Fifteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2023) STL combining LSTM for long-term predicting airport traffic flow

Ziming Wang, Yanjun Wang, Yaoshuai Zhao, Mark Hansen, Daniel Delahaye

Fifteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2023)

STL combining LSTM for long-term predicting airport traffic flow

Ziming Wang, Yanjun Wang
College of Civil Aviation
Nanjing Uni. of Aero. & Astro.
Nanjing 210016, China
{zimingwang, ywang}@nuaa.edu.cn

Yaoshuai Zhao
Research Institute of Travel Sky
Beijing 100010, China
yszhao@travelsky.com

Mark Hansen
Dept. of Civil & Envi. Eng.
UC Berkeley
Berkeley CA94720, USA
mhansen@ce.berkeley.edu

Daniel Delahaye
ENAC Research Lab
Ecole Nationale de l'Aviation Civile
Toulouse 31400, France
daniel@recherche.enac.fr

***Abstract*—Airport traffic flow exhibits significant periodicity on a daily scale, few studies have given attention to periodicity when predicting airport traffic flow. In this article, we propose a novel model that combines long short-term memory (LSTM) and seasonal-trend decomposition procedure based on loess (STL) to predict the arrival/departure traffic flow at the airport. A sinusoidal template-matching method based on Fréchet distance is used to restack the periodic input variables. A time series decomposition algorithm STL is used to decompose the traffic flow time series into trend, seasonal, and remainder components to identify its periodic structure. LSTM model is trained using historical airport operation data, strategic flight schedule data, and meteorological data from Beijing Capital International Airport, Guangzhou Baiyun International Airport, and Shanghai Pudong International Airport in 2019. Our results demonstrate that both the adaptive restack of input variables and the time series decomposition algorithm STL can improve the prediction performance. Our proposed method shows superior performance in long-time prediction (720-time steps). In particular, STL combined LSTM method achieves an R-squared of 0.97 and a mean absolute error (MAE) of less than 1.58 for all three airports.**

***Keywords*—long short-term memory (LSTM); time-series prediction; airport traffic flow; seasonal-trend decomposition procedure based on loess (STL)**

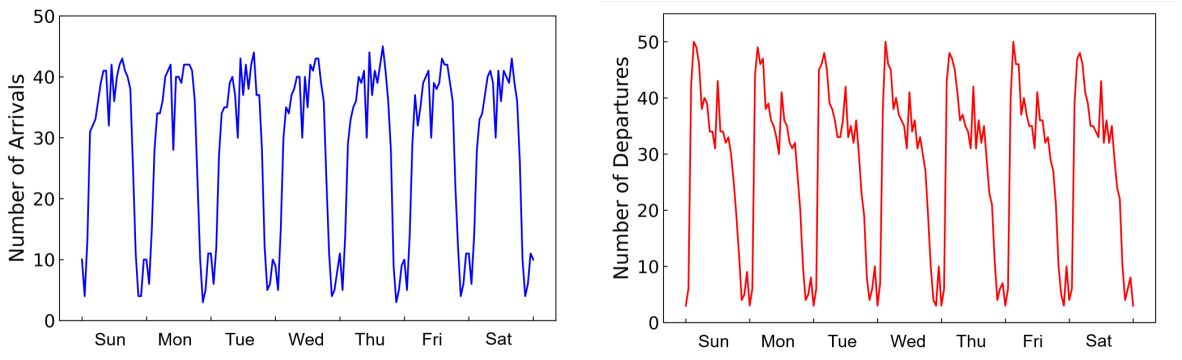
I. Introduction

More than 4.5 billion passengers were transported by passenger flights in 2019. Although the air transportation industry has been impacted by the COVID-19 pandemic, it is undoubtedly that air traffic would soon be recovering strongly [1]. As one of the most important components of the air transportation industry, airports provide services and infrastructure for airlines and passengers. These services and infrastructure cover a wide range, from passenger check-in, security scanning, through baggage systems, to gate operations and lighting systems. Resource allocation and collaborative operations play vital roles in improving the performance of the airport system. There is a growing body of literature that address various aspects of airport operations, for example, airport capacity and demand management [2], gate assignments [3], immigration and custom staff scheduling [4], [5], and airport delay prediction [6]–[8]. Most of these studies attempt to prepare and allocate airport resources to meet flights and passengers' demands. Accurately predicting departure and arrival flight flow is fundamental to airport management and air traffic flow management, which is however less investigated.

Air traffic flow at the airport exhibits high daily patterns and seasonal regularities due to the nature of air transport demand. Fig. 1 shows the actual departure/arrival traffic flow at Guangzhou Baiyun International (ICAO code: ZGGG) airport from Sunday to Saturday. A clear daily pattern can be observed with low traffic in the early morning and in late night hours, and higher traffic during the daytime. In most busy airports, flight schedules are published several months in advance. To manage the demand-capacity imbalance, airport slots are allocated under the guidance of Worldwide Airport Slot Guidelines (WASG) that is issued by the International Air Transport Association (IATA) [9]. Airlines develop flight schedules after obtaining airport slots. However, the actual number of flights arrival/departure at the airport may vary from the scheduled ones because of the uncertainty of tactical operations. A particular issue is whether that can reduce airport capacities thus limiting the number of arrivals or departures. For instance, the minimum standards for takeoff at the airport are given as follows: for one and two-engine aircraft, the visibility is 1600 meters; for three and four engines aircraft, the visibility is 800 meters [10]. Thus, it is still very difficult to accurately predict airport departure/arrival traffic flow.

Research methods on airport traffic flow prediction can be classified into mathematical modeling, simulations, and machine learning. In [11], the authors propose a timed stochastic Colored Petri Net (CPN) model of a single runway to analyze the effect of taxiways availability on runway traffic flow. In [12], the authors develop a traffic flow pattern classification model based on the clustering of multi-airport flight trajectories to capture the traffic flow patterns. An empirical model for arrival capacity estimation is proposed based on historical traffic patterns. The second group of studies to predict airport traffic flow is using simulation software. SIMMOD is a discrete-event simulation model that captures airport system evolving in time through a mathematical model, supporting the prediction of airport traffic flow [13].

This work is supported by the National Natural Science Foundation of China under grand number: U2033203.



(a) (b)

Figure 1: ZGGG arrival/departure flow during Sun - Sat

Aviation systems record vast amounts of data which enables machine learning to predict traffic flow and flight delays. Several neural network models have been proposed to predict traffic flow, including Artificial Neural Networks (ANN) [14], Long Short-Term Memory (LSTM) [15], [16], Gated Recurrent Units (GRU) [17], Recurrent Neural Network (RNN) [18] and Convolutional Neural Networks (CNN) [19]. RNNs are widely used for sequence modeling, but their training based on gradient descent algorithms may blow up [20]. Typical RNNs fail to recognize the time lag that exists for long periods of time between the input and the target output [21]. In recent years, LSTM and GRU have been successfully applied in predicting periodic time series. A work in [22] proposes Stacked AutoEncoder (SAE), LSTM, and GRU models to predict traffic flow at the Beijing Capital International Airport for the 45min time horizon. These three models tested on the airport north line, airport west road, and airport expressway, SAE model exhibits superior performance. In [15], the authors propose a method combining LSTM and eXtreme Gradient Boosting (LSTM-XGBoost) to predict airport arrival flow for the 30min, 60min, and 120 min time horizon. XGBoost is developed to obtain the ranking of the features. In [23], the authors propose a new deep learning-based framework called Airport Traffic Flow Prediction Network (ATFPNet) to capture the spatial-temporal dependence of historical airport departure and arrival traffic flows. The time horizon of the model prediction are 30min, 45min, 60min, and 120 min. A real-world airport traffic flow data set is applied to validate the proposed model. The results show that the proposed method achieves up to 17% MAE improvement compared to baseline models.

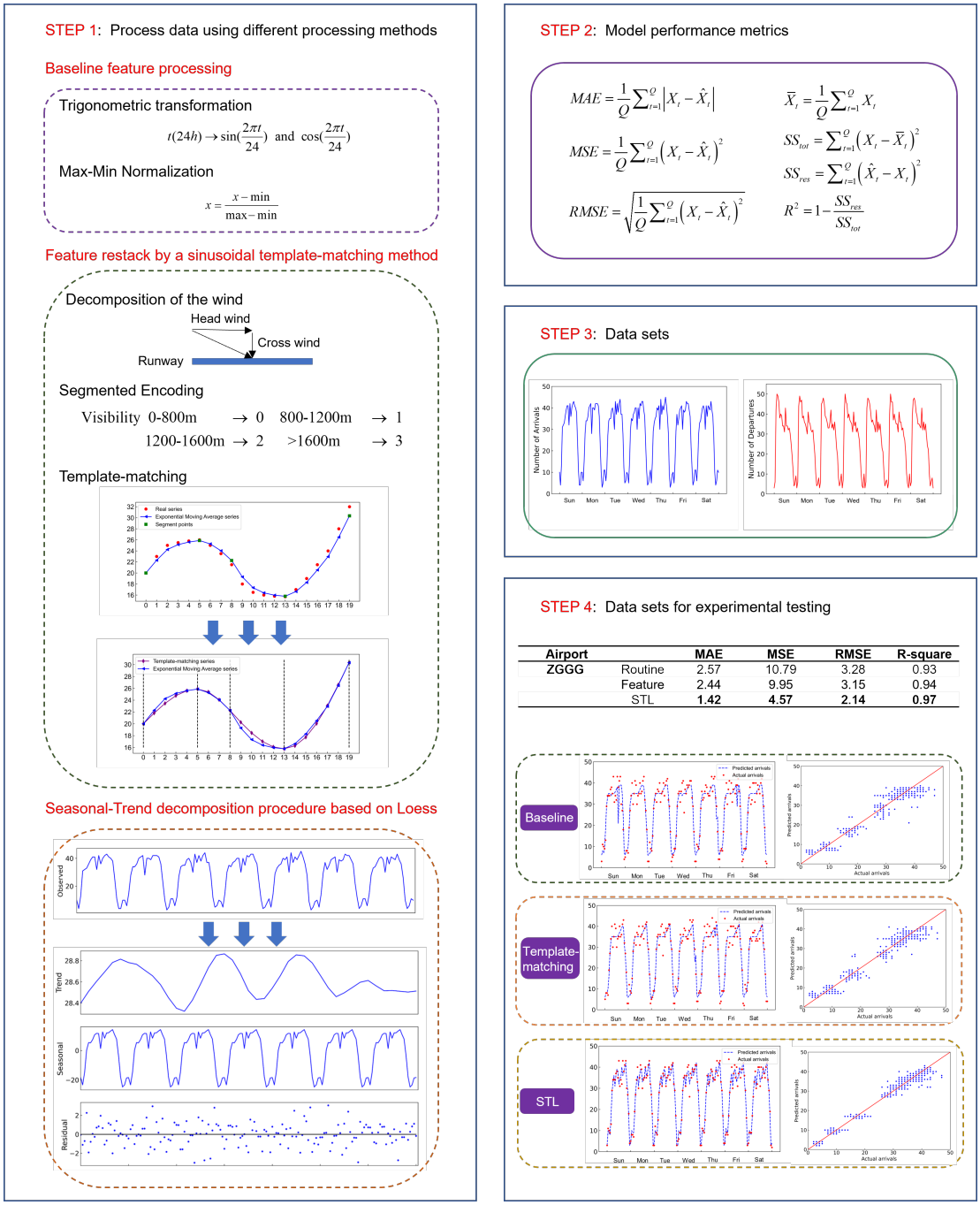
We notice that most of the work is focusing on short-range traffic flow prediction. In this paper, we focus on long-range traffic flow prediction (about 1 month prior to the day of operation, 720-time steps). A sinusoidal template-matching method based on Fréchet distance is used to restack periodic input weather variables adaptively. A time series decomposition algorithm, seasonal-trend decomposition procedure based on loess (STL), is used to decompose the departure/arrival flow time series into trend, seasonal, and remainder components, to identify the periodic structure of the flow. Subsequently, machine learning models are used to learn the trend components. The predicted trend components are summed to the seasonal components to obtain the predicted departure/arrival traffic flow. The results demonstrate that both the adaptive restack of input variables and the time series decomposition algorithm STL can improve the prediction performance. Especially the time series decomposition algorithm STL can significantly improve the prediction performance.

The main contents of the paper are shown in Fig. 2. Step 1 is processing experimental data using three methods; step 2 is selecting evaluation metrics; step 3 is collecting and analyzing the data set; step 4 is conducting experiments to compare the performance of different methods. The structure of this paper is organized as follows. Section II describes the data used in this paper. Section III introduces the sinusoidal template-matching method and time series decomposition algorithm STL. Section IV introduces the machine learning algorithms and compares the performance of algorithms. Finally, conclusions and discussions are given in Section V.

II. Data

This paper takes Guangzhou Baiyun International Airport, Beijing Capital International Airport and Shanghai Pudong International Airport as the subjects of the study, covering historical airport operation data, strategic flight schedule data, and meteorological data from March 31, 2019, to October 26, 2019. Meteorological data are obtained from National Oceanic and Atmospheric Administration (NOAA) [24]. The strategic flight schedule data is used to obtain the number of scheduled hourly departures/arrivals at the airport, and the historical airport operation data is used to obtain the airport departure/arrival hourly traffic flow, the target to be predicted. Meteorological data includes elements such as visibility, temperature, and pressure. The features extracted from each data source and used for training and prediction are shown in Table I. The extracted departure/arrival hourly traffic flow and weather data tables are merged with the date and time.

The daily departure/arrival hourly traffic flow of Guangzhou Baiyun International Airport is shown in Fig. 3. Fig. 3 shows that the hourly departure and arrival



Airport		MAE	MSE	RMSE	R-square
ZGGG	Routine	2.57	10.79	3.28	0.93
	Feature	2.44	9.95	3.15	0.94
	STL	**1.42**	**4.57**	**2.14**	**0.97**

Figure 2: Main contents of the paper

TABLE I. Feature description for the airport flow prediction

Feature	Feature description
Schedule	Month, Day of week, Hour, Day of month, Scheduled arrivals and scheduled departures
Weather	Wind direction[deg], Wind speed[m/s], Opaque layer height[m], Temperature[°C], Visibility[m], Dew point[°C], Pressure[hpa], Precipitation[mm], Cloud height[m] and Cloud cover

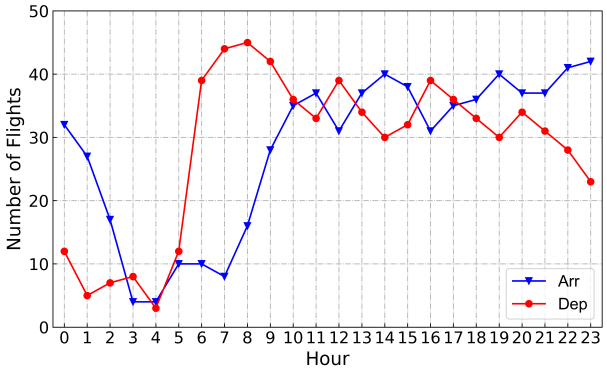


Figure 3: ZGGG arrival/departure hourly flow

traffic flow of the airport are different. In terms of arrival traffic flow, it is at a low level in the early hours of the day and increases significantly after 7:00 a.m., and then remains at 30-40 hourly traffic flow with oscillations. In terms of departure traffic flow, it starts to increase notably after 4:00 a.m. to reach a peak at 8:00 a.m., and then starts to decrease after 16:00 p.m..

III. Methods

A. Feature engineering and seasonal-trend decomposition

1) Baseline feature processing: Trigonometric functions are employed to convert month, day of week, hour, and day of month, to keep the nature of periodicity [25]. For a particular hour, t in one day, the trigonometric functions $sin(2\pi t/24)$ and $cos(2\pi t/24)$ are applied to ensure 24-hour periodicity. After trigonometric conversion, t = 24:00 and t = 01:00 will be continuous hours. Similarly, the periodicity of the month, day of week, and day of month are 12, 7, and 30 respectively.

Other numerical variables such as wind direction and wind speed are normalized. That is the data are mapped uniformly to the interval $[0, 1]$. Normalization will change the dimensional expression into a dimensionless expression. The normalization method used here is Max-Min standardization, which is the linear transformation of original data. The transformation function is given as follows:

$$x = \frac{x - \min}{\max - \min} \tag{1}$$

where x is the original data, min is the minimum value in the original data and max is the maximum value in the original data.

2) Feature restack by a sinusoidal template-matching method: The meteorological wind (wind direction and wind speed) is decomposed into *headwind* that is parallel to the airport runway direction and *crosswind* that is perpendicular to the airport runway direction. The segmentation coding method is used to encode visibility, cloud height, and opaque layer height variables. For example, airport visibility below 800m is coded as 0; visibility between 800m and 1200m is coded as 1; visibility between 1200m and 1600m is coded as 2; visibility above 1600m is coded as 3. The selection of 800, 1200, and 1600m as segmentation coding threshold is mainly based on airport runway operation standards and the regulations for the establishment and implementation of minimum standards for the operation of aircraft. The segmentation coding process of each feature is shown in Table II.

Temperature and pressure features of the weather exhibit significant periodic characteristics. A sinusoidal template-matching method based on Fréchet distance is used to restack the periodic input variables. Template matching is a representative algorithm in the field of image recognition, which identifies the matching location in the target image by a specified template [26]. To preserve the periodic changes of the weather time series, we match the time series using a standard sinusoidal image by extending the template-matching method.

Given a time series of temperature or pressure $X = \{x_1, x_2, \cdots, x_n\}$, where n here refers to the length of the data set (the number of days times 24), the Exponential Moving Average (EMA) is applied to ensure the time series will not fluctuate too much due to outliers. EMA equation is given as

$$v_t = \begin{cases} x_1 & t = 1 \\ \alpha v_{t-1} + (1-\alpha)x_t & t > 1 \end{cases} \tag{2}$$

where t is the sliding time window, x is the original data , and v is the data after EMA processing, α indicates the speed of weighted decline.

The time series after EMA processing denotes as $V = \{v_1, v_2, \cdots, v_n\}$, which will be further splited at the point where the concavity or monotonicity changes. Define $M = \{m_1, m_2, \cdots, m_{n-1}\}$, $C = \{c_1, c_2, \cdots, c_{n-2}\}$, where $m_i = v_{i+1} - v_i$, $c_i = m_{i+1} - m_i$. When $m_i m_{i-1} < 0$, it indicates a change of the monotonicity of the time series; when $c_i c_{i-1} < 0$, it indicates a change of the concavity of the time series. Dividing the time series when $m_i m_{i-1} < 0$ or $c_i c_{i-1} < 0$ to obtain four types of time series, including monotonically increasing convex, monotonically decreasing convex, monotonically increasing concave and monotonically decreasing concave time series (see Fig. 4 (a)). For each segment, the template-matching process is performed by using the standard sinusoidal image, which consists of four quarter-cycle sinusoidal images, as shown in Fig. 4 (b). The process of template-matching is done by stretching, compressing, and shifting the sinusoidal templates to minimize the Fréchet distance from the time series in this segmentation interval. The specific process of sinusoidal template-matching is as follows.

Given a four quarter-cycle sinusoidal templates $sin(x)$, define the following three transformation processes:

- Stretching $P(x)$: $P(x) = sin(x/duration)$;
- Compressing $Q(x)$: $Q(x) = amplitude * sin(x)$;
- Shifting $S(x)$: $S(x) = sin(x) + pan$.

Let $T = (P(x), Q(x), S(x))$ be the transforming process that contains the above three processes, and $J(x)$ be the transformed time series. Let $F(J(x), V(x))$ be the Fréchet distance between $J(x)$ and $V(x)$. The Fréchet distance is defined as:

$$F(J(x), V(x)) = \min \|L\| \tag{3}$$

TABLE II. Segmentation coding process of each feature

Feature	Segmentation coding
Visibility	0-800m →0, 800-1200m →1, 1200-1600m →2, >1600m →3
Cloud height	0-100m →0, 100-300m →1, 300-500m →2, 500-1000m →3, 1000-1500m →4, >1500m →5
Opaque layer height	0-80m →0, 80-120m →1, 120-200m →2, >200m →3

$$\|L\| = \max_{i=1,\ldots,n, a=1,\ldots,m} d\left(v_i, j_a\right) \tag{4}$$

where (v_i, j_a) is the pair of data in the templates and target time series, d is the Euclidean distance. In a sequence of two different sampling points, Fréchet distance tries to find a path that minimizes the sum of the distances of the values paired with each other. The objective of the sinusoidal template-matching method is to minimize $F(J(x), V(x))$. Fig. 4 (b), Fig. 4 (c), and Fig. 4 (d) demonstrate the process of minimizing the Fréchet distance by transforming the four quarter-cycle sinusoidal templates. The sinusoidal template-matching method pseudo code is shown in Fig. 5. With the special note that to restack input variables faster, we initialize transformation consisting of stretching, compression, and shifting shown in Fig. 5 line 7-9. Finally, restack the input by a sinusoidal template-matching method, and transformation Q, P, S is set to get the minimal Fréchet distance in a list $parameter_T$. An example of the original temperature series and its sinusoidal template-matching is shown in Fig. 6. As can be seen from the figure, the temperature series shows a periodicity of a week. The matched series successfully capture the periodic characteristics of the temperature data.

3) Seasonal-Trend decomposition procedure based on loss: STL algorithm is used to decompose the series of departure/arrival flow into trend, seasonal, and remainder components thus identifying its periodic structure. It uses robust locally weighted regression as a smoothing method [27]. Given a departure/arrival time series $Y = \{y_1, y_2, \cdots, y_n\}$, STL algorithm decomposes Y into trend component T, seasonal component S, and remainder component R (shown in Fig. 7), such that $Y_v = T_v + S_v + R_v$, $v = 1, 2, \cdots, n$. Once the original series has been decomposed, machine learning algorithms will be developed to predict the trend components T'. The seasonal component S is then added to T' to obtain the predicted time series Y'. STL algorithm presumes that the remainder component R is very small. Since there has already residual errors when predicting T', STL then preserves the seasonality of the original time series.

STL algorithm consists of two parts: inner loop and outer loop. The inner loop is for trend fitting and the calculation of the seasonal component, while the outer loop improves the robustness of the algorithm. The inner loop can be described in six steps.

1) Detrending. Initially, set $T_v^1 = 0$. Then $Y_v - T_v^1$.
2) Cycle-subseries smoothing. Set $n_{(p)} = 24$, then $Subseries\ \ one = \left\{y_1, y_{1+n_{(p)}}, y_{1+2n_{(p)}}, \cdots\right\}$, $Subseries\ \ two = \left\{y_2, y_{2+n_{(p)}}, y_{2+2n_{(p)}}, \cdots\right\}$, $\cdots$. $Loess(Subseries\ (one - twentyfour)) = C_v^2$.
3) Low-pass. $Moving\ average_{windowsize=n_{(p)}, n_{(p)}, 3}(C_v^2) = D_v^2$. $Loess(D_v^2) = L_v^2$.

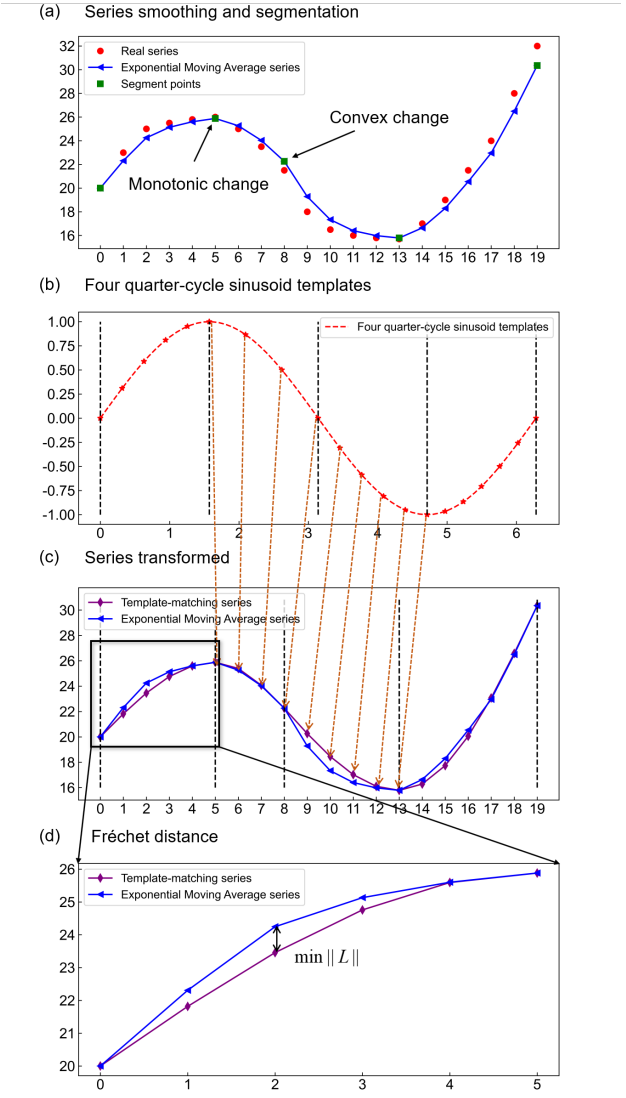


Figure 4: Feature restack by a sinusoidal template-matching method

4) Detrending of smoothed Cycle-subseries. $S_v^2 = C_v^2 - L_v^2$.
5) Deseasonalizing. $R_v^2 = Y_v - S_v^2$.
6) Trend Smoothing. $Loess(R_v^2) = T_v^2$, return Step 1 until convergence.

Step1: Detrending, Y_v subtract the trend component $T_v^{(k)}$ obtained at the end of the last inner loop, where k indicates at the end of kth path, initially $T_v^{(k)} = 0$; Step2: Cycle-subseries Smoothing, a Locally weighted regression (Loess) is done on the subseries (a sequence of sample points at the same position for each cycle, in this paper is the sequence

Algorithm 1 Sinusoidal Template-Matching

Input :

Time series X

1 : Exponential moving average X $\rightarrow$ V

2 : **if** $m_p m_{p-1} < 0 \cup c_p c_{p-1}$ **do**

3 : Divide V $\rightarrow$ $\{v_1, v_2, ..., v_p\}$ and $\{v_{p+1}, v_{p+2}, ..., v_n\}$

4 : V $\rightarrow$ Segment: {Segment one, Segment two,...,Segment last}

5 : **end if**

6 : **for** N_Seg in Segment **do**

7 : **Initialize** $Q = abs\left(begin\left(\text{N\_Seg}\right) - end\left(\text{N\_Seg}\right)\right)$

8 : $$P = \frac{\left(\text{N\_Seg}_{index_{i+1}} - \text{N\_Seg}_{index_i}\right) * 2}{\pi}$$

9 : $$S = begin(\text{N\_Seg}) - Q * \left(sin\left(\text{N\_Seg}_{indexi+1} - \text{N\_Seg}_{indexi}\right) / P\right)$$

10 : $T = (Q, P, S)$

11 : D = [], $parameter_T$ =[]

12 : **for** j , k, l = $-\delta$ to δ **do**

13 : D append Calculated $F(T(\sin(x), \text{N\_Seg})$

14 : $parameter_T$ append $(\text{Q} * (1+j), \text{P} * (1+k), \text{S} * (1+l))$

14 : **end for**

15 : Min d in D **return** $parameter_T$

16 : **end for**

Figure 5: Pseudo code of sinusoidal template-matching method

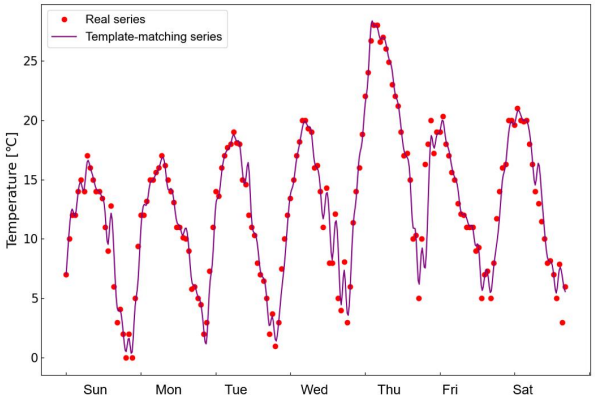


Figure 6: Original temperature series and its sinusoidal template-matching

of departure/arrival traffic flow at the same hour) and the result is noted as $C_v^{(k+1)}$; Step3: Low-Pass, the result $C_v^{(k+1)}$ is filtered by a moving average with a window of size $n_{(p)}$ (number of the subseries), $n_{(p)}$, 3, then a Loess is performed to get the output $L_v^{(k+1)}$. Step4: Detrending of smoothed Cycle-subseries, calculate $S_v^{(k+1)} = C_v^{(k+1)} - L_v^{(k+1)}$; Step5: Deseasonalizing, a deseasonalized series $Y_v - S_v^{(k+1)}$ is computed. Step6: Trend Smoothing, a Loess is done on the deseasonalized series to obtain the trend component $T_v^{(k+1)}$. The outer loop is mainly used to improve the robustness of the algorithm. When there is an outlier in the time series, the neighborhood weights need to be weighted so that the weights of the large outlier in the data are very small or taken as 0 to weaken the effect of the outlier when loess is performed in steps 2-6. In fact, there is no significant outlier in the departure/arrival flow time series data, and the outer

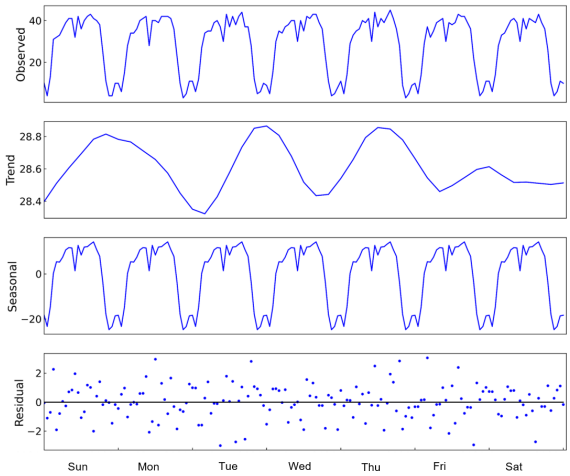


Figure 7: STL decomposition algorithm example diagram in a week

loop is not really uesful. but in order to retain the robustness of the algorithm, the outer loop process is retained in this paper.

B. LSTM-attention algorithm

We denote the i-th feature of the airport at time step t as $x_{i,t} \in \mathbb{R}^C$, where C denotes the traffic flow types (e.g., arrival traffic flow, departure traffic flow). Then, the airport traffic flow of N features at time step t is represented as $X_t = (x_{0,t}, x_{1,t}, ..., x_{N-1,t}) \in \mathbb{R}^{N \times C}$.

In this paper, we plan to predict the airport traffic flow of the next Q time steps $\hat{Y} = (\hat{X}_1, \hat{X}_2, ..., \hat{X}_Q) \in \mathbb{R}^{Q \times N \times C}$, given the airport traffic flow data of N features at previous P time steps $\mathcal{X} = (X_0, X_1, ..., X_{P-1}) \in \mathbb{R}^{P \times N \times C}$. In this paper, each time step is 1 hour, and we use the data of the previous 720 hours (P=720) to predict the flight delay of the next 720 hours (Q=720).

1) Attention: The task of the attention mechanism is to obtain information of local interest, and the introduction of attention makes certain locations in the input data more worthy of attention [28].

Given airport traffic flow data of N features at previous P time steps $\mathcal{X} = (X_0, X_1, ..., X_{P-1}) \in \mathbb{R}^{P \times N \times C}$, multiply the matrices $W^q \in \mathbb{R}^{N \times N}$, $W^k \in \mathbb{R}^{N \times N}$ and $W^v \in \mathbb{R}^{N \times N}$ separately to obtain Q, K and V (as shown in Equation 5). For example, X_i after the multiplication we get Q^i, K^i and V^i. Here Q, K, and V correspond to Query, Key, and Value. The original concept is derived from the information retrieval system. Take each query Q and key K, perform attention transformation to get the attention score. Then, the score is normalized by dividing by $\sqrt{d_k}$ (as shown in Equation 6). Softmax is applied to get the weights (Equation 7). The weights are multiplied by the value matrix V and then summed to obtain the final output (Equation 8). After the attention mechanism, we transfer airport traffic flow data of N features at previous P time steps $\mathcal{X} = (X_0, X_1, ..., X_{P-1}) \in \mathbb{R}^{P \times N \times C}$ to $\mathcal{A} = (A_0, A_1, ..., A_{P-1}) \in \mathbb{R}^{P \times N \times C}$.

$$Q^i = W^q X_i \quad K^i = W^k X_i \quad V^i = W^v X_i \tag{5}$$

$$\alpha_{i,p} = \frac{Q^i K^T}{\sqrt{d_k}} \tag{6}$$

$$\hat{\alpha}_{i,p} = \frac{e^{\alpha_{i,p}}}{\sum_p e^{\alpha_{i,p}}} \tag{7}$$

$$A^i = \sum_p \hat{\alpha}_{i,p} V^p \tag{8}$$

2) LSTM: A long short-term memory (LSTM) is a variant of the RNN structure, which is designed to be able to learn information between long separated events. It was proposed by Hoschreiter and Schmidhuber to cope with the long-term dependency problem of RNNs [29]. The reason that LSTM can find the correlation between long separated events is that LSTM has three multiplication gates: input gate, forgetting gate and output gate. LSTM controls the retention of the previous moment cell state by input gate and forgetting gate. The input gate, forgetting gate, output gate and cell status of the input unit are calculated as follows,

$$i_p = \sigma\left(W_i \cdot [h_{p-1}, A_p] + b_i\right) \tag{9}$$

$$f_p = \sigma\left(W_f \cdot [h_{p-1}, A_p] + b_f\right) \tag{10}$$

$$o_p = \sigma\left(W_o \cdot [h_{p-1}, A_p] + b_o\right) \tag{11}$$

$$\tilde{c}_p = \tanh\left(W_c \cdot [h_{p-1}, A_p] + b_c\right) \tag{12}$$

where σ is the sigmoid function, and $tanh$ is the $tanh$ function. $W_i \in \mathbb{R}^{N\times D}$, $W_f \in \mathbb{R}^{N\times D}$, $W_o \in \mathbb{R}^{N\times D}$, and $W_c \in \mathbb{R}^{N\times D}$ are the input gate, forgetting gate, output gate, and cell status of the input unit weight matrix. b_i, b_f, b_o, and b_c is the bias item, h_{p-1} is the LSTM output value at the previous moment, while A_p is the current network input value obtained from attention mechanism.

The state of the cell at the current moment is given as,

$$\mathbf{c}_p = f_p \circ c_{p-1} + i_p \circ \tilde{c}_p \tag{13}$$

where c_{p-1} is the status of the last output cell, while $\circ$ denotes multiplication of the elements.

The output of the current cell is obtained from the output gate and the cell state at the current moment with equation 14. We last use fully-connected layer to project the data into N-dimensional representation as $\hat{Y} = (\hat{X}_1, \hat{X}_2, ..., \hat{X}_Q) \in \mathbb{R}^{Q\times N\times C}$.

$$\mathrm{h}_p = o_p \circ \tanh\left(c_p\right) \tag{14}$$

A schematic diagram of the LSTM-attention framework is shown in Fig. 8. The general architecture of the network is shown in Fig. 8 (a), in which the network inputs are $\mathcal{X} = (X_0, X_1, ..., X_{P-1}) \in \mathbb{R}^{P\times N\times C}$ and the network outputs are $\hat{Y} = (\hat{X}_1, \hat{X}_2, ..., \hat{X}_Q) \in \mathbb{R}^{Q\times N\times C}$. Fig. 8 (b) shows an illustration of an LSTM model described in III-B2. Fig. 8 (c) illustrates a self-attention model described in III-B1.

3) Performance metrics: The metrics used to evaluate the performance of the models developed in this paper are the Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE), and R-square. The performance metrics are defined as follows:

$$MAE = \frac{1}{Q}\sum_{t=1}^{Q}\left|X_t - \hat{X}_t\right| \tag{15}$$

$$MSE = \frac{1}{Q}\sum_{t=1}^{Q}\left(X_t - \hat{X}_t\right)^2 \tag{16}$$

$$RMSE = \sqrt{\frac{1}{Q}\sum_{t=1}^{Q}\left(X_t - \hat{X}_t\right)^2} \tag{17}$$

$$\begin{aligned} \overline{X}_t &= \frac{1}{Q}\sum_{t=1}^{Q} X_t \\ SS_{tot} &= \sum_{t=1}^{Q}\left(X_t - \overline{X}_t\right)^2 \\ SS_{res} &= \sum_{t=1}^{Q}\left(\hat{X}_t - X_t\right)^2 \\ R^2 &= 1 - \frac{SS_{res}}{SS_{tot}} \end{aligned} \tag{18}$$

IV. Results

A. Model training

The case study data are divided into training data set and test data set. Specifically, we divide the training set and the test set in date sequence, and the ratio of the training set to the test set is 4:1. The training data is forward propagated into the LSTM network, and the output is generated by the network described in Fig. 8 (a). The network input is 720-time steps and the output is 720-time steps, which is an m-m network structure. The loss function is calculated using the MAE of the difference between the actual and predicted output values, and the error is back-propagated to update the model parameters by the back-propagation algorithm. The hyperparameters of the network are obtained by Grid Search. The LSTM model is trained using traffic flow and meteorological data from Beijing Capital International Airport, Guangzhou Baiyun International Airport and Shanghai Pudong Airport in 2019.

B. Results analysis

In Tables III and IV, the best performance metrics of each model are compared. Fig. 9 - 11 show the actual arrival flow and predicted flow (left) and compares the flow predictions of the three models on the test data with the actual hourly flow (right) of ZGGG arrival. This figure (right) can be used to analyze the performance of the models qualitatively. The diagonal line in the figure indicates the state where the error is zero since the predicted value and the actual value match. The greater the difference between the actual and predicted values the greater the deviation of the points from the diagonal.

The results in Tables III - IV and Fig. 9 - 11 show that the LSTM network under baseline feature processing can already learn daily traffic trends. The airport flow starts to increase after 4:00 a.m. and peaks in the afternoon, then starts to decrease at 20:00 p.m.. But the LSTM network is too conservative in order to get a smaller MAE, and it is

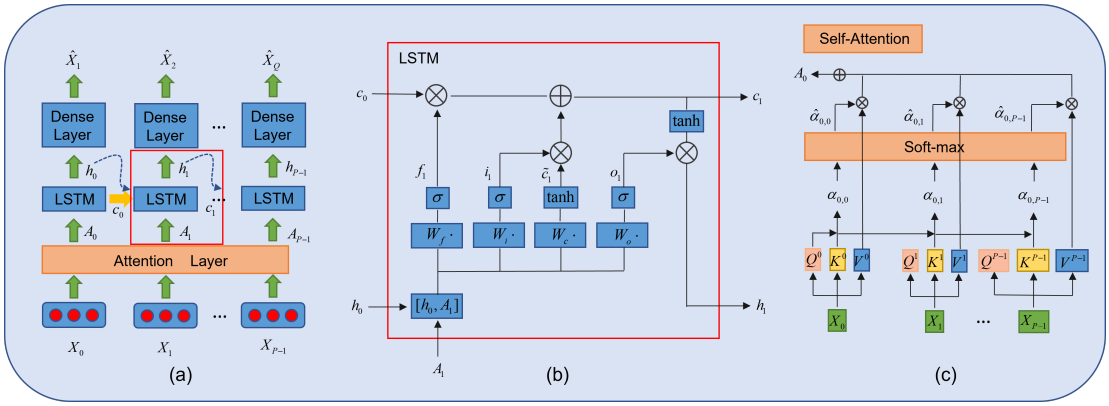


Figure 8: Schematic diagram of lstm-attention

difficult to identify the traffic changes near the hour, which may be related to the error propagation under the long time series prediction of LSTM. Traffic flow prediction under the STL decomposition method shows optimal performance in both arrival and departure of all three airports, with an MAE less than 1.58 and an R-squared performance of 0.97 above. Especially, in the ZGGG hourly arrival flow prediction, template-matching gives **5.06** % MAE reduction (2.57 ↓ 2.44), and STL gives **44.75** % MAE reduction (2.57 ↓ 1.42). In the ZGGG hourly departure flow prediction, template-matching gives **6.11** % MAE reduction (2.62 ↓ 2.46), and STL gives **52.29** % MAE reduction (2.62 ↓ 1.25). STL yields a **49.32** % averaged MAE reduction among the three airports in the arrival flow prediction, **47.93** % averaged MAE reduction among the three airports in the departure flow prediction. In most scenarios, the template-matching method has only a slight performance improvement over the baseline. Template-matching yields a **16.05** % averaged MAE reduction among the three airports in the arrival flow prediction, **6.56** % averaged MAE reduction among the three airports in the departure flow prediction. A more detailed analysis of why just slight performance improvement is discussed in the next section. Departure traffic flow prediction outperforms better than arrival traffic flow prediction.

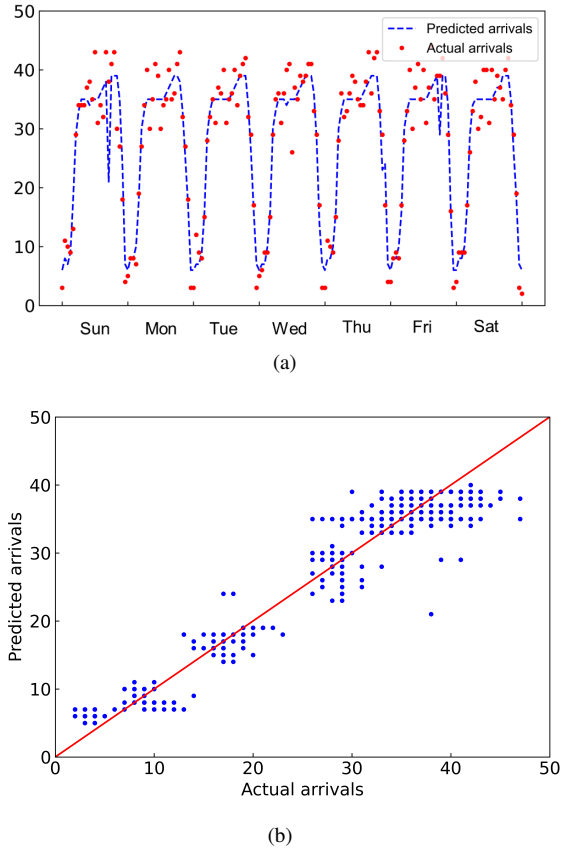


Figure 9: Actual arrival flow and estimated flow of ZGGG under baseline feature processing

V. Discussion and Conclusion

A. Discussion

Lundberg and Lee [30] proposed the SHAP (SHapley Additive exPlanation) method, an additive explanatory model. The SHAP value implements a sample-based approach to quantify the effect of features on the model output. Fig. 12 present a summary plots of SHAP values for all features for all samples at Guangzhou Baiyun Airport (The SHAP values of departure flow prediction has similar results. Due to limited space, the figure is not shown.). Each point corresponds to a sample and is associated with a SHAP value (i.e., the horizontal coordinate value in the plot). The color indicates the magnitude of the feature value, with blue indicating a smaller value for the feature, and red indicating a larger value for the feature. In Fig. 12, the features are sorted by the sum of the SHAP values, the features at the top of the graph have the greatest impact on the algorithm's output, while those at the bottom of the graph have the least.

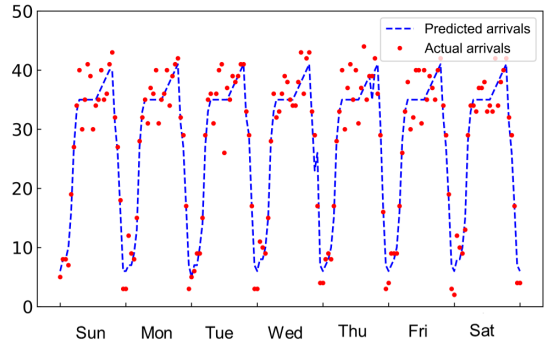
From the figure, we can see that the feature hour have a strong influence on the model output in arrival and departure traffic flow prediction. This means that the LSTM model itself has been able to learn some of the periodic features of the daily variation of traffic flow. Also note that the weather features cloud cover, cloud height and visibility have a large impact on the output. Some weather features like temperature, dew point and precipitation have a slight impact on the output,

TABLE III. Performance metrics of hourly arrival flow prediction

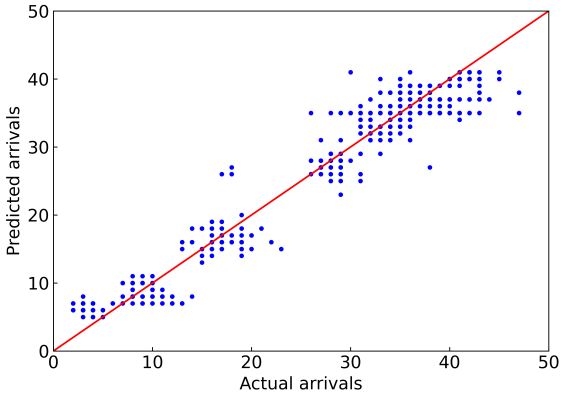
Airport		MAE	MSE	RMSE	R-square
ZGGG	Baseline	2.57	10.79	3.28	0.93
	Template-matching	2.44	9.95	3.15	0.94
	STL	**1.42**	**4.57**	**2.14**	**0.97**
ZBAA	Baseline	3.30	19.08	4.37	0.95
	Template-matching	2.01	15.7	3.96	0.96
	STL	**1.58**	**6.67**	**2.58**	**0.98**
ZSPD	Baseline	2.74	12.44	3.53	0.93
	Template-matching	2.63	11.68	3.42	0.93
	STL	**1.34**	**3.21**	**1.79**	**0.98**

TABLE IV. Performance metrics of hourly departure flow prediction

Airport		MAE	MSE	RMSE	R-square
ZGGG	Baseline	2.62	12.21	3.49	0.93
	Template-matching	2.46	11.29	3.36	0.93
	STL	**1.25**	**3.06**	**1.75**	**0.98**
ZBAA	Baseline	2.58	11.46	3.38	0.97
	Template-matching	2.48	10.32	3.21	0.97
	STL	**1.22**	**2.59**	**1.61**	**0.99**
ZSPD	Baseline	1.96	6.33	2.52	0.97
	Template-matching	1.77	5.05	2.25	0.98
	STL	**1.20**	**2.76**	**1.66**	**0.99**

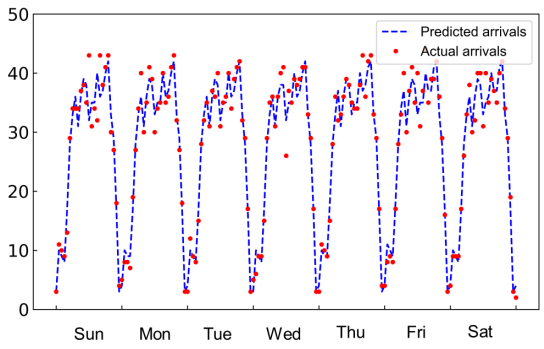


(a)

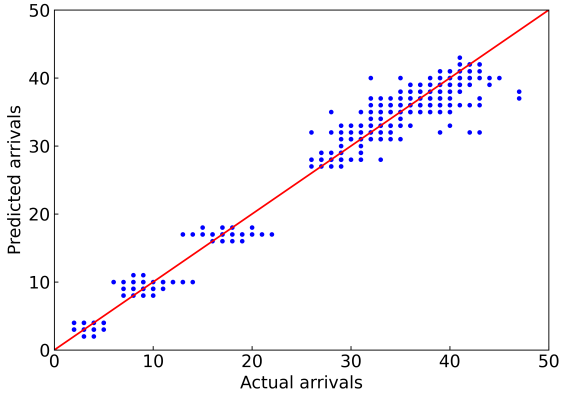


(b)

Figure 10: Actual arrival flow and estimated flow of ZGGG under sinusoidal template-matching method



(a)



(b)

Figure 11: Actual arrival flow and estimated flow of ZGGG under STL algorithm

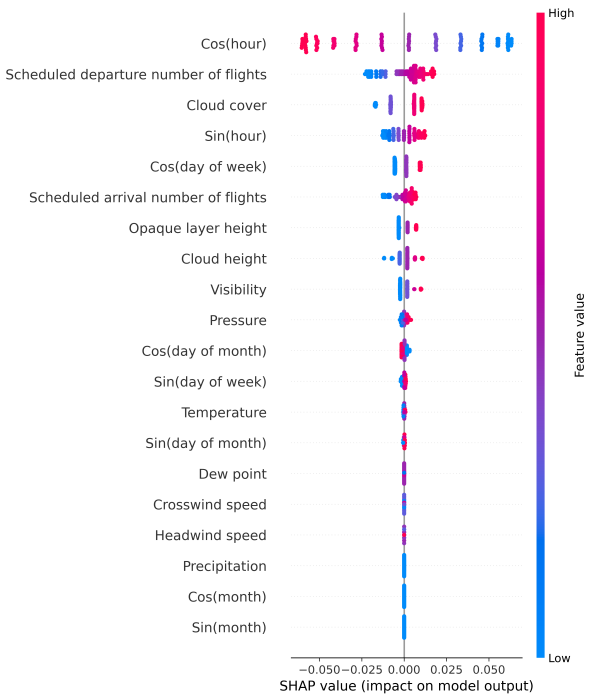


Figure 12: SHAP values of the features used for arrival flow prediction

this explains why feature restack by a sinusoidal template-matching method shows a lower performance improvement over the baseline.

B. Conclusion

The LSTM algorithm is used to estimate the hourly departure and arrival traffic at airports. The model is trained using weather data and historical operational data and tested on data from Guangzhou Baiyun Airport, Beijing Capital Airport, and Shanghai Hongqiao Airport. Some evaluation metrics, such as MAE and RMSE, are selected and the model performance is evaluated in actual compared to predicted plots. A feature processing method is also proposed to better retain the feature information of the training data, including segment coding of features and features restack by a sinusoidal template-matching method. Also, to better retain the periodic variation of the traffic flow, the flow data are decomposed into seasonal components, trend components and residual components using STL decomposition algorithm. The results show that both the treatment of features and the consideration of the periodic pattern of airport traffic flow can improve the prediction performance, especially the consideration of the periodic pattern of airport traffic can significantly improve the prediction performance. In the three airport prediction performances, the average MAE is less than 1.58, and the R-square performance reaches 0.97 above. More meaningfully, these two methods can be combined with arbitrary machine learning methods to achieve better prediction of periodic data. Feature restack by a sinusoidal template-matching method potentially could be a generalized method for smoothing periodic data.

Through this study, traffic flow could help to operate terminals more efficiently and further improve the efficiency of air traffic flow management. For example, airport congestion can be quantitatively analyzed in conjunction with the hourly traffic flow of the airport. Due to the uncertainty of weather forecasts, weather scenarios can be divided to build airport traffic prediction models, making the models more robust.

References

[1] International Air Transport Association, "World air transport statistics 2021," [WebPage], 2021, https://www.iata.org/contentassets/a686ff624550453e8bf0c9b3f7f0ab26/wats-2021-mediakit.pdf.

[2] A. Jacquillat and A. R. Odoni, "An integrated scheduling and operations approach to airport congestion mitigation," *Operations Research*, vol. 63, no. 6, pp. 1390–1410, 2015.

[3] M. Şeker and N. Noyan, "Stochastic optimization models for the airport gate assignment problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 2, pp. 438–459, 2012.

[4] A. J. Mason, D. M. Ryan, and D. M. Panton, "Integrated simulation, heuristic and optimisation approaches to staff scheduling," *Operations research*, vol. 46, no. 2, pp. 161–175, 1998.

[5] S. Alodhaibi, R. L. Burdett, and P. K. Yarlagadda, "Framework for airport outbound passenger flow modelling," *Procedia Engineering*, vol. 174, pp. 1100–1109, 2017.

[6] Y. Wang, M. Z. Li, K. Gopalakrishnan, and T. Liu, "Timescales of delay propagation in airport networks," *Transportation Research Part E: Logistics and Transportation Review*, vol. 161, p. 102687, 2022.

[7] B. Yu, Z. Guo, S. Asian, H. Wang, and G. Chen, "Flight delay prediction for commercial air transport: A deep learning approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 125, pp. 203–221, 2019.

[8] A. Jacquillat and A. R. Odoni, "Endogenous control of service rates in stochastic and dynamic queuing models of airport congestion," *Transportation Research Part E: Logistics and Transportation Review*, vol. 73, pp. 133–151, 2015.

[9] International Air Transport Association, "Worldwide airport slot guidelines," [WebPage], 2020, https://www.iata.org/en/policy/slots/slot-guidelines/.

[10] Civil Aviation Administration of China, "Regulations for the establishment and implementation of minimum standards for the operation of aircraft airports," [WebPage], 2017, http://www.gov.cn/gongbao/content/2017/content_5189036.htm.

[11] A. Kovacs, E. Németh, and K. Hangos, "Modeling and optimization of runway traffic flow using coloured petri nets," in *2005 International Conference on Control and Automation*, vol. 2. IEEE, 2005, pp. 881–886.

[12] M. C. R. Murca and R. J. Hansman, "Identification, characterization, and prediction of traffic flow patterns in multi-airport systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1683–1696, 2018.

[13] Federal Aviation Administration, "Simmod manual: How simmod works. faa capacity modeling & analysis group," [WebPage], 2017, https://www.tc.faa.gov/acb300/how_simmod_works.pdf.

[14] T. Cheng, D. Cui, and P. Cheng, "Data mining for air traffic flow forecasting: a hybrid model of neural network and statistical analysis," in *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, vol. 1. IEEE, 2003, pp. 211–215.

[15] Z. Yang, Y. Wang, J. Li, L. Liu, J. Ma, and Y. Zhong, "Airport arrival flow prediction considering meteorological factors based on deep-learning methods," *Complexity*, vol. 2020, 2020.

[16] G. Wei and W. Zhengyi, "Short-term airport traffic flow prediction based on lstm recurrent neural network," *Journal of Aeronautics, Astronautics and Aviation*, vol. 49, no. 4, pp. 299–307, 2017.

[17] D. Zhang and M. R. Kabuka, "Combining weather condition data to predict traffic flow: a gru-based deep learning approach," *IET Intelligent Transport Systems*, vol. 12, no. 7, pp. 578–585, 2018.

[18] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, pp. 64–67, 2001.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[20] M. Xu, Y. Yang, M. Han, T. Qiu, and H. Lin, "Spatio-temporal interpolated echo state network for meteorological series prediction," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 6, pp. 1621–1634, 2018.
[21] J. Wang, L. Zhang, Q. Guo, and Z. Yi, "Recurrent neural networks with auxiliary memory units," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 5, pp. 1652–1661, 2017.
[22] J. Li and J. Wang, "Short term traffic flow prediction based on deep learning," in *CICTP 2019*, 2017, pp. 2457–2469.
[23] Z. Yan, H. Yang, F. Li, and Y. Lin, "A deep learning approach for short-term airport traffic flow prediction," *Aerospace*, vol. 9, no. 1, p. 11, 2021.
[24] N. C. for Environmental Information, "Climate monitoring," [Web-Page], 2022, https://www.noaa.gov/.
[25] Y. Horiguchi, Y. Baba, H. Kashima, M. Suzuki, H. Kayahara, and J. Maeno, "Predicting fuel consumption and flight delays for low-cost airlines," in *Twenty-Ninth IAAI Conference*, 2017.
[26] C. Li and K. C. Hui, "Feature recognition by template matching," *Computers & Graphics*, vol. 24, no. 4, pp. 569–582, 2000.
[27] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition," *J. Off. Stat*, vol. 6, no. 1, pp. 3–73, 1990.
[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[30] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.