Explainable Multi-task Flight Arrival Delay Prediction

Tianqi Wang[1] and Lei Lin[2] and Jing Gao[3]

***Abstract*— Flight delays could disturb passengers' travel plans and cause financial loss to the airlines. Therefore, efforts have been spent in the past to conduct flight delay prediction, which could assist passengers, airports and airlines in better planning. Existing methods focus on predicting whether a flight will delay and how long the delay will be, but lack an effective explanation revealing possible reasons causing the delay, which is useful for timely decisions. Motivated by the demand of such cause-aware flight delay prediction, we propose an explainable multi-task framework to predict not only the delay duration but also the delay causes. The proposed framework consists of three modules: (1) flight record encoder that derives record embeddings, (2) flight sequence encoder integrating useful signals from other related flights, and (3) flight delay predictor that outputs delay duration as well as the possible causes of the delay. The proposed framework is evaluated on three real-world datasets. The results show that the proposed model outperforms multiple baselines on the flight arrival delay prediction task and is able to provide the causes behind the flight delay.**

***Index Terms*— Deep Learning, Multi-task Learning, Flight Arrival Delay Prediction**

I. Introduction

Flight delays leading losses to both the passengers and airlines have become a serious and widespread problem. Therefore, many efforts have been devoted towards solving the critical task of flight delay prediction. Among them, machine learning models [1], [2], [3], [4], [5] have demonstrated to be effective in predicting flight delays. These methods focus on predicting whether the flight delay will take place and the duration of the possible delays. Limited efforts have been spent on investigating flight delay causes. In [6], the authors model the causal factors for the flight delay and derive the weights of the causal factors. However, the causal factor analysis is conducted on the feature level. It is still hard to explain the delay causes as the weights are for individual features but not derived at a more meaningful semantic level. In addition, the interactions among features were not considered. In [2], the delay causes of the previous flight is used as additional input features to improve the current flight delay prediction. However, the model cannot predict the delay causes for the future flights.

According to the U.S. Bureau of Transportation Statistics [7], the causes of flight delay can be divided into five categories, which are air system cause, security cause, weather cause, airline cause and late aircraft cause. The delay duration caused by each category (i.e., cause-specific delay) is recorded for each past flight record. This provides a valuable information source for the task of flight delay prediction, but is seldom explored in existing work. The prediction on possible delay causes and their contributions to the duration of flight delay can provide important information to airline companies so that they can investigate these causes to mitigate possible delays. In addition, compared with existing models, a prediction model built upon cause-specific delays may have better prediction performance as the delay causes could be regarded as additional supervision to guide the prediction model.

Motivated by the aforementioned benefits, we propose an explainable deep learning framework for flight arrival delay prediction, which is named as Cause Aware Prediction of Flight Arrival Delay (CAP-FAD). There are three integral modules in the proposed CAP-FAD framework: (1) a flight record encoder that derives the embeddings of the raw features, (2) a flight sequence encoder which learns the representations of a sequence of related flights, and (3) a flight delay predictor that makes predictions. The flight delay predictor is based on a multi-task learning framework, where the representations are shared among the predictions on different delay types. The information about cause-specific delays is introduced into the CAP-FAD as additional supervision, which can not only improve the delay prediction performance but also explain possible delay causes. Another important source of information is departure delay as the objective is to predict arrival delay and departure delays are highly correlated with arrival delays. The experiments are conducted on three real-world datasets and the results show that the proposed CAP-FAD framework outperforms existing baselines and can predict the delay causes. The main contributions of this paper can be summarized as follows:

- We demonstrate the importance of incorporating possible delay causes into the fight delay prediction model, which leads to improvement in both the prediction performance and the model interpretability.
- We propose a novel multi-task deep learning framework for cause-specific delay prediction. It can not only predict if and how long the flight will delay, but also explain possible causes of the flight arrival delay,
- Comprehensive experiments are conducted on three real-world datasets. The results show that the proposed framework outperforms four existing methods and can predict the flight delay causes.

II. Related Work

In this section, we will review existing work related with flight delay prediction and multi-task learning.

[1]T. Wang is with Department of Computer Science and Engineering, University at Buffalo, Buffalo, New York, 14260 (email: twang47@buffalo.edu).

[2]L. Lin is with Goergen Institute for Data Science, University at Rochester, Rochester, New York, 14627 (email: Lei.Lin@rochester.edu)

[3]J. Gao is with School of Electrical and Computer Engineering, Purdue University, West Lafayette, 47907 (email: jinggao@purdue.edu)

Flight Delay Prediction: Various flight delay prediction models have been studied, ranging from classical statistical models to machine learning models. With respect to the statistical models, a multiple linear regression model is proposed to predict flight arrival delay based on departure delay and route distance in [8]. In [2], the Cox Proportional Hazards survival model is applied to capture the effects of multiple factors on flight departure and arrival delay. In terms of the machine learning models, Rebollo and Balakrishnan (2014) applied random forest model for flight departure delay classification and regression [9]. Balakrishna et al. (2010) investigated a reinforcement learning based approach to predict aircraft taxi-out delay, which is a major component in flight departure delay [10]. More recently, the Long Short-Term Memory (LSTM) [11] has been utilized to capture the impact of delay from previous flights to predict an aircraft's future departure delay [3].

Multi-Task Learning (MTL): MTL aims to optimize several learning tasks simultaneously. There are mainly two approaches of applying MTL in deep learning: hard parameter sharing and soft parameter sharing [12]. The former allows various tasks to share the same neural network layers for feature extraction and use task-specific layers to handle different tasks. The latter employs independent networks for tasks but applies message passing between specific layers. The concept of MTL can be extended to auxiliary learning in which tasks are classified as a primary task and additional auxiliary tasks [13]. The auxiliary tasks serve as regularizers to improve the generalization ability of the primary task to unseen data [13], [14].

We design a multi-task deep learning framework in this paper where departure and cause-specific arrival delay predictions are introduced as auxiliary tasks. These auxiliary tasks can serve as regularizers to enhance the performance of the primary flight arrival delay task. Meanwhile those auxiliary tasks reveal the possible delay causes explicitly, thus also explain the reasons behind the flight arrival delay.

III. Problem Statement

In this section, we first introduce important notations in this paper and then define the flight arrival delay prediction problem.

We denote the set of airports, the set of airlines and the aircraft set as $\mathcal{P}$, $\mathcal{L}$ and $\mathcal{C}$. For a flight F_j, it has an aircraft ID represented as c_j where $c_j \in \mathcal{C}$. The airline that F_j belongs to is denoted as l_j. The original and arrival airports of F_j are o_j and a_j, where o_j and $a_j \in \mathcal{P}$. The scheduled departure and arrival time of flight F_j are t_j^d and t_j^a. Thus the scheduled duration of flight F_j can be defined as t_j^f, where $t_j^f = t_j^a - t_j^d$. In addition, the distance between the origin airport and the arrival airport is represented by b_j. We use $\mathbf{F}_j =< c_j, l_j, o_j, a_j, t_j^d, t_j^a, t_j^f, b_j >$ to denote the general information of the flight F_j.

In addition to the general flight information, the weather condition of the origin airport at the scheduled departure time is also included in the record. We denote the predicted weather condition of the origin airport as $\mathbf{W}_j$. We use $\mathbf{R}_j =< \mathbf{F}_j, \mathbf{W}_j >$ to denote the flight record j. The set of all flight records is denoted as $\mathcal{R}$.

Let $\overline{t_j^d}$ and $\overline{t_j^a}$ denote the actual departure and arrival time of flight F_j. Thus the departure and arrival delay can be defined using the difference between the actual and scheduled departure and arrival time. The departure delay $d_d^j = max\{0, \overline{t_j^d} - t_j^d\}$. As defined by the commercial aviation industry, the flight arrival delay is the time of duration that a flight is late or postponed. The flights that arrive more than 15 minutes after its scheduled gate arrival time are considered as delayed flights [7]. Thus the arrival delay d_t^j of flight F_j is defined as:

$$d_t^j = \begin{cases} 0, & \overline{t_j^a} - t_j^a \leq 15 \\ \overline{t_j^a} - t_j^a, & \overline{t_j^a} - t_j^a > 15 \end{cases} \quad (1)$$

Meanwhile, let $d_j = \mathbb{1}(d_t^j > 0)$ denote if the flight F_j is delayed, where $\mathbb{1}(x)$ is the indicator function.

According to the U.S. Bureau of Transportation Statistics [7], there are five different causes that lead to the flight arrival delays: air system delay, security delay, airline delay, late aircraft delay, and weather delay. For the flight F_j, we use d_a^j, d_s^j, d_l^j, d_c^j and d_w^j to denote its duration of delay caused by air system, security, airline, late aircraft, and weather accordingly.

Two more concepts need to be explained: pre-order flight and flight record sequence. Pre-order flight is the previous flight which shares the same aircraft of the target flight. In another word, the destination of the pre-order flight is the departure airport of the target flight. We use $F_k \rightarrow F_j$ to represent that flight F_k is the pre-order flight of F_j. Flight record sequence related with F_j is the sequence of flight records that departure before the flight F_j from the same airport. The sequence is sorted by the scheduled departure times and represented as $\mathbf{S}_j =< \mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_j >$.

Finally, the flight arrival delay prediction can be defined as given the flight record set $\mathcal{R}$, predict if the flight will delay or not (i.e. d_j) and how long the flight will delay (i.e. d_t^j).

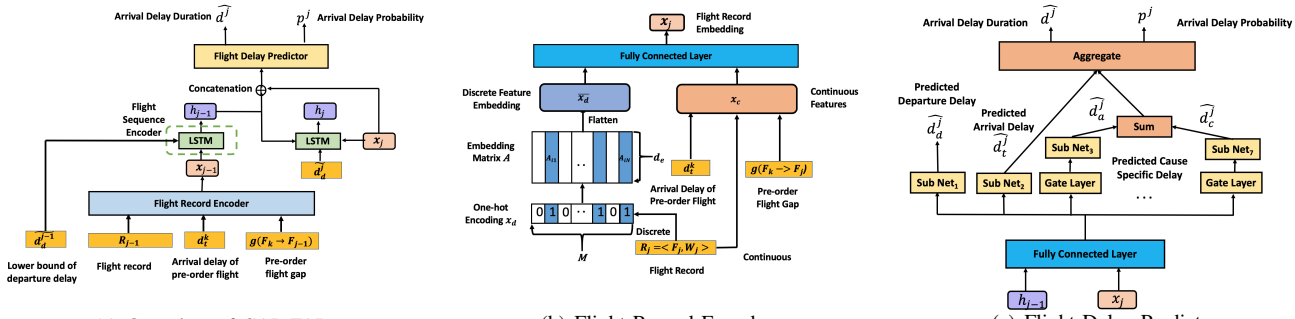
IV. Methodology

The architecture of the proposed CAP-FAD is shown in Figure 1(a). There are three modules in the proposed CAP-FAD framework: a flight record encoder, a flight sequence encoder and a flight delay predictor. The details of those modules are introduced in the following part.

A. Flight Record Encoder

The flight record encoder is used to encode the features of the target flight record $\mathbf{R}_j$ and the features related with the delay of its pre-order flight F_k. The inputs of the flight record encoder are: the flight record $\mathbf{R}_j =< \mathbf{F}_j, \mathbf{W}_j >$, the arrival delay d_t^k of the pre-order flight F_k, and the pre-order flight gap $g(F_k \rightarrow F_j)$ between the scheduled departure time t_j^d of F_j and the actual arrival time $\overline{t_k^a}$ of flight F_k.

The features $\mathbf{R}_j$, d_t^k and $g(F_k \rightarrow F_j)$ can be separated into discrete and continuous features. As shown in Figure 1(b), the flight record encoder encodes the discrete features into high-deminsional dense vectors and then concatenate the



(a) Overview of CAP-FAD (b) Flight Record Encoder (c) Flight Delay Predictor

Fig. 1: The overview of CAP-FAD and architecture of its flight record encoder and delay predictor.

embeddings of discrete features with the continous features. The concatenation is fed into a fusion layer to obtain the representation of the flight record. We denote the set of discrete features as $\mathcal{F}_D$ and the set of continuous features as $\mathcal{F}_C$. Assume there are N discrete features and M distinct values of all the N discrete features in $\mathcal{F}_D$. We denote the number of continuous features as $|\mathcal{F}_C|$. Thus the inputs of the flight record encoder can be represented as $\mathbf{x}_d \oplus \mathbf{x}_c$. $\mathbf{x}_d \in \{0,1\}^M$ represents the discrete features after the binary encoding. Each binary element in $\mathbf{x}_d$ represents the presence of that feature value. $\mathbf{x}_c \in \mathbb{R}^{|\mathcal{F}_C|}$ are the values of continuous features. $\oplus$ denotes the vector concatenation operation.

The flight record encoder learns an embedding matrix $\mathbf{A} \in \mathbb{R}^{d_e \times M}$, where the colomun i represents the embedding of the i_{th} value in the binary discrete feature representation $\mathbf{x}_d$ and d_e is the dimension of the embedding vector. We use $\overline{\mathbf{x}_d} \in \mathbb{R}^{d_e \times N}$ to denote the dense representation of $\mathbf{x}_d$:

$$\overline{\mathbf{x}_d} = (\mathbf{A}(i_1) \oplus \mathbf{A}(i_2) \oplus \cdots \oplus \mathbf{A}(i_N))^T \tag{2}$$

where $\mathbf{A}(i_1), \mathbf{A}(i_2), \cdots, \mathbf{A}(i_N)$ are the column $i_1, i_2, \cdots, i_N$ of the embedding matrix $\mathbf{A}$, and $i_1, i_2, \cdots, i_N$ represent the indices of element 1 in $\mathbf{x}_d$.

Then the flight record encoder flattens the dense representation of the discrete features $\overline{\mathbf{x}_d}$ and concatenate it with the continuous features $\mathbf{x}_c$ to get the embedding of the flight record. The embedding of the flight record can be denoted as $\mathbf{x} = \text{ReLU}(\mathbf{W}_r \times [\overline{\mathbf{x}_d} \oplus \mathbf{x}_c]^T + \mathbf{b}_r)$. $\mathbf{x} \in \mathbb{R}^{d_r}$ denotes the embedding of the flight record and d_r is the number of dimensions of the vector. $\mathbf{W}_r \in \mathbb{R}^{d_r \times (N d_e + |\mathcal{F}_c|)}$ and $\mathbf{b}_r \in \mathbb{R}^{d_r}$ are the trainable weights and bias. $ReLU(x) = max\{x, 0\}$ denotes the Rectified Linear Unit activation function.

B. Flight Sequence Encoder

It is observed that the delay of flights at the same airport during a period may not be independent from each other and the delay of an aircraft often propagates. Thus, we propose the flight sequence encoder to leverage the information of the flights related with the target flight to help predict its arrival delay.

We use $\mathbf{X} =< \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_j >$ to denote the embeddings obtained from the flight record encoder of the flight record sequence $\mathbf{S}_j =< \mathbf{R}_1, \mathbf{R}_2, \cdots, \mathbf{R}_j >$, where $\mathbf{R}_j$ is the record of the target flight. Since the sequence is ordered according to their scheduled take off times, the actual departure time may not follow that order. Some flights may departure after the target flight even they have earlier scheduled departure times. To handle this issue, we use the lower bound of the departure delay instead of actual departure delays of those flights to make predictions. What we know is that those flights have not departured until the time to make prediction. Thus the actual departure delay should be larger than the gap between the time when making prediction and the scheduled departure time. Let $\tilde{d_d^j}$ denote the lower bound of the actual departure delays of the flight record $\mathbf{R}_j$, which can be represented as:

$$\tilde{d_d^j} = min\{t_{cut} - t_j^d, d_d^j\} \tag{3}$$

where t_{cut} is the time of the prediction. t_j^d is the scheduled departure time of flight f_j and d_d^j is the actual departure delay of flight f_j.

The sequence of concatenation of the flight record embedding and lower bound of the departure delay is fed into a LSTM [11] to generate the representation of the flight sequence S. The outputted hidden state $\mathbf{h}_{j-1} \in \mathbb{R}^h$ defined in Eq. 4 by the LSTM integrates the historical information of the first $(j-1)$ flights and can be used to predict the delay of the F_j, where h is the dimension of the hidden states.

$$\mathbf{h}_{j-1} = LSTM(\mathbf{x}_{j-1} \oplus \tilde{d_d^{j-1}}) \tag{4}$$

C. Flight Delay Predictor

The flight delay predictor aims to predict if and how long the flight will delay on arrival. The flight arrival delay is highly related with the departure delay and can be caused by different reasons. In the proposed flight delay predictor, we utilize a multi-task learning architecture that takes the flight arrival delay as the primary task, and the departure delay and cause-specific delay prediction as the auxiliary tasks. The architecture of the flight delay predictor is shown in Figure 1(c). The inputs of the flight delay predictor are the hidden state vector $\mathbf{h}_{j-1}$ integrating the historical flight information of the first $(j-1)$ flights and the flight record embedding $\mathbf{x}_j$. There are seven sub-networks in the predictor and the final prediction is generated by integrating the prediction of those sub-networks. More details are described in the following part.

To integrate the historical information and the target flight record, a summary vector $\mathbf{v}_j \in \mathbb{R}^{d_s}$ is generated as:

$$\mathbf{v}_j = \text{ReLU}(\mathbf{W}_s \times [\mathbf{h}_{j-1} \oplus \mathbf{x}_j] + \mathbf{b}_s) \tag{5}$$

where $\mathbf{W}_s \in \mathbb{R}^{d_s \times (h+d_r)}$ and $\mathbf{b}_s \in \mathbb{R}^{d_s}$ are trainable matrices and d_s is the dimension of $\mathbf{v}_j$.

Then the summary vector is fed into seven different sub-networks to predict different types of flight delay. The seven sub-networks are used to predict the departure delay, arrival delay, air system delay, security delay, airline delay, weather delay and late aircraft delay separately. The subnetworks to predict the departure delay and arrival delay share the same architecture, and the five subnetworks to predict cause-specific delay share the same architecture. All those subnetworks share the same input $\mathbf{v}_j$.

Let $SN(x)$ denote the shared subnetwork architecture for departure and arrival flight delay prediction, where

$$SN(x) = FC(ReLU(FC(x))) \tag{6}$$

$FC(x)$ is the fully connected layer. Let $\hat{d_t^j}$ denote the predicted arrival delay of flight record $\mathbf{R}_j$ and $\hat{d_d^j}$ is the predicted departure delay. Then we have that:

$$\begin{aligned} \hat{d_t^j} &= SN_t(\mathbf{v}_j), p_t^j = \sigma(\hat{d_t^j}) \\ \hat{d_d^j} &= SN_d(\mathbf{v}_j), p_d^j = \sigma(\hat{d_d^j} - 15) \end{aligned} \tag{7}$$

where SN_t and SN_d, sharing the architecture with SN, are the sub-networks for predicting flight departure delay and arrival delay separately. p_t^j and p_d^j are the probability that the flight will delay on arrival and departure. $\sigma(x) = \frac{1}{1+e^{-x}}$ is the Sigmoid activation function.

Different from the architecture of the sub-networks for departure and arrival delay prediction, a gate layer is added to those sub-networks to filter out useful signals for the specific causes. Then the filtered signals will be fed into the fully connected network SN for the cause-specific flight delay prediction. Take air system delay as an example, let $\hat{d_a^j}$ and p_a^j be the predicted duration and probability, which can be derived as:

$$\begin{aligned} p_a^j &= \sigma(\hat{d_a^j}), \hat{d_a^j} = SN_a(\overline{\mathbf{v}_j}^a) \\ \overline{\mathbf{v}_j}^a &= \mathbf{v}_j \otimes \mathbf{v}_g^a, \mathbf{v}_g^a = \sigma(\mathbf{W}_g^a \mathbf{v}_j^T + \mathbf{b}_g^a) \end{aligned} \tag{8}$$

where SN_a is the sub-network sharing the same architecture of SN for air system delay prediction. $\mathbf{W}_g^a \in \mathbb{R}^{d_s \times d_s}$ and $\mathbf{b}_g^a \in \mathbb{R}^{d_s}$ are the trainable weights of the gate layer for the air system delay prediction. $\otimes$ represents the element-wise multiplication.

Similarly the predicted duration of security delay ($\hat{d_s^j}$), airline delay ($\hat{d_l^j}$), weather delay ($\hat{p_w^j}$) and aircraft delay ($\hat{d_c^j}$); as well as their corresponding predicted probability $p_s^j, p_l^j, p_w^j, p_c^j$ can be derived.

As shown in Fig. 1(c), the total arrival delay can be also represented by the sum of the delays of different causes. Thus we can aggregate the predicted arrival delay and the sum of cause-specific delays to make the final prediction. The final predicted flight arrival delay can be described as:

$$\hat{d^j} = \alpha \hat{d_t^j} + (1-\alpha)(\hat{d_a^j} + \hat{d_s^j} + \hat{d_l^j} + \hat{d_w^j} + \hat{d_c^j}) \tag{9}$$

α denotes the weighted factor that needs to be learned. And the probability that the flight will delay on arrival is represented as $p^j = \sigma(\hat{d^j} - 15)$.

D. Loss function

The CAP-FAD framework is trained based on losses from both primary and auxiliary tasks, which are arrival delay prediction, departure delay prediction, air system delay prediction, security delay prediction, airline delay prediction, weather delay prediction, late aircraft delay prediction and the aggregated final prediction. Let $\mathcal{T} = \{T_1, T_2, \cdots, T_8\}$ denotes the task set of those 8 tasks.

The losses for each prediction task include the classification loss and regression loss, where the classification loss is based on the cross-entropy loss and the regression loss is based on the mean square error (MSE).

The classification loss of task T_i is denoted as :

$$\mathcal{L}_{T_i}^c = \frac{-\sum(y_{T_i}^j log(p_{T_i}^j) + (1-y_{T_i}^j) log(1-p_{T_i}^j))}{|\mathcal{R}|} \tag{10}$$

$y_{T_i}^j$ and $p_{T_i}^j$ are the groundtruth label and the predicted probability that the flight will delay of record $\mathbf{R}_j$ in task T_i.

The regression loss for each prediction task is defined as:

$$\mathcal{L}_{T_i}^r = \frac{1}{|\mathcal{R}|}(max\{0, \hat{d_{T_i}^j} - t\} - d_{T_i}^j)^2 \tag{11}$$

$\hat{d_{T_i}^j}$ is the predicted delay of record $\mathbf{R}_j$ in task T_i. $t = 15$ for the arrival delay prediction and $t = 0$ for all other tasks.

The total loss $\mathcal{L}_{T_i}$ for task $T_i \in \mathcal{T}$ is the weighted sum of the classification loss and regression loss, which is:

$$\mathcal{L}_{T_i} = \mathcal{L}_{T_i}^c + \lambda \mathcal{L}_{T_i}^r \tag{12}$$

where λ is the balance weight.

Then the total loss of the proposed framework can be defined as the sum of the losses of those eight tasks, which can be defined as: $\mathcal{L} = \sum_{i=1}^{8} \mathcal{L}_{T_i}$ where $\mathcal{L}$ denotes the total loss of the proposed framework.

V. Experiment

A. Datasets

The datasets include two parts: one is the flight information and the other is the airport weather conditions.

The flight datasets were collected by the U.S. Department of Transportation's Bureau of Transportation Statistics[1]. There are three datasets including the on-time performance of domestic flights connecting 322 airports in the U.S. from January to March 2015 separately. Each dataset includes $4,898$ aircrafts from 14 airlines.

We collected the weather data of airport stations via Weather Underground[2]. The duration is from 1st January 2015 to 31st March 2015. The temporal resolution of the scrawled weather data is an hour. The collected weather information includes the temperature, weather condition, wind speed and precipitation.

The preprocessing steps are combining the flight dataset and weather information according to the airport and time,

[1] https://www.bts.dot.gov/topics/airlines-and-airports/understanding-reporting-causes-flight-delays-and-cancellations

[2] https://www.wunderground.com/

	Jan.	Feb.	Mar.
# of flight records	418,034	374,221	451,991
# of arrival delayed flights	84,653	84,238	84,163
mean and std of arrival delay	58±61	61 ±66	59±64
# of air system delayed flights	50,587	50,488	47,536
mean and std of air system delay	23±28	24±30	23 ±32
# of security delayed flights	269	207	269
mean and std of security delay	22±27	20 ±25	23 ± 32
# of airline delayed flights	46,514	46,162	49,315
mean and std of airline delay	33±58	34±59	33±59
# of weather delayed flights	5,884	8,180	4,108
mean and std of weather delay	40±59	46±76	50±81
# of late aircraft delayed flights	46,744	45,481	46,055
mean and std of late aircraft delay	42 ±47	43 ±50	42±50
# of departure delayed flights	161,155	158,333	177,034
mean and std of departure delay	32 ±50	33±53	31±51

TABLE I: Statistics of Datasets

and removing those flight records with missing values or without weather condition, or the flights that are cancelled. The detailed statistics of each dataset after preprocessing is shown in Table. I. All the delays are described in minutes.

B. Baselines

1) Linear regression (LR): The inputs of the model are the raw features. For the discrete features, one-hot encoding is applied.
2) XGBOOST: XGBOOST is a powerful aggregation model for the regression task. The inputs of the model are the same as the LR.
3) Cox proportional hazards (COX) [2]: Cox proportional hazards (COX) [15] is a classical survival analysis model. The COX model is proposed for flight delay propagation [2]. We use the COX model with the available features in the datasets as the baseline.
4) LSTM [3]: It considers the delay propagation and takes a 24-hour flight sequence as the input.

In addition to the above existing methods, we also compare with two variations of the CAP-FAD to show the effectiveness of designing subnetworks in it. The two variations are:

1) FAD: In FAD, the five cause specific subnetworks are removed (i.e. α in Eq. 9 is set to 1).
2) CAP-FAD-T: In this model, the subnetwork for direct arrival delay prediction (SN_t) is removed (i.e. α in Eq. 9 is set to 0).

C. Experimental Setup and Implementation Details

Each dataset is divided into the training dataset and testing dataset according to the date of the records. The flight records of the first 24 days of each month are used as the training dataset and the left are used for testing. The flight records are sorted by the scheduled departure time at each airport everyday. We use the flight sequence 30 minutes before the scheduled departure time of the target flight for prediction.

Tensorflow [16] is used to implement the model. The model is trained using a mini-batch with a size of 256. The dimension of the feature embeddings is set to 10 (i.e. $d_e = 10$) and the dimension of the hidden state is set to 50 (i.e. $h = 50$). The maximum number of epochs is set to 100. All the parameters are initialized randomly. The loss function is optimized by Adam Optimizer [17] with the initial learning rate set to 0.001. To reduce overfitting, we adopt dropout with a rate of 0.2 and perform early stop as well.

D. Results

The accuracy, recall, precision, F1 value, RMSE and MAE of the flight arrival delay prediction on the three datasets are shown in Fig. 2(a) to Fig. 2(f) separately. The unit of the RMSE and MAE is minute. We can observe that the pattern of model performance on all the metrics are highly consistent and the proposed framework outperforms all other baselines on all the metrics. It demonstrates the effectiveness of the proposed framework by considering the cause specific delay in the flight arrival delay prediction task. It is worth noting that although inferior to the CAP-FAD framework, its two variations FAD and CAP-FAD-T beat all other baselines.

The performance of LR is the worst. Since the LR model does not consider the delay propagation, which provides important signal for flight delay prediction task. In addition, the linear assumption may also limit its performance. XGBOOST achieves the second best performance among those baselines, and is only inferior to the LSTM model. This may be attributed to the power of model aggregation in the XGBOOST. The performance of COX is also worse than the performance of the LSTM model. It shows that although the COX model takes the propagation of flight delay into consideration, LSTM has better ability to capture useful signals for the flight delay prediction task.

E. Ablation Study

To further investigate the effectiveness of aggregating the direct arrival delay prediction and cause specific delay prediction, we conduct the ablation study. The proposed CAP-FAD framework is compared with its two variations: FAP which is without the cause specific prediction and CAP-FAD-T which only use the cause specific delay to predict the arrival delay. As shown in Fig. 2, it is can be noted that the performance metrics of those two models are comparable with each other while worse compared with the CAP-FAD model. It is possible that different information is captured by different sub-networks, so when aggregating them the performance will be improved. Moreover, it demonstrates the effectiveness and the necessity of aggregating the sub-networks for flight arrival delay prediction.

F. Delay Cause Analysis

In this section, we will evaluate the explainability of the proposed model. The prediction accuracies of all the cause specific delays on all the three datasets are beyond 90%, which demonstrates the ability to interpret the arrival delay causes of the proposed framework. Here is an example of the cause specific delay prediction shown in Fig. 2(g). The left and right plot present the pattern of the actual and predicted cause specific delay. We can find that the two patterns are very similar. The actual arrival delay is mainly caused by the air system, airline and weather, and the air system issue is the main reason of the arrival delay. It is consistent with the predicted distribution.

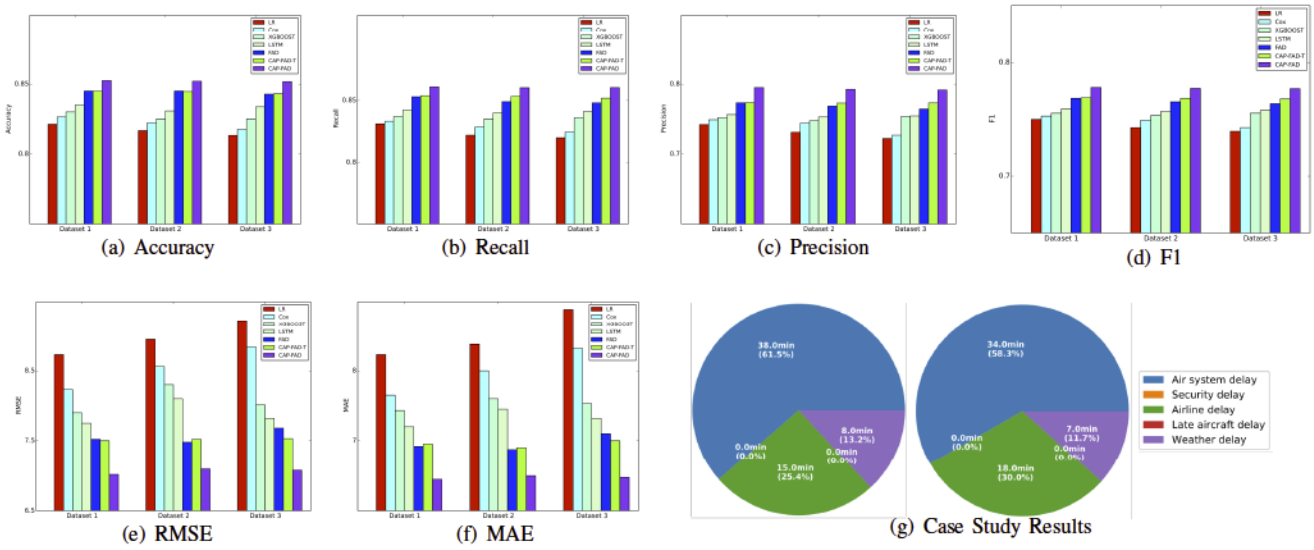


Fig. 2: Flight Arrival Delay Prediction and Case Study Results

VI. Conclusion

Accurate flight delay predictions are important for both travelers and commercial aviation companies. Although extensive machine learning techniques have been proposed for this task, little work has been done to capture the delay causes, which could be useful to aviation companies. In this paper, we proposed a novel multi-task deep learning framework CAP-FAD that provides cause-aware prediction of flight arrival delays. By this framework, the cause specific delay predictions and the departure delay prediction are integrated as auxiliary tasks so that better feature representations and useful signals for the primary flight arrival prediction can be captured. Comprehensive experiments are conducted on three real-world datasets. The results show that the CAP-FAD can not only provide an explanation of the delay causes but also achieve better prediction performance than baselines, suggesting the importance of considering the delay causes in the flight arrival delay prediction.

Acknowledgment

This work is supported in part by the US National Science Foundation under grant NSF-IIS 1956017. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s)and do not necessarily reflect the views of the National Science Foundation.

References

[1] 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.

[2] J.-T. Wong and S.-C. Tsai, "A survival model for flight delay propagation," *Journal of Air Transport Management*, vol. 23, pp. 5–11, 2012.

[3] N. McCarthy, M. Karzand, and F. Lecue, "Amsterdam to dublin eventually delayed? lstm and transfer learning for predicting delays of low cost airlines," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 9541–9546.

[4] W. Shao, A. Prabowo, S. Zhao, S. Tan, P. Koniusz, J. Chan, X. Hei, B. Feest, and F. D. Salim, "Flight delay prediction using airport situational awareness map," in *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2019, pp. 432–435.

[5] 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, p. 2749, 2020.

[6] S. Cheng, Y. Zhang, S. Hao, R. Liu, X. Luo, and Q. Luo, "Study of flight departure delay and causal factor using spatial analysis," *Journal of Advanced Transportation*, vol. 2019, 2019.

[7] Bureau of Transportation Statistics, "Airline on-time performance and causes of flight delays," https://www.bts.gov/topics/airlines-and-airports/airline-time-performance-and-causes-flight-delays, 2020, last accessed 16 July 2020.

[8] Y. Ding, "Predicting flight delay based on multiple linear regression," in *IOP Conference Series: Earth and Environmental Science*, vol. 81, no. 1, 2017, pp. 1–7.

[9] J. J. Rebollo and H. Balakrishnan, "Characterization and prediction of air traffic delays," *Transportation research part C: Emerging technologies*, vol. 44, pp. 231–241, 2014.

[10] P. Balakrishna, R. Ganesan, and L. Sherry, "Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: A case-study of tampa bay departures," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 6, pp. 950–962, 2010.

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

[12] Y. Liu, B. Zhuang, C. Shen, H. Chen, and W. Yin, "Auxiliary learning for deep multi-task learning," *arXiv preprint arXiv:1909.02214*, 2019.

[13] S. Liu, A. Davison, and E. Johns, "Self-supervised generalisation with meta auxiliary learning," in *Advances in Neural Information Processing Systems*, 2019, pp. 1679–1689.

[14] P. Vafaeikia, K. Namdar, and F. Khalvati, "A brief review of deep multi-task learning and auxiliary task learning," *arXiv preprint arXiv:2007.01126*, 2020.

[15] N. E. Breslow, "Analysis of survival data under the proportional hazards model," *International Statistical Review/Revue Internationale de Statistique*, pp. 45–57, 1975.

[16] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation)*, 2016.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.