Multi-Step Regression Network With Attention Fusion for Airport Delay Prediction

Zhenchun Wei, *Member, IEEE*, Siwei Zhu, Zengwei Lyu, Yan Qiao, Xiaohui Yuan, *Senior Member, IEEE*, Yang Zhao, and Hao Zhang

***Abstract*— As part of airport behavior decisions, the accurate prediction of airport delay is highly significant in optimizing flight takeoff and landing sequences. However, the combination of various influencing factors affects airport delay prediction strongly, which would bring severe challenges in prediction. This paper introduces the sequence-to-sequence network and proposes a multi-step regression prediction method for the airport delay (DA-BILSTM) to accurately predict the airport delay. Rather than only considering a single kind of airport delay influencing factors, we design an attention fusion network for learning the sequence and condition correlation features adaptively. Moreover, the Bayesian optimization algorithm is introduced to optimize DA-BILSTM's hyperparameters. The method is applied individually to two datasets for predicting the airport's delays. The experiment results show that the prediction performance of DA-BILSTM is better than many state-of-the-art methods including the autoregressive integrated moving average model (ARIMA), long short-term memory (LSTM), gated recurrent unit(GRU), CNN-BILSTM, and TS-LSTM. When using DA-BILSTM in the two datasets, the average MAE of airport delay prediction in the next 5 hours is about 10 minutes, and the average RMSE is 20 minutes.**

***Index Terms*— Airport delay prediction, attention fusion network, bidirectional long short-term memory network, sequence to sequence.**

I. INTRODUCTION

THE limited airspace and airport ground resources, the rapid growth of passengers, and the increasingly dense flight scheduling worsen the airport delay problem daily [1]. Airport delay seriously endangers the interests of airports, airlines, air traffic controllers, passengers, etc. When an airport is delayed, the delay will spread to the downstream airports of the flight, to spread to all airports, resulting in a large-scale delay of the whole civil aviation transportation network. Accurate prediction of airport delays can assist airport managers to make decisions in advance so that the chain propagation of flight delays can be effectively restrained, and the serious consequences caused by large-scale airport delays can be avoided [2]. Accurate prediction of airport delay can also indirectly reflect the relationship between airport flight capacity and demand in the future [3], which is essential for optimizing the sequence of flight takeoff and landing, ensuring the safety and orderliness of all aspects of flights, and improving the robustness of civil aviation transportation system [4].

In the research of using machine learning to predict airport delays, most previous studies divide the airport delay into different categories according to the range of airport delays and then predict the delay category [5], [6], [7]. Classification methods give a coarse prediction of the delay level of flights but cannot determine the specific amount of airport delay time [8]. In practical applications, accurate prediction of airport-specific delays is crucial in flight scheduling [9]. Accurate airport delay prediction methods can be broadly classified into two categories: single-step forecasting and multi-step forecasting. Single-step prediction involves forecasting only a single future value per prediction, rendering it well-suited for short-term predictions. On the other hand, multi-step prediction enables the prediction of multiple future values in a single instance, making it more suitable for long-term forecasts [10]. When addressing the challenge of airport delay prediction, single-step predictions exhibit relatively high accuracy, while multi-step prediction offers a wealth of information that proves invaluable for airport decision-making and proactive scheduling [11]. In recent years, only a few studies focused on predicting the specific time of airport delay, especially multi-step airport delay prediction. The methods mentioned above may not fully account for the varying degrees of influence exerted by numerous factors contributing to airport delays. Some factors are strongly correlated with airport delays, while others may have a lower degree of impact on airport delay prediction [12]. Hence, airport delays caused by sequence and condition factors should be treated differently. By considering sequential and conditional factors and the strength of these factors, much-improved airport delay prediction can be achieved.

Airport operation is jointly affected by various conditions, such as weather conditions, scheduling conflicts, and overuse of airport capacities [13]. Sequence factors can also affect

Manuscript received 13 January 2022; revised 19 July 2022, 12 December 2022, 27 February 2023, and 9 July 2023; accepted 16 December 2023. Date of publication 16 January 2024; date of current version 2 July 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62002097; in part by the Natural Science Foundation of Anhui Province under Grant 2108085MF202; and in part by the Fundamental Research Funds for the Central Universities of China under Grant PA2023GDSK0048, Grant PA2022GDGP0028, and Grant PA2023GDGP0044. The Associate Editor for this article was R. Arghandeh. *(Corresponding author: Zengwei Lyu.)*

Zhenchun Wei, Siwei Zhu, Zengwei Lyu, Yan Qiao, and Yang Zhao are with the School of Computer and Information, Hefei University of Technology, Hefei, Anhui 230009, China, and also with the Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education, Hefei, Anhui 230009, China (e-mail: lzw@hfut.edu.cn).

Xiaohui Yuan is with the Department of Computer Science and Engineering, University of North Texas, Denton, TX 76210 USA.

Hao Zhang is with the Research and Development Department, Feeyo Technology Company Ltd., Hefei 230088, China.

Digital Object Identifier 10.1109/TITS.2023.3345630

airport operations [14]. An airport delay triggers a chain reaction and affects other airports. Hence, it is challenging to predict airport delays [11], [15], [16]. Firstly, analyzing all the factors that affect airport delays can be time-consuming and it is challenging to identify and focus on factors that have a greater impact on airport delays. The second challenge is how to maintain high prediction accuracy when making long-time airport delay predictions. Thirdly, manually setting hyperparameters is often based on experience and analysis of data, resulting in a large consumption of time and manpower. Hence, it is necessary to design a hyperparameter optimization method.

This paper formulates airport delay prediction as a multi-step prediction problem to address these challenges. The average delay per hour at the airport is used to represent the degree of delay [17]. An attention fusion network-based predicting method of airport delay is proposed, which consists of a sequence-to-sequence (Seq2Seq) framework and an attention fusion network. Seq2Seq consists of an encoder and a decoder, which extract airport delay features. The attention fusion network is composed of a sequence attention network and a condition attention network. The sequence attention network is designed to mine historical time series factors, which affect airport delays and generate the sequence correlation feature vector. The condition attention network is designed to capture the internal mechanism of various condition factors and generate the condition correlation feature vector. The fusion feature vector is obtained and delivered to the decoder for airport delay prediction by fusing the sequence and condition correlation feature vectors. The main contributions of this paper can be summarized as the following two points.

(1) A multi-step regression prediction method for airport delay is proposed in this paper, which can capture the historical sequence features and condition features on airport delay, portraying the influence mechanisms of different factors on airport delay from different dimensions.

(2) A Bayesian optimization method is devised to select hyperparameters for the DA-BILSTM, which avoids the subjectivity of manual parameter adjustment and the uncertainty of parameter adjustment time.

The remainder of this study is organized as follows. Section II gives a brief overview of airport delay prediction. Section III introduces the airport delay prediction model based on the attention fusion network. Section IV shows how the Bayesian optimization algorithm optimizes DA-BILSTM. Evaluations and comparisons are presented and discussed in Section V, and Section VI concludes this study and suggests potential future work.

II. Related Work

Airport delay, regarded as crucial indicators in transportation systems, have received substantial attention in the field of air transportation through extensive research.

Wu et al. [18] analogized the process of spreading delay status among airports to the process of spreading infectious diseases and divided airports into susceptible and infected states, according to which the susceptible airports in the process of spreading airport delays are predicted. Considering the continuous updating of airport flight data, Rodriguez-Sanz et al. [19] introduced a method to model the causal relationship between airport arrival performance indicators and meteorological events, which could be used to quantify the impact of weather on airport arrival conditions, predict the evolution of airport operational scenarios and support the airport decision making process. Mukherjee et al. [20] proposed a logistic regression and decision tree-based airport delay prediction method considering meteorological conditions and planned traffic demand. Validation of actual data from Newark Liberty International Airport and San Francisco International Airport showed that the proposed airport delay prediction method outperformed existing stochastic prediction methods. Wang et al. [21] focused on the influence of weather factors on airport delays, introduced Gaussian mixture models and OTSU methods to improve the weather-influenced traffic index model, and proposed an improved weather-influenced traffic index model based on the improved weather-influenced traffic index model. Mo et al. [22] considered the demand for predicting medium and long-term flight delays and established an ARIMA model based on a time series prediction algorithm. Then they validated the prediction performance of the proposed method using historical data of flights at a sizeable domestic airport. Xie et al. [23] used an error back propagation neural network to classify airport departure delays by developing two classes based on delay frequency and delay time delineation method.

Statistical-based airport delay prediction methods presented above infer airport delay by analyzing the relationship between variables. Although the calculation is simple and the time complexity is low, the prediction accuracy could be worse, and the stability of the prediction method is poor in the condition of large and high-dimensional airport delay data [24]. Fortunately, the development of machine learning has brought a new turn to the airport delay prediction problem. Recurrent Neural Network (RNN) [25] has excellent advantages in learning nonlinear features of sequences by introducing the concept of sequence into the network structure. The LSTM network [26] has been improved for the network gradient explosion and gradient disappearance problems of RNN and is more suitable for processing and predicting long-distance time sequence information with long intervals and delays in time sequences and outperforms other mainstream networks in airport delay prediction problems. Liu et al. [27] proposed a multi classification airport delay prediction method based on LSTM to solve the difficulty of capturing delay sequence information in traditional prediction models. Li and Jing [7] developed a prediction framework that utilizes LSTM units to capture temporal characteristics related to crowdedness and weather conditions. They employed Random Forest as a classifier to predict flight delays, achieving a classification accuracy of up to 92.39%. Li et al. [6] considered the spatial correlations between airports along with the temporal correlations among timestamps and proposed a CNN-LSTM deep learning framework to predict airport delays. The performance of the proposed method was experimentally demonstrated to be better than several benchmark models.

However, the above research methods based on statistics and machine learning tend to classify airport delays as classification problems, only rating the delay level of the airport. The information provided by predicting the delay category is not enough to meet the needs of airport decision-making, etc. For organizations such as airlines and logistics companies that require making complex business decisions, delay category prediction may not provide sufficiently accurate data for detailed business impact analysis [28]. In the context of practical applications, the precise estimation of delay times specific to each airport is more informative in facilitating the scheduling process at airports. The method proposed in [29] predicted the airport delay time but lacked the mining of airport delay state characteristics and only predicted the airport delay time based on the temporal sequence of airport delay data. A long short-term memory network of delay prediction with an attention mechanism is proposed in [12] to predict flight delays, which can focus on input data combined with the attention vector to capture the critical time points, making the prediction more accurate. Wang et al. [30] proposed a novel "Bubble" mechanism, which utilizes clustering analysis and data-driven sampling methods to construct new spatiotemporal features. They combined outbound and inbound prediction with trajectory pattern prediction using random forests, aiming to accurately forecast the mid-term Estimated Time of Arrival for Multi-Airport System. Yan et al. [31] proposed a method named MASTNet that combines multiple views to control information propagation between airports for airport arrival flow prediction. They automatically learn an adjacency matrix using flight duration and schedule factors, which is then input into specialized graph convolutional blocks to capture temporal dependencies. Qu et al. [32] considered the spatial dimensional information within the data and proposed a flight delay prediction model based on Att-Conv-LSTM. In this model, the LSTM network is utilized to capture temporal characteristics, the convolutional neural network is employed to extract spatial features, and an attention mechanism module is incorporated to enhance the network's computational efficiency. The aforementioned methods have not investigated multi-step prediction of airport delays, which can give the airport a longer flight planning and scheduling time.

This paper builds a Seq2Seq framework and designs a sequence attention network and a condition attention network to capture the effects of historical sequence factors and condition factors on airport delay time, which considers the multi-dimension of airport delay influence factors. Airport decision-making can be improved by multi-step prediction. The Bayesian optimization algorithm is introduced to obtain the optimal hyperparameters of the proposed method.

The comparisons between our work and the previous works are listed in Table I.

III. Methodology

This section describes the structure of the proposed airport delay prediction model DA-BILSTM in detail. The structure of DA-BILSTM can be divided into the following two parts: the Seq2Seq network and the attention fusion network. The Seq2Seq network consists of an encoder and a decoder. The encoder is responsible for learning the hidden features of the airport delay data. Furthermore, the decoder is responsible for connecting the hidden state of the decoder's current time step with the fusion feature vector to assist the airport delay prediction. The fusion feature vector is calculated according to the fusion network. The attention fusion network is composed of a sequence attention network and a condition attention network, which are respectively responsible for extracting the deep time sequence features and condition features of airport delay history data, and fusing the sequence correlation feature vector and condition correlation feature vector to obtain the fusion feature vector. The airport delay prediction model based on the attention fusion network is shown in Fig. 1.

A. Problem Formulation

The method to predict the airport delay is studied in this paper. This paper defines airport delay as the average hourly arrival delay of flights in an airport. In airport delay prediction, supposing that the input sequence is $\{x_{t-D+1}, x_{t-D+2}, \cdots, x_t\}$, where $x_i \in R^n$ represents the observed value at time i and D is the window size. Then, the task is to predict the airport delay $y_{t+\Delta}$, where Δ is the fixed horizon [33]. And the predicted value of airport delay is denoted as $\hat{y}_{t+\Delta}$.

B. Sequence to Sequence Network

As the airport delay data is naturally chronological, our method adopts the Seq2Seq network [34]. According to the delay propagating chain, the airport delay at the predicted time is jointly affected by the delays before and after it. Moreover, the bidirectional long short-term memory network (BILSTM) [35] can model context information and has stronger information memory ability. Hence, BILSTM is used for encoding and decoding in this paper [36], [37], [38], and [39].

As shown in Fig. 2, BILSTM consists of two unidirectional and oppositely oriented LSTM networks. $\{x_{t-D+1}, x_{t-D+2}, \cdots, x_t\}$, $x_i \in R^n$ represents the input data sequence and h_i is the output of the hidden layer at each time step. The i-th hidden state of BILSTM h_i combines the hidden states in the forward and backward directions and can be represented as $h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}]$, which can synthesize the modeling context information and enhance the long-term information memory capability [40].

Suppose the input of the encoder is x_t and the hidden state of the previous step is h_{t-1}, the output is the hidden state of the current moment h_t, then,

$$h_t = \text{BILSTM}_{enc}(x_t, h_{t-1}). \tag{1}$$

The decoder is responsible for receiving the output sequence of the previous time step y_{t-1}, the hidden state s_{t-1} of the decoder in the previous time step $t-1$ and the fusion feature vector $c_{mix,t}$, and calculating the hidden state s_t for time t, as shown in Equation (2):

$$s_t = \text{BILSTM}_{dec}(y_{p,t-1}, c_{mix,t}, s_{t-1}). \tag{2}$$

TABLE I
SUMMARY OF PREVIOUS RESEARCH ON AIRPORT DELAY PREDICTION

Reference	Airport system		Prediction horizon		Prediction target		Method
	Single airport	Multi airport	Single-step	Multi-step	Delay Severity Level	Precise Delay Value	
Mukherjee et al. [20]	√		√		√		Logistic Regression
Mo et al. [22]	√		√			√	ARIMA
Xie et al. [23]	√		√		√		BP Neural Network
Liu et al. [27]	√		√		√		LSTM
Li et al. [7]		√	√		√		LSTM + Random Forest
Li et al. [6]		√	√		√		CNN-LSTM
Li et al. [29]	√			√		√	TS-LSTM
Wang et al. [12]	√		√			√	LSTM + Attehtion
Wang et al. [30]		√		√		√	ARIMA + BISLTM
Yan et al. [31]		√		√		√	MASTNet
Qu et al. [32]	√		√			√	Att-Conv-LSTM
our work	√			√		√	DA-BILSTM

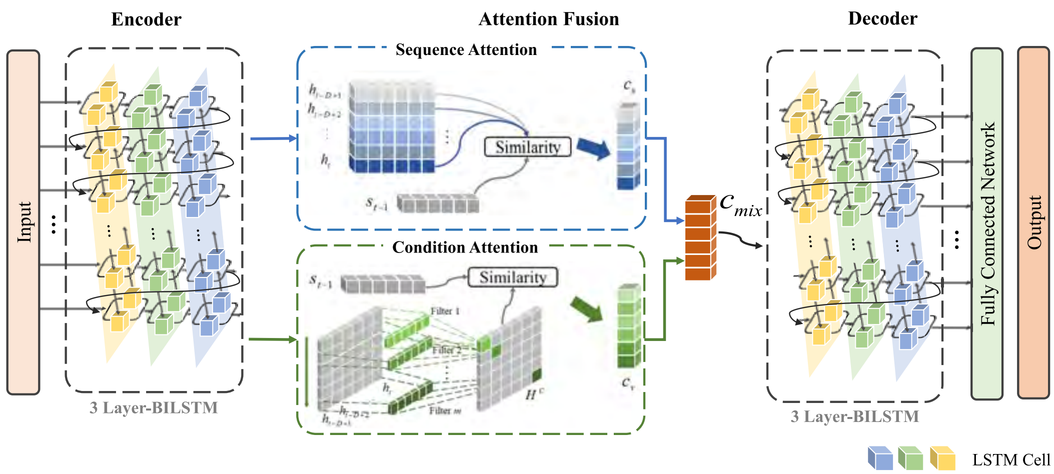


Fig. 1. Airport delay prediction model based on attention fusion network.

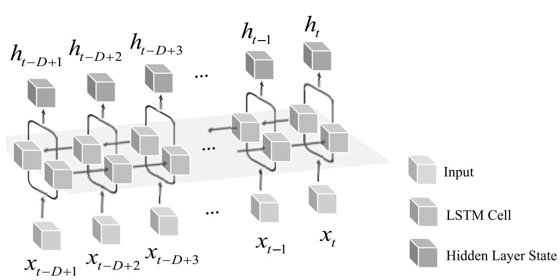


Fig. 2. BILSTM network structure.

Finally, the fusion feature vector $c_{mix,t}$ is concatenated with the hidden layer state output s_t of the BILSTM decoder [41], and the prediction results are computed by a layer of the fully connected network as follows:

$$\hat{y}_i = \tanh(W_c[c_{mix,t}; s_t]), \tag{3}$$

where W_c is the weight.

The complete process of DA-BILSTM is described in Algorithm 1.

Algorithm 1 DA-BILSTM

Require: The input sequence $\{x_{t-D+1}, x_{t-D+2}, \cdots, x_t\}$;
Ensure: The predicting value $\hat{y}_i$;

Initialize: Encoder, Decoder;
for time step $t = 1$ to n **do**
  if $t = 1$ **then**
    Encode $\{x_{t-D+1}, x_{t-D+2}, \cdots, x_t\}$ to obtain the encoder output h_t;
    Calculate the sequence correlation feature vector $c_{s,t}$ and the condition correlation feature vector $c_{v,t}$;
    Fusing $c_{s,t}$ and $c_{v,t}$ to obtain the fusion feature vector $c_{mix,t} = c_{s,t} + c_{v,t}$;
    Get decoder output $s_t = \text{BILSTM}_{dec}(c_{mix,t}, h_t)$;
  else
    $s_t = \text{BILSTM}_{dec}(y_{p,t-1}, c_{mix,t}, s_{t-1})$;
  end if
  Obtain the prediction value $\hat{y}_i = \tanh(W_c[c_{mix,t}; s_t])$;
end for

C. Attention Fusion Network

Due to a large amount of airport data and the variety of influencing factors, the dimension of the input data is high,

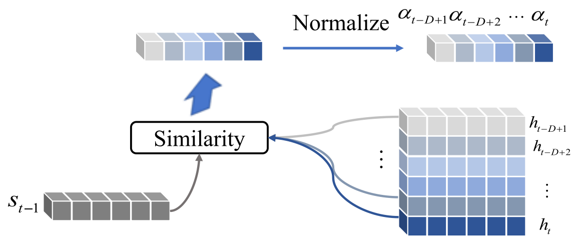


Fig. 3. Sequential attention network structure.

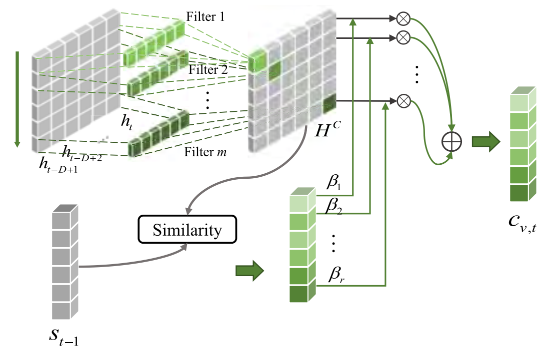


Fig. 4. TPA network structure.

and it is difficult to obtain effective data information directly. Therefore, this paper designs the attention fusion network, which is composed of a sequence attention network and a condition attention network.

1) Sequence Attention Network: The sequence attention network introduces an attention mechanism to automatically and dynamically capture the influence mechanism of the airport delay history information. Assuming that the data window is $[t-D+1, t]$, where D is the data window size to be determined, and h_i is the hidden state of the encoder at the i-th time step, where $h_i \in \mathbb{R}^m$. The structure of the sequence attention network is shown in Fig. 3.

For the decoding time step t, the hidden state of the decoder is s_{t-1}, and the scoring function is calculated as follows:

$$f(h_i, s_{t-1}) = {s_{t-1}}^{\mathrm{T}} W_\alpha h_i, t-D+1 \leq i \leq t, \tag{4}$$

where W_α is the sequence attention network parameter. The weight α_i of each h_i is computed by

$$\alpha_i = \frac{\exp(f(h_i, s_{t-1}))}{\sum_{j=t-D+1}^{t} \exp(f(h_j, s_{t-1}))}, \tag{5}$$

where α_i represents the sequence correlation feature weights of different airport delay data. The hidden state h_i is weighted by α_i and the sequence correlation feature at time step t is obtained as follows:

$$c_{s,t} = \sum\nolimits_{i=t-D+1}^{t} \alpha_i h_i, \tag{6}$$

where $c_{s,t} \in \mathbb{R}^m$.

2) Condition Attention Network: There are many potential factors that cause delays, such as time condtition, weather condition, air traffic control signals, and many others. These various factors do not equally affect airport delay and are intertwined together to affect the airport delay condition. The sequence attention network fails to detect condition patterns useful for airport delay prediction. In this case, this paper introduces the Temporal Pattern Attention (TPA) network [33] for local feature learning to dynamically capture the impact of different factors on the execution of that flight. The TPA network structure is shown in Fig. 4.

The original airport delay data is processed by the encoder to obtain the hidden state h_i at each time step, to obtain the hidden state matrix $H = \{h_{t-D+1}, h_{t-D+2}, \cdots, h_t\}$. For the hidden state matrix, the row vector represents a state factor at all time steps such as the weather condition while the column vector represents all state factors in a certain time step.

In the TPA network, the input hidden layer is firstly subjected to feature extraction through a one-dimensional convolutional layer, and the size of the convolutional kernel is set to $1 \times T$. T indicates the range covered by the attention mechanism, which is believed $T = D$. The number of convolutional kernel C is m. The convolutional kernels are convolved along the row vectors of the output matrix of the hidden layer to obtain the condition feature matrix H^C, where $H^C \in \mathbb{R}^{m\times k}$, which is calculated as:

$$H^C_{r,k} = \sum_{l=1}^{D} C_{k,T-D+l} * H_{r,(t-D-1+l)}, \tag{7}$$

where, $H^C_{r,k}$ denotes the condition feature value obtained by convolving the r-th row vector with the k-th convolution kernel. Then, scoring the condition feature matrix can obtain the impact weights of different factors on airport delay, which can be calculated as:

$$\beta_r = sigmoid(f(H^C_r, s_{t-1})), \tag{8}$$

where H^C_r and s_t are both m-dimensional vectors, then the scoring function $f(H^C_r, s_{t-1})$ is defined as follows:

$$f(H^C_r, s_{t-1}) = (H^C_r)^{\mathrm{T}} s_{t-1}. \tag{9}$$

The condition relevance feature vector $c_{v,t}$ at time step t is obtained by making β_r and the corresponding hidden layer output vector H^C_r multiply and sum, as shown in Equation (10), where H^C_r is the r-th row of the state feature matrix.

$$c_{v,t} = \sum\nolimits_{i=1}^{r} \beta_i H^C_i. \tag{10}$$

3) Attention Fusion Procedure: As shown in Fig. 5, the vector obtained from the sequence attention network and the output of the condition attention network are fused to obtain the fusion feature vector $c_{mix,t}$, which is calculated as follows:

$$c_{mix,t} = c_{s,t} + c_{v,t}, \tag{11}$$

where $c_{s,t}$, $c_{v,t}$, and $c_{mix,t} \in \mathbb{R}^m$. Finally, the fusion feature vectors are delivered to the decoder, where the decoder hidden layer states are connected for decoding to obtain the prediction results of airport delay time.

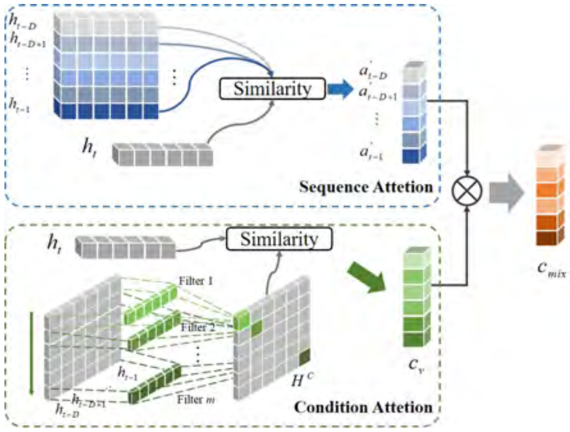


Fig. 5. Attention fusion network structure.

IV. Hyperparameter Optimization Algorithm

The selection of hyperparameter values in a deep learning model is crucial to the model's performance [42]. The hyperparameters in deep learning are often determined manually based on experience and data analysis [43]. Moreover, grid search, random search, and Bayesian optimization algorithms [16] are also commonly used for hyperparameter optimization.

Grid search determines the optimal value by finding all points on the grid in the hyperparameter space. Grid search is computationally intensive, especially when the number of hyperparameters is enormous. Random search finds the global optimum by taking random sample points in the search range. Random search is generally faster than grid search. However, each search step of grid search and random search did not consider the situation of the explored points. These methods usually require extensive experiments to determine the optimal combination of hyperparameters for the model. The Bayesian optimization algorithm uses the available experimental data, dramatically improves the efficiency and effectiveness of hyperparameter estimation, and has a more sound theoretical basis and convergence guarantee. Compared with other hyperparametric optimization methods, the Bayesian optimization algorithm finds optimal solutions faster on average, requires less computational resources, and can identify reasonably good solutions.

The Bayesian optimization algorithm uses a continuously updating probabilistic model and updates the posterior probabilities of the optimization function a few times of the objective function evaluation [44]. Finally, the optimal combination of network model hyperparameters is obtained. The objective function is denoted as $f_{obj}(u)$. The set of hyperparameter combinations is denoted as $U = \{u_1, u_2, \cdots, u_n\}$, and u^* denotes the combination of hyperparameters that makes $f_{obj}(u)$ achieving the maximum value, then the Bayesian optimization formulation [45] can be expressed as:

$$u^* = \arg\max_{u \in U} f_{obj}(u). \tag{12}$$

Based on the Bayesian theorem, the probability distribution of the Bayesian optimization process is obtained to satisfy:

$$P(f|D_{1:t}) \propto P(D_{1:t}|f)P(f), \tag{13}$$

where f denotes the unknown objective function. $D_{1:t} = [(u_1, y_{observe,1}), (u_2, y_{observe,2}), \cdots, (u_t, y_{observe,t})]$ denotes the set of observed parameters and observations, where u_t denotes the observed parameter combination and $y_{observe,t}$ denotes the observations. $P(f|D_{1:t})$ denotes the posterior probability. $P(D_{1:t}|f)$ denotes the likelihood distribution, and $P(f)$ denotes the prior probability of f.

In this paper, the Bayesian optimization algorithm is used to optimize DA-BILSTM. The steps of optimizing the DA-BILSTM using the Bayesian optimization algorithm are as follows.

Step 1: Determine which of the DA-BILSTM's hyperparameters should be optimized. Set the hyperparameter range, randomly generate initialized sample points, input the initialized sample points into the Gaussian process, train the DA-BILSTM, and use the loss function value of the objective function of the DA-BILSTM to modify the proxy Gaussian model, so that the proxy model is gradually closer to the real function distribution.

Step 2: Select the next set of sample points to be evaluated in the modified proxy model using the acquisition function. And input the new sample points into DA-BILSTM for training to obtain the new output value of the objective function $D_{1:t}$. Thereby updating the set of samples and the proxy model.

Step 3: If the loss value of the objective function corresponding to newly selected sample points u_i is the minimum, the algorithm will be terminated. The optimal combination of parameters for the current model is obtained.

Step 4: If the objective function value corresponding to the newly selected sample point u_i does not meet the requirement, $(u_i, y_{observe,i})$ is added to the sample set. Then proceed Step 2 to continue the correction of the Gaussian model until the requirement is met.

V. Experiments and Analyses

The experiment is based on the deep learning framework PyTorch 1.10.1 and written in Python language. The computer hardware configurations used are as follows: the processor is AMD Ryzen 53500 $\times$ 6-core processor. The memory is 16.00 GB, and the operating system is windows 10. The workflow of the data pre-processing, and how the DA-BILSTM is obtained is shown in Fig. 6.

A. Datasets

In order to accurately test the proposed method, this paper conducts simulation experiments based on two real flight datasets.

Dataset 1 is the operational records of Atlanta airport in 2015, in which the operational flight records were obtained from the Bureau of Transportation Statistics (BTS) of the United States 2015. Airline On-Time Performance (AOTP) records and meteorological records are from Quality Controlled Local Climatological Data (QCLCD) provided by the National Climatic Data Center (NCDC). We employ the dataset fusion method [46] to combine flight records and meteorological records. Dataset 2 is obtained from the operational records of an airport in China in 2019 from Feiyou Technology

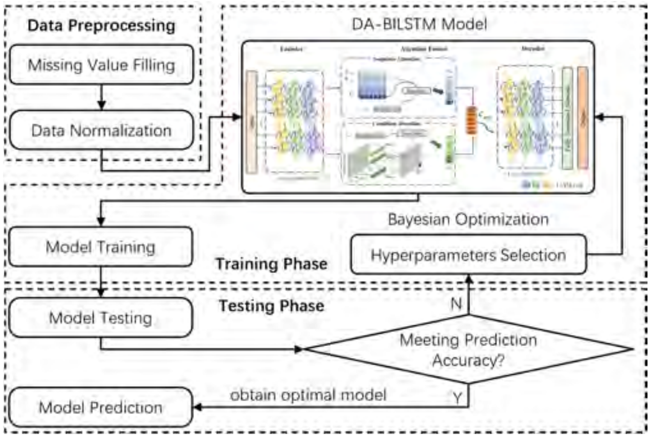


Fig. 6. The workflow of DA-BILSTM.

TABLE II
CLASSIFICATION OF THE INFLUENCING FACTORS IN DATASETS

Factor Classification	Specific Factors
Flight Factors	The average route distance, average sailing time, average transit time, average inbound taxiing time, and average outbound taxiing time.
Weather Factors	The average temperature, average humidity, average wind speed, average air pressure, and average visibility of the airport per hour.
Time Factors	The month, day of the month, day of the week, and week of the year when the flight takes off.

Company (https://data.variflight.com), which contains 208,152 records. Table II presents the three major categories of airport delay factors.

Flight factors: The flight factors record the flight operations. For example, the average route distance represents the average route distances of all the flights in this airport per hour. And the average sailing time represents the average sailing time of all the flights in this airport per hour.

Weather Factors: The weather factors consider weather changes, such as humidity, temperature, etc. Weather conditions affect the operation of airport flights during that period. For example, airport flights have to be delayed during extreme weather to ensure the safety of airport flights.

Time Factors: The time factors analyze the specific time of the flight from different dimensions, which can also influence the airport flight operation. Due to the difference in flight volumes between weekdays and weekends during the week, the probability of delays at the airport will also change as the number of flights changes.

When dealing with missing values, the direct deletion of missing values may lose important information. Given the temporal nature of the data in the dataset, we replace the missing values with the average of the data before and after the column in which the missing values are located. To avoid the impact of the difference in magnitudes in the data on the prediction accuracy, the Min-Max Normalization method is adopted in the datasets, which is calculated as follows:

$$x^* = (x - min) \,/\, (max - min)\,. \tag{14}$$

Fig. 7. The flowchart for splitting data for regression.

We divide both datasets into two parts at a ratio of 8:2, with one part used for training and the other part as the test set [47]. During training, the K-fold (K=8) cross-validation is used to split the training dataset into 8 equal subsets, conducting 8 experiments by selecting one subset as the validation set and the remaining subsets as the training set each time. The flowchart for splitting data is shown in Fig. 7.

B. Verification of DA-BILSTM Structure

In this paper, the backpropagation algorithm is used to train the network, and the RMSprop optimizer is selected to train and obtain the values of the model parameters. To evaluate the performance of the model, mean absolute error (MAE), root mean squared error (RMSE), mean squared error (MSE), and normalized root mean squared error (NRMSE) are chosen as the evaluation criteria of the method performance [23] in this paper, and the calculation formulas are shown in Equation (14), Equation (15), Equation (16), Equation (17) and Equation (18) [13]. $\hat{y}_i$ denotes the predicted value of airport delay time and y_i denotes the real delay time.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|(\hat{y}_i - y_i)|, \tag{15}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}, \tag{16}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2, \tag{17}$$

$$NRMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\widehat{y_i} - y_i)^2} \Big/ \frac{1}{N}\sum_{i=1}^{N} y_i, \tag{18}$$

$$MAPE = \frac{1}{N}\left(\sum_{i=1}^{N}|\frac{\hat{y}_i - y_i}{y_i}|\right). \tag{19}$$

The proposed DA-BILSTM uses a three-layer BILSTM network as the encoder-decoder network, and to seek the best neural network structure and to determine the degree of influence of different attention mechanisms on the model performance, the network structures are divided into four components:

(1) Seq2Seq: the original sequence-to-sequence network without an attention network.

(2) Seq2Seq+SA: the Seq2Seq network with sequence attention network.

TABLE III
COMPARISON OF MODEL STRUCTURE EFFECTS ON PREDICTION OF A SINGLE TIME STEP

Network	Dataset	MAE(min)	RMSE(min)	MSE	NRMSE
Seq2Seq	Dataset 1	10.94	12.09	146.17	0.13
	Dataset 2	14.42	15.33	235.01	0.30
Seq2Seq+SA	Dataset 1	6.40	7.21	51.98	0.09
	Dataset 2	12.52	13.26	175.83	0.17
Seq2Seq+CA	Dataset 1	6.72	7.95	63.20	0.04
	Dataset 2	12.51	13.37	178.76	**0.06**
DA-BILSTM	Dataset 1	**4.90**	**5.36**	**28.73**	**0.02**
	Dataset 2	**9.26**	**9.84**	**96.83**	0.14

Note: MAE represents the prediction error of a single time step, and the bold number represents the lowest error value.

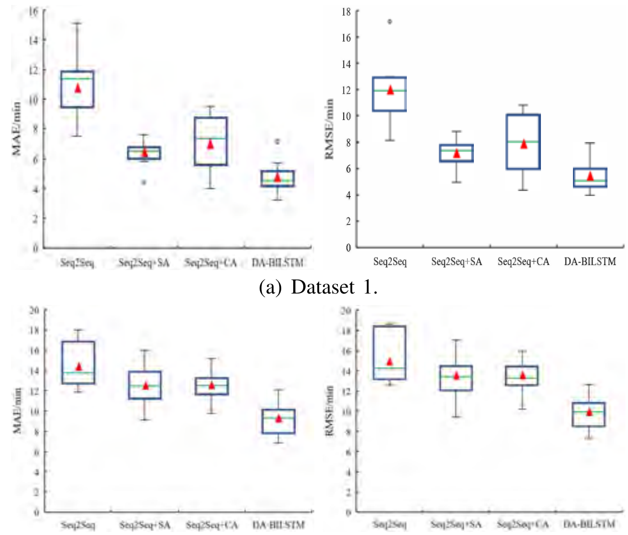
(3) Seq2Seq+CA: the Seq2Seq network with condition attention network.

(4) DA-BILSTM: the Seq2Seq network with both sequence attention network and condition attention network.

Based on Dataset 1 and Dataset 2, the performance of the Seq2Seq network without attention network, Seq2Seq+SA network with only sequence attention network, Seq2Seq+CA network with only condition attention network, and DA-BILSTM network with both sequence attention network and condition attention network are compared for single-step prediction of airport delay, and the results of 100 single-step prediction experiments are shown in Table III.

Table III illustrates that the MAE values of the delay prediction on the two datasets for Seq2Seq without attention mechanism are 10.94 minutes and 14.42 minutes respectively. The RMSE values on the two datasets are 12.09 minutes and 15.33 minutes respectively. The MSE values on the two datasets are 146.17 and 235.01, and the NRMSE values are 0.13 and 0.30 respectively. After adding the sequence attention network and condition attention network respectively to Seq2Seq, the MAE values, the RMSE values, the MSE values, and the NRMSE values of the delay time prediction on the two datasets are reduced. When the attention fusion network is embedded into the original Seq2Seq model, the prediction performance of the model is significantly improved.

To analyze the impact of the attention network on the airport delay prediction model, the results of DA-BILSTM with 100 sets of single-step predictions are plotted in Fig. 8. It can be seen that the median MAE values and RMSE values of the Seq2Seq model without an attention network are the highest among the four models on both datasets. The data distribution is relatively scattered, and the prediction is the least accurate. The introduction of the sequence attention network and condition attention network reduces the median MAE values and RMSE values on the two datasets. When the attention fusion network is embedded, the model extracts the effects of both airport historical data and condition data on airport delay time, which results in the best model performance with the lowest median MAE values and RMSE values on both datasets. Moreover, the model is the most robust and maintains accuracy in predicting airport delay time for different moments.



(a) Dataset 1.

(b) Dataset 2.

Fig. 8. Box diagram of the prediction error of different model structures.

TABLE IV
BAYESIAN OPTIMIZATION PARAMETERS

Parameter Setting Method	BILSTM unit	Step	Batch size	Dropout	Learning rate
Manual Setting	256	5	128	0.1	0.01
Bayesian Optimization	64	20	200	0.3	0.01
Grid search [16]	32	5	64	0.1	0.01
Random search [16]	128	10	256	0.4	0.1

TABLE V
COMPARISON OF NETWORK MODEL LOSS VALUE WITH DIFFERENT PARAMETER SETTING METHODS

Parameter Setting Method	K-fold	Dataset 1		Dataset 2	
		Loss	Epoch	Loss	Epoch
Manual Setting	K=8	0.40	800	0.40	1000
Bayesian Optimization	K=6	0.38	375	0.44	250
	K=8	0.35	400	0.40	375
	K=10	0.36	600	0.40	600
Grid Search	K=6	0.56	800	0.60	800
	K=8	0.50	1000	0.58	1000
	K=10	0.46	1200	0.52	1000
Random Search	K=6	0.66	400	0.72	300
	K=8	0.64	475	0.66	375
	K=10	0.64	500	0.70	425

Note: Loss represents the prediction error of a single time step after normalization, and Epoch represents the epoch value when the loss curve converges.

C. Verification of Bayesian Optimization

To verify the performance of different optimization methods on the model, four hyperparameter optimization methods, the manual setting, Bayesian Optimization, Grid Search, and Random Search, were chosen in this article. Based on the datasets, the Bayesian optimization algorithm, the Grid Search and the Random Search were used to optimize the number of LSTM neurons, time steps, batch size, dropout, and learning rate, which have a large impact on the network. The number of LSTM neurons is set in the range of (1, 512). The time step is set in the range of (1, 64). The size of batch size is set in the range of (16, 256). The size of dropout is set in the range of (0.1, 0.8), and the learning rate is set in the range of (0.00001, 0.1). The manual setting method selects the settings by turning the reference experience. Table IV shows the optimal values of the hyperparameters of the DA-BILSTM model obtained by the manual setting method and Bayesian optimization.

The different models obtained by the four parameter setting methods are iteratively trained 1000 times on Dataset 1 and Dataset 2, with K-fold values set to 6, 8, and 10, respectively.The comparative loss values of the validation process are obtained as shown in Table V. When the hyperparameter values are determined by the manual method, the converged loss value is 0.40 on both datasets. The Bayesian optimization algorithm and the Random search algorithm achieve the optimal tuning effect when the value of K is set to 8, while the Grid search algorithm achieves the optimal tuning effect when the value of K is set to 10. When K is set to 8, the Bayesian optimization algorithm results in converged loss values of 0.35 and 0.40 on the two datasets, making it the best-performing among the four parameter setting methods. Hence,the Bayesian optimization algorithm is used to optimize the parameters and K is set to 8 in this article.

In summary, the DA-BILSTM model based on Bayesian optimization converges with fewer iterations than the original DA-BILSTM model, and the loss values of the model after convergence are smaller. What is more, the manual setting method of setting hyperparameters needs to keep trying suitable combinations of hyperparameter values based on experience, and it causes a lot of time consumption.

D. Model Comparison Experiment

To validate the prediction performance of our proposed method DA-BILSTM for airport delay, DA-BILSTM is compared with five methods [9], [22], [26], [29], [48]. ARIMA and LSTM are classic time series data prediction algorithms and are frequently used as airport delay prediction baselines in many studies [22], [26]. GRU, CNN-BILSTM, and TS-LSTM have been state-of-the-art methods in the field of airport delay prediction with excellent performance in recent years [9], [29], [48].

(1) ARIMA: autoregressive integrated moving average method, which converts the time series into a smooth time series after differential processing and then fits the prediction.

(2) LSTM: a variant of RNN, commonly used for predicting time series data.

(3) GRU: a method used in [9], which performs well in predicting flight departure time.

(4) CNN-BILSTM: a method used in [48], which uses CNN-BILSTM to integrate information in both spatio-temporal directions and focuses on similar data through an attention network, performs well in predicting time series data.

(5) TS-LSTM: a method proposed in [29], which considers the correlation of associated airports in time and space and is able to capture the delay propagation mechanism from high-dimensional variables adequately.

TABLE VI
COMPARISON OF THE AVERAGE PREDICTION ERROR OF DIFFERENT METHODS

Model	Dataset 1			Dataset 2		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
ARIMA	13.80	25.50	1.30	16.05	28.64	2.21
LSTM	10.55	19.47	1.02	13.76	24.13	1.59
GRU	9.52	19.24	0.98	11.94	23.76	1.12
CNN-BILSTM	8.97	**17.32**	0.21	11.76	21.45	0.39
TS-LSTM	9.23	18.87	**0.16**	12.07	21.85	**0.29**
DA-BILSTM	**8.60**	17.73	0.18	**10.28**	**20.84**	0.35

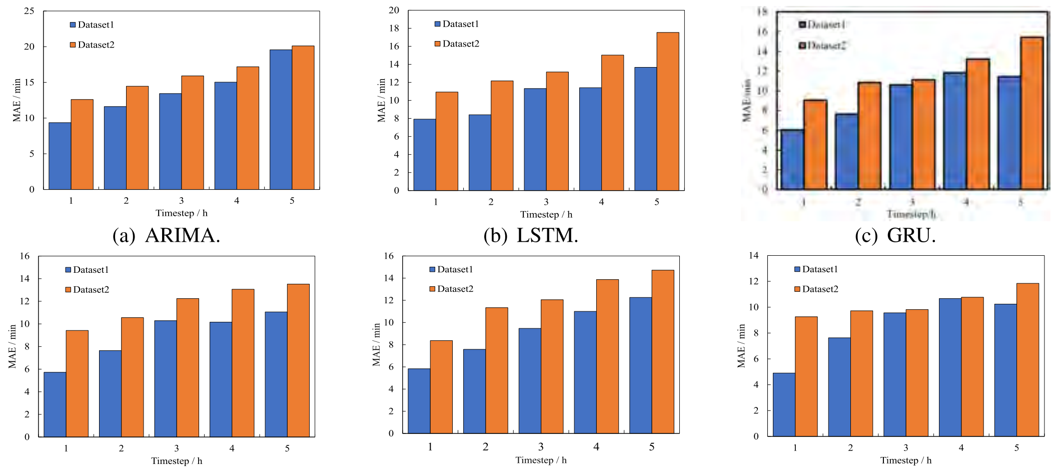
Note: MAE(min) and RMSE(min) represent the average of the prediction errors in the forward five time steps (from t+1 to t+5), and the bold number represents the lowest error value.

To ensure a fair comparison, the Bayesian optimization algorithm is likewise used to optimize the hyperparameters of those baseline methods. The hyperparameters used by the baseline methods are tuned to achieve their best predicting performance. The optimal hyperparameters are as follows.

In the ARIMA method, the auto-regressive term number is 3, the sliding average number of terms is 1, and the step number is 3. In the LSTM method, the LSTM unit is 64, the timestep is 20, the batch size is 200, the dropout is 0.3, and the learning rate is 0.01. In the GRU method, the GRU unit is 50, the timestep is 20, the batch size is 32, the dropout is 0.3, and the learning rate is 0.01. In the CNN-BILSTM method, the filter number is 64, the kernel size is 20, the LSTM unit is 64, the timestep is 20, the batch size is 200, and the learning rate is 0.01. And in the TS-LSTM method, the LSTM unit is 80, the LSTM layer is 2, the timestep is 20, the batch size is 200, and the learning rate is 0.01.

Based on the two real datasets, the overall prediction performance of DA-BILSTM is compared with other comparison methods. The experiment results are shown in Table VI, where the average error MAE and RMSE results are shown for the multi-step prediction of airport delay in the next five time steps from $t+1$ to $t+5$ for the DA-BILSTM and the comparison methods. Table VI shows that the DA-BILSTM outperforms the other comparison methods in terms of the multi-step prediction performance of airport delay.

Compared with the shallow learning model, the DA-BILSTM maintains the highest prediction performance for the next five steps from $t+1$ to $t+5$. That is because ARIMA cannot capture the complex relationships among airport delay factors. The benchmark deep learning model, LSTM and GRU, predict more accurately than ARIMA but predict less accurately than other models compared in this paper. Because LSTM and GRU only capture the sequence information but fails to learn the condition factors. A comparison of deep learning models shows that the overall prediction accuracy of DA-BILSTM is higher than CNN-BILSTM and TS-LSTM. That is because TS-LSTM cannot learn the influence weights of factors adaptively



(a) ARIMA. (b) LSTM. (c) GRU.
(d) CNN-BILSTM. (e) TS-LSTM. (f) DA-BILSTM.

Fig. 9. Comparison of airport delay prediction errors of different methods.

TABLE VII
COMPARISON OF THE PREDICTION ERROR OF DIFFERENT METHODS

Model	MAE (Dataset 1)					MAE (Dataset 2)				
	t+1	t+2	t+3	t+4	t+5	t+1	t+2	t+3	t+4	t+5
ARIMA	9.36	11.62	13.48	15.03	19.56	12.60	14.46	15.90	17.19	20.11
LSTM	7.92	8.40	11.32	11.41	13.67	10.93	12.16	13.15	15.03	17.54
GRU	6.05	7.66	10.61	11.83	11.46	9.07	10.86	11.13	13.21	15.45
CNN-BILSTM	5.72	7.64	10.28	10.16	11.05	9.42	10.56	12.24	13.06	13.52
TS-LSTM	5.83	**7.58**	**9.47**	11.00	12.26	**8.37**	11.34	12.05	13.87	14.72
DA-BILSTM	**4.90**	7.63	9.56	**10.67**	**10.24**	9.26	**9.72**	**9.82**	**10.77**	**11.84**

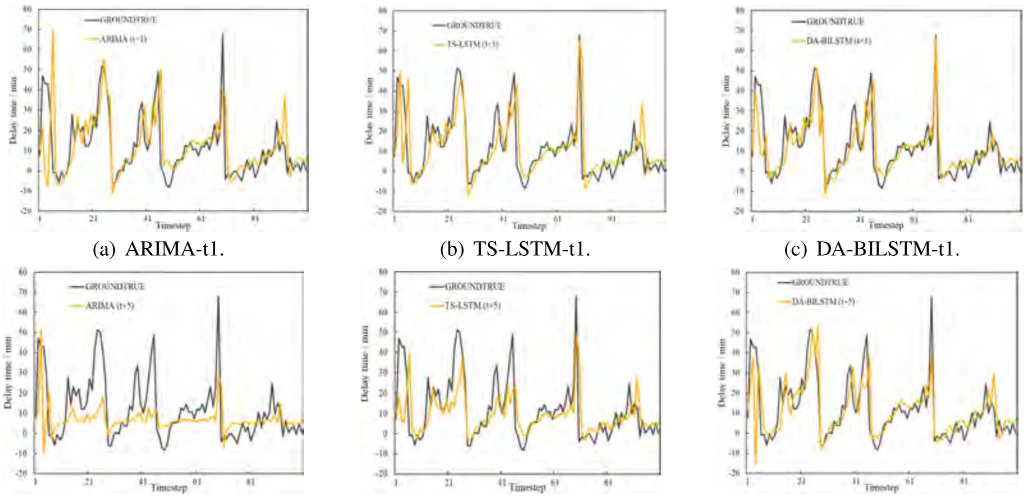
Note: MAE represents the prediction error of a single time step, and the bold number represents the lowest error value.

without an attention mechanism. And CNN-BILSTM cannot learn the influence weights of condition factors adaptively. DA-BILSTM is a deep learning model, which is able to portray the complex relationship between the factors of airport delays. Moreover, the attention fusion network of DA-BILSTM is more suitable for capturing the historical information on airport delay and the intrinsic relationship between the factors on the airport delay status that needs to be predicted.

Based on the comparison of the overall average prediction error of the two datasets, the comparison and analysis of the multi-step prediction performance of the two datasets are continually conducted. Table VII presents the MAE values of DA-BILSTM and other algorithms for predicting airport delay at time steps $t+1$, $t+2$, $t+3$, $t+4$, and $t+5$. Fig. 9 visually depicts different algorithms' comparative multi-step prediction performance on the two datasets. From Table VII and Fig. 9, it can be observed that DA-BILSTM outperforms ARIMA, LSTM, GRU, CNN-BILSTM, and TS-LSTM in terms of prediction performance at all five-time steps. ARIMA performs the worst among the compared methods due to its limited capability of capturing complex temporal features. As benchmark deep learning models, LSTM and GRU show better prediction performance than ARIMA, but they underperform compared to other deep learning methods. This is because LSTM and GRU can only capture time dependencies and fail to capture sequence and condition features that may affect airport delays. Especially in the case of long-time steps, the prediction performance of ARIMA and LSTM decreases significantly. LSTM-based state-of-the-art models like CNN-BILSTM-ATTENTION and TS-LSTM maintain high prediction accuracy under short time steps. However, the prediction error gradually increases when the time steps get longer. DA-BILSTM can still perform well in multi-step prediction, particularly exhibiting significant superiority over other algorithms at time steps $t+4$ and $t+5$.

By applying the identical model to both datasets, we observe that ARIMA, LSTM, GRU, CNN-BILSTM, TS-LSTM, and DA-BILSTM demonstrate superior overall predictive performance on Dataset 1 compared to Dataset 2. Despite both datasets containing variables of the same nature, there exists a disparity in data balance between them. Dataset 1 exhibits a higher level of balance in contrast to Dataset 2, as the number of observations varies significantly across different delays, resulting in an uneven distribution of data. Notably, DA-BILSTM achieves superior performance on both Dataset 1 and Dataset 2, indicating its robust generalization ability. However, it is worth noting that the prediction performance of DA-BILSTM diminishes in the presence of imbalanced data within the dataset.

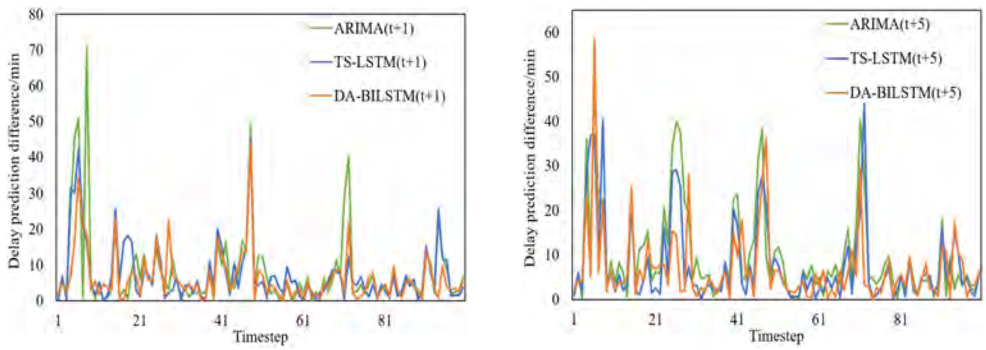
According to the experiment results, it can be found that the prediction time step has a significant influence on the prediction performance of the model. We further analyze and compare the prediction performance of ARIMA, which represents the shallow learning model, TS-LSTM, which represents the deep learning model, and DA-BILSTM for different time steps on Dataset 1. Fig. 10 illustrates the prediction errors of the three methods in predicting delay times under single-step prediction $t+1$ and multi-step prediction $t+5$ conditions. The horizontal axis indicates the observed time step, and the vertical axis indicates the delay time value. From Fig. 10, it can be seen that DA-BILSTM prediction accuracy outperforms ARIMA and TS-LSTM for different time-step



(a) ARIMA-t1. (b) TS-LSTM-t1. (c) DA-BILSTM-t1.

(d) ARIMA-t5. (e) TS-LSTM-t5. (f) DA-BILSTM-t5.

Fig. 10. Comparison of ground truth value and different time steps (t+1 and t+5) predicted delay time value of different models.



(a) (b)

Fig. 11. Comparison of airport delay prediction differences of different methods.

prediction cases. Furthermore, the disparity in performance among the various methods in single-step prediction is minimal. However, as the time steps increase, both ARIMA and LSTM experience a significant decrease in performance, while only DA-BILSTM maintains a relatively higher accuracy of airport delay prediction.

Fig. 11 illustrates the delay prediction differences among ARIMA, TS-LSTM, and DA-BILSTM under single-step prediction $t+1$ and multi-step prediction $t+5$ conditions. The green line represents the absolute difference between the predicted and actual delay times using ARIMA. Similarly, the blue line represents this difference for TS-LSTM, while the orange line represents it for DA-BILSTM. Analysis of Fig. 11 reveals that in the single-step prediction scenario, DA-BILSTM exhibits significantly lower prediction differences compared to the other methods, indicating higher accuracy. In the multi-step prediction condition, the three models' prediction difference is more significant than the single-step prediction. ARIMA exhibits the highest variability in its prediction difference curve, while DA-BILSTM demonstrates the least amount of variation, indicating relatively consistent performance in multi-step prediction.

VI. Conclusion and Future Work

In this paper, we present an attention fusion network-based deep learning framework DA-BILSTM for multi-step airport delay prediction. Specifically, an attention fusion network is designed to capture the correlation characteristics of historical information and condition information of airport delays. The Bayesian algorithm is introduced to optimize the model hyperparameters. The experimental results demonstrate that DA-BILSTM optimized by the Bayesian algorithm converges at lower loss values with fewer iterations.

The proposed DA-BILSTM is evaluated with two real-world datasets and compared against state-of-the-art methods. Our model achieved superior performance in multi-step airport delay prediction. The embedding of the attention fusion network dramatically improves the accuracy and robustness of the model for predicting airport delays, with average MAE reductions of 55.21% and 35.78% for the two datasets, respectively. The use of Bayesian optimization algorithms resulted in a 50% and 62.5% increase in model convergence speed on the two datasets. Compared with the optimal mainstream method, the average MAE of the two datasets is reduced by 4.12% and 12.59%, respectively. Furthermore, the average MAE is reduced by 7.33% and 12.42% for the multi-step prediction.

The proposed method could be extended further in future studies if relevant data is available. For example, considering the fact that spatial factors, such as the mutual propagation of delay between different airports, also influence airport delay prediction. It would be interesting to integrate the temporal and spatial factors to predict airport delay. Moreover, besides being suitable for airport delay prediction, the proposed model could also be applied to other temporal forecasting tasks.

References

[1] M. Kim and S. Park, "Airport and route classification by modelling flight delay propagation," *J. Air Transp. Manag.*, vol. 93, Jun. 2021, Art. no. 102045.

[2] N. Pyrgiotis, K. M. Malone, and A. Odoni, "Modelling delay propagation within an airport network," *Transp. Res. C, Emerg. Technol.*, vol. 27, pp. 60–75, Feb. 2013.

[3] X. Wei, Y. Li, R. Shang, C. Ruan, and J. Xing, "Airport cluster delay prediction based on TS-BiLSTM-attention," *Aerospace*, vol. 10, no. 7, pp. 580–595, Jun. 2023.

[4] G. G. N. Sandamali, R. Su, and Y. Zhang, "Flight routing and scheduling under departure and en route speed uncertainty," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1915–1928, May 2020.

[5] Q. Li and R. Jing, "Generation and prediction of flight delays in air transport," *IET Intell. Transp. Syst.*, vol. 15, no. 6, pp. 740–753, Jun. 2021.

[6] Q. Li, X. Guan, and J. Liu, "A CNN-LSTM framework for flight delay prediction," *Exp. Syst. Appl.*, vol. 227, Oct. 2023, Art. no. 120287.

[7] Q. Li and R. Jing, "Flight delay prediction from spatial and temporal perspective," *Exp. Syst. Appl.*, vol. 205, Nov. 2022, Art. no. 117662.

[8] J. Yi, H. Zhang, H. Liu, G. Zhong, and G. Li, "Flight delay classification prediction based on stacking algorithm," *J. Adv. Transp.*, vol. 2021, pp. 1–10, Aug. 2021.

[9] H. Zhou, W. Li, Z. Jiang, F. Cai, and Y. Xue, "Flight departure time prediction based on deep learning," *Aerospace*, vol. 9, no. 7, pp. 394–408, Jul. 2022.

[10] S. Saha, A. Haque, and G. Sidebottom, "Analyzing the impact of outlier data points on multi-step internet traffic prediction using deep sequence models," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 2, pp. 1345–1362, 2023.

[11] H. Zhang, C. Song, J. Zhang, H. Wang, and J. Guo, "A multi-step airport delay prediction model based on spatial–temporal correlation and auxiliary features," *IET Intell. Transp. Syst.*, vol. 15, no. 7, pp. 916–928, Jul. 2021.

[12] F. Wang, J. Bi, D. Xie, and X. Zhao, "Flight delay forecasting and analysis of direct and indirect factors," *IET Intell. Transp. Syst.*, vol. 16, no. 7, pp. 890–907, Jul. 2022.

[13] M. Güvercin, N. Ferhatosmanoglu, and B. Gedik, "Forecasting flight delays using clustered models based on airport networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 3179–3189, May 2021.

[14] T. Wang and S.-C. Chen, "Multi-task local–global graph network for flight delay prediction," in *Proc. IEEE 23rd Int. Conf. Inf. Reuse Integr. Data Sci. (IRI)*, Aug. 2022, pp. 49–54.

[15] Q. Li, R. Jing, and Z. S. Dong, "Flight delay prediction with priority information of weather and non-weather features," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 7149–7165, Jul. 2023.

[16] B. Bischl et al., "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *WIREs Data Mining Knowl. Discovery*, vol. 13, no. 2, pp. 1–43, Mar. 2023.

[17] B. Ye, B. Liu, Y. Tian, and L. Wan, "A methodology for predicting aggregate flight departure delays in airports based on supervised learning," *Sustainability*, vol. 12, no. 7, pp. 2749–2761, Apr. 2020.

[18] X. Wu, H. Yang, and S. Han, "Analysis of properties and delay propagation of air traffic based on complex network," *Acta Aeronautica et Astronautica Sinica*, vol. 38, no. s1, p. 721473, 2017.

[19] Á. Rodríguez-Sanz, J. Cano, and B. Rubio Fernández, "Impact of weather conditions on airport arrival delay and throughput," *Aircr. Eng. Aerosp. Technol.*, vol. 94, no. 1, pp. 60–78, Jan. 2022.

[20] A. Mukherjee, S. R. Grabbe, and B. Sridhar, "Predicting ground delay program at an airport based on meteorological conditions," in *Proc. 14th AIAA Aviation Technol., Integr., Operations Conf.*, Jun. 2014, pp. 1–18.

[21] K. Wang, J. Li, and Y. Tian, "Airport delay prediction method based on improved weather impacted traffic index," in *Proc. IEEE 1st Int. Conf. Civil Aviation Saf. Inf. Technol. (ICCASIT)*, Oct. 2019, pp. 73–78.

[22] F. Mo, Z. Zhao, and T. Yin, "Research on airport flight delay prediction technology based on arima model," *Aeronaut. Comput. Technique*, vol. 48, no. 3, pp. 68–73, 2018.

[23] H. Xie, Y. Li, M. Hu, and H. Chen, "Prediction of airport departure delay level based on BP neural network," *Aeronaut. Comput. Technique*, vol. 40, no. 4, pp. 71–74, 2019.

[24] Z. Wang, W. Xue, Y. Niu, Y. Cui, Q. Sun, and X. Hei, "Research on berth occupancy prediction model based on attention mechanism," *J. Commun.*, vol. 41, no. 12, pp. 182–192, 2020.

[25] A. C. Tsoi and S. Tan, "Recurrent neural networks: A constructive algorithm, and its properties," *Neurocomputing*, vol. 15, nos. 3–4, pp. 309–326, Jun. 1997.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[27] J. Liu, J. Zhao, Y. Lu, and Y. Wu, "Flight delay prediction method based on LSTM," in *Proc. Int. Conf. Autom., Robot. Comput. Eng. (ICARCE)*, Dec. 2022, pp. 1–4.

[28] P. Malighetti, C. Morlotti, R. Redondi, and S. Paleari, "The turnaround tactic and on-time performance: Implications for airlines' efficiency," *Res. Transp. Bus. Manag.*, vol. 46, Jan. 2023, Art. no. 100874.

[29] J. Li, W. Zeng, D. Liu, Z. Yang, and W. Cong, "Spatial–temporal long short-term memory networks for airport flight delay prediction," in *Proc. World Transp. Conf.*, 2019, pp. 504–517.

[30] L. Wang, J. Mao, L. Li, X. Li, and Y. Tu, "Prediction of estimated time of arrival for multi-airport systems via 'Bubble' mechanism," *Transp. Res. C, Emerg. Technol.*, vol. 149, Apr. 2023, Art. no. 104065.

[31] Z. Yan, H. Yang, Y. Wu, and Y. Lin, "A multi-view attention-based spatial–temporal network for airport arrival flow prediction," *Transp. Res. E, Logistics Transp. Rev.*, vol. 170, Feb. 2023, Art. no. 102997.

[32] J. Qu, M. Xiao, L. Yang, and W. Xie, "Flight delay regression prediction model based on Att-Conv-LSTM," *Entropy*, vol. 25, no. 5, pp. 770–785, May 2023.

[33] S.-Y. Shih, F.-K. Sun, and H.-Y. Lee, "Temporal pattern attention for multivariate time series forecasting," *Mach. Learn.*, vol. 108, nos. 8–9, pp. 1421–1441, Sep. 2019.

[34] R. Wu, Y. Zhuang, K. Xu, H. Zhang, and B. Chen, "PQ-NET: A generative part Seq2Seq network for 3D shapes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 826–835.

[35] 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.

[36] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 3104–3112.

[37] Y. Keneshloo, T. Shi, N. Ramakrishnan, and C. K. Reddy, "Deep reinforcement learning for sequence-to-sequence models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2469–2489, Jul. 2020.

[38] N. Forti, L. M. Millefiori, P. Braca, and P. Willett, "Prediction of vessel trajectories from AIS data via sequence-to-sequence recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 8936–8940.

[39] N. Khan et al., "Ab-Net: A novel deep learning assisted framework for renewable energy generation forecasting," *Mathematics*, vol. 9, no. 19, pp. 2456–2473, 2021.

[40] S. U. Khan and R. Baik, "MPPIF-Net: Identification of plasmodium falciparum parasite mitochondrial proteins using deep features with multilayer bi-directional LSTM," *Processes*, vol. 8, no. 6, pp. 725–740, Jun. 2020.

[41] S. Du, T. Li, Y. Yang, H. Wang, P. Xie, and S.-J. Horng, "A sequence-to-sequence spatial–temporal attention learning model for urban traffic flow prediction," *J. Comput. Res. Develop.*, vol. 57, no. 8, pp. 1715–1728, Aug. 2020.

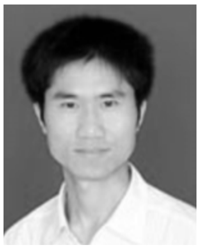
[42] B. Singh, P. K. Dubey, and S. N. Singh, "Recent optimization techniques for coordinated control of electric vehicles in super smart power grids network: A state of the art," in *Proc. IEEE 9th Uttar Pradesh Sect. Int. Conf. Electr., Electron. Comput. Eng. (UPCON)*, Dec. 2022, pp. 1–7.

[43] B. Singh and P. K. Dubey, "Distributed power generation planning for distribution networks using electric vehicles: Systematic attention to challenges and opportunities," *J. Energy Storage*, vol. 48, Apr. 2022, Art. no. 104030.

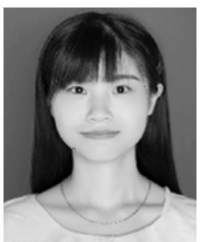
[44] W. Zhang, C. Wu, H. Zhong, Y. Li, and L. Wang, "Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization," *Geosci. Frontiers*, vol. 12, no. 1, pp. 469–477, Jan. 2021.

[45] H. M. Torun, M. Swaminathan, A. K. Davis, and M. L. F. Bellaredj, "A global Bayesian optimization algorithm and its application to integrated system design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 4, pp. 792–802, Apr. 2018.

[46] R. Wu, T. Zhao, and J. Qu, "Flight delay prediction model based on deep SE-DenseNet," *J. Electron. Inf. Technol.*, vol. 41, no. 6, pp. 1510–1517, 2019.
[47] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.
[48] L. Wang et al., "CNN-BiLSTM-attention model in forecasting wave height over south-east China seas," *Comput., Mater. Continua*, vol. 73, no. 1, pp. 2151–2168, 2022.



Zhenchun Wei (Member, IEEE) received the Ph.D. degree from the Hefei University of Technology, China, in 2007. He is currently an Associate Professor with the School of Computer and Information, Hefei University of Technology. His research interests include the Internet of Things, wireless sensor networks, distributed systems, and prediction methods.

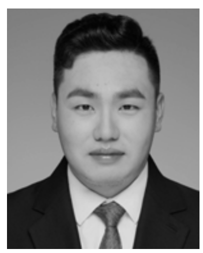


Siwei Zhu received the M.S. degree from the School of Computer and Information, Hefei University of Technology, China, in 2022. Her research interests include airport flight delay prediction methods.

Zengwei Lyu received the B.S., M.S., and Ph.D. degrees from the School of Computer and Information, Hefei University of Technology, China, in 2012, 2015, and 2019, respectively. His research interests include air transportation systems diagnostics and optimization methods.

Yan Qiao received the Ph.D. degree in computer science from the Beijing University of Posts and Telecommunications in 2012. She was a Post-Doctoral Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore. She is currently an Associate Professor with the School of Computer and Information, Hefei University of Technology, China. Her research interests include network monitoring and fault diagnosis for internet.

Xiaohui Yuan (Senior Member, IEEE) received the Ph.D. degree from Tulane University in 2004. He is currently an Associate Professor with the University of North Texas. His research interests include computer vision, data mining, machine learning, and artificial intelligence. He serves as the chair for several international conferences. He serves on the editorial board of several international journals. He is the Editor-in-Chief of the *International Journal of Smart Sensor Technologies and Applications*.



Yang Zhao received the B.S. degree from the Hefei University of Technology, Hefei, China, in 2019, where he is currently pursuing the M.S. degree with the School of Computer and Information. His research interests include airport flight delay prediction methods.

Hao Zhang received the master's degree from Tsinghua University in 2012. He is currently an Algorithm Engineer with Feeyo Technology Company Ltd., China. His research interests include application of computer vision and intelligent hardware in airports.