_1$, and fine-tune the network parameters in the same way as in Step 4;
  Step 7: Repeat steps 4–6 until reaching the maximum iteration $T$ or the optimal performance in the validation dataset $A_2$.
Inputs:
  • The dynamic trajectory information of the aircraft in the ATCS at time t.
Outputs:
  • Sector congestion level at time t.
On-line recognition:
  Step 1: Determine whether there are overlapping areas between any two aircraft protection areas and build an aircraft network;
  Step 2: Calculate the values of seven congestion indicators $X(t) = \{x_1(t), x_2(t), \cdots, x_7(t)\}$ of the sector according to Equations (2)–(7), and obtain the feature vector of the sector congestion degree at time t;
  Step 3: The max-min normalization method is used to normalize $X(t)$, get $\hat{X}(t) = \{\hat{x}_1(t), \hat{x}_2(t), \cdots, \hat{x}_7(t)\}$, and input it to the trained model, and output the sector congestion level.
```

4. Experimental Results and Analysis

This section verifies the validity and superiority of the method in this paper. First, the data used in the experiments and the model's performance evaluation index are briefly introduced. Secondly, the optimal parameters select through experiments, and the sensitivity of the parameters is analyzed. Thirdly, the validity of the sample labeling strategy proposed in the model is verified. Finally, the effectiveness and superiority of our proposed model are confirmed by testing in multiple sectors and comparing it with mainstream methods.

4.1. Data

In the subsequent experiments of this paper, we selected three representative sectors for case analysis and marked them as “S1”, “S2”, and “S3” (as shown in Figure 4). The experimental data selection period is from 1 September 2016 to 28 September 2016 6:00–24:00. Take 1 min as the time step to sample data. The total data collected by each sector are 30,240. Each piece of data contains attribute information such as flight number, longitude, latitude, and altitude. According to the calculation formula of the congestion evaluation indicators in Section 2, we will obtain a sample set of experimental data. Each sample is represented by a 7-dimensional vector composed of the values of congestion indicators. After constructing the feature vectors of each sample, it is necessary to calibrate the samples’ congestion level. In the following experiments, we divide the congestion levels into three categories, which are represented by “1”, “2”, and “3”, which correspond to the three congestion levels of low, normal, and high, respectively. “1” and “2” are calibrated by air traffic controllers. To reduce the workload of the label and reduce the redundancy of labeled data, we use the process of active learning and use the sample selection method of basic information entropy. Each time, 300 samples are selected from the unlabeled sample set to mark the congestion level for 30 consecutive times. Each sector received 9000 labeled samples. Figure 4 shows the proportion of each level. Further, from the labeled sample set, 7000 are randomly selected as the training set, and the remaining 2000 samples are used as the test set. Unlike mainstream supervised learning methods, this article belongs to a class of active learning methods. The training set includes 7000 labeled samples and 21,240 unlabeled samples.

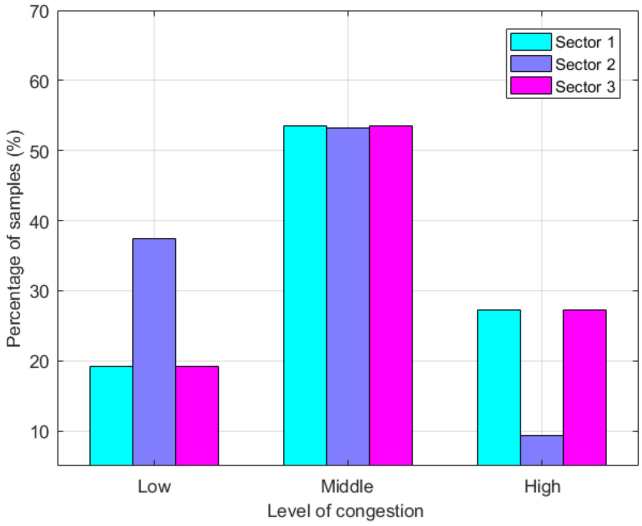


Figure 4. Percentage of samples of three congestion levels in three representative sectors.

4.2. Evaluation Metric

The proposed DAL method belongs to a class of iterative supervised learning classification methods. To this end, we use four evaluation measures: accuracy, precision, recall, and F-measure to evaluate the performance of the model. Accuracy is the percentage of correctly classified samples. Precision is the percentage of correctly classified samples in each class. The recall rate is the percentage of correctly classified samples in each category. The *F-measure* combines accuracy and recall into one evaluation index. Let $CM_{i,i}$ denote the number of samples with the congestion degree of a class i being labeled as i, and $CM_{i,j}$ denote the number of samples with the congestion degree of a class i being labeled as i, where $i, j = 1, 2, 3$. Each metric is defined as:

$$Accuracy = \frac{\sum_{i=1}^{3} CM_{i,i}}{\sum_{i=1}^{3}\sum_{j=1}^{3} CM_{i,j}} \times 100\%, \tag{14}$$

$$Precision(k) = \frac{CM_{k,k}}{\sum_{i=1}^{3} CM_{i,k}} \times 100\%, \tag{15}$$

$$Recall(k) = \frac{CM_{k,k}}{\sum_{j=1}^{3} CM_{k,j}} \times 100\%, \tag{16}$$

$$F\text{-}measure(k) = \frac{2 \times Precision(k) \times Recall(k)}{Precision(k) + Recall(k)}, \tag{17}$$

where $Precision(k)$, $Recall(k)$ and $F\text{-}measure(k)$ denote the precision, recall, and F-measure of the k-th class, respectively.

The four indices mentioned above reflect the performance of the model from different aspects. The accuracy rate reflects the overall situation of model performance. The evaluation is most effective when the number of samples is relatively uniform. It can be seen from Figure 4 that the samples with "high" and "low" congestion levels account for a relatively small proportion. Precision and recall rate measurement can reflect each category's local situation and make up for the lack of accuracy to a certain extent. The precision and recall rate tend to show the opposite trend. It can be regarded as the harmonic average of accuracy and recall, further reflecting the overall situation of the model's performance.

4.3. Parameter Settings

The parameters of DAL can divide into network structure parameters and model training parameters. The network structure parameters include the number of hidden layers and the number of hidden neurons. Model training parameters consist of the learning rate, the maximum number of iterations, the training stop threshold, and the number of samples each time in active learning. Among these two types of parameters, network structure parameters depend on specific application scenarios and training data sets, which have the most significant impact on model performance. In contrast, model training parameters have less dependence on specific application scenarios and training data sets. Some general settings are used for parameter setting. In this experiment, the initial learning rate sets to 0.01, the number of active learning iterations sets to 50, and the number of samples per acquisition parameter sets to 300.

Like all other deep learning models, this paper uses a large number of experiments to select network structure parameters. Considering the dimensionality of the sample and the total number of participating samples, the range of the number of hidden layers is $\{1, 2, 3, 4, 5, 6, 7\}$, and the number of hidden layer neurons is $\{16, 32, 64, 128, 256\}$. Within the value range, 500 groups select randomly from all possible network structure configurations for experiments. The parameter configuration corresponding to the smallest error is selected as the optimal parameter setting. The network structure's optimal parameters are: the number of hidden layers is 3, and the number of neurons corresponding to each layer is 256, 128, and 64.

After obtaining the optimal parameters of the model, the network structure parameters' sensitivity will be analyzed below, that is, investigating a specific parameter, with all other fixed parameters, and the impact of parameter changes on the accuracy of the model will be examined. Figure 5 is a graph of model accuracy as a function of the number of hidden layers. When using a hidden layer, the precision of the model is about 80%. The main reason for the lower accuracy is that there are too few weight parameters in the neural network. There is insufficient ability to represent the sample's statistical feature information. When the number of hidden layers is 3, the accuracy of the current sample labeling model has reached the maximum. Continuously increasing the number of layers leads to over-fitting the model, which affects the model's generalization ability, and the accuracy rate in the test set shows a downward trend.

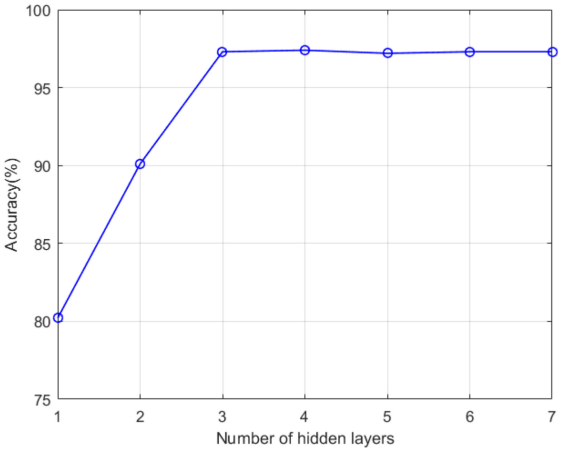


Figure 5. Impact of the number of hidden layers on model performance.

Finally, we fix the number of hidden layers and analyze the influence of the number of neurons on the performance of the model. Due to a large number of neuron combinations, it is not possible to display all of them. The neurons of each hidden layer are analyzed separately. We fix neurons in other hidden layers and examine the impact of model performance when the hidden layer changes. Figure 6 shows the experimental results of the number of neurons on the model's performance. When the number of neurons in two layers is fixed, the farther the number of neurons in the other layer is from the optimal value, the worse the model performance. It is further found that the number of neurons in each layer decreased from the first layer to the third hidden layer. For the first hidden layer, the accuracy rate increases with the number of neurons. This result is mainly due to the effect of the sparse encoder. Because the input data dimensions are small, to characterize the coupling relationship between the attributes of the input data better, the first hidden layer needs more neurons to participate in the representation. For the third hidden layer, the number of neurons is small, closely related to the congestion level's classification into three categories. Since the output layer has only three neurons, it is obtained by applying the softmax function to the third layer's hidden layer. If the number of neurons in the hidden layer is too large, softmax classification becomes difficult.

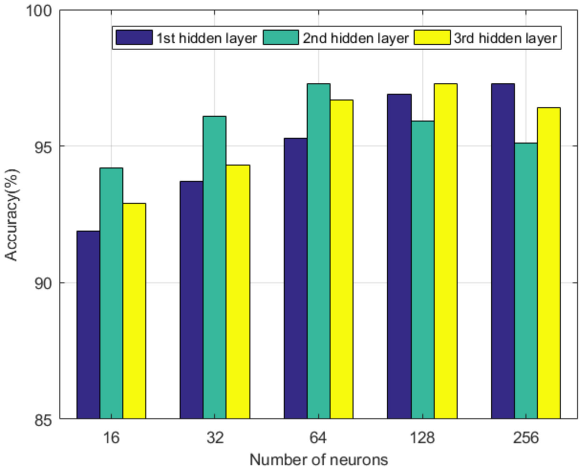


Figure 6. Effect of the number of neurons on model performance.

4.4. Performance Analysis

In this section, we first analyze the impact of sample collection strategy and initial sample size on model performance. We then compare it with mainstream methods to verify the effectiveness and superiority of the model.

4.4.1. Effect of Sample Label Strategy on Model Performance

This paper uses active learning to train the congestion recognition model based on a deep sparse autoencoder. During the model training process, each iteration needs to use a sample label strategy to select samples from the unlabeled sample set for labeling. The following experiments verify the impact of different labeling strategies on model performance. S3 was chosen as an experimental case. Each time, 300 samples were selected from the unlabeled sample set for labeling. Different sample label strategies are utilized for training the models and comparing their performance during the iteration process.

Figure 7 shows the experimental results of four label strategies, where the horizontal ordinate indicates the iteration number of the active learning process. This figure suggests that each of the four label strategies has an increasing trend with the increase of iterations. Random sampling has the worst performance. The performance of least confidence, marginal sampling, and information entropy is not much different. The main reason for this result is that the samples selected by minimum confidence, marginal sampling, and information entropy are more representative than random sampling. Thus, the labeled sample set has less data redundancy.

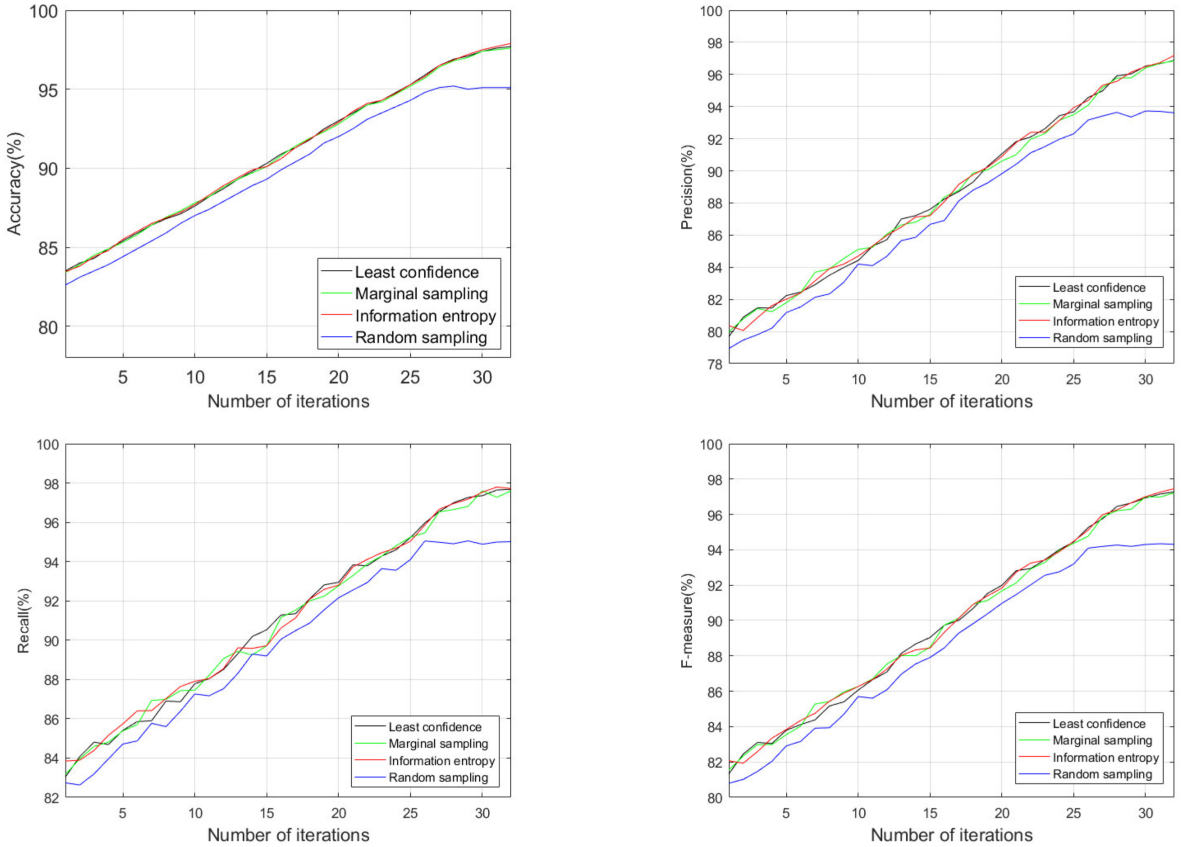


Figure 7. Performance comparison of four active label strategies.

It is further found that when the number of iterations reached 26, the curve of the label strategy based on random sampling started to flatten, and the model performance reached its peak. In contrast, the other three label strategies slowed down but still showed an upward trend. The optimal value is reached when the number of iterations is 33. After 26 iterations, the unlabeled sample set itself has much redundancy by analyzing the remaining unlabeled samples. Because samples selected by the random sampling method are highly similar to the labeled sample set, it will not help improve the recognition rate even if one continues to increase the labeled training set. Consequently, the other three label strategies can still find samples that differ from the features of the labeled set from the remaining unlabeled samples. In general, the sample labeling strategies of minimum confidence, marginal sampling, and information entropy are better than random sampling. Information entropy is slightly better than minimum confidence and marginal sampling. Given this, in the following experiments, the information entropy is utilized to label samples.

4.4.2. Effect of the Number of Initial Labeled Samples on Model Performance

The method proposed in this paper is inherently iterative semi-supervised learning. Before the iteration, the hidden layer features of the deep learning model are constructed by initial training of the model, that is, the sparse autoencoding method is used to extract the features of labeled and unlabeled samples. Further, the initial labeled samples are utilized for training the parameters of the regression layer. The purpose of the active learning method is to train a model that meets the required performance using as few labeled samples as possible. The number of initial labeled samples on the performance of the model is analyzed experimentally to provide a basis for selecting the initial labeled samples.

Figure 8 is the experimental result of the initial labeled sample number in the range of 100 to 2000. When the number of initial samples is 100, the accuracy, precision, recall, and F-measure values are 90.30%, 87.44%, 90.41%, and 88.90%, respectively. The performance of the model is relatively good. The main reason is that there are continually new labeled samples to modify the model during the active learning iteration process. Due to too few initial marking samples, the initial model's performance is weak, which is far from the ideal requirement. As the number of initial samples increases, the performance of the initial model improves, and the final model also adjusts accordingly. When the initial sample size sets to 1000, the model performance reaches the optimal state. If the number of initial labeled samples increases, the performance improvement is minimal. By considering the relationship between model performance and the initial labeling sample, it is more appropriate to select 1000 labeled samples when training the model.

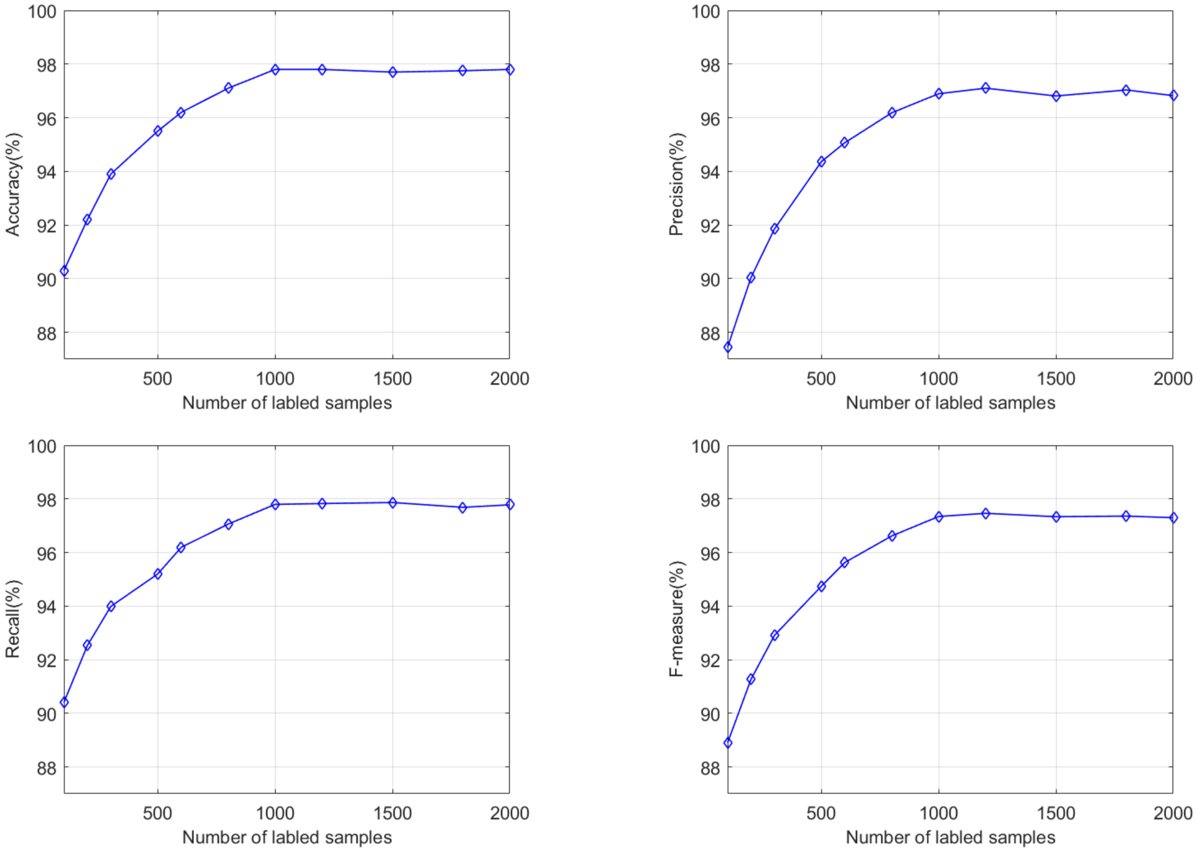


Figure 8. Impact of the number of initial labeled samples on model performance.

4.4.3. Comparative Analysis of Different Methods

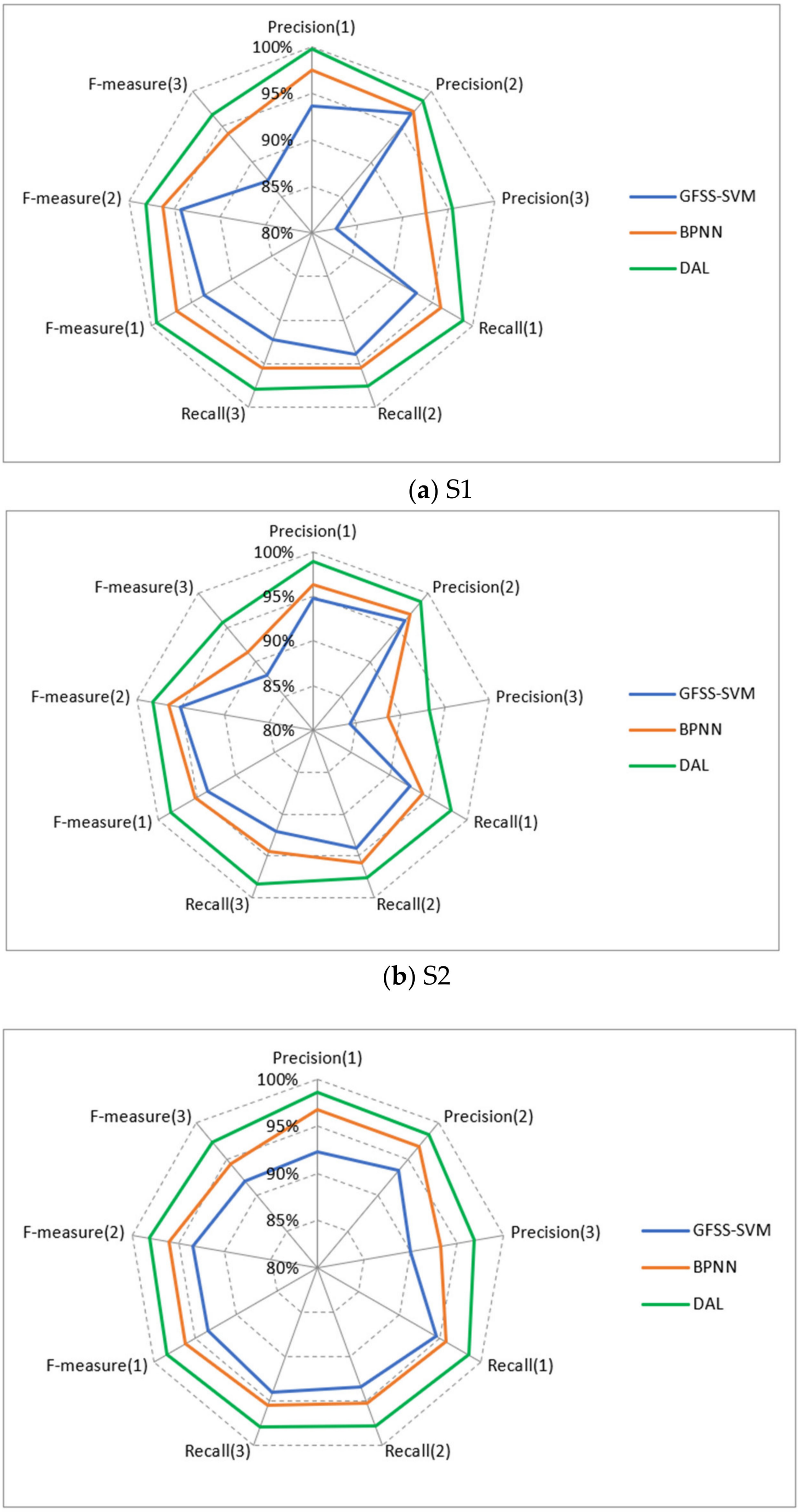
To verify the superiority of our proposed method, the deep active learning method (referred to as DAL in this experiment) is compared with PCA-BPNN [23] and GFSS-SVM [24]. Table 2 and Figure 9 show the experimental results. According to the results presented in Table 2, as a whole, DAL maintained above 97% in the four indices. In contrast, although the accuracy, precision, recall, and measurement values of PCA-BPNN and GFSS-SVM are all over 90%, PCA-BPNN and GFSS-SVM are lower than DAL by about 4.5% and 2.5%, respectively. It is because PCA-BPNN belongs to shallow feed-forward neural networks with limited ability to characterize input features, resulting in lower model performance. Similarly, the essence of GFSS-SVM is to use the SVM method for classification. SVM only supports two classifications. Multi-classification must combine multiple support vector machines. For large sample classification problems, training is time-consuming, and performance is relatively poor.

Table 2. Experimental results of different methods in three representative sectors.

Sector	Metric	GFSS-SVM	BPNN	DAL
S1	Accuracy	93.40%	95.60%	97.90%
	Precision	91.14%	93.63%	97.19%
	Recall	93.28%	95.47%	97.82%
	F-measure	92.20%	94.55%	97.50%
S2	Accuracy	93.30%	95.10%	97.80%
	Precision	91.51%	93.92%	97.02%
	Recall	93.09%	94.80%	97.94%
	F-measure	92.29%	94.36%	97.47%
S3	Accuracy	93.50%	95.40%	98.00%
	Precision	91.65%	94.14%	97.16%
	Recall	93.50%	95.44%	98.02%
	F-measure	92.57%	94.78%	97.59%

Figure 9 shows the recognition results of three representative sectors using different methods. For S2, the accuracy of GFSS-SVM and BPNN is less than 90%, and the accuracy of DAL is about 94%. The main reason for this result is the class imbalance of the sample set. Compared with congestion categories 1 and 2, congestion category 3 is heavily congested and belongs to a few extreme cases in daily operation. The proportion of samples available for model learning is less than 10%, so the trained model has a relatively low accuracy for the congestion category of 3. S3 belongs to a class of busy sectors, and the congestion is moderately severe. The proportion of samples with category three participating in training exceeds 25%. Therefore, in S3, the effect on the congestion category of 3 is better than that of S1 and S2.

In general, DAL is superior to the mainstream GFSS-SVM and BPNN in both the overall performance and each type of congestion level. Additionally, for sectors with different busy levels, DAL can still achieve a relatively high recognition rate due to the imbalance of samples, proving that our approach has a good generalization ability.



(**a**) S1

(**b**) S2

(**c**) S3

Figure 9. Radar charts of experimental results of different methods in three representative sectors.

5. Conclusions

In order to improve the generalization ability of the model and overcome the high cost of sample labeling, this paper proposes an ATCS congestion recognition method based on deep active learning, which obtains a high recognition rate with the least labeled samples. This method has strategic and tactical significance. It can not only provide a basis for the merging and addition of sectors, but also identify the congestion of sectors in real-time, and provide a more intuitive reference for controllers to provide control services. The proposed method consists of two modules, the congestion recognition model based on deep sparse autoencoding and the sample labeling strategy, which adopts an iterative semi-supervised manner.

The network structure of the congestion recognition model based on deep sparse autoencoding consists of a feature input layer, multiple hidden layers, and a regression output layer. Before iterative training of the model, we employed a sparse autoencoder to perform feature learning on all labeled and unlabeled samples, constructed a hidden layer in a stacked manner, and used labeled samples to determine the regression layer. Because the model uses only a few labeled samples in the initial training stage, the trained model has relatively poor performance. It needs to select a certain number of samples from the unlabeled sample set for labeling. To this end, we introduced three sample labeling strategies: minimum confidence, marginal sampling, and information entropy. Samples with substantial feature differences from the labeled sample set are chosen from the unlabeled sample set. This can reduce the data redundancy in the labeled sample set, thereby achieving the desired model performance with smaller label samples.

Finally, this paper used three representative sectors with different airspace structures and flight flows. The experimental results demonstrated that the number of initial labeled samples affects the performance of the model. For the three types of congestion levels, it is appropriate to choose 1000 initial label samples. At the same time, we verified the three sample label strategies. The results showed that the three sampling strategies are almost equivalent and better than the commonly used random sampling method. The information entropy is slightly better than the minimum confidence and marginal sampling. To verify the superiority of our proposed method, we compared it with two mainstream methods. The experimental results suggested that our approach is superior to the mainstream approaches in accuracy, precision, recall rate, and F-measure.

Author Contributions: Conceptualization, X.T. and Y.S.; methodology, X.T. and W.Z.; investigation, X.T. and Z.Q.; data curation, W.Z. and X.T.; supervision, Y.S.; validation, W.Z. and X.T.; writing—original draft preparation, X.T. and W.Z.; writing—review and editing, X.T. and Z.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is supported by the State Key Laboratory of Air Traffic Management System and Technology (NO: SKLATM202007); the National Natural Science Foundation of China (No: 62076126).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This study did not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rytter, A.; Skorupski, J. The concept of initial air traffic situation assessment as a stage of medium-term conflict detection. *Procedia Eng.* **2017**, *187*, 420–424. [CrossRef]
2. Bloem, M.; Gupta, P. Configuring airspace sectors with approximate dynamic programming. In Proceedings of the 27th International Congress of the Aeronautical Sciences 2010, Nice, France, 25 August 2013.
3. Cao, X.; Zhu, X.; Tian, Z.; Chen, J.; Wu, D.; Du, W. A knowledge-transfer-based learning framework for airspace operation complexity evaluation. *Transp. Res. Part C Emerg. Technol.* **2018**, *95*, 61–81. [CrossRef]
4. Jiang, X.; Wen, X.; Wu, M.; Song, M.; Tu, C. A complex network analysis approach for identifying air traffic congestion based on independent component analysis. *Phys. A Stat. Mech. Its Appl.* **2019**, *523*, 364–381. [CrossRef]

5. Prandini, M.; Piroddi, L.; Puechmorel, S.; Brázdilová, S.L. Toward air traffic complexity assessment in new generation air traffic management systems. *IEEE Trans. Intell. Transp.* **2011**, *12*, 809–818. [CrossRef]
6. Wanke, C.; Song, L.; Zobell, S.; Greenbaum, D.; Mulgund, S. Probabilistic congestion management. In Proceedings of the 6th Europe-USA ATM Seminar, Baltimore, MD, USA, 26–30 June 2005.
7. Mulgund, S.; Wanke, C.; Greenbaum, D.; Sood, N. A genetic algorithm approach to probabilistic airspace congestion management. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO, USA, 21–24 August 2006; p. 6226.
8. Sood, N.; Mulgund, S.; Wanke, C.; Greenbaum, D. A multi-objective genetic algorithm for solving airspace congestion problems. In Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, SC, USA, 20–23 August 2007; p. 6357.
9. Djokic, J.; Lorenz, B.; Fricke, H. Air traffic control complexity as workload driver. *Trans. Res. Part C Emerg. Technol.* **2010**, *18*, 930–936. [CrossRef]
10. Van Paassen, M.J.; d'Engelbronner, G.; Mulder, M. Towards an air traffic control complexity metric based on workspace constraints. In Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics, Istanbul, Turkey, 10–13 October 2010; pp. 654–660.
11. Lee, K.; Feron, E.; Pritchett, A. Air traffic complexity: An input-output approach. In Proceedings of the 2007 American Control Conference, IEEE, New York, NY, USA, 9–13 July 2007; pp. 474–479.
12. Laudeman, I.V.; Shelden, S.G.; Branstrom, R.; Brasil, C. *Dynamic Density: An Air Traffic Management Metric*; NASA: Chicago, IL, USA, 1998.
13. Kopardekar, P.; Schwartz, A.; Magyarits, S.; Rhodes, J. Airspace complexity measurement: An air traffic control simulation analysis. In Proceedings of the 7th USA/Europe Air Traffic Management R&D Seminar, Barcelona, Spain, 2–5 June 2007; pp. 295–315.
14. Chatterji, G.; Sridhar, B. Measures for air traffic controller workload prediction. In Proceedings of the 1st AIAA, Aircraft, Technology Integration, and Operations Forum, Los Angeles, CA, USA, 16–18 October 2001; p. 5242.
15. Delahaye, D.; Puechmorel, S. Air traffic complexity: Towards intrinsic metrics. In Proceedings of the Third USA/Europe Air Traffic Management R & D Seminar, Napoli, Italy, 13–16 June 2000.
16. Delahaye, D.; Puechmorel, S.; Hansman, J.; Histon, J. Air traffic complexity map based on non linear dynamical systems. *Air Traffic Control Q.* **2004**, *12*, 367–388. [CrossRef]
17. Lee, K.; Delahaye, D.; Puechmorel, S. Describing air traffic flows using stochastic programming. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Chicago, IL, USA, 10–13 August 2009; p. 6198.
18. Bilimoria, K.; Lee, H. Analysis of aircraft clusters to measure sector-independent airspace congestion. In Proceedings of the AIAA 5th ATIO and16th Lighter-Than-Air Sys Tech. and Balloon Systems Conferences, Arlington, VA, USA, 26–28 September 2005; p. 7455.
19. Surakitbanharn, C.; Wei, P.; Landry, S.; Sun, D. Evaluation of Stream Air Traffic Operations by Adapting Dynamic Density Complexity Measure. In Proceedings of the 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Indianapolis, IN, USA, 17–19 September 2012; p. 5614.
20. Brinton, C.R.; Pledgie, S. Airspace partitioning using flight clustering and computational geometry. In Proceedings of the 2008 IEEE/AIAA 27th Digital Avionics Systems Conference, St. Paul, MN, USA, 26–30 October 2008; pp. 3.B.3-1–3.B.3-10.
21. Nguyen, A.T. Identification of Air Traffic Flow Segments via Incremental Deterministic Annealing Clustering. Ph.D. Thesis, University of Maryland, College Park, MD, USA, 2012.
22. Wang, H.; Song, Z.; Wen, R.; Zhao, Y. Study on evolution characteristics of air traffic situation complexity based on complex network theory. *Aerosp. Sci. Technol.* **2016**, *58*, 518–528. [CrossRef]
23. Gianazza, D. Forecasting workload and airspace configuration with neural networks and tree search methods. *Artif. Intell.* **2010**, *174*, 530–549. [CrossRef]
24. Xi, Z.; Xianbin, C.; Kaiquan, C. Measuring air traffic complexity based on small samples. *Chin. J. Aeronaut.* **2017**, *30*, 1493–1505.
25. Lordan, O.; Sallan, J.M.; Simo, P. Study of the topology and robustness of airline route networks from the complex network approach: A survey and research agenda. *J. Transp. Geogr.* **2014**, *37*, 112–120. [CrossRef]
26. Wen, X.; Tu, C.; Wu, M.; Jiang, X. Fast ranking nodes importance in complex networks based on LS-SVM method. *Phys. A Stat. Mech. Its Appl.* **2018**, *506*, 11–23. [CrossRef]
27. Zanin, M.; Lillo, F. Modelling the air transport with complex networks: A short review. *Eur. Phys. J. Spec. Top.* **2013**, *215*, 5–21. [CrossRef]
28. Costa, L.D.F.; Rodrigues, F.A.; Travieso, G.; Boas, P.R.V. Characterization of complex networks: A survey of measurements. *Adv. Phys.* **2007**, *56*, 167–242. [CrossRef]
29. Parisot, S.; Ktena, S.I.; Ferrante, E.; Lee, M.; Guerrero, R.; Glocker, B.; Rueckert, D. Disease prediction using graph convolutional networks: Application to Autism Spectrum Disorder and Alzheimer's disease. *Med. Image Anal.* **2018**, *48*, 117–130. [CrossRef] [PubMed]

30. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F.-Y. Traffic Flow Prediction with Big Data: A Deep Learning Approach. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 865–873. [CrossRef]
31. Ranganathan, H.; Venkateswara, H.; Chakraborty, S.; Panchanathan, S. Deep active learning for image classification. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3934–3938.
32. Wang, D.; Shang, Y. A new active labeling method for deep learning. In Proceedings of the 2014 International Joint Conference on Neural Networksf (IJCNN), Beijing, China, 6–11 July 2014; pp. 112–119.