Flight Delay Prediction Based on Aviation Big Data and Machine Learning

Guan Gui, *Senior Member, IEEE*, Fan Liu, *Student Member, IEEE*, Jinlong Sun, *Member, IEEE*, Jie Yang, *Member, IEEE*, Ziqi Zhou, *Student Member, IEEE*, and Dongxu Zhao, *Student Member, IEEE*

***Abstract*—Accurate flight delay prediction is fundamental to establish the more efficient airline business. Recent studies have been focused on applying machine learning methods to predict the flight delay. Most of the previous prediction methods are conducted in a single route or airport. This paper explores a broader scope of factors which may potentially influence the flight delay, and compares several machine learning-based models in designed generalized flight delay prediction tasks. To build a dataset for the proposed scheme, automatic dependent surveillance-broadcast (ADS-B) messages are received, pre-processed, and integrated with other information such as weather condition, flight schedule, and airport information. The designed prediction tasks contain different classification tasks and a regression task. Experimental results show that long short-term memory (LSTM) is capable of handling the obtained aviation sequence data, but overfitting problem occurs in our limited dataset. Compared with the previous schemes, the proposed random forest-based model can obtain higher prediction accuracy (90.2% for the binary classification) and can overcome the overfitting problem.**

***Index Terms*—Flight delay prediction, ADS-B, machine learning, LSTM neural network, random forest.**

I. Introduction

AIR traffic load has experienced rapid growth in recent years, which brings increasing demands for air traffic surveillance system. Traditional surveillance technology such as primary surveillance radar (PSR) and secondary surveillance radar (SSR) cannot meet requirements of the future dense air traffic. Therefore, new technologies such as automatic dependent surveillance broadcast (ADS-B) have been proposed, where flights can periodically broadcast their current state information, such as international civil aviation organization (ICAO) identity number, longitude, latitude and speed [1]. Compared with the traditional radar-based schemes, the ADS-B-based scheme is low cost, and the corresponding ADS-B receiver (at 1090 MHz or 978 MHz) can be easily connected to personal computers [2]. The received ADS-B message along with other collected data from the Internet can constitute a huge volumes of aviation data by which data mining can support military, agricultural, and commercial applications. In the field of civil aviation, the ADS-B can be used to increase precision of aircraft positioning and the reliability of air traffic management (ATM) system [3]. For example, malicious or fake messages can be detected with the use of multilateration (MLAT) [1], allowing open, free, and secure visibility to all the aircrafts within airspace [2]. Thus, the ADS-B provides opportunity to improve the accuracy of flight delay prediction which contains great commercial value.

The flight delay is defined as a flight took off or arrived later than the scheduled time, which occurs in most airlines around the world, costing enormous economic losses for airline company, and bringing huge inconvenience for passenger. According to civil aviation administration of China (CAAC), 47.46% of the delays are caused by severe weather, and 21.14% of the delays are caused by air route problems. Due to the own problem of airline company or technical problems, air traffic control and other reasons account for 2.31% and 29.09%, respectively. Recent studies have been focused on finding a suitable way to predict probability of flight delay or delay time to better apply air traffic flow management (ATFM) [4] to reduce the delay level. Classification and regression methods are two main ways for modeling the prediction model. Among the classification models, many recent studies applied machine learning methods and obtained promising results [5]–[7]. For instance, L. Hao *et al.* [8] used a regression model for the three major commercial airports in New York to predict flight delay. However, several reasons are restricting the existing methods from improving the accuracy of the flight delay prediction. The reasons are summarized as follows: the diversity of causes affecting the flight delay, the complexity of the causes, the relevancy between causes, and the insufficiency of available flight data.

In [6], a public dataset named VRA [9] was used to compare the performance of several machine learning models including k-nearest neighbors (K-NN) [10], support vector machines (SVM) [11], naive Bayes classifier, and random forests for predicting flight delay, and achieved the best accuracy of 78.02% among the four schemes. However, the air route information (e.g., traffic flow and size of each route) was not considered

Manuscript received September 24, 2019; revised October 23, 2019; accepted November 14, 2019. Date of publication November 18, 2019; date of current version January 15, 2020. This work was supported in part by the Project Funded by the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant TC190A3WZ-2, in part by the National Natural Science Foundation of China under Grant 61901228, in part by the Jiangsu Specially Appointed Professor Program under Grant RK002STP16001, in part by the Summit of the Six Top Talents Program of Jiangsu under Grant XYDXX-010, in part by the Program for High-Level Entrepreneurial and Innovative Talents Introduction under Grant CZ0010617002, and in part by the 1311 Talent Plan of the Nanjing University of Posts and Telecommunications. The review of this article was coordinated by Prof. J. Wang. *(Corresponding authors: Jinlong Sun; Jie Yang.)*

The authors are with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: guiguan@njupt.edu.cn; 1018010402@njupt.edu.cn; sunjinlong@njupt.edu.cn; jyang@njupt.edu.cn; 1218012005@njupt.edu.cn; 1218012004@njupt.edu.cn).

Digital Object Identifier 10.1109/TVT.2019.2954094

in their model, which prevents them from obtaining higher accuracy. In [4], D. A. Pamplona *et al.* built an artificial neural network with 4 hidden layers, and achieved the highest accuracy of 87%; their proposed model suggested that the day of the week, block hour, and route has great influence on the flight delay. This model did not consider meteorological factors, so there is room for improvement. Y. J. Kim *et al.* [12] proposed a model with two stage. The first stage is to predict day-to-day delay status of specific airport by using deep RNN model, where the status was defined as an average delay of all flights arrived at each airport. The second stage is a layered neuron network model to predict the delay of each individual flight using the day-to-day delay status from the first stage and other information. The two stages of the model achieved accuracies of 85% and 87.42%, respectively. This study suggested that the deep learning model requires a great volumes of data. Otherwise, the model is likely to end up with poor performance or overfitting [13].

To address the problems in ATM, the received ADS-B messages can be coupled with weather information, traffic flow information, and other information to constitute an aviation data lake, which provides an opportunity to find a better approach to accurately predict the flight delay. Meanwhile, machine learning have made great progress and have obtain amazing performance in many domains, such as internet of things [14], heterogeneous network traffic control [15], autonomous driving [16], unmanned aerial vehicle [17]–[21], wireless communications [22]–[28], and cognitive radio [29]–[31]. The above successes motivate us to apply machine learning in the field of air traffic data analytic applications [12], [32]. Compared with the scenarios in wireless communications, the air traffic also faces dynamic environment and can be affected by many dynamic factors. Therefore, it is worthy to apply machine learning models for the flight delay prediction by making full use of the aviation data lake. By combining the advantages of all the available different data, we can feed the entire dataset into specific deep learning models, which allows us to find optimal solution in a larger and finer solution space and gain higher prediction accuracy of the flight delay.

Our work benefits from considering as many factors as possible that may potentially influence the flight delay. For instance, airports information, weather of airports, traffic flow of airports, traffic flow of routes. The contributions of this paper can be summarized as follows:

- We explore a broader scope of factors which may potentially influence the flight delay and quantize those selected factors. Thus we obtain an integrated aviation dataset. Our experimental results indicate that the multiple factors can be effectively used to predict whether a flight will delay.
- Several machine learning based-network architectures are proposed and are matched with the established aviation dataset. Traditional flight prediction problem is a binary classification task. To comprehensively evaluate the performance of the architectures, several prediction tasks covering classification and regression are designed.
- Conventional schemes mostly focused on a single route or a single airport [4], [6], [12]. However, our work covers all routes and airports which are within our ADS-B platform.

II. Methodology

A. ADS-B Message Based Aviation Big Data Platform

Air traffic flow are increasing rapidly in recent years. The number of aircraft will be doubled if the growth of general purpose aircraft and unmanned aerial vehicle resulting from civilian demand are considered [33]. Traditional technology of aircraft positioning and tracking relies on radar system (i.e., PSR and SSR), whose performance can be greatly reduced when the aircrafts are in transoceanic and remote area due to the limited working range of radar. In order to achieve the goal of global tracking and monitoring for the aircrafts especially flights, international telecommunications union (ITU) has reached a consensus on the implementation of integrated space-space-earth positioning and tracking system by using the ADS-B.

ADS-B system is a communication and surveillance integrated system for air traffic management (ATM) where flights periodically broadcast location and other information on the same frequency band [34]. Compare to the traditional surveillance technology, the ADS-B system can obtain higher location precision, lower cost of deployment, and simpler maintain system. The ADS-B system overcomes the effects brought by clouds and low visibility, and thus improves the surveillance ability and enhances the flight security.

The ADS-B system can be divided into ADS-BOUT subsystem and ADS-BIN subsystem. In the ADS-BOUT subsystem, flight transmitters periodically send their own information (e.g., identity, position, velocity) to other flights and ground stations. And in the ADS-BIN subsystem, the flight receivers receive out-message from other flights and the ground stations. Since the free visibility of the ADS-B messages, ADS-B receivers for personal use can be connected to standard computer [2], which allows the ADS-B messages be collected for the sake of our study.

Fig. 1 shows a typical deployment of the ADS-B-based ground station. As shown in Fig. 1, an ADS-B omnidirectional antenna directly connects to the ground station, and constantly receives ADS-B messages from flights within a diameter of 300 km. The integrated aviation surveillance ground station conducts the ADS-B signals received by antenna, and visualizes the flight path, flight numbers and other information. Hence, the ground stations continuously receiving ADS-B messages can upload data to a central cloud server.

The proposed architecture of the aviation big data platform is shown in Fig. 2. In our big data platform, 14 ADS-B ground stations have been deployed at major cities of eastern China. The saved ADS-B message contains the following flight information: date of the message, time-stamped of the message, ICAO identity number, flight number, longitude, latitude, height, heading, velocity, and other information. And other data from the Internet such as weather and flight schedule are also collected by the centralized cloud server for our study. Fig. 2 also depicts

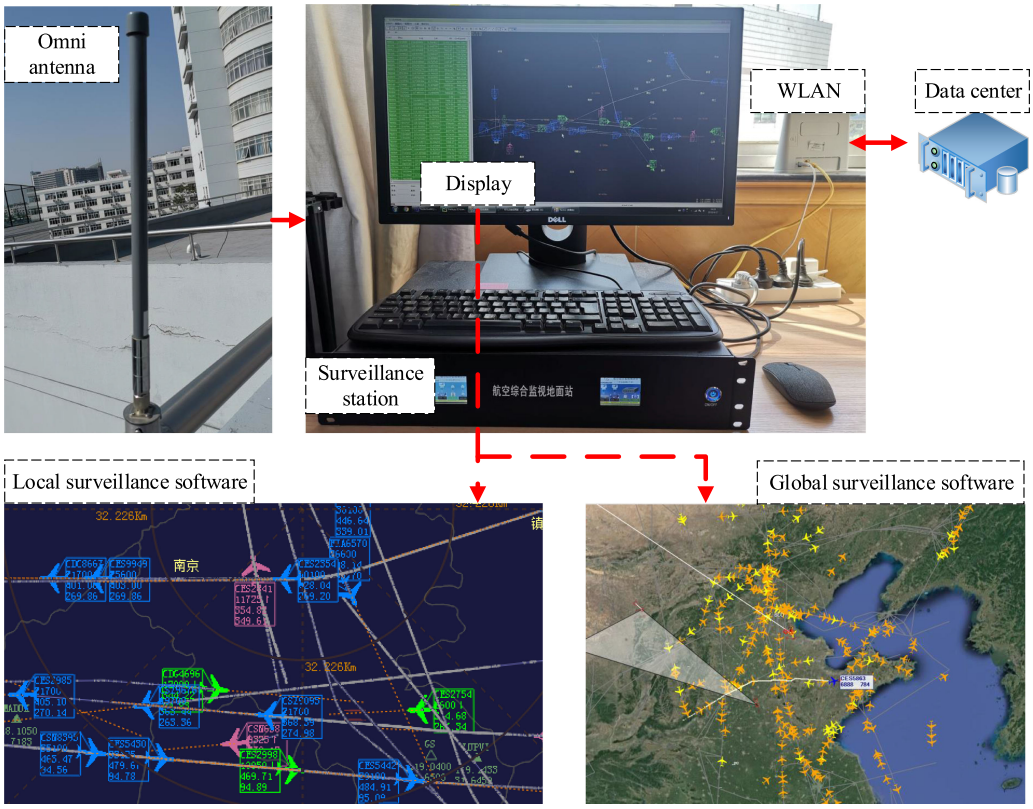


Fig. 1. Deployment of an ADS-B ground station: ADS-B omnidirectional antenna (left top), integrated aviation surveillance ground station (right top), and software running in the station and visualizing flight trajectories (left bottom, right bottom).

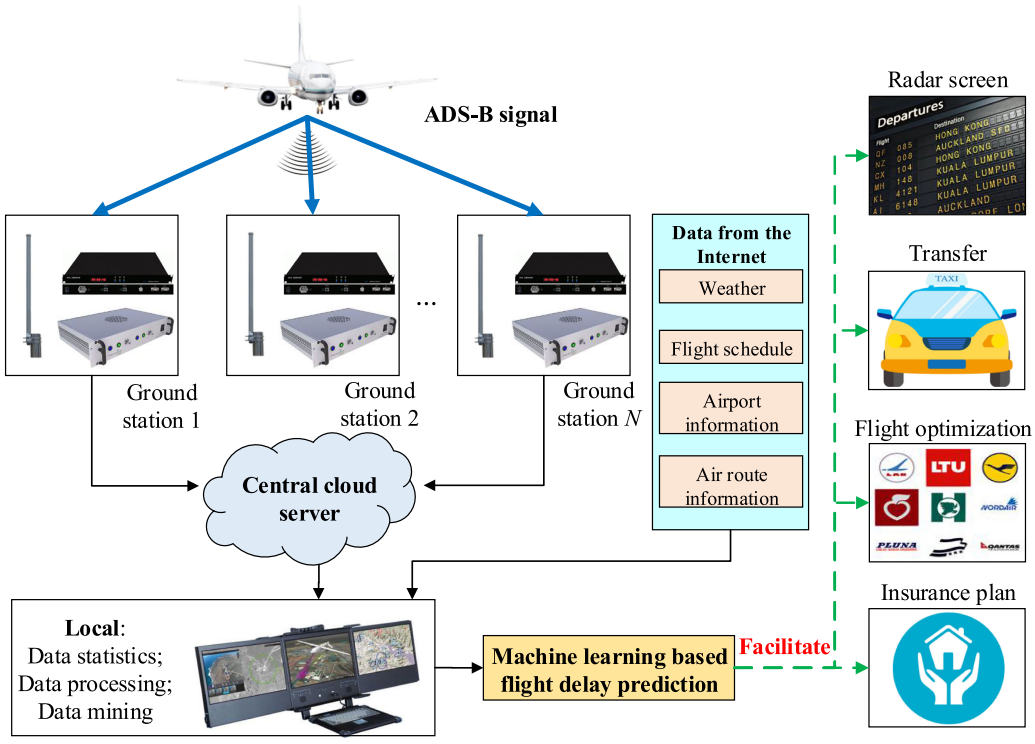


Fig. 2. Architecture of aviation big data platform. The ground station continuously receives ADS-B messages and uploads to central cloud server. Data from the Internet is collected and integrated with the ADS-B messages by the central server.

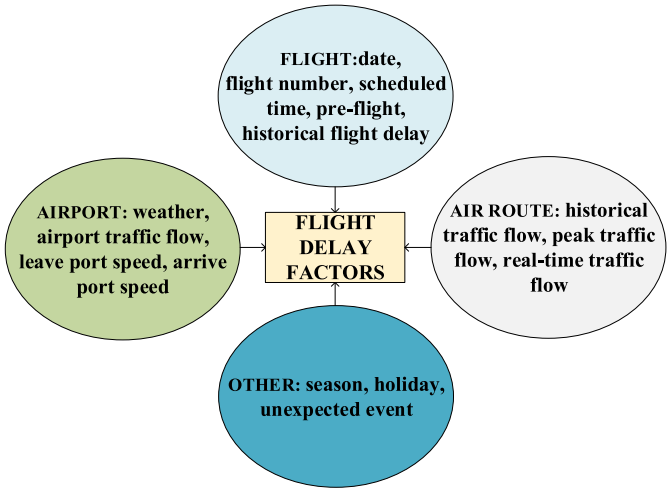


Fig. 3. Flight delay causes are divided into four main categories.

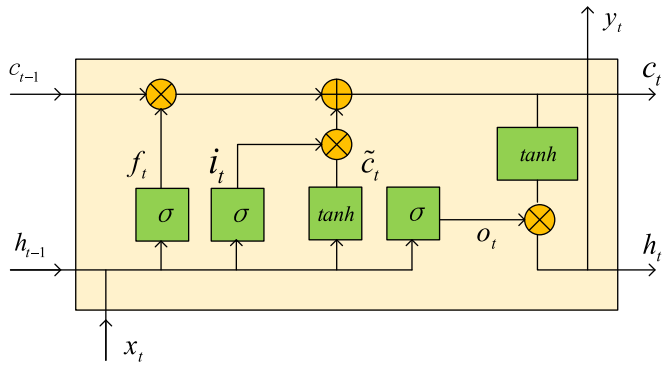


Fig. 4. LSTM cell which is structured as four gates named forget gate, candidate gate, input gate and output gate.

services and applications which may be facilitated by the proposed architecture.

B. Flight Prediction Formulation

According to CAAC, a flight arriving more than 15 minutes later than scheduled is considered as a delayed flight. As shown in Fig. 3, the causes of flight delay can be divide into four main categories: In this paper, we extract some attributes that affecting the flight delay, and form them as an input vector $\mathbf{x}$ in the proposed model. Before giving the definition of $\mathbf{x}$, we first give the definition of weather vector $\mathbf{w}$, time-stamped vector $\mathbf{t}$, and flight schedule vector $\mathbf{s}$ as follows:

$$\mathbf{w} = [w_1, w_2, w_3, w_4, w_5, w_6] \tag{1}$$

where w_1, w_2, w_3, w_4, w_5 and w_6 are the weather condition of departure airport, weather condition of destination airport, wind direction of departure airport, wind power of departure airport, wind direction of destination airport, and wind power of destination airport, respectively.

$$\mathbf{t} = [t_1, t_2, t_3, t_4] \tag{2}$$

where t_1, t_2, t_3 and t_4 are the day of month, month, day of week, and season, respectively.

$$\mathbf{s} = [s_1, s_2, s_3, s_4] \tag{3}$$

where s_1, s_2, s_3 and s_4 are the departure airport, scheduled time of departure, destination airport, scheduled time of arrival, respectively.

Thus, we can define the input vector $\mathbf{x}$ as

$$\mathbf{x} = [d, n_1, n_2, f, \mathbf{w}, \mathbf{t}, \mathbf{s}] \tag{4}$$

where d denotes the date of the flight, n_1 and n_1 are ICAO identity number and flight number, respectively. f denotes the traffic flow of the route. $\mathbf{w}$, $\mathbf{t}$ and $\mathbf{s}$ are the weather vector, time-stamped vector, and flight schedule vector as we have mentioned before. As shown in Fig. 3, the input vector $\mathbf{x}$ contains vast majority of factors affecting the flight delay. However, some factors such as unexpected events and pre-order flights cannot be measured and quantized straightforwardly, so these factors are not considered in our model.

Since the relationship between the flight delay and the factors is generally complicated and nonlinear, it is difficult to establish an accurate mathematical prediction model. Meanwhile, deep learning has made great progress and has obtain amazing performance in many domains, such as wireless communications [23]–[25] and cognitive radio [29], [30]. Therefore, it is worthy to apply deep learning methods to the flight delay prediction task.

The main goal in this paper is to obtain an acceptable accuracy of the flight delay prediction. If we handle the prediction as a regression task, the metric $\hat{\delta}_T$ can be used to indicate the delay in minutes. And if the prediction is handled as a classification task, the vector $\mathbf{l}$ consisting probability of each class can be defined as

$$\mathbf{l} = [p_1, p_2, \ldots, p_n] \tag{5}$$

where p_n is the predicted probability of a specific delay class and n refers to the index of the class.

$$\hat{\delta}_T = h_1(x) \tag{6}$$

$$\mathbf{l} = h_2(x) \tag{7}$$

As shown in (6) and (7), the proposed model attempts to fit two hidden functions h_1 and h_2 for mapping $\mathbf{x}$ into $\hat{\delta}_T$ (for a regression model) and $\mathbf{l}$ (for a classification model), respectively.

III. Proposed LSTM Based Method

Recurrent neural network (RNN) is suited for sequential data [35] and has been widely applied in the field of natural language processing (NLP). Specifically, long short-term memory (LSTM) network is one of most powerful RNNs with more complex cell structure [36], and overcomes the gradient vanishing problem in RNNs. We consider an input sequence $x = [x_1, x_2, \ldots, x_T]$, and the LSTM model computes the hidden state $h = [h_1, h_2, \ldots, h_T]$ for each time step and gets output sequence $y = [y_1, y_2, \ldots, y_T]$ [35].

Fig. 4 shows that the single LSTM cell [37] contains four gate structures, namely forget gate, candidate gate, input gate and output gate. The hidden layer function H is implemented

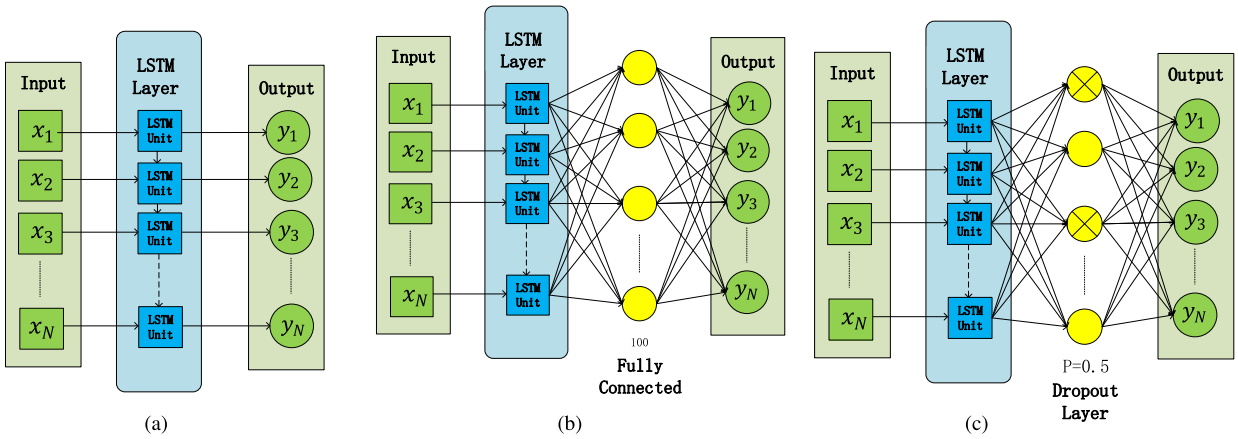


Fig. 5. Three different network architectures based on LSTM: a standard architecture (a); a fully-connected layer is added (b); a dropout layer is added (c).

by the following functions represented by the gates [35]:

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right) \quad (8)$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \quad (9)$$

$$\tilde{c}_t = \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \quad (10)$$

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \quad (11)$$

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right) \quad (12)$$

$$h_t = o_t \tanh\left(c_t\right) \quad (13)$$

where σ is the activation function sigmoid, and f_t is the forget gate that decides what information should be thrown away from the past cell state. i_t and $\tilde{c}_t$ are the input gate and candidate gate, respectively, and they together decide what information should be added from the current cell state to the new cell state [36]. o_t is the output gate deciding the degree of revealing the cell state.

The structure stacked by LSTM cells just looks like a gradient freeway allowing gradient uninterruptedly flow from upstream to downstream during back-propagation [38]. As can be seen in Fig. 5, three different network architectures based on the LSTM cells are implemented in this paper. The first is a standard LSTM-based architecture, the second is combined with a fully-connected layer, the third is integrated with an additional dropout layer.

To create a dataset from the ADS-B messages for training and testing process, data pre-processing has been done:

Data Cleaning: Because the proposed model is based on a supervised learning method which refers to a process of adjusting the hyper-parameters with a dataset with known labels, it is significant to obtain a clean dataset and label the samples elaborately. First, the ADS-B message dataset is divided by flight number and date, and then is filtered according to the flights height (data below 1500 feet are ignored). Thus, data arrays of every flight on every day are gotten, and the items in each array are sorted by time. The last item of each sorted array is considered as the arrival data of each flight on a specific day. When the arrival data is extracted, the delay of the flight can be calculated. Since the delay prediction results can be presented with higher resolution, the labels can be divided into more than two classes (delayed and not delayed). Therefore, we consider no-delay as Class 0, delay within one hour as Class 1, delay within two hours as Class 2, and delay over 2 hours as Class 3. After every items of each array are labeled, a preliminary dataset is established.

Data Integration: An ADS-B message includes basic information of a flight such as ICAO identity number, position, and velocity. However, it does not include weather information, traffic flow, departure airport, destination airport, scheduled departure time or scheduled arrival time. To create an integrated dataset for our study, the weather information of airports are collected from a website [39]. The weather information includes wind direction, wind speed, and weather condition (sunshine, rain, snow etc.), which has potential influence on the flight delay. Other information such as scheduled flight time, departure airport, and destination airport are collected from Ctrip [40] by searching the identity number of each flight. Another significant factor of the flight delay is traffic flow control. A flight will not allowed if the traffic flow of the route reaches a critical value. To compute the traffic flow, we define each air route as a corridor with a width of 10 km that connects destination and departure airports, and we count the flights hourly from the received ADS-B messages within each corridor.

Data Transformation and Balancing: Table I shows the attributes used as input variables of the proposed model in detail. Since some attributes such as weather condition cannot be fed into the training model straightforwardly, the idea of data quantization is natural. Furthermore, data normalization is also implemented in the process of data quantization, which improves the convergence speed and prediction accuracy of the model. The attributes of airports such as Nanjing is transformed into digit code by mapping the English letters (A to Z) into digit numbers (1 to 26), and the ICAO identity numbers and flight numbers are transformed similarly. To enrich the content of the dataset, the time stamps are split into four parts: Month, day of month, day of week, and season. Other attributes with limited categories such as weather condition, wind direction, and wind speed are encoded by using enumeration encoding. For example, the categories of the weather condition include sunshine, light rain, moderate rain, and hard rain are coded as

TABLE I
ATTRIBUTES INTEGRATED IN THE DATASET TO TRAIN THE PROPOSED MODEL

Attributes	Input variables
Flight	Departure airport Destination airport Flight number ICAO identity number Departure scheduled time Arrival scheduled time
Time-stamped	Day of month Month Season Day of week
Weather	Weather condition Wind direction Wind speed
Airport	Airport name
Air route	Traffic flow of air route

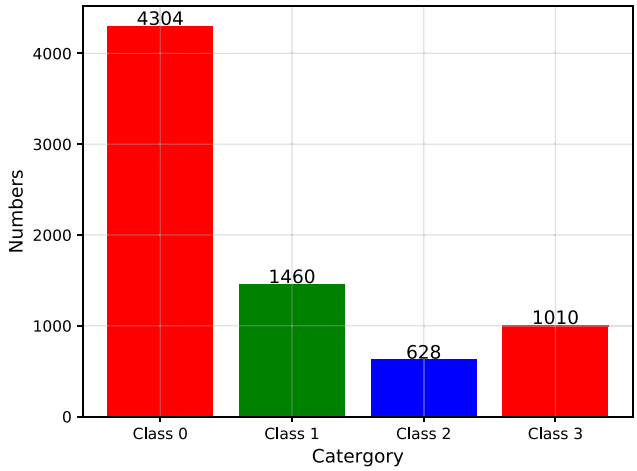


Fig. 6. Distribution of the flight delay class, where the no-delay class (Class 0) is the majority class occupying 58.15% of the samples, while Class 2 just occupies 8.48% samples.

0, 1, 2, and 3, respectively. As shown in Fig. 6, the statistical distribution of the flight delay are imbalanced, and the no-delay class (i.e., the major class) occupies approximately 58.15% of all the labels. However, models trained by this imbalanced dataset may result in unfair decisions: samples in small classes may be incorrectly classified as the major class, but the overall prediction accuracy still looks good. Therefore, random under-sampling strategy is implemented in this model to obtain fairer prediction results. However, the defect of this strategy is quiet obvious; some important information concerning the majority class may also be removed and cannot be learnt by the model [41].

IV. PROPOSED RANDOM FOREST-BASED MODEL

Random forest is an ensemble learning method, which uses decision tree [42] as as sub-classifiers, and introduces random attributes selection into the decision tree. The algorithm of building a random forest-based classifier is as follows:

Algorithm 1: TrainRandomForest.

Input: data set $D = \{d_1, d_2, ..., d_m\}$, attribute set $A = \{a_1, a_2, ..., a_d\}$, the number of decision tree N.
Output: trained random forest classifier
for *i in 1,2,,N* **do**
  D_i = subset(D);
  A_j = subset(A);
  T_i = TreeGenerate(D_i,A_j);
  put back D_i;
end
classifier = ensemble(T_1,T_2,...,T_N);
return classifier;

Algorithm 2: TreeGenerate.

Input: data set $D = \{d_1, d_2, ..., d_m\}$, attribute set $A = \{a_1, a_2, ..., a_d\}$.
Output: root node of a decision tree
n = new node;
if d *in* D == *same class* C **then**
  n = leaf node C;
  return n;
end
if A == *empty OR* d *in* D == *same attribute value* **then**
  C = majority class of D;
  n = leaf node C;
  return n;
end
a_* = best attribute(A);
for *attribute value* a_*^k *in* a_* **do**
  n = new brach B_k;
  D_v = data set{attribute value == a_*^k in D};
  if D_v == *empty* **then**
    C = majority class of D;
    n = leaf node C;
    return n
  end
  else
    B_k = TreeGenerate(D_v,A/{a_*});
    return B_k;
  end
end

It should be noted that D_i was generated by random under-sampling the full training dataset and A_j were random selected from full attribute set [6] for building T_i.

As you can see in Fig. 7, after the random forest classification architecture is constructed, the ensemble classifier uses the most voted result of the N sub-classifiers as its prediction. The ability of each sub-classifier and the independence of the sub-classifiers jointly improve the model accuracy. The randomness in the under-sampled training dataset for generating subsets, selecting attributes, and building sub-classifiers avoids the probability of overfitting greatly [43].

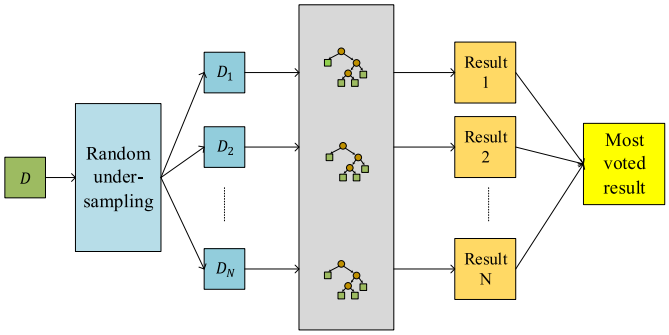


Fig. 7. The train and test process of random forest, N decision trees were constructed by using N subsets respectively, most voted result was considered as final result.

TABLE II
PERFORMANCE ANALYSIS OF DIFFERENT NETWORK ARCHITECTURES WHEN THE MEMORY DEPTH IS 3

Architecture	Training accuracy and mini-batch	Testing accuracy and elapsed time	Epochs
Standard	88%	36.53%	3000
	10%	14min13sec	
Added fully-connected	88%	36.1%	3000
	10%	14min17sec	
Added dropout	67%	41.5%	3000
	20%	12min35sec	

V. EXPERIMENT RESULTS

In our experiment, the LSTM-based and random forest-based prediction models are trained separately. And then we make a comparison between the two flight delay prediction models in terms of prediction accuracy and other concerns.

A. Result of LSTM-Based Classifier

Before training the LSTM-based classifier, the dataset was divided by flight and date. Considering several successive days of each flight and the corresponding data items, we can establish input sequences for the LSTM-based classifier with specific memory depth. We use MATLAB to construct the classifier. In this paper, the total number of the input sequences is 1542 if we set the memory depth as 3, and the number is 1186 for a memory depth of 5, and it means 942 if the memory depth is set as 7. To find the best parameters when training the classifier, a grid-search strategy was used and serval parameters were selected and adjusted as follows:

- Epochs: 1500, 2000, and 3000.
- Mini-batch size: 5% of training dataset, 10% of training dataset, and 20% of training dataset.
- Memory depth: 3, 5, and 7.

Table II–IV show the parameters and accuracies of different network architectures when the memory depth is set as 3, 5 and 7, respectively. As can be seen, the standard LSTM-based architecture and the architecture with an additional fully-connected layer present similar prediction accuracy on the testing dataset, which is lower than the accuracy of the architecture with an additional dropout layer. Meanwhile, the architecture with an additional dropout layer requires larger mini-batch size and longer training time to converge. By comparing the different memory depth, we can see that the longer memory depth means the higher accuracy on the training dataset. It suggests that the flight delay has strong correlation with time. Fig. 8 shows the training process of the standard architecture. And as shown in Fig. 8, when the iterations increases, the accuracy for the training dataset increases and the loss drops. Unfortunately, all the accuracies in Table II∼IV for the testing dataset get worse dramatically compared to those for the training dataset. It suggests the models end up overfitting, even if in the architecture where an additional dropout layer is added. The dropout layer devitalizes certain artificial neurons with a specific probability, which can reduce independency between neurons and simplify the network. Hence, the LSTM-based architecture with dropout layer alleviates the overfitting problem, to some extent. As a result, the testing accuracy of the architecture with a dropout layer is better than that of the

TABLE III
PERFORMANCE ANALYSIS OF DIFFERENT NETWORK ARCHITECTURES WHEN THE MEMORY DEPTH IS 5

Architecture	Training accuracy and mini-batch	Testing accuracy and elapsed time	Epochs
Standard	99%	32.8%	1500
	5%	7min6sec	
Added fully-connected	99%	34.3%	1500
	10%	5min48sec	
Added dropout	80%	37.1%	3000
	20%	10min7sec	

TABLE IV
PERFORMANCE ANALYSIS OF DIFFERENT NETWORK ARCHITECTURES WHEN THE MEMORY DEPTH IS 7

Architecture	Training accuracy and mini-batch	Testing accuracy and elapsed time	Epochs
Standard	99%	31.2%	1500
	10%	4min57sec	
Added fully-connected	99%	31.7%	1500
	10%	5min19sec	
Added dropout	88%	33.1%	3000
	20%	9min12sec	

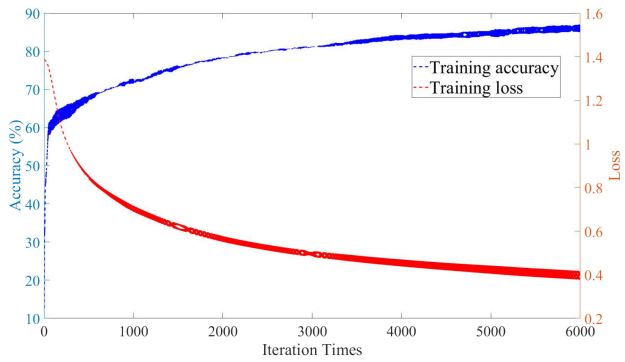
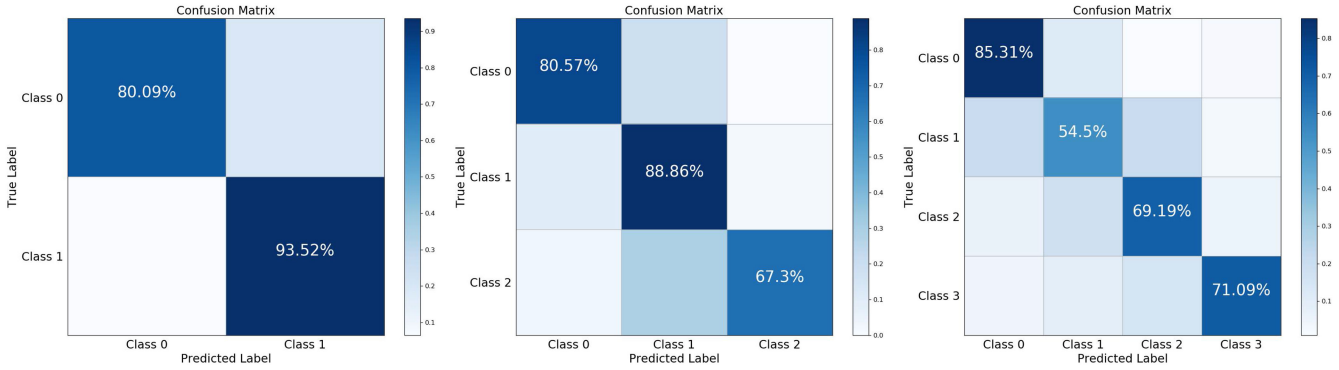


Fig. 8. Training process of the standard LSTM-based architecture, where the blue line is the accuracy (%) for the training dataset versus iteration, and the orange line is the corresponding loss versus iteration.



(a) (b) (c)

Fig. 9. Confusion matrix of the classification tasks based on random forest: (a) binary classification task; (b) three categories classification task; (c) four categories classification task.

architecture with a fully-connected layer. Compared to the standard LSTM, the architecture involving fully-connected layers increases the model complexity and introduces worse overfitting problem. As mentioned before, deep learning model requires a great volumes of data. The overfitting problem may due to the limited training data, which just consists 771 and 593 sequences (the training data set is half the size of full data set) for memory depth of 3 and memory depth of 5, respectively.

B. Result of Random Forest-Based Classifier

To make a fair comparison between different models, the classification accuracy is considered as the evaluation metric. And the definition of the classification accuracy is as follows:

$$accuracy = \frac{1}{n}\sum_{i=1}^{n} I\left(f\left(x_i\right), y_i\right) \tag{14}$$

$$I\left(y_1, y_2\right) = \begin{cases} 1 & y_1 = y_2 \\ 0 & y_1 \neq y_2 \end{cases} \tag{15}$$

where f is the classification model, $f(x_i)$ denotes the predicted label of sample i, y_i is the true label of sample i, n is the total number of all the samples in dataset, and I is the judge function.

The full dataset we generate includes 5761 items and their period is from December 2018 to May 2019. The no-delay class is the major class which includes 3368 items. We use scikit-learn [44] to construct the random forest-based classifier. To adjust the training parameters, a grid-search strategy was used and serval parameters were selected as follows:

- Numbers of estimators: 20∼150.
- Max depth of decision tree: 3∼20.
- Max feature of decision tree: log2, auto, and sqrt.

The max feature is the maximum number of features that can be used in a sub-classifier of the random forest. And the auto means that the max feature is the square root (sqrt) of the total number of the features.

Classification tasks considering binary categories, three categories, and four categories were implemented, respectively.

TABLE V
PERFORMANCE ANALYSIS OF DIFFERENT RANDOM FOREST-BASED TASKS

Task	Accuracy	Estimators	Max depth	Max feature
Binary	90.2%	35	12	'auto'
Three	81.4%	70	13	'auto'
Four	70.0%	55	10	'auto'

After an under-sampling process, each class has the same number of sequences in the four categories classification task, and the number is 248. To make a fair comparison, the binary and three categories classification tasks were also executed on the same under-sampled dataset. And to apply this dataset to other classification tasks, an operation named merge class were performed as follows: Delays within 1 hour and 2 hours were merged into the same class for the three categories classification; delays within 1 hour and 2 hours and delays over 2 hours were merged into the same class for the binary classification. And the dataset was random split into training dataset and testing dataset for training and testing process, respectively. Table V shows the best parameters and accuracy of different classification tasks using the testing dataset. This paper and previous work in [6], [12] share the same definition of accuracy, and the datasets in the previous work contain more attributes (26 and 19 versus 17). Therefore, the experimental results fairly show that our binary classification task obtained the best accuracy of 90.2%, which exceeds the classification accuracy of the previous work. And the accuracy of the three and four categories classification are 81.4% and 70.0%, respectively.

To further evaluate the performance in detail, confusion matrices of the above classification tasks are shown in Fig. 9. The X label of the confusion matrix is the true label of each sample, and the Y label is the predicted label. The darker main diagonal of the matrix indicates higher prediction accuracy. Therefore, the confusion matrix shown in Fig. 9(a) shows that the overall accuracy of binary classification task is over 90%. It suggests that the weak ability in distinguishing delays within 1 hour from delays within 2 hours, to a great extent, degrades the accuracy of the four categories classification.

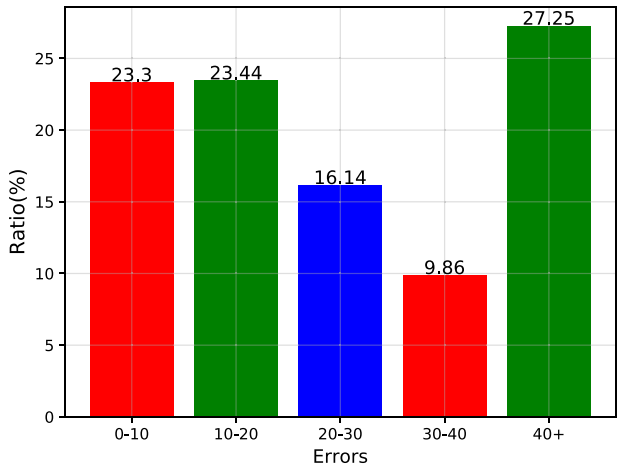


Fig. 10. Statistics histogram of predicted delay minute errors by using the regression model.

For the regression task, the predicted delay minutes are output by the trained model. We evaluate the prediction performance by comparing the predicted delay minutes and real delay minutes, and correspondingly, histogram of the errors are depicted in Fig. 10. As shown in Fig. 10, errors which are less than 10 minutes, between 10 and 20 minutes, between 20 and 30 minutes, between 30 and 40 minutes, and over 40 minutes account for 23.3%, 23.44%, 16.14%, 9.86%, and 27.25%, respectively.

C. Comparison of the Proposed Architectures

The random forest-based architecture obtained a testing accuracy of 90.2% for the binary classification, which is considered a promising result and demonstrate the strong ability of the ensemble learning. For the four categories classification task, the random forest-based architecture obtained an testing accuracy of 70.0%. While the LSTM-based network with memory depth of 7 can obtain accuracy of 99% for the training dataset, but all the LSTM-based architectures present poor performance on the testing dataset. The LSTM-based architectures captured the time correlation of the flight delay while the random forest-based architectures did not, however, the former ones ended up with overfitting due to the limited training dataset. It suggests that the generalization ability of the random forest-based method is stronger than the LSTM-based one in our dataset. However, there are reasons to believe that the LSTM-based method can obtain promising performance if one or more years data can be utilized and the overfitting problem can be overcame.

VI. Conclusion and Future Work

In this paper, random forest-based and LSTM-based architectures have been implemented to predict individual flight delay. The experimental results show that the random forest-based method can obtain good performance for the binary classification task and there are still room for improving the multi-categories classification tasks. The LSTM-based architecture can obtain relatively higher training accuracy, which suggests that the LSTM cell is an effective structure to handle time sequences. However, the overfitting problem occurred in the LSTM-based architecture still needs to be solved. In summary, the random forest-based architecture presented better adaptation at a cost of the training accuracy when handling the limited dataset. In order to overcome the overfitting problem and to improve the testing accuracy for multi-categories classification tasks, our future work will focus on collecting or generating more training data, integrating more information like airport traffic flow, airport visibility into our dataset, and designing more delicate networks.

References

[1] M. Leonardi, "ADS-B anomalies and intrusions detection by sensor clocks tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 5, pp. 2370–2381, Oct. 2019, doi: 10.1109/TAES.2018.2886616.
[2] Y. A. Nijsure, G. Kaddoum, G. Gagnon, F. Gagnon, C. Yuen, and R. Mahapatra, "Adaptive air-to-ground secure communication system based on ADS-B and wide-area multilateration," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3150–3165, May 2015.
[3] J. A. F. Zuluaga, J. F. V. Bonilla, J. D. O. Pabon, and C. M. S. Rios, "Radar error calculation and correction system based on ADS-B and business intelligent tools," in *Proc. IEEE Int. Carnahan Conf. Secur. Technol.*, 2018, pp. 1–5.
[4] D. A. Pamplona, L. Weigang, A. G. de Barros, E. H. Shiguemori, and C. J. P. Alves, "Supervised neural network with multilevel input layers for predicting of air traffic delays," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2018, pp. 1–6.
[5] S. Manna, S. Biswas, R. Kundu, S. Rakshit, P. Gupta, and S. Barman, "A statistical approach to predict flight delay using gradient boosted decision tree," in *Proc. IEEE Int. Conf. Comput. Intell. Data Sci.*, 2017, pp. 1–5.
[6] L. Moreira, C. Dantas, L. Oliveira, J. Soares, and E. Ogasawara, "On evaluating data preprocessing methods for machine learning models for flight delays," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2018, pp. 1–8.
[7] J. J. Rebollo and H. Balakrishnan, "Characterization and prediction of air traffic delays," *Transp. Res. Part C, Emerg. Technol.*, vol. 44, pp. 231–241, 2014.
[8] L. Hao, M. Hansen, Y. Zhang, and J. Post, "New York, New York: Two ways of estimating the delay impact of New York airports," *Transp. Res. Part E, Logistics Transp. Rev.*, vol. 70, pp. 245–260, 2014.
[9] ANAC, "The Brazilian national civil aviation agency," 2017. [Online]. Available: http://www.anac.gov.br/
[10] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1774–1785, May 2017.
[11] J. Sun, Z. Wu, Z. Yin, and Z. Yang, "SVM-CNN-based fusion algorithm for vehicle navigation considering a typical observations," *IEEE Signal Process. Lett.*, vol. 26, no. 2, pp. 212–216, Feb. 2018.
[12] Y. J. Kim, S. Choi, S. Briceno, and D. Mavris, "A deep learning approach to flight delay prediction," in *Proc. IEEE Digit. Avionics Syst. Conf.*, 2016, pp. 1–6.
[13] Y. Cong, J. Liu, B. Fan, P. Zeng, H. Yu, and J. Luo, "Online similarity learning for big data with overfitting," *IEEE Trans. Big Data*, vol. 4, no. 1, pp. 78–89, Mar. 2017.
[14] F. Tang, Z. M. Fadlullah, B. Mao, and N. Kato, "An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: A deep learning approach," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5141–5154, Dec. 2018.
[15] N. Kato *et al.*, "The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective," *IEEE Wireless Commun.*, vol. 24, pp. 146–153, Jun. 2017.
[16] J. Wang, J. Liu, and N. Kato, "Networking and communications in autonomous driving: A survey," *IEEE Commun. Surv. Tuts.*, vol. 21, no. 2, pp. 1243–1274, Apr.–Jun. 2019.
[17] Y. Kawamoto, H. Nishiyama, N. Kato, F. Ono, and R. Miura, "Toward future unmanned aerial vehicle networks: Architecture, resource allocation and field experiments," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 94–99, Feb. 2019.
[18] D. Takaishi, Y. Kawamoto, H. Nishiyama, N. Kato, F. Ono, and R. Miura, "Virtual cell-based resource allocation for efficient frequency utilization in unmanned aircraft systems," *IEEE Trans. Veh. Technol.*, vol. 67, no. 4, pp. 3495–3504, Apr. 2018.

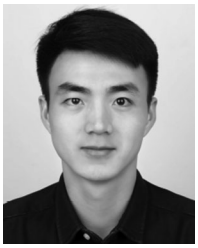
[19] F. Tang, Z. M. Fadlullah, N. Kato, F. Ono, and R. Miura, "AC-POCA: Anti-coordination game based partially overlapping channels assignment in combined UAV and D2D based networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1672–1683, Feb. 2018.
[20] M. Liu, J. Yang, and G. Gui, "DSF-NOMA: UAV-assisted emergency communication technology in a heterogeneous Internet of Thing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5508–5519, Jun. 2019.
[21] W. Shi *et al.*, "Multi-drone 3D trajectory planning and scheduling in drone assisted radio access networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8145–8158, Aug. 2019.
[22] G. Gui, Y. Wang, and H. Huang, "Deep learning based physical layer wireless communication techniques: Opportunities and challenges," *J. Commun.*, vol. 40, no. 2, pp. 19–23, 2019.
[23] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, Apr. 2019.
[24] J. Sun, W. Shi, Z. Yang, J. Yang, and G. Gui, "Behavioral modeling and linearization of wideband RF power amplifiers using BiLSTM networks for 5G wireless systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10348–10356, Nov. 2019, doi: 10.1109/TVT.2019.2925562.
[25] G. Gui, H. Huang, Y. Song, and H. Sari, "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8440–8450, Sep. 2018.
[26] H. Huang, Y. Song, J. Yang, and G. Gui, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, Mar. 2019.
[27] N. Cheng *et al.*, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.
[28] H. Huang, Y. Peng, J. Yang, W. Xia, and G. Gui, "Fast beamforming design via deep learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1065–1069, Jan. 2020.
[29] M. Liu, T. Song, J. Hu, J. Yang, and G. Gui, "Deep learning-inspired message passing algorithm for efficient resource allocation in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 641–653, Jan. 2018.
[30] M. Liu, T. Song, and G. Gui, "Deep cognitive perspective: Resource allocation for NOMA-based heterogeneous IoT with imperfect SIC," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2885–2894, Apr. 2018.
[31] Q. Peng, A. Gilman, N. Vasconcelos, P. C. Cosman, and L. B. Milstein, "Robust deep sensing through transfer learning in cognitive radio," *IEEE Wireless Commun. Lett.*, vol. 9, no. 1, pp. 38–41, Jan. 2020.
[32] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. Big Data*, vol. 2, no. 1, pp. 1–21, 2015.
[33] M. Strohmeier, M. Schafer, V. Lenders, and I. Martinovic, "Realities and challenges of nextgen air traffic management: The case of ADS-B," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 111–118, May 2014.
[34] W. Wang, R. Wu, and J. Liang, "ADS-B signal separation based on blind adaptive beamforming," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6547–6556, Jul. 2019.
[35] L. Jin, S. Li, and B. Hu, "RNN models for dynamic matrix inversion: A control-theoretical perspective," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 189–199, Jan. 2018.
[36] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, "LSTM-based flight trajectory prediction," in *Proc. Int. Joint Conf. Neural Netw.*, 2018, pp. 1–8, 2018.
[37] S. O. Sahin and S. S. Kozat, "Nonuniformly sampled data processing using LSTM networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1452–1461, May 2019.
[38] T. Ergen and S. S. Kozat, "Online training of LSTM networks in distributed systems for variable length data sequences," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 5159–5165, Oct. 2018.
[39] TQHB, "Tian Qi Hou Bao." [Online]. Available: www.tianqihoubao.com/lishi
[40] CTRIP, "Flight schedule." [Online]. Available: flight.ctrip.com/domestic/schedule
[41] J. Hu, H. Yang, M. R. Lyu, I. King, and A. M.-C. So, "Online nonlinear AUC maximization for imbalanced data sets," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 882–895, Aug. 2018.
[42] X. Wang, "Decision-tree-based relay selection in dualhop wireless communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6212–6216, Jun. 2019.
[43] Y. Wang, S.-T. Xia, Q. Tang, J. Wu, and X. Zhu, "A novel consistent random forest framework: Bernoulli random forests," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3510–3523, Aug. 2018.
[44] J. V. den Bossche, "scikit-learn 0.21.2," 2019. [Online]. Available: https://scikit-learn.org/stable/

Guan Gui (SM'18) received the Dr.Eng. degree in information and communication engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2012. From 2009 to 2014, he was with the Wireless Signal Processing and Network Laboratory (Prof. Adachi Laboratory), Department of Communications Engineering, Graduate School of Engineering, Tohoku University, as a Research Assistant as well as Postdoctoral Research Fellow. From 2014 to 2015, he was an Assistant Professor with the Department of Electronics and Information System, Akita Prefectural University. Since 2015, he has been a Professor with the Nanjing University of Posts and Telecommunications, Nanjing, China. He has authored and coauthored more than 200 international peer-reviewed journal/conference papers. His research interests include deep learning, compressive sensing, and advanced wireless techniques. Dr. Gui was the recipient of Member and Global Activities Contributions Award in IEEE ComSoc and seven best paper awards, i.e., International Conference on Electronic Information and Communication Technology 2019, Advanced Hybrid Information Processing 2018, Communications, Signal Processing, and Systems 2018, ICNC 2018, ICC 2017, ICC 2014, and VTC 2014-Spring. He was also selected for Jiangsu Specially Appointed Professor in 2016, Jiangsu High-Level Innovation and Entrepreneurial Talent in 2016, Jiangsu Six Top Talent in 2018, and Nanjing Youth Award in 2018. He was an Editor for the *Security and Communication Networks* (2012∼2016). He has been the Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY since 2017, IEEE ACCESS since 2018, *Physical Communication* since 2019, *KSII Transactions on Internet and Information Systems* since 2017, and *Journal on Communications* since 2019.

Fan Liu (S'18) is currently working toward the master's degree in signal processing for wireless communications with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China. His research interest includes deep learning and its applications.

Jinlong Sun (M'18) received the M.S. and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 2014 and 2018, respectively. He is currently an Assistant Professor with the Nanjing University of Posts and Telecommunications, Nanjing, China. His current research interests include signal processing for wireless communications, machine learning, and integrated navigation systems.

Jie Yang (M'17) received the B.Sc., M.Sc., and Ph.D. degrees in communication engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2003, 2006, and 2018, respectively. She is currently an Assistant Professor with the Nanjing University of Posts and Telecommunications.



Ziqi Zhou (S'18) is currently working toward the master's degree in signal processing for wireless communications with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China. His research interest includes deep learning and its applications.

Dongxu Zhao (S'18) is currently working toward the master's degree in signal processing for wireless communications with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China. His research interest includes deep learning and its applications.