A deep sequence-to-sequence method for accurate long landing prediction based on flight data

Zongwei Kang1,2  |  Jiaxing Shang1,2  |  Yong Feng1,2  |  Linjiang Zheng1,2  |  Qixing Wang1,2  |  Hong Sun3  |  Baohua Qiang4,5  |  Zhen Liu6

1 College of Computer Science, Chongqing University, Chongqing, China
2 Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing University, Chongqing, China
3 Flight Technology and Flight Safety Research Base, Civil Aviation Flight University of China, Guanghan, China
4 Guangxi Key Laboratory of Optoelectronic Information Processing, Guilin University of Electronic Technology, Guilin, China
5 Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, China
6 Chongqing Adtech Science and Technology Co., Ltd

Abstract
In civil aviation industry, runway overrun is a typical landing safety incident concerned by both airlines and authorities. Among various contributing factors to the runway overrun incident, long landing plays an important role. However, existing studies for long landing prediction mainly depend on classic machine learning methods and handcrafted features. As a result, they usually require much expert knowledge and provide unsatisfactory results. To address these problems, this paper proposes an innovative deep sequence-to-sequence model which utilizes QAR (Quick Access Recorder) data for accurate long landing prediction. Specifically, to cope with the high heterogeneity of QAR dataset, a data pre-processing procedure is first proposed which includes data cleaning, interpolation and normalization steps. Second, to avoid the noises incurred by too many QAR parameters and relieve the reliance on expert experience, the GBDT (gradient boosting decision trees) model is employed to choose the most relevant parameters as features. Then a CNN-LSTM and TG-attention encoder-decoder architecture is proposed to accurately predict future aircraft ground speed and radio height sequences, based on which the touchdown distance can be finally calculated. Experimental results on a large QAR dataset with 44,176 A321 flights validate effectiveness of the proposed method.

1 | INTRODUCTION
In civil aviation industry, flight safety is a vital important issue concerned by both airlines and authorities. According to the Boeing statistics [1], the landing phase, which only occupies 4% of the overall flight time, contributes to 48% of the fatal accidents during 2007 to 2016. This phase also has the largest accident probability and severity, contributing to the highest risk of flight accidents [2]. Therefore, early warning during the landing phase is of great significance to avoid accidents and improve flight safety. Runway overrun, which is defined as situations when aircraft extends beyond the end of runway, is a typical safety incidents concerned by pilots and airlines during this phase. Based on work [3] and a recent study [4], runway overrun...
Accidents mainly owe to the following risk factors: long landing, unstable approach, high approach speed, visual approach, etc. Long landing, in which case an aircraft consumed too much runway length when it touched the ground, that is, the touchdown distance significantly exceeds a given threshold, can largely contribute to runway overrun accidents and it is regarded as the main risk factor among all the risk factors. Therefore, studying the long landing events and predict the touchdown distance play a vital important role to avoid the overrun accidents and ensure flight safety.

Nowadays, many commercial aircraft are equipped with the Quick Access Recorder (QAR)\(^1\) sensory system, which is able to collect comprehensive flight parameters in real-time, including aircraft status parameters (e.g. ground speed, altitude, weight, pitch), pilot operational parameters (e.g. pitch/roll command), environmental parameters (e.g. temperature, wind speed/direction) etc. Therefore, the QAR dataset provide a digital reproduction of the flight dynamics. Based on these data, the flight process of aircraft can be analysed to assist airline decision making and provide flight safety warnings. With the recent advance in big data methodologies and applications, many researchers began to investigate flight safety with QAR dataset, such as detecting abnormal flights [5, 6], investigating risk models [7, 8], predicting safety incidents [9, 10] etc. However, existing works are mainly based on traditional machine learning methods or classic statistical models, such as the ANOVA analysis and multiple regression method [11], the SVM-based landing risk model [2] etc. These studies face two major disadvantages: First, they require a lot of expert experience to design and handcraft features, which can be tedious and challenging. Second, classic machine learning models may give unsatisfactory results since they are incapable of capturing deep temporal information from the QAR dataset. Recently, Tong et al. [12, 13] used LSTM-based deep learning models for ground speed and hard landing prediction, but their methods cannot predict the trend of near future. Meanwhile the traditional LSTM-based deep models are difficult to capture both spatial inter dependence among different QAR parameters and temporal interdependence among different time steps. As a result, the traditional LSTM-based deep models cannot provide early flight safety warnings, making themselves impractical in real applications.

Motivated by the above observations, this paper proposes an innovative CNN-LSTM and TG-attention sequence-to-sequence model for aircraft landing distance prediction from QAR time series data. Unlike traditional end-to-end deep learning models, we take an indirect approach which first predicts future ground speed and radio height sequences, and then calculates the landing distance according to the predicted sequences. It takes advantage of CNN-LSTM deep neural network to effectively capture the temporal interdependence and spatial features of QAR parameter sequences. Besides, TG-attention (Temporal and Global attention) is proposed to capture the impact of local and global trends of the input sequence on the future output sequence, thereby significantly improves the prediction accuracy. Specifically, the method proposed in this paper consists of the following steps: First, to cope with the high heterogeneity of QAR parameters, a preprocessing procedure is proposed which includes data cleaning, Lagrange interpolation and normalization. Second, to avoid the noises incurred by too many QAR parameters and relieve the reliance on expert experience, the GBDT model is employed to choose the most relevant parameters as features. Finally, the proposed CNN-LSTM and TG-attention encoder-decoder architecture is used to predict future ground speed and radio height sequences. Based on the predicted ground speed and radio height sequences, the touchdown point can be located by inspecting when the predicted radio height values reduce to zero, and the landing distance can be calculated by accumulating over the ground speed sequence. Extensive experiments are conducted on a QAR dataset which include 44,176 Airbus A321 aircraft flights and the results show that the proposed CNN-LSTM and TG-attention encoder-decoder model achieves significantly lower prediction error (in RMSE, MAE and MAPE) than state-of-the-art baselines.

In sum, the contributions of this paper are as follows:

- An innovative deep CNN-LSTM and TG-attention sequence-to-sequence model is proposed for accurate touchdown distance prediction, where the CNN-LSTM module can capture both temporal and spatial interdependence, while the TG-attention module can capture the impact of both local and global trends.
- Instead of handcrafting the features, the GBDT model is employed to conduct correlation analysis the most relevant flight parameters are selected as features. Experimental results show that GBDT-based features give better performance than experience-based features.
- The proposed method is evaluated on large QAR dataset, and the experimental results show that the proposed CNN-LSTM and TG-attention encoder-decoder model achieves significantly lower prediction error than state-of-the-art baselines.

The following parts are organized as follows. Section 2 gives a brief introduction about related works on flight safety and QAR dataset analytics. In Section 3, the method for QAR dataset pre-processing is introduced. Section 4 gives detailed illustration of the CNN-LSTM and TG-attention encoder-decoder model. Experiments and discussions are shown in Section 5. The paper is concluded in Section 6.

### 2 RELATED WORK

A lot of research works have been devoted to investigate flight safety. This thesis briefly reviews the related works that are based on flight data.

Some research works have focused on landing safety analysis using traditional risk models or machine learning methods.

---

\(^1\) All abbreviations in this paper have been listed in Table 1.
Wang et al. [11] conducted ANOVA (analysis of variance) on QAR dataset to see how the pilot flare operation affects landing incidents such as long touchdown distance. Through a step-wise linear regression, they found that five parameters can be used as significant features for landing distance prediction, including flare height, flare time, ground speed, vertical acceleration and descent rate. However, some of their features (e.g. flare time) can be calculated only when the aircraft has already touched the ground, which makes the model unable to provide long landing warnings in a proactive manner. Khatwa et al. [14] investigated two landing incidents, that is, runway overrun and landing undershoot, and divide the multiple risks factors into three categories, including flight control, environment, and pilot operations. Gui et al. [15] proposed a random forest-based model to predict the flight delay, and the classification accuracy of the model reached 90.2%. In [4], Eduardo et al. considered the runway overrun risk in different scenarios and proposed a BN (Bayesian network) [16, 17] based risk assessment model. Lukas et al. [18] investigated the runway overrun and loss of control accidents by combining physical models of aircraft dynamics with statistical dependence analysis, based on which they developed advanced FDM algorithms for safety management.

Some other works focused on predicting or analysing safety incidents with deep learning methods. For sequential data analysis, many studies employed the two classic RNN (recurrent neural networks) based models, that is, LSTM and GRU, due to their power in capturing the temporal interdependence of sequential data [19]. In [12], Tong et al. first performed stationarity randomness tests on the QAR dataset, and then proposed a LSTM model for aircraft landing speed prediction. They used random forest algorithm and PCA for feature selection and dimensionality reduction, and then feed the features to train the LSTM model. In another work [13], they also applied the LSTM model for hard landing incident prediction. In [20], Janakiraman developed a precursor mining method based on GRU deep network and multiple instance learning. This method was able to identify events from the multidimensional time series that are correlated with the high-speed exceedance incidents. Chen et al. [21] proposed a LSTM deep model for landing pitch prediction so as to reduce the tail strike risk. One deficiency of the LSTM and GRU models is that they only generate predictions for the next moment and cannot provide predictions for the trend of near future. For sequence generation, Wu et al. [22] proposed a LSTM model based on hierarchical attention for Chinese lyric generation. Belhadi et al. [23] used the RNN model for urban long-term traffic flows prediction based on multiple data sources (e.g., weather and contextual information). Kang et al. [24] proposed a deep sequence-to-sequence method for aircraft landing ground speed sequence prediction. In [25], a hybrid model is proposed by Xu et al. who combined deep belief network and linear regression for time series data forecasting. Future air-ground integrated vehicular network (AGVN) is a vital part of the expected ubiquitous communication system. Sun et al. [26] proposed a conceptual enhanced AGVN by introducing a so-called surveillance plane to promote the management in the control of plane, while AI techniques are involved in the new plane.

Besides the aforementioned methods, there are also many other QAR-based research works, such as the studies focused on hard landing [27, 28], exceedance events analysis [29, 30], abnormal flight detection [31, 32], traffic network flow analysis [33] etc.

3 | DATA PREPROCESSING

3.1 | QAR dataset

The QAR dataset used in this paper are collected from a China commercial airline. The original dataset includes more than 40,000 Airbus A321 aircraft flights, whose departure or arrival places belong to two domestic airports during January 2018². After decoding, each flight is represented as a CSV file storing the time series flight parameters throughout the flight. Since some raw data may be incorrect due to sensory or decoding errors, the data cleaning procedure is first performed to remove flights with errors and complete missing information. After a series of data cleaning steps, 44,176 flight samples with each sample containing 39 parameters are obtained.

3.2 | Landing interval extraction

For each flight, due to the fact that the landing phase only occupies about 1% of the entire flight time, we only use data of this phase to predict the touchdown distance. To be specific, in order to extract the landing interval, it is necessary to know when an aircraft descends to the altitude of 50 feet (above runway) during landing. To this end, the radio height parameters (left and right radio height) parameters are tracked to locate the 50 feet point, as shown in c1. Then 4 s forward and 4 s backward from this point are taken to obtain an 8 s interval, from which flight parameters are extracted and treated as input sequence of the deep learning model. After that, we consider the 10 s interval (4–14 s after 50 feet point) and utilize ground speed and radio height parameters in this interval as the output sequences. Meanwhile, the flight distance of the aircraft from the 50 feet point to the touchdown point is calculated and defined as the touchdown distance, as shown in Figure 1. The touchdown point is obtained by monitoring when the state of landing gears first switched from ‘AIR’ to ‘GROUN’. There are two reasons for choosing 50 feet as a key radio height point: First, pilots usually perform flare operations at an altitude of around 50 feet, which means the information around this point can be important for touchdown distance prediction. Second, it is generally acknowledged that the aircraft is right above the runway threshold when its radio height drops to 50 feet, which means the flight distance from 50 feet to touchdown can be used to approximate the runway length consumed by the aircraft when it touches the ground. The 4–14 s zone covers around 99.94% of the flight landing time (the time from 50 feet to landing) in the dataset. For this reason, we generate sequence

² For the sake of privacy, the airline name, airports and QAR dataset cannot be disclosed.
predictions for the 4–14 s interval after the 50 feet point to ensure the touchdown point can be covered.

3.3 Data transformation and interpolation

QAR parameters exhibit strong heterogeneity, so they need to be preprocessed to improve the usability. Most QAR parameters are numerical, such as radio height, airspeed, pitch angle, etc., while values of some other parameters are discrete and belong to discrete variables, such as the status of landing gears, flap level, auto throttle mode, etc. For the discrete state variables, we convert them to discrete integers. For example, the switch variables are transferred into 1 and 0 corresponding to the states of on and off respectively. In addition, the vast QAR parameters are collected at various sampling rates (1/2, 1, 2, 4, 8 Hz). In the preprocessing process, with elaborately designed down and up-sampling methods, different parameters are unified to make sure they have the same sampling rates of 4 Hz. Specifically, on the one hand, we take the average of every two consecutive frames to achieve the purpose of down-sampling the parameters with a sampling rate of 8 Hz. On the other hand, two up-sampling methods are further considered for parameters with a sampling rate lower than 4 Hz. The first strategy is to utilize the correlation among different parameters and calculate from the related high sampling rate parameters to substitute the corresponding low sampling rate parameters. Take $IV \_CA$ which represents the vertical descent rate for example, the original parameter is inaccurate and sampled at a low rate (1 Hz), so it is replaced with the derivative of radio height, which is more accurate and sampled with a higher rate (4 Hz). Similarly, the ground speed is replaced with the integral of longitudinal acceleration. For the second strategy, interpolation is utilized to insert parameter values between consecutive data frames until the sampling rate requirement is satisfied. In this paper, the Lagrange interpolation [34] method is used. This method has been widely utilized in a lot of practical applications. Specifically, Lagrange interpolation is used to insert data between successive frames. Mathematically speaking, assume that the observed data points are $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, the interpolation point is $(x, y)$, where $x_i$ represents the time point and $y_i$ represents the corresponding parameter value, then:

$$I_i(x) = \prod_{j=0, j \neq i}^{n} \frac{x - x_j}{x_i - x_j} \quad (1)$$

$$L_n(x) = \sum_{i=0}^{n} y_i I_i(x) = \sum_{i=0}^{n} \left\{ \prod_{j=0, j \neq i}^{n} \frac{x - x_j}{x_i - x_j} \right\} \quad (2)$$

By solving the above two equations, the interpolated value $y$ corresponding to $x$ can be obtained. After interpolation, features that are critical to ground speed and radio height from the flight samples are extracted.

The workflow of our data process procedure is summarized in Figure 2. Through these preprocessing operations, we not only improve usability of the data, but also retain as much useful information as possible.

4 METHODOLOGY

4.1 Problem formulation

This paper mainly aims to develop a proactive method to provide accurate and early warning of long landing events. To this end, the proposed model should be able to forecast the touchdown distance, that is, the runway length to be consumed by the aircraft, several seconds before touchdown. The touchdown distance is defined as follows:

**Definition 1 (Touchdown Distance)**. The touchdown distance is defined as the flying distance of an aircraft from the radio height of 50 feet to touchdown during the landing phase,
where the flying distance is calculated by aggregating over the ground speeds.

Based on the definition of touchdown distance, the touchdown distance prediction problem is further defined as follows:

**Definition 2** (Touchdown Distance Prediction). Given the 8 s interval around the 50 feet point during lading (as shown in Figure 1) and the corresponding QAR flight parameters as input sequences, the touchdown distance prediction problem aims at predicting the touchdown distance based on the given input sequences.

To solve the above problem, instead of directly predicting the touchdown distance, a two-step indirect method is utilized, that is, the predictions for the ground speed and radio height sequences are first generated based on the input parameter sequences with a deep CNN-LSTM and TG-attention sequence-to-sequence model, and then the touchdown distance is calculated based on the predicted sequences. The framework is integrated and shown in Figure 3, and detailed instructions will be given in the following section.

### 4.2 Feature selection

To avoid the noises incurred by too many QAR parameters and relieve the reliance on expert experience, appropriate feature selection is needed to get the useful flight parameters. As aforementioned, after data processing, 39 flight parameters are obtained, but not all parameters are beneficial to prediction performance and results. Traditional methods mainly rely on flight experts’ experience to select the features, which can be a laborious task. In this paper, the gradient boosting decision trees (GBDT) model is used to seek the parameters that are most closely related to the ground speed and radio height sequences. As an ensemble learning algorithm, GBDT consists of multiple decision tree iterations. It exhibits well acknowledged advantages over other machine learning models in regression, classification and feature selection. The ensemble learning characteristic of GBDT gives itself a natural advantage for the discovery of feature importance and feature combination [35]. Here, this paper separately considers the ground speed and radio height parameters, and utilize the GBDT method to compute feature importance and rank the important features accordingly. Then, flight parameters, that is, features, with higher importance scores are selected and used as the features of input sequences, which are further feed to the downstream CNN-LSTM and TG attention sequence-to-sequence model.

### 4.3 Traditional LSTM codec architecture

The LSTM encoder-decoder architecture is a traditional encoding and decoding structure, originally used for text translation [36, 37]. As shown in the Figure 4, it can read and generate arbitrary length sequences. The LSTM encoder-decoder mainly includes two parts: the encoding part reads the input sequence and passes it to the LSTM for processing, and the decoding part decodes the vector passed from the encoder, and finally outputs the predicted sequence. Specifically, $x_1, x_2, \ldots, x_T$ are passed into the encoder as an input sequence, and the encoder LSTM unit performs $T$ recursive updates to generate a state vector $C_t$ based on the input sequence. Then, $C_T$ is passed to the decoder as the initial unit state of the decoder. As shown in the Figure 4, the decoder iteratively utilizes the output of the previous time step as the input of the next time, and finally generates a complete prediction sequence after $T$ times of execution.

For the LSTM unit in the encoder, the LSTM network stores important characteristic information of the input data through a memory unit. Each network neuron contains the core cell and three gate units, which are input gate, forget gate and output gate. At time $t$, the value of input gate $i_t$ and the value of candidate cell state $\hat{C}_t$ are calculated as follows:

$$i_t = \sigma (W_i^i (h_{t-1}, x_t) + b_i)$$

$$\hat{C}_t = \tanh (W_C^r (h_{t-1}, x_t) + b_c)$$

Similarly, the activation value $f_t$ of the forget gate at time $t$ can be calculated:

$$f_t = \sigma (W_f^f (h_{t-1}, x_t) + b_f)$$

Based on the previous calculations, the regenerative cell state $C_t$ is as follows:

$$C_t = f_t \# C_{t-1} + i_t \# \hat{C}_t$$

Finally, the value of the output gate $O_t$ is calculated. Based on the value of the output gate and the current cell state, the current hidden state $b_t$ is obtained at the current moment $t$.

$$O_t = \sigma (W_o^O (h_{t-1}, x_t) + b_o)$$

$$b_t = O_t \# C_t$$
In the above equations, $\sigma$ represents the sigmoid activation function. $W_i$, $W_o$, $W_f$ and $W_o$ are weight matrices, $b_i$, $b_o$, $b_f$ and $b_o$ represent bias vectors. Through the above calculations, LSTM encoder can effectively utilize the temporal interdependence among the input sequence to have a long-term memory. In the decoding stage, the decoder predicts the next output $y_t$ on account of $c_T$ and the output sequence $y_1, y_2, y_3, \ldots, y_t$ at the previous time. In fact, the joint probability of generating $y = y_1, y_2, \ldots, y_T$ sequence can be broken down into continuous conditional probabilities.

$$p(y_1, \ldots, y_T | x_1, \ldots, x_T) = \prod_{t=1}^{T} p(y_t | c_T, y_1, \ldots, y_{t-1})$$

where $c_T$ represents the context vector.

The advantages of using CNN-LSTM and TG-attention Encoder-Decoder architecture are two folds: First, with the CNN-LSTM module the model can capture both temporal and spatial interdependence among the flight parameters. Second, with the TG-attention module the model is able to capture the impact of both local and global trends of the input sequence on the future output sequence.

We separately use two CNN-LSTM and TG-attention encoder-decoder models to predict the ground speed and radio height sequences respectively. For convenience, the network structure is same for the two models, as seen in Figure 5.

It is worthwhile to note that we did not use one model to simultaneously generate predictions for both ground speed and radio height sequences since the importance of features with respect to the two sequences are quite different (as will be seen in Figures 6 and 7), training them separately can benefit the
**FIGURE 5** Detailed CNN-LSTM and TG-attention encoder–decoder architecture

**FIGURE 6** TG-attention architecture
prediction accuracy. The encoder whose sequence length is set to 32 is made up of one input layer followed by a CNN layer and two LSTM layers. The size of CNN convolution kernel is set to $1 \times 2$, which changes the step size from the original two-dimensional movement to a one-way movement along different parameters. In this way, the model is able to capture the spatial interdependence of different flight parameters and get a spatial local feature representation at each single time point, as shown in the left part of Figure 5. Therefore, the CNN-LSTM module ensures deep extraction of spatial features while retaining the temporal information of the flight data. Each LSTM layer is constructed with 256-dimensional cell.

The TG_attention module consists of temporal attention and global attention, as shown in Figure 8. The attention can be described as the different effects of different parts of the previous time series on the current time step. The proposed temporal attention is enlightened by self-attention mechanism, which uses both past and future data of time $t$. But for flights, the current aircraft status is only determined by information from the past. Therefore, we modify the structure of self attention and propose temporal attention, as shown in Figure 8. The temporal attention mechanism can be summarized as a three-part paradigm including query $Q$, key $K$ and value $V$. $Q$ determines the key-value pair to focus on. $K$ is utilized to calculate the attention score, and $V$ is used to calculate the context vector.

$$W_{i,j}(Q, K, V) = \frac{K_T \cdot Q_j}{\sqrt{d_k}}$$

where $i, j = 1, 2, \ldots, T$, $d_k$ represents the dimension of $K$ and $W_{i,j}(Q, K, V)$ represents the attention score. In order to eliminate the influence of the future flight data information, the following formula is used to set the influence of the future time to 0.

$$W_{i,j}(Q, K, V) = \begin{cases} W_{i,j}(Q, K, V), & i \geq j \\ 0, & i < j \end{cases}$$

Then, softmax function is applied on the first dimension of $W_{i,j}(Q, K, V)$ to normalize them. Finally, as shown in Figure 8, the sum of the weights of each row of $\text{softmax}(W_{i,j}(Q, K, V))$ is calculated to represent the importance of each time step.

$$W'_{t,i} = \sum_{j=1}^{i} \text{softmax}(W_{i,j}(Q, K, V))$$

where $W'_{t,i}$ represents the importance of data at time step $t$. Temporal attention characterizes the importance of previous states to each current input data. It captures the dependency between inputs and can be used to characterize pilot operations at each moment. Besides, in order to obtain the global influence of the input sequence on the future predicted sequence, the following global attention is added:

$$e_{i,t} = \overline{W}^T \tanh (W^T (i_t, h_t))$$

where $e_{i,t}$ represents the correlation between the hidden vector $i_t$ of the decoder output sequence and the hidden vector $h_t$ of the encoder. $\overline{W}$ and $W^T$ are the parameters to be learned. Then, the softmax function is applied to $e_{i,t}$.

$$a_{i,t} = \frac{\exp (e_{i,t})}{\sum_{t'=1}^{T} \exp (e_{i,t'})}$$

Finally, as shown in Figure 8, the context vector $\overrightarrow{C}$ is obtained by using the dot product to combine temporal attention and global attention, which can capture the influence of local temporal information, as well as the global trend of the entire flight trajectory, so as to better predict future time series data (ground speed and radio height) sequences.

$$\overrightarrow{C} = \sum_{i=1}^{T} W'_{t,i} a_{i,t} h_t$$

where $W'_{t}$ indicates the importance of input sequence QAR dataset at time step $t$ from temporal attention, and $h_t$ represents the hidden vector to the encoder.
The decoder is made up of two LSTM layers, and the sequence length is set to 40. LSTM layer is constructed with 256-dimensional cell, followed by a 256-dimensional fully connected layer and an output layer. The decoder utilizes the context vector $\vec{C}$ transmitted from the TG-attention as its initial cell states. The decoder structure is presented in Figure 9.

4.5 Touchdown distance calculation

Through the deep CNN-LSTM and TG-attention sequence-to-sequence model, the ground speed and radio height sequences can be predicted, based on which we can further calculate the touchdown distance, as shown in Algorithm 1. Line 4–6 calculates the flight distance from 50 feet point to 4 s after the 50 feet point based on the real ground speed sequence. Here the ground speed $S_i$ is divided by four because each second contains four frames. Line 7–12 finds the touchdown point $p$, that is, we go through the predicted radio height sequence and watch when the radio height firstly falls below a predefined threshold $\theta$ (close to zero), and then mark the index as the touchdown point. Then the touchdown distance can be calculated by aggregating over the predicted ground speed sequence, as shown in line 13–15, where the ground speed $s_i$ is divided by four because each second contains four frames.

5 EXPERIMENTS

5.1 QAR dataset

The dataset used in this paper include 44,176 samples of Airbus A321 flights. Each flight took off or landed from two domestic air ports during January 2018, with 39 flight parameters. We divide the dataset into a training set and a test set, randomly selecting 4,417 (about 10%) flight samples as the test set, and the remaining 39,759 flights as the training set.

Algorithm 1 Training of the network using SGD

Algorithm 1 Touchdown Distance Calculation

1: Input: Real ground speed sequence $S_1, S_2, ..., S_T$ (from 50 feet point to 4 s later), predicted ground speed sequence $s_1, s_2, ..., s_T$ and radio height sequence $h_1, h_2, ..., h_T$, touchdown threshold $\theta$.
2: Output: Predicted touchdown distance $l$
3: Initialize $l = 0$, touchdown point $p = 0$.
4: for $i = 1$ to $T$ do:
5:     $l \leftarrow l + \frac{S_i}{4}$
6: end for
7: for $i = 1$ to $T$ do:
8:     if $h_i \leq \theta$ then
9:         $p \leftarrow i$
10:     break
11: end if
12: end for
13: for $i = 1$ to $p$ do
14:     $l \leftarrow l + \frac{s_i}{4}$
15: end for
16: return $l$

5.2 Baselines

The following baselines are used for comparison:

- RF-LSTM: This algorithm was proposed by Tong et al. [12], and based on the traditional random forest and LSTM model, which used to predict the landing velocity. We use this baseline for the comparison in the ground speed series prediction part. Like the authors did in their original work, PCA dimensionality reduction is performed on the 21 features selected by GBDT. The number of hidden layers is set to 4 and the dimensionalities are 128, 64, 32, and 8 respectively.

- Stepwise Linear Regression: Wang et al. [11] proposed the stepwise linear regression algorithm to predict the touchdown distance. The algorithm used flare time, flare height, ground speed, descent rate and vertical acceleration as features. In our experiments, we remove the flare time (which is the time-span from 50 feet altitude to touchdown) since it implicitly requires the touchdown information.

- Traditional Machine Learning Algorithms: Classical machine learning methods include decision tree (DT), linear regression (LR), gradient boosting decision tree (GBDT), random forest (RF), neural network (NN) and SVM, etc. In our experiment, we first consider stitching 21 parameters with 32 time steps (8 s and four frames per second), and the composed feature length is up to 672, and then feed the features to downstream machine learning models for training. However, the performances of these models trained through this method are extremely poor, so we eventually only use the 21 flight parameters at the 50 feet point as the features so as to ensure that each machine learning method can exert the best performance.
• Expert CNN-LSTM and TG-attention: For this baseline, we use exactly the same deep learning model as proposed in this paper. The only difference is that this baseline uses a total number of 32 features selected through experience of flight experts rather than GBDT. Specifically, the model considers as many parameters as possible based on the flight expert experience.
• LSTM and TG-attention: This baseline removes the CNN block to verify the role of CNN.
• CNN-LSTM and attention Encoder- Decoder utilizes traditional attention mechanism, which can verify the role of TG-attention we proposed.

5.3 Evaluation metrics

This paper uses MAE, RMSE and MAPE to evaluate the performance of comparing methods. These metrics are widely used to measure the accuracy of regression models. The definitions of these metrics are as follows:

\[
RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2}
\]

(19)

\[
MAE = \frac{1}{m} \sum_{i=1}^{m} |y_i - \hat{y}_i|
\]

(20)

\[
MAPE = \frac{100}{m} \sum_{i=1}^{m} \left| \frac{y_i - \hat{y}_i}{y_i} \right|
\]

(21)

where \(y_i\) and \(\hat{y}_i\) are real and predicted values, respectively, and \(m\) is the number of the testing samples.

5.4 Feature selection results

As aforementioned, the GBDT method is employed to rank and select important features from the 39 flight parameters. Specifically, it first computes weight values for all the parameters (features) on each flight sample, and then accumulates the weights over the 44,176 samples to obtain the overall weight (i.e. importance) for each feature. Table 2 shows the regression accuracy scores of GDBT on ground speed and radio height respectively.

5.4.1 Feature selection results for ground speed

Figure 6 presents the top features that are important for ground speed (those insignificant parameters have been deleted). The number of trees in GBDT is set to 6570. From Figure 6 it is observed that wind direction (\(WIN\_DIR\)) and aircraft magnetic heading (\(HEAD\_MAG\)) are more important than other parameters. At the same time, the feature importance of gravitational potential energy (\(P\_ENERGY\)) is also very high, which means that this feature is also important for ground speed. It is worthwhile to note that (\(P\_ENERGY\)) is not a sensory parameter contained in the raw data, and this feature is computed as follows:

\[
P\_ENERGY = GW\_C \times g \times RALT
\]

(22)

where \(GW\_C\) is the aircraft gross weight, \(g\) is the acceleration due to gravity, and \(RALT\) is the radio height. Similarly, the kinetic energy of the aircraft, that is, \(K\_ENERGY\), is also taken into account:

\[
K\_ENERGY = \frac{1}{2} \times GW\_C \times GS \times GS
\]

(23)

where \(GS\) is the aircraft ground speed. Since the \(K\_ENERGY\) feature itself incorporates the ground speed parameter, it is not considered as the candidate feature for ground speed feature importance analysis.

5.4.2 Feature selection results for radio height

Figure 7 shows the rank of feature importance for the radio height parameter (the insignificant features are re-moved), where the number of trees in GBDT is set to 8000 with the depth of the tree set to eight. Unlike ground speed, the most important parameters for radio height are latitude (\(LAT\_PC\)), descent rate (\(IVV\_CA\)), kinetic energy (\(K\_ENERGY\)), and longitude (\(LON\_PC\)). Again we see that energy plays an important role, and the gravitational potential energy parameter (\(P\_ENERGY\)) is not included due to similar reasons.

### TABLE 1 Abbreviations and explanations

| Abbreviation | Explanation |
|--------------|-------------|
| QAR | Quick access recorder |
| GBDT | gradient boosting decision trees |
| TG-attention | Temporal and Global attention |
| ANOVA | Analysis of variance |
| LSTM | Long Short-Term Memory |
| CNN | Convolutional Neural Network |
| BN | Bayesian network |
| RF | Random Forest |
| LR | Linear regression |
| AGVN | Air-ground integrated vehicular network |
| AIR | The state of the landing gear of the airliner |
| GROUDN | The state of the landing gear of the airliner |

### TABLE 2 Results of GBDT regression accuracy

| Parameters | \(R^2\) |
|------------|---------|
| Ground speed | 82.8% |
| Radio height | 72.3% |
According to the feature importance results of GBDT, finally 21 parameters are selected as features from 39 candidate flight parameters to generate predictions for ground speed and radio height sequences respectively. Table 3 shows the 21 most important parameters selected.

### Table 3  Top 21 features selected based on GBDT

| Parameters     | Explanation                        | Frequency |
|----------------|-------------------------------------|-----------|
| VAPP           | Reference landing speed             | 1 Hz      |
| HEAD_MAG       | Magnetic heading                    | 1 Hz      |
| WIN_SPD        | Wind speed                          | 1 Hz      |
| WIN_DIR        | Wind direction                      | 1 Hz      |
| LONPC          | Longitude                           | 1 Hz      |
| LATPC          | Latitude                            | 1 Hz      |
| ROLL           | Roll                                | 2 Hz      |
| ROLL_CPT       | Captain roll control                | 8 Hz      |
| PITCH          | Aircraft pitch angle                | 4 Hz      |
| PITCH_CPT      | Captain pitch control               | 8 Hz      |
| IAS            | Indicated air speed                 | 1 Hz      |
| GS             | Ground speed                        | 1 Hz      |
| RALT_LH        | Radio height (left)                 | 4 Hz      |
| RALT_RH        | Radio height (right)                | 4 Hz      |
| IVV_CA         | Decline rate                        | 1 Hz      |
| N11            | No. 1 engine speed ratio            | 1 Hz      |
| N12            | No. 2 engine speed ratio            | 1 Hz      |
| P_ENERGY       | Aircraft potential energy           | 1 Hz      |
| K_ENERGY       | Aircraft kinetic energy             | 1 Hz      |
| GW_C           | Aircraft gross weight               | 1 Hz      |
| VRTG           | Vertical load                       | 8 Hz      |

#### 5.5 Touchdown distance prediction

Before the selected features are fed to the CNN-LSTM and TG-attention sequence-to-sequence model, the features is first standardized with zero-mean normalization to eliminate the possible impact of different feature scales. The normalization method is as follows:

\[
Z = \frac{x - \mu}{\sigma}
\]  

where \(\mu\) is the average value, \(\sigma\) is the variance, and \(Z\) is the transformed data.

##### 5.5.1 Ground speed prediction results

For the training set, 39,759 processed flights samples are used to train the ground speed sequence prediction model. The Adam optimization algorithm is employed to learn model parameters and use 4,417 test samples to evaluate model performance. As mentioned earlier, the input feature sequence of the CNN-LSTM encoder is extracted from an 8 s interval, and all parameters unified to 4 Hz sampling rate so that the length of the feature sequence input to the CNN-LSTM encoder is set to 32. Similiarly, the length of the ground speed sequence output by the decoder is set to 40 (10 s multiplied by 4 Hz).

The batch size of the QAR training data is set to 256, and the number of epoch iterations are set to 6000. Figure 10 shows the performance of the final training model by showing the relationship between the predicted ground speed and the real value, where the x axis represents the real ground speed value and the y-axis represents the predicted value. A total number of 176,680 points are displayed in the figure, because the test set contains 4,417 samples, and the predicted sequence length of the sample is 40. It can be seen from the figure that the data points are clustered closely around the \(y = x\) line, which indicates that the predicted ground speed is very close to the real value, especially when the real ground speed is greater than 110 knots. Figure 11 shows the relative error of ground speed at each data point, where x-axis is the indices of the data points.
TABLE 4

| Models                  | MAPE(%) | RMSE  | MAE  |
|------------------------|---------|-------|------|
| CNN-LSTM and TG-attention | 0.21    | 0.13  | 0.63 |
| LSTM and TG-attention   | 0.41    | 0.32  | 0.71 |
| CNN-LSTM and attention  | 1.71    | 1.07  | 0.83 |
| Expert CNN-LSTM TG-attention | 0.59    | 0.28  | 0.19 |
| LSTM Encoder-Decoder    | 2.20    | 1.32  | 1.03 |
| LSTM Tong et al.        | 1.96    | 1.50  | 1.13 |
| Linear Regression       | 12.57   | 11.01 | 8.43 |
| Decision Tree           | 5.70    | 4.66  | 3.51 |
| GBDT                    | 5.79    | 5.22  | 3.95 |
| Random Forest           | 9.75    | 8.48  | 6.55 |
| Neural Networks         | 21.86   | 21.17 | 16.11|
| SVM                     | 7.83    | 6.55  | 5.02 |

Here the data points are sorted in increasing order according to the real ground speed values, so as to provide a better view of the relative errors at different speed values. It can be seen that when the real ground speed is small (110–120 knots), the relative error is larger. As the real value increases, the relative error reduces and becomes stable (at around ±5 knots). When the real value is larger than 160 knots, the relative error becomes almost invisible.

Table 4 shows the comparison results of the ablation experiment, as well as the comparison results of other different methods and our proposed model. The evaluation metrics are RMSE, MAE, and MAPE, and the best results are shown in bold. It can be seen from the comparison results that the CNN-LSTM and TG attention encoder-decoder model exhibits significantly better performance in terms of prediction accuracy, especially when compared to classic machine learning models. Among all the models, the five-layer shallow neural network has the worst results, which reflects the power of deep learning models. For classic machine learning, the performance of tree-based models (GBDT and DT) are better than other models. In our opinion, their relatively better performance largely owes to the generalization ability of tree-based models. The most competitive benchmark among the models proposed by other authors is the RF-LSTM based method [12], which is still largely inferior to ours. In addition, by comparing CNN-LSTM and TG attention encoder-decoder with Expert CNN-LSTM and TG attention encoder-decoder, it is shown that using GBDT instead of expert experience for feature selection can significantly improve the prediction performance of the model. Moreover, as shown in the ablation experiments, after removing the CNN block, the experimental results of the LSTM and TG-attention Encoder-Decoder model are worse than that of CNN-LSTM and TG-attention Encoder-Decoder model. This shows that adding CNN block can indeed capture the spatial local characteristics of time series QAR dataset at the same time, thereby improving the prediction performance of the model. By comparing traditional attention with the proposed TG-attention, we can see that our TG-attention prediction performance is significantly improved.

5.5.2 Consequence of radio height prediction

We use the same model structure as the ground speed prediction model to predict the radio height sequence. The model is trained with 8000 iterations and the prediction results of radio height sequence on test set are shown in the Figure 12, where x-axis indicates the real radio height while y-axis indicates the predicted values. Similar to the results of ground speed prediction, there is a total number of 176,680 data points exhibited in this figure. The figure shows that data points are located closely along the y = x line, which indicates that the predicted radio heights are very close to the real values. Figure 13 shows the relative error of radio height at each data point. Similar to Figure 11, x-axis indicates indices of the data points sorted in increasing order according to the real radio height values. It can be seen from the figure that when the radio height is higher...
than 10 feet, the model's prediction performance is more accurate. When the aircraft's radio height drops to 5 feet, the model's prediction accuracy decreases, which shows different patterns compared to Figure 11.

Table 5 presents the results of radio height for different models on the test set in terms of RMSE and MAE. Here the MAPE measure is not used because there are a lot of data points with zero real height values, according to Equation 21, the MAPE values can be infinite. From the results we see that compared with all other baselines, our model exhibits significant better prediction performance. All classic machine learning models show poor prediction accuracy, among which the Linear Regression model has the worst performance, implying the need for deep learning model. As in the case of ground speed prediction, the tree based models (RF, DT and GBDT) shows better performance than other traditional machine learning models. The performance of Experience CNN-LSTM TG-attention Encoder-Decoder is the closest to our model, however, its performance is still inferior to ours, which demonstrates the effectiveness of GBDT to select features instead of experience. After removing the CNN block, the performance of the model has a certain decline, which shows that CNN can benefit the model performance. Using traditional attention to replace the TG-attention we proposed, it can be seen that the performance of the model is significantly reduced, indicating that TG-attention can better predict future time series.

### Table 5 Results of RMSE and MAE for radio height prediction

| Models                          | RMSE  | MAE  |
|--------------------------------|-------|------|
| CNN-LSTM and TG-attention      | 0.25  | 0.17 |
| LSTM and TG-attention          | 0.33  | 0.25 |
| Expert CNN-LSTM TG-attention   | 0.28  | 0.21 |
| CNN-LSTM and attention         | 0.91  | 0.59 |
| LSTM encoder-decoder           | 1.11  | 0.68 |
| Linear Regression              | 8.75  | 5.40 |
| Decision Tree                  | 2.66  | 1.64 |
| GBDT                           | 2.40  | 1.54 |
| Random Forest                  | 2.48  | 1.59 |
| Neural Networks                | 3.15  | 2.02 |
| SVM                            | 4.02  | 2.96 |

5.5.3 Result of touchdown distance

Based on the predicted radio height and ground speed sequences from CNN-LSTM and TG-attention Encoder-Decoder, the touchdown distance of aircraft from the 50 feet point to the ground is calculated according to Algorithm 1 (where the threshold $\theta$ is set to 1.9 feet through small-scale experiments). The predicted and real touchdown distance of the aircraft from 50 feet to touchdown is shown in the Figure 14, where x-axis represents the real touchdown distances while y-axis indicates the predicted values. As aforementioned, the test set contains 4,417 samples, so there are totally 4,417 data points shown in the figure. In general, from the results we see that the predicted distances are very close to the real distances, especially when the real distance is less than 700 m. Figure 15 shows the relative error of landing touchdown distance at each data point, where each data point represents a flight sample. From an overall point of view, the error between the real and predicted landing touchdown distance values is about 26 m, which is smaller than the existing methods, and it is even smaller as the real value decreases, indicating that the proposed model has a better fitting property.

Table 6 shows RMSE, MAE, and MAPE of the touchdown distance prediction results of different models on the test set. From the table we can conclude that the CNN-LSTM and TG-attention encoder-decoder based methods significantly outperform all the other methods, indicating the superiority of using deep learning models. For example, the prediction error of the stepwise linear regression is almost three times as that of our model (CNN-LSTM and TG-attention encoder-decoder). Comparing with experience-based feature selection, the GBDT-based feature selection can further improve the performance of our model.
6 | CONCLUSION

In civil aviation industry, flight safety is a widely concerned issue and long landing prediction is important to avoid runway overrun accidents. To improve the prediction performance, this paper proposed an innovative and accurate long landing prediction model based on the CNN-LSTM and TG-attention Encoder-Decoder architecture. The proposed TG-attention combined CNN with global attention to predict ground speed and radio height sequences respectively, based on which the touchdown distance was further calculated. Experimental results on large QAR data validated effectiveness of the proposed method. In the future, our work can be extended in the following directions. First, we may apply the sequence-to-sequence model to other flight safety related problems, such as hard landing, tail strike etc. Second, despite our method can provide good prediction accuracy, it lacks interpretability, which can be very important in some scenarios. How to combine deep learning and interpretability in flight safety research is a topic worth studying in the future.

ACKNOWLEDGMENTS
This work was supported in part by: National Key R&D Program of China (Nos. 2020YFC0811000, 2018YFB2101203), National Natural Science Foundation of China (Nos. U2033213, 61966008), Fundamental Research Funds for the Central Universities (No. 2020CDCGJJS041), Key Research and Development Program of Chongqing (No. cste2019jscx-fxydX0071), Zhejiang Lab (No. 2021KE0AB01), Guangxi Key Laboratory of Trusted Software (No. kx201702, kx202006).

ORCID
Jiaxing Shang  https://orcid.org/0000-0002-3152-1760

REFERENCES
1. Boeing (2008). Statistical summary of commercial jet airplane accidents. Available at: http://www.boeing.com
2. Lv, H., Yu, J., Zhu, T: A novel method of overrun risk measurement and assessment using large scale QAR data. In: 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService), Bamberg, Germany, pp. 213–220 (2018)
3. Van Es, G.W.H., Tritschler, K., Tauss, M.: Development of a landing overrun risk index, 21st annual European Aviation Safety Seminar (EASS) Nicosia, Cyprus, NLR Air Transport Safety Institute, Report NLR-TP-280, (2009)
4. Ayra, E.S., Rios Insua, D., Cano, J.: Bayesian network for managing runway overruns in aviation safety. J. Aerospace Inf. Syst. 16(12), 546–558 (2019)
5. Sheridan, K., et al.: An application of dbscan clustering for flight anomaly detection during the approach phase. In: AIAA Scitech 2020 Forum. p. 1851 (2020)
6. Básora, L., Olive, X., Duhot, T.: Recent advances in anomaly detection methods applied to aviation. Aerospace 6(1), 117 (2019)
7. Calle Alonso, F., Pérez, C.J., Ayra, E.S.: A bayesian-network-based approach to risk analysis in runway excursions. The J. Navig. 72(S), 1121–1139 (2019)
8. Wang, L., et al.: A landing operation performance evaluation system based on flight data. Engineering Psychology and Cognitive Ergonomics: Cognition and Design. Springer, Berlin, pp. 297–305 (2017)
9. Wang, L., Wu, C., Sun, R.: An analysis of flight quick access recorder (QAR) data and its applications in preventing landing incidents. Reliab. Eng. Syst. Safety 127, 86–96 (2014)
10. Hu, C., et al.: The study on hard landing prediction model with optimized parameter SVM method. In: 2016 35th Chinese Control Conference (CCC). Chengdu, China, pp. 4283–4287 (2016)
11. Wang, L., Ren, Y., Wu, C.: Effects of flare operation on landing safety: A study based on ANOVA of real flight data. Saf. Sci. 102, 14–25 (2018)
12. Tong, C., et al.: A novel deep learning method for aircraft landing speed prediction based on cloud-based sensor data. Future Gener. Comput. Syst. 88, 552–558 (2018)
13. Tong, C., et al.: An innovative deep architecture for aircraft hard landing prediction based on time-series sensor data. Appl. Soft Comput. 73, 344–349 (2018)
14. Khatwa, R., Helmreich, R.L.: Analysis of critical factors during approach and landing in accidents and normal flight. Flight Safety Digest 17(11–12), 1–2 (1999)
15. Gui, G., et al.: Flight delay prediction based on aviation big data and machine learning. IEEE Trans. Veh. Technol. 69(1), 140–150 (2019)
16. Nielsen, T.D., Jensen, F.V.: Bayesian networks and decision graphs. Springer Science & Business Media, Berlin (2009)
17. Pearl, J.: Fusion, propagation, and structuring in belief networks. Artif. Intell. 29(3), 241–288 (1986)
18. Höhndorf, L.: Statistical Dependence Analyses of Flight Data for Safety Management. Technische Universität München (2019)
19. Park, S.H., et al.: Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture. In: 2018 IEEE Intelligent Vehicles Symposium (IV), Changsha, pp. 1672–1678 (2018)
20. Janakiraman, V.M.: Explaining aviation safety incidents using deep temporal multiple instance learning. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, pp. 406–415 (2018)
21. Chen, H., et al.: A deep learning method for landing pitch prediction based on flight data. In: 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Weihai, China, pp. 199–204 (2020)
22. Wu, X., et al.: Hierarchical attention based long short-term memory for chinese lyric generation. Appl. Intell. 49(1), 44–52 (2019)
23. Belhadi, A., et al.: A recurrent neural network for urban long-term traffic flow forecasting. Applied Intelligence (2020)
24. Kang, Z., et al.: A deep sequence-to-sequence method for aircraft landing speed prediction based on QAR data. In: International Conference on Web Information Systems Engineering, Amsterdam, Netherlands, pp. 516–530 (2020)
25. Xu, W., et al.: A hybrid modelling method for time series forecasting based on a linear regression model and deep learning. Appl. Intell. 49(8), 3002–3015 (2019)
26. Sun, J., et al.: Surveillance plane aided air-ground integrated vehicular networks: Architectures, applications, and potential. IEEE Wireless Commun. 27(6), 122–128 (2020)

27. Li, X., et al.: Curvecluster: Automated recognition of hard landing patterns based on QAR curve clustering. In: 2019 IEEE Smart-World, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (Smart World/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, 602–609 (2019)

28. Qiao, X., et al.: A prediction model of hard landing based on rbf neural network with k-means clustering algorithm. In: 2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Indonesia, pp. 462–465 (2016)

29. Wang, L., et al.: Risk cognition variables and flight exceedance behaviors of airline transport pilots. In: International Conference on Engineering Psychology and Cognitive Ergonomics. Springer, Las Vegas, NV, USA, pp. 725–737 (2018)

30. Liu, S., Zhang, Y., Chen, J.: A system for evaluating pilot performance based on flight data. In: International Conference on Engineering Psychology and Cognitive Ergonomics. Springer, Las Vegas, NV, USA, pp. 605–614 (2018)

31. Li, L., et al.: Anomaly detection via a gaussian mixture model for flight operation and safety monitoring. Transp. Res. Part C: Emerging Technologies 64, 45–57 (2016)

32. Zhang, X., Chen, J., Gan, Q.: Anomaly detection for aviation safety based on an improved KPCA algorithm. J. Electr. Comput. Eng. 2017, 4890921 (2017)

33. Ren, P., Li, L.: Characterizing air traffic networks via large-scale aircraft tracking data: A comparison between china and the us networks. J. Air Transport Manage. 67, 181–196 (2018)

34. Berrut, J.P., Trefethen, L.N.: Barycentric lagrange interpolation. SIAM Rev. 46(3), 501–517 (2004)

35. He, X., et al.: Practical lessons from predicting clicks on ads at facebook. In: Proceedings of the Eighth International Workshop on Data Mining for Online Advertising. New York, NY, USA, 1–9 (2014)

36. Cho, K., et al.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:14061078 (2014)

37. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:150804025 (2015)