Attention-based Deep Learning Model for Flight Delay Prediction using Real-time Trajectory

Tanaya Chaudhuri, Sheng Zhang, Yicheng Zhang
Institute for Infocomm Research (I^2R), A*STAR, Singapore
tanaya_chaudhuri@i2r.a-star.edu.sg; zhang_sheng@i2r.a-star.edu.sg; zhang_yicheng@i2r.a-star.edu.sg

***Abstract*—This paper presents a deep learning model termed LSTM-Attention based Time-dependent Flight-delay Classifier (LATTICE) for real-time flight arrival delay classification. Initially, this model incorporates a comprehensive set of factors influencing flight delays, including weather conditions, flight information, and en-route real-time trajectory data provided by ADS-B technology. Subsequently, LATTICE leverages a full-sequenced LSTM network for the extraction of deep temporal trajectory features and employs an attention network for the allocation of weights and mapping of relevant information. Ultimately, the model utilizes a masking layer to address the challenges posed by varying trajectory lengths, and experimental results demonstrate a significant enhancement in the accuracy of flight delay predictions as a result of these integrated measures. The model classifies incoming flights into On-Time/Late and Early/Punctual/Late. On being evaluated against historical data, it achieves about 91% accuracy and 0.96 AUC at predicting delay, yielding better predictions compared to baseline models. Trajectory inputs improve the prediction by about 15%. The model is real-time via ADS-B technology, robust via adaptive improvement with continuous training, and able to handle both late and early arrivals. This paper demonstrates that the real-time trajectory inferred from ADS-B messages can add significantly to the reliability of delay prediction.**

***Keywords*—Flight delay prediction, deep learning, sequential neural networks, LSTM, attention, ADS-B, real-time**

I. Introduction

The estimation and management of flight delays is a key performance indicator in the aviation sector [1]. With the ever-growing global aviation networks and their interdependence, and with the higher demand for global travel in the recovery from COVID, flight delay prediction is becoming increasingly crucial. Between 2015-2020, the worldwide aviation passenger count grew by over 30% from 3.5k to 4.7k mil/year [2] while in Singapore alone, it grew from 55.45 to 68.3 mil/year [3]. However, in 2017, about 19% of the flights were cancelled or delayed over 30min in Brazil [4]. In 2018, 21% of flights suffered more than 15min delays in the USA [5], and recently in 2023 Q1, about 24% of flights in Europe delayed over 15min [6]. The negative impact of these delays include passenger dissatisfaction, penalty to airlines, additional costs at the operational level and even environmental issues due to extra fuel consumption [7], [8].

On-time-performance is a major measure of airline and airport efficiency. Based on the nature of the flight, delays can be divided into departure and arrival delays. Aircrafts that experienced departure delays can propagate into arrival delays; average arrival delay is approximately the sum of the average departure and enroute delays [9]. Therefore, we focus mainly on arrival delays in this paper.

A. Related works

Sternberg et al. [10] and Carvalho et al. [1] have provided extensive literature review on flight delay prediction. Many initial works are based on statistical approaches. Clustered models have been adapted to historical data to forecast delays [11]. Non-parametric function to model delays have been used to analyze the USA airports' efficiency [12]. Mueller et al. [9] used probabilistic density functions to model departure, enroute and arrival delays. Besides, other classical approaches include network representation methods based on graph theory, or operational research based methods such as optimization, simulations and queue theory [13]–[15].

These classical methods are valuable to understand the delay factors, but they can fall short in terms of accuracy for individual flight delay predictions [16]. With the availability of massive air traffic data, an increasing number of learning-based works are being performed with promising results, as summarized in Table I. Broadly, delay can be modelled as a regression or classification task. Random forest (RF) was adopted in several studies [17]–[19]. While the first study used both weather and flight information, the second study used flight data alone; the third study analysed at a 2-hr forecast horizon with a test error of 19% in a binary classification of 60min delay. However, the 60min threshold is far from meeting the delay threshold of 15min set by the International Civil Aviation Organization (ICAO) [20]. In [21], support vector machine (SVM) models were used to predict delays using weather, airport demand and capacity, and flight schedule related factors. Pamplona et al. [22] employed artificial neural networks (ANN) to model flight information to classify delay from no-delay based on 15min threshold.

Recently, several studies have used deep learning methods as well. Yu et al. [16] proposed a deep belief network (DBN) combined with support vector regression (SVR) that predicts delay at 25min error tolerance. Factors like airport crowdedness, air route situation reflecting weather, and delay propagation were suggested beside the common airline factors. Zhu and Li [23] developed a spatial weighted recurrent neural network (RNN) model to predict the delays using ADS-B, weather and airline record data. However, the ADS-B data was used only for the purpose of estimating the actual arrival times. Li et al. [24] combined convolutional neural network

TABLE I. PRIOR STUDIES ON LEARNING-BASED DELAY PREDICTION

Study and year	Type of prediction			Considered factors			Methodology		
	Reg.	Classification		Flight	Weather	Traj.	ML	DL	Algorithm(s)
		Late	*Early/Late*						
[19] '14	✓	✓		✓	✓		✓		RF
[17] '16		✓		✓	✓		✓		RF
[26] '16		✓		✓	✓			✓	LSTM
[22] '18		✓		✓			✓		ANN
[16] '19	✓			✓				✓	DBN,SVR
[25] '19		✓		✓	✓			✓	RF,LSTM
[21] '20		✓		✓	✓		✓		SVM
[23] '21	✓			✓	✓			✓	LSTM
[11] '21	✓			✓			✓		ARIMA
[18] '21	✓			✓			✓		RF
[24] '23		✓		✓	✓			✓	CNN-LSTM,RF
Ours		✓	✓	✓	✓	✓		✓	**ALSTM,ANN**

Note:- Regression: predicting the delay time; Classification *Late*: predicting whether the flight will delay; Classification *Early/Late*: predicting whether the flight will be early/punctual/late; ML: Machine learning; DL: Deep learning; RF:Random Forest; DBN:Deep belief network; SVR(M):Support vector regression(machine); CNN:Convolutional neural network; ALSTM:Attention-based LSTM

(CNN), Long Short-Term Memory (LSTM) network and RF for classifying delays using flight schedule, aircraft capacity, distance between airports and previous flight delay where LSTM was used to draw temporal features from weather. Gui et al [25] studied RF and LSTM separately for predicting individual flight delay. LSTM was used to model weather and air route features for temporal correlation extraction.

B. Contributions

Although several promising works are done on learning-based approaches to predict flight delay, there are some concerns and areas of improvement as summarized in Table I. Firstly, in the current literature only late arrivals are investigated, even though early arrivals can also have negative impacts on the traffic flow. Arrivals before the planned schedule affects the surface management on aerodrome as well as the terminal management. In this paper, we analyse both late and early arrivals, which can contribute to smoother operations.

Secondly, it is more common to employ static ground factors such as flight information and weather, that do not represent real-time status of the flight in motion. Delay prediction methods are crucial to help the airline operators to correctly comprehend the current and future status of flights and help the air controllers make prompt decisions. Therefore, delay predictions should run in real-time for better outcomes. This could be achieved by incorporating dynamic airspace factors. In this study, we propose a novel idea to incorporate the en-route real-time flight trajectory, a dynamic airspace factor, to enable predictions to be real-time and more reliable. We extract it from surveillance system namely, Automatic Dependent Surveillance-Broadcast (ADS-B) communication messages.

The ADS-B system is a promising technology in air traffic control. As ADS-B signals are time-series data, they have been used for estimation of flight coordinates [27] and flight trajectory [28]. They have been used in [25] to estimate the arrival time based on the time of last signal received due to the unavailability of the actual arrival time in their study. In another study [29], ADS-B data was used to extract the flight's location only at the range ring instead of its entire trajectory. In our study, we explore the entire trajectory to enhance predictions.

Thirdly, classical methods and shallow ML approaches cannot well-capture the temporal aspects of flight movements. Given that trajectory is time-series data, deep sequential neural networks like the LSTM would be apt for modelling in our study. Current delay prediction studies have used LSTM networks only for modelling weather [24], delay states [23], or weather along with flight schedules [26]. Several advantages of the LSTM such as the ability to capture time dependencies, handling long sequences and temporal feature learning make it suitable for modelling the trajectory in our study. However, modelling long flight sequences can suffer from information loss. We suggest using attention mechanism [30] to prevent such loss, besides providing better interpretation and appropriate information mapping. The contributions of this paper can be summarized as:

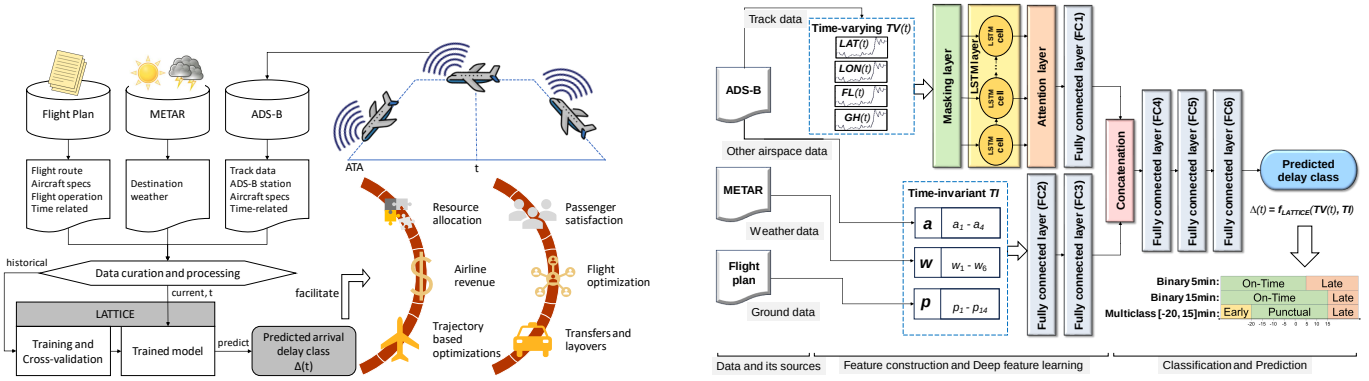
- A novel deep learning model for real-time arrival delay classification of flights named as **L**STM-**A**ttention based **T**ime-dependen**T** fl**I**ght-delay **C**lassifi**E**r (LATTICE) is proposed. A full-sequenced LSTM network enables deep temporal trajectory features extraction, while an attention network facilitates adequate weight assignment and relevant information mapping from the time-steps. In addition, the model employs a masking layer to address the challenges with varying trajectory lengths.
- While most of the literature focus on late arrivals, our model differentiates between early and late arrivals as well. Specifically, LATTICE classifies incoming flights into on-time/late as well as early/punctual/late.
- Along with weather condition and flight information (ground static factors), we propose using en-route real-time trajectory data (airspace dynamic factors) via ADS-B technology to enhance predictions. We explore a broad scope of factors and propose several new features.
- The model efficacy is validated using real historical data of Singapore Changi airport, against data-driven algorithms used in prior studies.

Incorporating real-time track data makes our approach reliable, while using data-driven strategy based on periodic retraining can render robustness. To the best of our knowledge, this study is the first to use real-time trajectory of the enroute flight through ADS-B technology as a dynamic factor, along with static factors as weather and flight information, powered by a deep learning model based on full-sequenced LSTM and attention networks for flight delay prediction for both early and late arrivals.

II. PROPOSED LATTICE MODEL

A. Overall Framework

Implementation of the proposed model is illustrated in Fig 1a. For an enroute flight, the model can be applied once the ADS-B communications start broadcasting. The selected airspace and ground data from flight plan, METAR and ADS-B will be gathered and features will be extracted. They will



(a) Model overview, implementation and applications (b) Model architecture

Figure 1. Proposed LATTICE model for real-time arrival flight delay classification

be fed to the trained model, which would predict the arrival flight delay class. The model would be employed cyclically because predictions would improve as more trajectory data is accumulated with time. This dynamic prediction could help the airport regulators to be aware of potential delays in advance and to develop management strategies to improve on-time-performance. Fig 1a indicates the several applications that could be facilitated by the proposed approach.

The architecture of the LATTICE model is illustrated in Fig. 1b. The objective is two-fold:

1) Binary classification: We predict whether the arriving flight will incur a delay. We define a flight is delayed if the Actual In-Block Time (AIBT) of the flight is lagging behind the Scheduled In-Block Time (SIBT) by a given threshold. ICAO states 5min and 15min as thresholds for arrival punctuality with respect to AIBT and SIBT [20]. Therefore, we run separate experiments with 5min and 15min as the delay threshold. The target class is defined as:

$$\Delta(t) = \begin{cases} \text{On-time} & \text{if AIBT - SIBT <= threshold} \\ \text{Late} & \text{otherwise} \end{cases}$$

2) Multiclass classification: We predict whether a flight will have an early arrival, be punctual or have a late arrival. We select 15min as the threshold for *late* arrivals in the multiclass task as it is more practically used and 5min would be rather too stringent. For *early* arrival, we use 20min as threshold to apply a stricter penalty for the *late* arrivals, as late arrivals have more consequences than the early arrivals. The target class is thus defined as:

$$\Delta(t) = \begin{cases} \text{Early} & \text{if AIBT - SIBT} < -20 \\ \text{Punctual} & \text{if} -20 <= \text{AIBT - SIBT} <= 15 \\ \text{Late} & \text{if AIBT - SIBT} > 15 \end{cases}$$

Broadly, the LATTICE model $f_{LATTICE}$ has two branches with two sets of input vectors- time-invariant $\boldsymbol{TI}$ and time-varying $\boldsymbol{TV}(t)$ (Fig. 1b). Processed track TV features is fed to an LSTM network through a masking layer, and filtered through an attention layer. Meanwhile, the TI features are fed to fully connected (FC) layers. The learned deep features from the two branches are concatenated and projected to the target class through a deep network of FC layers. The model can be symbolically represented as

$$f_{LATTICE} : \{\boldsymbol{TI}, \boldsymbol{TV}(t)\} \rightarrow \Delta(t) \quad (1)$$

B. Flight Delay Feature Space

We consider a wide range of factors as described in Table II-flight route, operation, aircraft, weather, time-related, ADS-B station, and trajectory. They can be categorised as ground and airspace information. 24 TI and 4 TV features are extracted from these factors. We define the feature space as follows:

1) Time-invariant features:

$$\boldsymbol{TI} = [\boldsymbol{p}, \boldsymbol{w}, \boldsymbol{a}] \quad (2)$$

where $\boldsymbol{p}$ is the feature vector from flight plan, $\boldsymbol{w}$ is the weather vector and $\boldsymbol{a}$ is the airspace vector:

$$\boldsymbol{p} = [p_1, p_2, p_3, ...p_{14}] \quad (3)$$

where the acronyms are described in Table II. For consistency and variability, we used ICAO codes for departure airport (p_1). The actual time of departure ATD (p_7) is encoded numerically as (hour*100 + minute). The time elapsed (p_{10}) between the creation of flight plan and the ATD is noted. Since flight plans are created closer to scheduled departure, larger gaps of p_{10} may indicate a delayed departure. Using the Estimated Off-Block Time (EOBT) and the Scheduled In-Block Time (SIBT), the departure ground delay (p_9) and the approximate flight duration (p_{11}) are computed as:

$$p_9 = \text{ATD} - \text{EOBT} \quad (4)$$

$$p_{11} = \text{SIBT} - \text{ATD} \quad (5)$$

The true air speed (p_{13}) is computed using mach number (p_{12}) and speed of sound c:

$$p_{13} = c * p_{12} \quad (6)$$

$$\boldsymbol{w} = [w_1, w_2, w_3, w_4, w_5, w_6] \quad (7)$$

Weather is a major cause of delays and flight cancellations. Singapore being a tropical country, witnesses bouts of heavy

TABLE II. FEATURES USED TO PREDICT THE FLIGHT ARRIVAL DELAY

Factors	Features (unit, type)	Acronym	TI/TV
Ground information from flight plan (FP)			
Route	departure airport (-, cat)	p_1	
	airline (-, cat)	p_2	
	callsign (-, cat)	p_3	
Aircraft	aircraft type (-, cat)	p_4	
	aircraft registration (-, cat)	p_5	
	wake turbulence category (-, cat)	p_6	
Time-related	actual time of departure (-, num)	p_7	
	actual day of week (-, num)	p_8	
	departure ground delay (min, num)	p_9	
	timespan: FP creation to ATD (min, num)	p_{10}	TI (24 singular features)
	approximate flight duration (min, num)	p_{11}	
Operation	mach number (-, num)	p_{12}	
	true air speed (m/s, num)	p_{13}	
	planned flight level (100 ft, num)	p_{14}	
Ground information from weather station			
Destination	air temperature ($^{\circ}F$, num)	w_1	
	relative humidity (%, num)	w_2	
	wind speed (knots, num)	w_3	
	wind direction (deg true N, num)	w_4	
	visibility (mi, num)	w_5	
	weather event (-, cat)	w_6	
Airspace information via ADS-B communications			
Aircraft	emitter category (-, num)	a_1	
Time-related	timespan: ATD to 1^{st} signal (min, num)	a_2	
ADS-B station	system area code (-, num)	a_3	
	system identification code (-, num)	a_4	
Trajectory	latitude (deg, num)	LAT	TV (4 sequence features)
	longitude (deg, num)	LON	
	flight level (100 ft, num)	FL	
	geometric height (ft, num)	GH	

Note:- cat: categorical, num: numerical, ATD: actual time of departure

rainfall and thunderstorms year-round, which can affect visibility at the airport. Weather at the destination airport as listed in Table II are noted. Weather event w_6 is represented by METAR codes as described in [31].

$$\boldsymbol{a} = [a_1, a_2, a_3, a_4] \tag{8}$$

Certain time-invariant airspace information received via ADS-B are noted. We introduce the time elapsed between the departure and the first ADS-B signal (a_4) as a feature.

2) Time-varying features:

$$\boldsymbol{TV(t)} = \{\boldsymbol{LAT}(t), \boldsymbol{LON}(t), \boldsymbol{FL}(t), \boldsymbol{GH}(t)\} \tag{9}$$

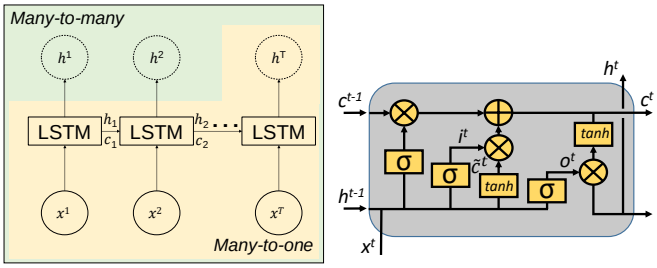
Flight trajectory parameters namely, the latitude $\boldsymbol{LAT}$, longitude $\boldsymbol{LON}$, flight level $\boldsymbol{FL}$ and geometric height $\boldsymbol{GH}$ sequence vectors of the flight are noted:

$$\boldsymbol{LAT}(t) = \{.., lat(t-2), lat(t-1), lat(t)\} \tag{10}$$
$$\boldsymbol{LON}(t) = \{.., lon(t-2), lon(t-1), lon(t)\} \tag{11}$$
$$\boldsymbol{FL}(t) = \{.., fl(t-2), fl(t-1), fl(t)\} \tag{12}$$
$$\boldsymbol{GH}(t) = \{.., gh(t-2), gh(t-1), gh(t)\} \tag{13}$$

While $\boldsymbol{GH}$ reflects flight altitude above mean sea level, $\boldsymbol{FL}$ indicates height in terms of pressure altitude.

C. Full-sequenced Long Short-Term Memory (LSTM)

We selected LSTM for its ability to capture long-term dependencies which makes it suitable to model trajectory [32]. However, trajectory lengths change with time and between different flights. Therefore, the TV input sequences are of variable length, which can be challenging to model directly using



(a) Types of LSTM based on output (b) Structure of an LSTM cell

Figure 2. Long Short-Term Memory (LSTM)

LSTM. Downsampling/upsampling/averaging to a fixed length could cause significant information loss as well. Therefore, we retain the original sequences by employing a masking layer.

The structure of LSTM layer can be many-to-one or many-to-many (Fig. 2a). LATTICE model employs a many-to-many or full-sequenced layout where hidden states from all time-steps of the input trajectory sequence are extracted. Each hidden state h carries information from the past cell as well as current input, thereby better capturing the temporal dependencies of delay on the trajectory. Fig. 2b describes a single LSTM cell. Assuming that x^t and h^t are the input and the hidden state, respectively, at time-step t, and c^{t-1} is the cell state, the LSTM network can be expressed as:

$$f^t = \sigma(w_f[h^{t-1}, x^t] + b_f) \tag{14}$$
$$i^t = \sigma(w_i[h^{t-1}, x^t] + b_i) \tag{15}$$
$$\tilde{c^t} = tanh(w_c[h^{t-1}, x^t] + b_c) \tag{16}$$
$$c^t = f^t * c^{t-1} + i^t * \tilde{c^t} \tag{17}$$
$$o^t = \sigma(w_o[h^{t-1}, x^t] + b_o) \tag{18}$$
$$h^t = o^t * tanh(c^t) \tag{19}$$

where, f^t is the forget gate, i^t is input gate, o^t is the output gate, w_f, w_i, w_c, and w_o are the weights, b_f , b_i, b_c, and b_o are the biases, and $\sigma(.)$ and *tanh(.)* are the sigmoid and tanh functions, respectively.

D. Attention mechanism

We use attention mechanism [30] because it can help alleviate the information loss from long sequences processed by LSTM. It can identify the most relevant amongst the extracted deep features from LSTM. Assuming there are m=500 cells in the LSTM layer, output will be 500 vectors each of length equal to the number of timestamps in the input TV sequence. For n samples, the output will be $n \times m \times$timestamps. The attention mechanism helps to identify the importance of both the input features and the time-steps by paying attention to every hidden state generated by the LSTM. Let the learned features by the LSTM be its hidden states $H = \{h_1, h_2, ...h_m\}$. Alignment scores are calculated for each encoded state by training a single unit feedforward network, and the attention scores are obtained as

$$r_i = tanh(W^T h_i + b) \tag{20}$$

where W is the weight matrix and b is the bias vector. Attention weights $\alpha_1, \alpha_2, ..\alpha_m$ are generated by applying *softmax* function to the scores. The final output of the attention layer is the weighted sum:

$$C = \alpha_1 * h_1 + \alpha_2 * h_2 + ... + \alpha_m * h_m \tag{21}$$

E. Concatenation and prediction

The learned deep features from TV and TI are concatenated and fed through another deep network of FC layers. Rectified Linear Unit $ReLU(x) = max(x, 0)$ is used as the activation function. Assuming that $(C)^k = \{(c)_1^k, (c)_2^k, ..\}$ is the input vector and $(Y_{FC})^k$ is the output vector for the k^{th} training sample, the FC layer can be formulated as:

$$V_i^k = ReLU\left(\sum w_{ij}(c)_j^k - bh_i\right) \tag{22}$$

$$(Y_{FC})_p^k = \sum V_i^k w_{pi} - bo_p \tag{23}$$

where, V_i^k is the output of hidden neuron i, w_{ij} is weight parameter from input layer neuron j to hidden layer neuron i, bh_i is the bias of hidden neuron i, w_{pi} is the weight parameter from hidden neuron i to output neuron p, and bo_p is the bias of output neuron p. In the final layer FC6, 1 output unit with *sigmoid* activation $\sigma(x) = \frac{1}{(1+e^{-x})}$ and 3 output units with *softmax* activation $s(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$ are used for binary and multiclass classifications, respectively. The LATTICE model can thus be represented as

$$\Delta(t) = f_{LATTICE}(\boldsymbol{TI}, \boldsymbol{TV}(t)) \tag{24}$$

III. Experimental Results and Discussion

A. Experimental settings

The flight plan and the ADS-B data are procured from the Civil Aviation Authority of Singapore (CAAS), while the weather data is obtained from METAR [31]. We considered all flights inbound Changi International Airport, Singapore between Nov 19 to Dec 31, 2019 which amounted to about 15,000 flights. The processed data is downsampled to minutes for consistency. Experiments are run at 3 different time-points namely, at 60min, 40min and 20min before arrival (ATA-60, ATA-40, ATA-20). The class distributions for all experiments are presented in Fig. 3. It is evident that there are class imbalances. To avoid majority class bias, first, class weights are added as shown in figure, and second, cross-validation with repeated stratified folds are used. Stratification ensures the folds preserve the sample distribution for each class.

Table III describes the hyperparameter settings used and the inputs to the layers of the LATTICE model. Cross-validation reduces noise and improves reliability of the trained model. We used Repeated Stratified KFold for cross-validation. Our dataset is split into 10 folds, with 2 repeats. Therefore, for each classification task, 20 sets of experiments (of train+validate+test) are run and the mean results are noted. The learning rate is tuned by grid search over $[10^{-5}, 10^{-3}]$, and the optimal rates as shown in the table are used. Maximum epochs

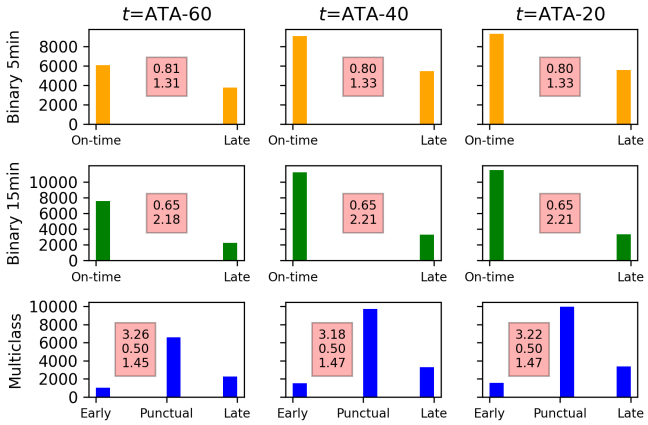


Figure 3. Distribution of the six types of experiments performed.

TABLE III. Hyperparameter settings of the LATTICE model

Layer	Attributes	Inputs
Masking	mask value: 0	TV input data
LSTM	units: 500, full-sequenced time-steps: trajectory length	Masked TV features
Attention	units: 500	Features learned by the LSTM
FC1	units: 100, activation: *ReLu*	Weights created by Attention
FC2	units: 500, activation: *ReLu*	TI input data
FC3	units: 100, activation: *ReLu*	Features learned by FC2
Concat	FC1 + FC3	Features learned by FC1, FC3
FC4	units: 100, activation: *ReLu*	Concat features of FC1+FC3
FC5	units: 50, activation: *ReLu*	Features learned by FC4
FC6	Binary (1 unit): *sigmoid*, *lr*:10^{-5} Multiclass (3 units): *softmax*, *lr*:10^{-4}	Features learned by FC5

of 500 are used. However, to prevent overfitting, regularization is added with early stopping based on the validation loss. A validation split of 20% is used for the training. The train loss functions used for the binary and multiclass tasks are the binary cross-entropy (BCE) and the categorical cross-entropy (CCE), respectively. We implemented the LATTICE model using the *Tensorflow* package in *Python*.

The models are evaluated using the classification accuracy and the AUC (Area Under the ROC Curve) metrics:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{25}$$

$$\text{AUC} = \int \text{TPR}\ d(\text{FPR}) \tag{26}$$

where, TPR $= \frac{TP}{TP+FN}$, FPR $= \frac{FP}{TN+FP}$ and, TP, TN, FP, FN, TPR and FPR are true positive, true negative, false positive, false negative, true positive rate and false positive rates, respectively. AUC metric applies for binary task only.

B. Baselines

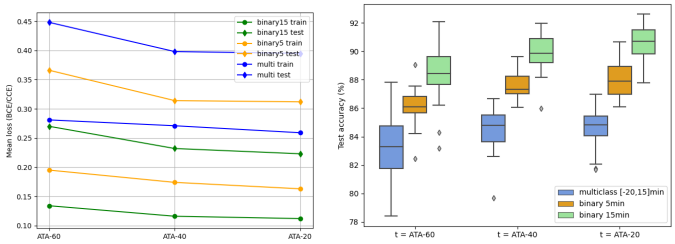
We implement several learning-based algorithms used in previous delay studies for comparative performance analysis:

1) Shallow learning algorithms (ML): L2 regularised Logistic Regression (LR-L2) [33], a popular method for predicting binary outcomes; Random Forest (RF) [17]–[19], [24], [25] and Extreme Gradient Boosting (XGB) [34], both decision tree algorithms; and Support Vector Machine (SVM) [16], [21] with Radial Basis Function kernel.

TABLE IV. CROSS-VALIDATED TEST RESULTS OF LATTICE MODEL

	$binary_5$		$binary_{15}$		multi-class
	Accuracy (%)	AUC	Accuracy (%)	AUC	Accuracy (%)
ATA-60	86.106±1.346	0.927±0.011	88.397±2.078	0.942±0.011	83.201±2.326
ATA-40	87.668±1.003	0.944±0.006	89.861±1.429	0.956±0.007	84.471±1.539
ATA-20	87.973±1.248	0.945±0.008	90.580±1.321	0.960±0.005	84.555±1.576

Note: Cross-validated mean ± s.d over 20 experiments (10 folds, 2 repeats).



(a) Mean train and test loss (b) Test accuracy

Figure 4. LATTICE model performance at different time-points

2) Deep learning algorithms (DL): Long Short-Term Memory (LSTM) [23], [26] and Gated Recurrent Unit (GRU) [35] networks with May-to-one structure; and 1D Convolutional Neural Network (1D-CNN) [24].

C. Effect of the reference time-point

The prediction results of the LATTICE model are presented in Table IV. Firstly, it is observed that the performance improves with reference points nearer to the ATA. This can be contributed to the additional 20min of TV information. With time nearing the ATA, TI features remain constant while more trajectory information gets updated and accumulated. Recent updates of the trajectory adds reliability to the prediction. The models are able to train better with more TV data as also seen from the improving training loss in Fig. 4a.

Secondly, it is to be noted that even 1hr prior (t =ATA-60), the LATTICE model is able to perform quite well at about 86% and 88% accuracy and 0.927 and 0.942 AUC for 5min and 15min cases, respectively. Thirdly, it is observed that a threshold of 15min fares better than 5min across all train and test metrics. Using 15min to define delay achieves best mean accuracy and AUC of 90.580% and 0.96, respectively. It indicates that a threshold of 5min may be a bit too stringent to segregate delayed arrivals.

These results are summarized graphically in Fig. 4b. Firstly, it is clearly seen that binary 15min task performs the best at all time-points. Multi-class task fares lower as it is harder to differentiate between 3 classes as compared to only 2 classes. At the time-point with the largest sample size (t =ATA-20), the binary 15min task achieves mean accuracy of 90.580%, which is about 3% higher than binary 5min (87.973%) and 7% higher than the multi-class task (84.555%). Secondly, the effect of the reference time-points can also be revealed from the illustration. For the binary 15min task, performance improves by 0.8% in 20min (from 40 to 20min prior ATA), and by 2.5% in 40min (from 60 to 20min prior ATA).

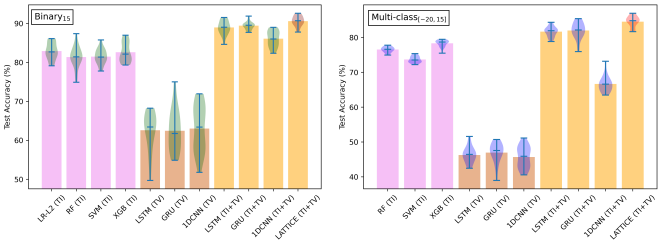


Figure 5. Performance comparison with the baselines

D. Model comparison

Since $binary_{15}$ task showed better performance than $binary_5$ task, from here on in this paper, we will present results only for the 15min case. The prediction results of the models are summarized in Table V and distributions are illustrated in Fig. 5. Comparisons are made at the reference point containing the highest flight data, t=ATA-20. In general, it is seen that the predictive performance P using TI+TV features is better than that using only TI features, which is better than that using only TV features, i.e. $P_{TI+TV} > P_{TI} > P_{TV}$.

First, using only TI features, the shallow models perform better than the deep models using only TV features ($P_{TI} > P_{TV}$). Moreover, with the addition of TI features, the same deep models witness significant improvements ($P_{TI+TV} > P_{TV}$). Based on the data in Table V, in LSTM, GRU and 1DCNN, improvements of 42.1%, 43.2% and 36.6%, respectively, are seen with the inclusion of TI features. In multi-class task, improvements of 76.5%, 74.7% and 45.9% are seen for LSTM, GRU and 1DCNN, respectively. This implies that the intrinsic factors such as flight information and weather are crucial. The standalone trajectory information without these intrinsic factors are not meaningful to relate to the delay.

However, with the TI and TV features combined, the deep models are able to perform much better than the shallow models using only TI features ($P_{TI+TV} > P_{TI}$). For binary task, the accuracy of the deep models range between 86-91% as compared to 81-83% for the shallow models. For multi-class task, the accuracy of the deep models range between 66-85% as compared to 73-78% for the shallow models. This reveals that while the intrinsic factors are vital to forecasting delay, the addition of real-time trajectory information improves the prediction and makes it more reliable and robust. The flight, weather, and trajectory information when combined is more meaningful to project delay.

Above all, it can be seen that the proposed LATTICE model outperforms the baseline methods at both binary and multi-class tasks (Fig. 5). Based on the data in Table V, the proposed LATTICE is 1.3% to 5.2% better than the deep models using TI+TV features, 9.3% to 11.3% better than the shallow models using TI features, and 43.7% to 45% better than the deep models using TV features, for the binary task. And it is 3.2% to 26.8% better than the deep models using TI + TV features, 8% to 14.7% better than the shallow models using TI features, and 80% to 85% better than the deep models using TV features,

TABLE V. COMPARISON OF PERFORMANCE BETWEEN THE PROPOSED MODEL AND THE BENCHMARK APPROACHES

		Shallow learning algorithms				Deep learning algorithms (t = ATA-20)						
		Input: TI (static factors)				Input: TV (dynamic factors: trajectory)			Input: TI + TV			
	Metric	LR-L2	RF	SVM	XGB	LSTM	GRU	1DCNN	LSTM	GRU	1DCNN	**LATTICE**
Binary	AUC	0.890±0.009	0.884±0.008	0.898±0.006	0.877±0.011	0.643±0.027	0.646±0.023	0.653±0.013	0.954±0.006	0.953±0.006	0.927±0.008	**0.960±0.005**
15min	Acc(%)	82.910±2.206	81.360±2.735	81.474±1.933	82.619±2.506	62.591±4.845	62.463±5.253	63.024±5.812	88.957±1.754	89.421±0.965	86.079±2.035	**90.580±1.321**
Multiclass	Acc(%)	NA	76.564±0.758	73.701±0.896	78.333±0.985	46.265±2.276	46.950±2.841	45.681±2.988	81.666±1.436	82.012±2.369	66.661±2.238	**84.555±1.576**

Note: Reported values are cross-validated mean ± s.d over 20 experiments (10 folds, 2 repeats). Best values are highlighted in bold.

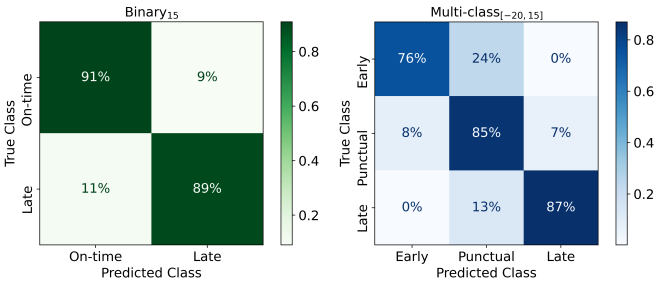


Figure 6. Accumulated confusion matrix of classifications by LATTICE.

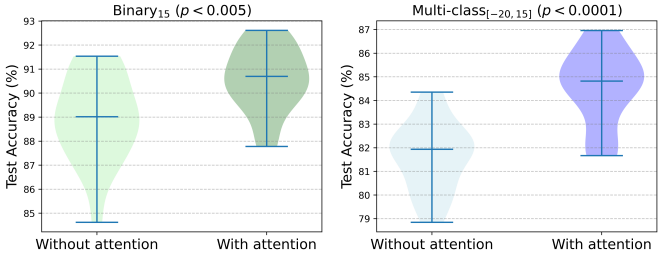


Figure 7. Significance of the attention mechanism in LATTICE.

TABLE VI. COMPUTATION TIME TO TRAIN 1000 AND TEST 1 SAMPLE(S)

Inputs	Model	Binary$_{15}$		Multi-class$_{[-20,15]}$	
		Train	Test	Train	Test
TV	1DCNN	0.07	0.005	0.03	0.002
	GRU	0.32	0.024	0.29	0.021
	LSTM	0.35	0.026	0.34	0.026
TI+TV	1DCNN	0.11	0.008	0.04	0.003
	GRU	0.45	0.033	0.32	0.024
	LSTM	0.49	0.037	0.35	0.026
	LATTICE	0.59	0.044	0.45	0.033

for the multi-class task. This suggests that the full sequenced LSTM layer and the attention mechanism can better capture the complex varying-range temporal dependencies of the flight trajectory and greatly improve the model performance. They help to extract the most relevant information without loss from the trajectory and efficiently combine it with the intrinsic information to project the delay.

For further evaluation of the predictive performance, Fig. 6 presents the accumulated confusion matrix over the 20 cross-validated experiments. The matrix diagonals denote the correct predictions. First, the heatmaps reveal high accuracies for all the evaluated classes. Second, closer class predictions (91%, 89%) in binary$_{15}$ task reveals a balanced performance between the two classes. In multi-class task, *Punctual* and *Late* classes are better forecasted as compared to *Early* class. Some *Early* cases are misclassified as *Punctual*, which largely contributed to the overall reduced prediction. However, it is observed that there are no misclassifications between end classes *Early* and *Late*, implying the model's reliability.

E. Significance of attention mechanism

Fig. 7 demonstrates importance of using the attention mechanism in the LATTICE model. The experiments with attention mechanism are clustered towards higher accuracies. The addition of the attention layer significantly improved the mean performances by 1.8% and 3.5% for the binary 15min and multiclass tasks, respectively. We performed a Welch T-test which revealed that these improvements are significant at the p-values mentioned in the figure. By assigning adequate weights, only the information most relevant to delay are extracted and relayed further down the layers by the attention mechanism. It processes not only the input trajectory features pertaining to the flight location but also processes the time-steps of the trajectory. In other words, it takes into account both the feature values as well as the temporal aspect of the features. Thereby it enhances the model performance.

F. Computation time of deep models

We present the computation times of the deep models on a workstation with 8 core CPUs of Intel i7-9700 3.60 GHz and a CUDA-enabled GPU of NVIDIA GeForce RTX 2080 in Table VI. The LATTICE model is expensive to train because it has the most complex structure and consequently, the largest set of trainable parameters. The 1DCNN model is non-RNN and therefore computes faster, and the models processing more features (TI+TV) takes longer to compute. With LATTICE, the time required to train 1000 flights is 0.59s and 0.45s, while to predict delay for single flight is only 0.044s and 0.033s, for binary and multi-class tasks, respectively. These computation times are quite reasonable especially given that the proposed model would be trained once in a while when a considerable amount of historical data has accumulated.

IV. CONCLUSIONS

In this paper, we proposed a deep learning model for the real-time arrival delay prediction of flights with the novel use of real-time trajectory data using ADS-B communications, besides the flight information and weather data. A total of 24 TI and 4 TV features were proposed including some new factors such as departure ground delay, approximate flight duration, time elapsed between flight plan creation and ATD, and between ATD and the first ADS-B signal. Among model

components, a full-sequenced LSTM network helped retain the temporal relevance of all time-steps in the trajectory, while the attention mechanism enabled adequate information mapping. In addition, the deep network of FC layers enabled an intensive feature extraction from the TI data. The model performed better than baseline shallow and deep learning algorithms on historical data. This study reveals that while ground information is vital, the addition of real-time flight trajectory makes prediction more reliable. On adding trajectory inputs, the prediction was greatly improved by about 8-15% and 44-85% as compared to TI and TV features alone, respectively. A 15min threshold was observed to yield better performance than 5min. Additionally, the computation times of the models were analysed for practicality.

By predicting both early and late arrivals, the proposed model can enhance airline operations; real-time trajectory makes our approach reliable, while learning based adaptive strategy makes it robust. However, the model can be implemented post departure provided the ADS-B communication has already begun. In future, we plan to extend our model to predict the actual delay time as regression task using algorithms like ensemble deep learning and transformers, and incorporate more factors based on spatial traffic complexity, multi-airport scenario, and weather data of entire flight route.

Acknowledgments

The authors would like to express their sincere appreciation to the Civil Aviation Authority of Singapore (CAAS) for providing valuable data and insights. This work is supported by the National Research Foundation (NRF), Singapore, and CAAS, under the Aviation Transformation Programme. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of NRF or CAAS.

References

[1] L. Carvalho *et al.*, "On the relevance of data science for flight delay research: a systematic review," *Transport Reviews*, vol. 41, no. 4, pp. 499–528, 2021.

[2] E. B. Salas. (2022) Global air traffic scheduled passengers 2004-2022. [Online]. Available: https://www.statista.com/statistics/564717/airline-industry-passenger-traffic-globally/

[3] Statista. (2022) Number of passengers at Changi Airport in Singapore 2014-2021. [Online]. Available: https://www.statista.com/statistics/867835/singapore-number-of-passengers-changi-airport/

[4] ANAC. (2017) Agância Nacional de Aviação Civil. Technical report. [Online]. Available: http://www.anac.gov.br/

[5] U.S. Dept. of Transp. (2018) Air Travel Consumer Report. [Online]. Available: https://www.transportation.gov/briefing-room/dot1219

[6] EUROCONTROL. (2023) All-causes delays to air transport in Europe Quarter 1 Tech. report. [Online]. Available: https://www.eurocontrol.int/publication/all-causes-delays-air-transport-europe-quarter-1-2023

[7] R. Britto *et al.*, "The impact of flight delays on passenger demand and societal welfare," *Transp. Res. E Logist. Transp. Rev.*, vol. 48, no. 2, pp. 460–469, 2012.

[8] M. S. Ryerson *et al.*, "Time to burn: Flight delay, terminal efficiency, and fuel consumption in the national airspace system," *Transp. Res. A Policy Pract.*, vol. 69, pp. 286–298, 2014.

[9] E. Mueller and G. Chatterji, "Analysis of aircraft arrival and departure delay characteristics," in *AIAA's Aircraft Technology, Integration, and Operations (ATIO) 2002 Technical Forum*, 2002, p. 5866.

[10] A. Sternberg *et al.*, "A review on flight delay prediction," *arXiv preprint arXiv:1703.06118*, 2017.

[11] M. Güvercin *et al.*, "Forecasting Flight Delays Using Clustered Models Based on Airport Networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 3179–3189, 2021.

[12] S. Pathomsiri *et al.*, "Impact of undesirable outputs on the productivity of us airports," *Transp. Res. E: Logist. Transp. Rev.*, vol. 44, no. 2, pp. 235–259, 2008.

[13] K. F. Abdelghany *et al.*, "A model for projecting flight delays during irregular operation conditions," *J. Air Transp. Manage.*, vol. 10, no. 6, pp. 385–394, 2004.

[14] A. Kim and M. Hansen, "Deconstructing delay: A non-parametric approach to analyzing delay changes in single server queuing systems," *Transp. Res. B Methodol.*, vol. 58, pp. 119–133, 2013.

[15] N. Pyrgiotis *et al.*, "Modelling delay propagation within an airport network," *Transp. Res. Part C Emerg.*, vol. 27, pp. 60–75, 2013.

[16] B. Yu *et al.*, "Flight delay prediction for commercial air transport: A deep learning approach," *Transp. Res. E: Logist. Transp.*, vol. 125, pp. 203–221, 2019.

[17] L. Belcastro *et al.*, "Using scalable data mining for predicting flight delays," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 8, no. 1, pp. 1–20, 2016.

[18] Z. Guo *et al.*, "A novel hybrid method for flight departure delay prediction using random forest regression and maximal information coefficient," *Aerosp. Sci. Technol.*, vol. 116, p. 106822, 2021.

[19] J. J. Rebollo and H. Balakrishnan, "Characterization and prediction of air traffic delays," *Transp. Res. Part C Emerg.*, vol. 44, pp. 231–241, 2014.

[20] ICAO. KPI Overview. [Online]. Available: https://www4.icao.int/ganpportal/ASBU/KPI

[21] E. Esmaeilzadeh and S. Mokhtarimousavi, "Machine learning approach for flight departure delay prediction and analysis," *Transp. Res. Rec.*, vol. 2674, no. 8, pp. 145–159, 2020.

[22] D. A. Pamplona *et al.*, "Supervised neural network with multilevel input layers for predicting of air traffic delays," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–6.

[23] X. Zhu and L. Li, "Flight time prediction for fuel loading decisions with a deep learning approach," *Transp. Res. Part C Emerg.*, vol. 128, p. 103179, 2021.

[24] Q. Li *et al.*, "A cnn-lstm framework for flight delay prediction," *Expert Systems with Applications*, vol. 227, p. 120287, 2023.

[25] G. Gui *et al.*, "Flight delay prediction based on aviation big data and machine learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 140–150, 2019.

[26] Y. J. Kim *et al.*, "A deep learning approach to flight delay prediction," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. IEEE, 2016, pp. 1–6.

[27] Y. Chen *et al.*, "Hybrid N-Inception-LSTM-Based Aircraft Coordinate Prediction Method for Secure Air Traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2773–2783, 2022.

[28] Z. Shi *et al.*, "4-D Flight Trajectory Prediction With Constrained LSTM Network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7242–7255, 2021.

[29] I. Dhief *et al.*, "Speed control strategies for E-AMAN using holding detection-delay prediction model," in *Proc. 10th EUROCONTROL SESAR Innov. Days*, 2020, pp. 1–10.

[30] D. Bahdanau *et al.*, "Neural machine translation by jointly learning to align and translate," *arXiv:1409.0473*, 2014.

[31] NOAA National Weather Service. Weather symbols. [Online]. Available: https://www.aviationweather.gov/metar/symbol

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

[33] R. Nigam and K. Govinda, "Cloud based flight delay prediction using logistic regression," in *International Conference on Intelligent Sustainable Systems (ICISS)*, 2017, pp. 662–667.

[34] G. Wang *et al.*, "A high-precision method of flight arrival time estimation based on xgboost," in *International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*. IEEE, 2020, pp. 883–888.

[35] T. Chaudhuri *et al.*, "An attention-based deep sequential GRU model for sensor drift compensation," *IEEE Sens. J.*, vol. 21, no. 6, pp. 7908–7917, 2020.