上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Aviation Flight Delay Prediction Based on ARIMA and TFT

AM417 Group2 Project Report

陈嘉懿 520120910145

陈琪颖 521120910160

金相恒 521120910042

张天成 521021910798

June 3, 2024

Contents

1 Introduction

Civil aviation, encompassing all non-military and non-state aviation, undeniably plays a crucial role in the transportation system. However, various factors, including but not limited to weather conditions, military control, major events, and emergencies, contribute to a significant likelihood of prolonged delays in civil flights. These delays can persist for several hours or even longer, causing significant inconvenience to people's lives. Recognizing this issue, we have decided to employ ARIMA and TFT methods to predict flight delays based on flight data, aiming to minimize the disruption caused by unforeseen flight delays.

1.1 Problem Description

Our prediction task entails utilizing the collected data to construct models for forecasting flight arrival delays. Given the distinct characteristics of different models, our ARIMA model will rely solely on historical arrival delays for its predictions. In contrast, our TFT model will incorporate a broader range of data for its forecasts, including flight airlines, departure and arrival airports, and weather conditions, among other factors.

2 Data Description

2.1 Data Source

The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics tracks the on-time performance of domestic flights operated by large air carriers.

2.2 Definition of Time Series

When it comes to define the time series, we notice that the flight data is precise to the second, reflecting instantaneous rather than daily cumulative information. Moreover, the data varies in flight duration and daily frequency. As illustrated in table 1 below, Flight UA72 frequently operates multi-stop routes within a single day. For instance, on January 5, 2015, it flew from HNL to SFO, and subsequently from SFO to DEN.

Table 1: Multi-destination flights within a day

Month	Day	Airline	Flight Number	Original Airport	Destination Airport
1	5	UA	72	HNL	SFO
1	5	UA	72	SFO	DEN

Therefore, it is not appropriate to directly construct a time series using natural time units such as days or hours from the raw data. To define the raw data as a standard time series, our exploration of the data has yielded two significant findings:

1. Scheduled Flights: 15.67% of flights operate on a fixed daily schedule, flying the same route. More specifically, the scheduled departure and arrival times, as well as the departure and arrival airports, remain consistent each day.

2. Circuit Flights: 78.95% of flights follow a set plan each day, flying fixed routes. This means that within a day, several routes are flown, with all routes being connected end-to-end, and the daily flying schedule is fixed. To visually illustrate this characteristic, we plot the six airports that flight AS65 traverses in a single day.

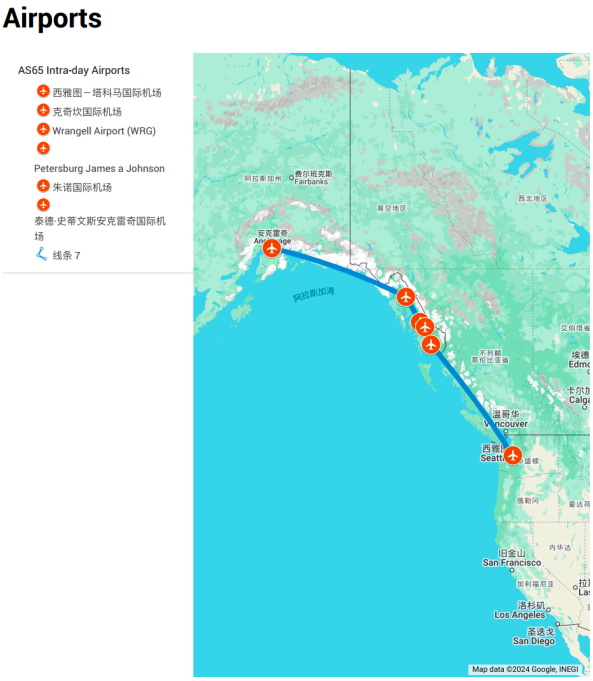

Figure 1: AS65 Intra-day Airports

From these observations, we can decompose the original problem into the prediction of delay times for scheduled flights and circuit flights, thus establishing separate time series for each. Scheduled flights take off at a fixed time each day, allowing their time series to be considered as one occurring on the natural calendar. In contrast, circuit flights have additional itinerary routes within a day, treating "one day" as a cycle, with each cycle comprising m time points.

Scheduled Flights:

$$\{Y_d\}, \quad d = 1 \dots 365 \tag{1}$$

Circuit Flights:

$$\{Y_{dt}\}, \quad d = 1 \dots 365, \quad t = 1 \dots m \tag{2}$$

Here, m represents the number of circuit routes within a day.

3 Model

3.1 ARIMA Model Construction

Based on the previous definition of flight delay time series, we hereby construct the ARIMA models.

Since ARIMA models cannot handle relationships between flights, we establish separate ARIMA models for each flight's data, allocating the first 80% of data as training data to independently estimate model parameters, and the remaining 20% as testing data to evaluate model performance. To facilitate subsequent presentation, we select 4 scheduled flights and 4 circuit flights as examples, with specific information as follows:

- Scheduled Flights:
    1. AA1042 (One flight record per day, totaling 365 entries for the year)
    2. AA1406 (One flight record per day, totaling 365 entries for the year)
    3. AA1486 (One flight record per day, totaling 365 entries for the year)
    4. AA1567 (One flight record per day, totaling 365 entries for the year)
- Circuit Flights:
    1. AS64 (Total of 425 flight records for the year, with 5 flight routes per day in the circuit route)
    2. AS65 (Total of 427 flight records for the year, with 5 flight routes per day in the circuit route)
    3. DL2452 (Total of 171 flight records for the year, with 2 flight routes per day in the circuit route)
    4. DL1270 (Total of 170 flight records for the year, with 2 flight routes per day in the circuit route)

Scheduled Flights For scheduled flights, we first draw the ACF and PACF plots to preliminarily determine our ARIMA model parameters and then refine the final model parameters based on AIC, AICc and BIC. Since no obvious seasonal patterns were detected from the plots, we rule out the use of SARIMA model.

Circuit Flights Since circuit flights have multiple flight routes within a single day, the delay of a previous route may lead to subsequent delays, i.e., $Y_{d,h} = \phi_1 Y_{d,1} + \cdots + \phi_{h-1} Y_{d,h-1}$, $h = 1, 2, \cdots, m$. Therefore, considering the autocorrelation, we prioritize $p = m - 1$ in the ARIMA model. We will use this as a basis, combined with the ACF and PACF plots, to preliminarily determine the ARIMA model parameters and further refine the final model parameters based on AIC, AICc and BIC. Additionally, through data exploration, we found no seasonality between different days, meaning the correlation between the same route on different days is trivial, hence we also rule out the SARIMA model in circuit flights.

3.2 TFT Model Construction

Temporal Fusion Transformer(TFT) is an attention-based DNN architecture for multi-horizon forecasting.

There are many problems in the application of the traditional time series model and machine learning model. For example, DNN and RNN models often fail to consider the different types of inputs commonly present in multi-horizon forecasting. However, in practical applications, the types of input variables to the model are often diverse (e.g. future known variables, variables known in the past but unknown in the future and static metadata). Therefore, multidimensional input can better express the actual problem. Another problem is, for those ML models, they are difficult to interpret, that is, they are "black box" models.

TFT model can solve the problems mentioned above. It contains processing modules for different input types. As shown in figure 2, static variables are processed by static covariate encoders, past-known variables are processed by LSTM encoders, pre-known variables are used by LSTM decoders. All the information is then integrated into interpretable multi-head attention block. In all processing processes, we can see the presence of gating mechanisms(Gated Residual Network), which can help the model to skip over any unused components of the architecture. Owe to the introduction of the attention layer, the model has remarkable interpretability.

Overall, TFT is able to analyze global temporal relationships and allows users to interpret global behaviors of the model on the whole dataset, for example, the identification of persistent patterns.

Another difference from traditional models is that TFT model can predict different quantiles of target values. Each quantile forecast takes the form:

$$\hat{y}_i(q, t, \tau) = f_q(\tau, y_{i,t-k:t}, z_{i,t-k:t}, x_{i,t-k:t+\tau}, s_i), \tag{3}$$

where $\hat{y}_{i,t+\tau}(q, t, \tau)$ is the predicted qth sample quantile of the τ-step-ahead forecast at

time t, and $f_q(.)$ is the prediction model. $y_{i,t}$ means the target variables, $z_{i,t}$ means the observed variables which are unknown beforehand, $x_{i,t}$ means the known variables which can be predetermined and s_i means the static variables. In line with other direct methods, the model simultaneously output forecasts for $\tau_{\max}$ time steps.

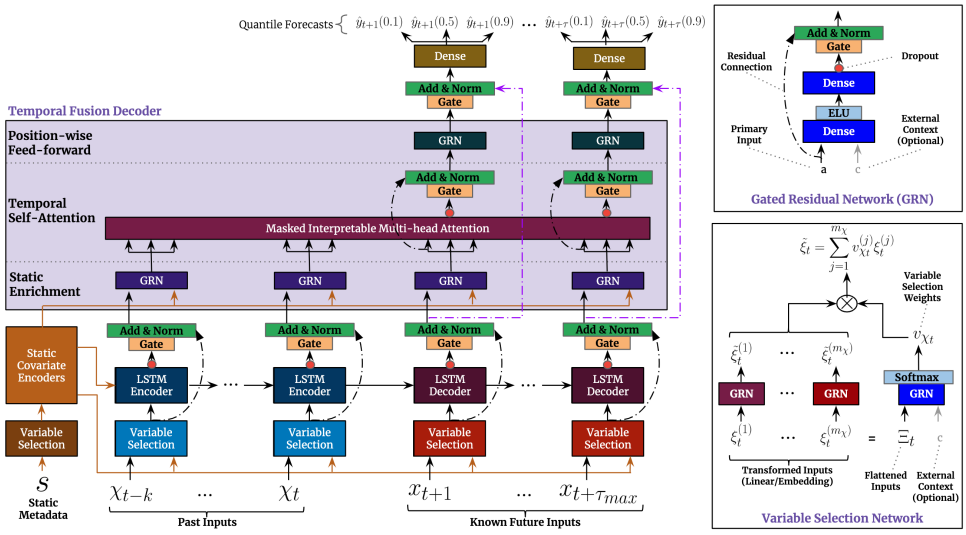

Figure 2: TFT Architecture

In our research problem, we will import the following variables into the model, shown in table 2.

Variable Type	Variable name	Meaning of Variable
Static real	DISTANCE	flying distance
Static categorical	FLIGHT_ID	flight number
	ORIGIN_AIRPORT_IATA	departure airport code (IATA)
	DESTINATION_AIRPORT_IATA	destination airport code (IATA)
Time varying known real	time_idx	timestamp
	SCHEDULED_TIME	planned flight time
	DAY_OF_WEEK	-
	XX_YYYY	XX: SD: scheduled departure, SA: scheduled arrival YYYY: YEAR: -, MONTH: -, DAY: -, H: hour, M: minute, S: second
Time varying known categorical	TAIL_NUMBER	aircraft unique identification number
Time varying unknown real	DEPARTURE_DELAY	total delay on departure
	TAXI_OUT	time from departure to wheel lift
	AIR_TIME	actual flight time
	TAXI_IN	landing to approach time
	ARRIVAL_DELAY	arrival delay time
	AIR_SYSTEM_DELAY	delay time due to flight congestion, major events, tower orders, etc
	SECURITY_DELAY	delay time due to national security and other reasons
	AIRLINE_DELAY	delay time caused by the route itself
	LATE_AIRCRAFT_DELAY	delay time caused by the plane itself
	WEATHER_DELAY	delay time due to extreme weather

Table 2: Variable Explanation

We train the TFT model on the first 300 days and use the rest for validation. At

each forecast point, we predict the next 14 days of delays based on the last 90 days of data. The selection of other hyperparameters is detailed in the code.

4 Data Analysis and Result Evaluation

4.1 ARIMA Data Analysis and Result Evaluation

To evaluate the rationality of the ARIMA model and its predictive performance, we sequentially conduct stationarity tests, residual analysis, and analysis of the final model's predictive performance indicators.

4.1.1 Stationarity Test

Since time series prediction with models requires the assumption that the analyzed time series is (weakly) stationary, we examine whether the data meets the stationarity assumption through ACF/PACF plots and ADF tests. Based on the ACF and PACF of the circuit and scheduled flight data (plots can be seen in appendix), combined with the ADF unit root test results in table 3, we draw the following conclusions:

- The selected four scheduled flight data are all stationary time series, with autocorrelation mainly occurring between closer terms (weakly dependent).
- 1. AS64 flight data is a stationary time series.
  2. AS65 flight data's first-order difference is a stationary time series.
  3. DL2452 flight data is a stationary time series.
  4. DL1270 flight data's first-order difference is a stationary time series.

Table 3: ADF test results

Circuit				Scheduled			
FlightID	d	Test-Statistic	p-value	FlightID	d	Test-Statistic	p-value
AS64	0	-9.4121	0.0000	AA1042	0	-6.8338	0.0000
AS65	1	-13.0887	0.0000	AA1406	0	-5.9511	0.0000
DL2452	0	-3.041	0.0312	AA1486	0	-14.2987	0.0000
DL1270	1	-10.9286	0.0000	AA1567	0	-11.3103	0.0000

4.1.2 Model Parameter Selection

Based on the observations and analyses, for the selected data of four scheduled flights and four rotational flights, we have determined the final parameters of the ARIMA models using the fitting results guided by AIC, AICc and BIC as information criteria. The final parameters are as follows:

Table 4: Final Selection of ARIMA Model Parameters

Circuit					Scheduled				
FlightID	ARIMA	p	d	q	FlightID	ARIMA	p	d	q
AS64	(4,0,3)	4	0	3	AA1042	(2,0,0)	2	0	0
AS65	(3,1,4)	3	1	4	AA1406	(2,0,1)	2	0	1
DL2452	(2,0,4)	2	0	4	AA1486	(1,0,0)	1	0	0
DL1270	(2,1,2)	2	1	2	AA1567	(1,0,0)	1	0	0

The parameters of the final model are in line with our expectations based on the visual observations.

Scheduled Flights The data itself is stationary, hence the differencing order d is 0 for all; the data has weak autocorrelation, mainly significant at close lags (weakly dependent), and the partial autocorrelation of AA1042, AA1486, and AA1567 shows a cut-off pattern after the first order (AA1486, AA1567) or second order (AA1042), thus the final fitted p and q are very small, with q being 0 except for AA1406.

Circuit Flights For AS64 and AS65, due to strong intra-day autocorrelation but weak inter-day autoregression, the final fitted model's p is slightly less than m (=5); for DL2452 and DL1270, with only two flights per day, they exhibit similar autocorrelation characteristics to scheduled flights, hence a smaller p (only 2); since circuit flights have more than one flight per day, the impact of the error term lasts for several periods, resulting in a larger q compared to scheduled flights, and some of the flights show non-stationary (first-order stationary) characteristics.

4.1.3 Residual Analysis

To assess the fit of the ARIMA model and to test whether the residuals are white noise, we analyze based on the ACF plots of the residuals and the Ljung-Box test. Based on the residual ACF plots in figure 3 and figure 4, and the results of the Ljung-Box test

in table 5, we consider that the model fits well for all flights, and the residuals can be regarded as white noise series.

Table 5: Ljung-Box Test Results

Circuit				Scheduled			
FlightID	ARIMA	Ljung-Box	Prob-Q	FlightID	ARIMA	Ljung-Box	Prob-Q
AS64	(4,0,3)	0.04	0.84	AA1042	(2,0,0)	0.04	0.83
AS65	(3,1,4)	0.00	0.98	AA1406	(2,0,1)	0.02	0.90
DL2452	(2,0,4)	0.01	0.93	AA1486	(1,0,0)	0.03	0.87
DL1270	(2,1,2)	0.12	0.73	AA1567	(1,0,0)	0.06	0.81

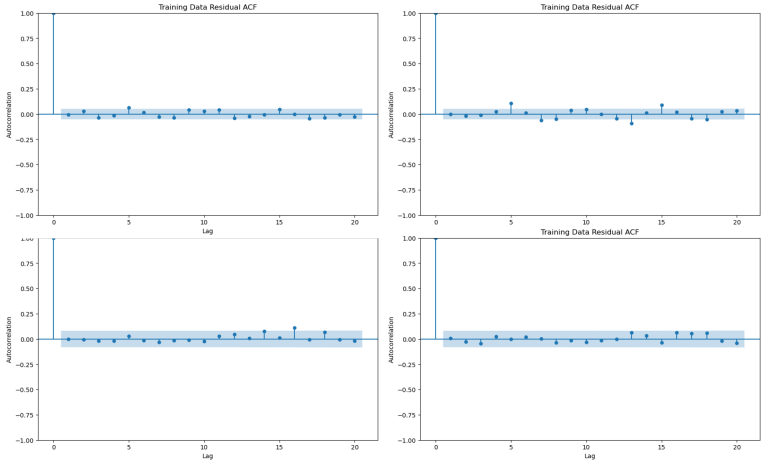

Figure 3: Residual Analysis of Circuit Flights

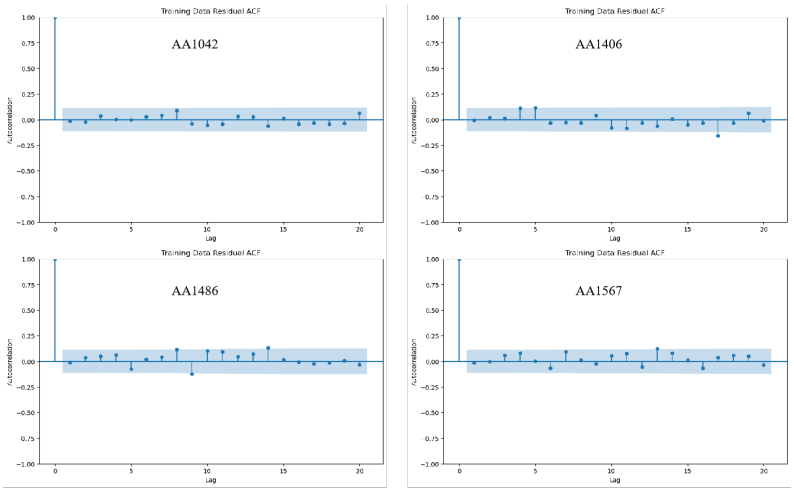

Figure 4: Residual Analysis of Scheduled Flights

4.1.4 Model Metrics

Based on the trained ARIMA models, we plotted the forecast results against the actual data for comparison, as can be seen in figure 5 and figure 6. Meanwhile, we present two model performance metrics, MAE and RMSE, as in table 6.

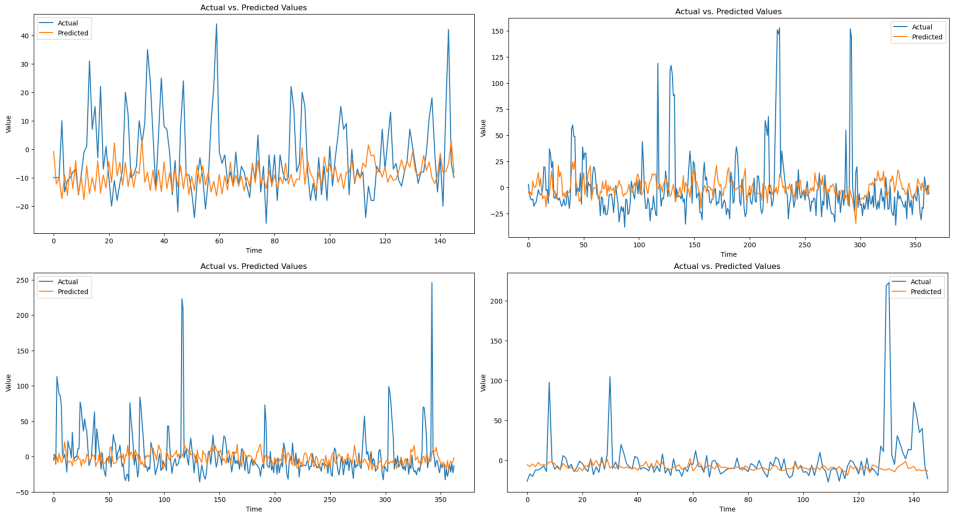

Figure 5: Forecast Results for Circuit Flights

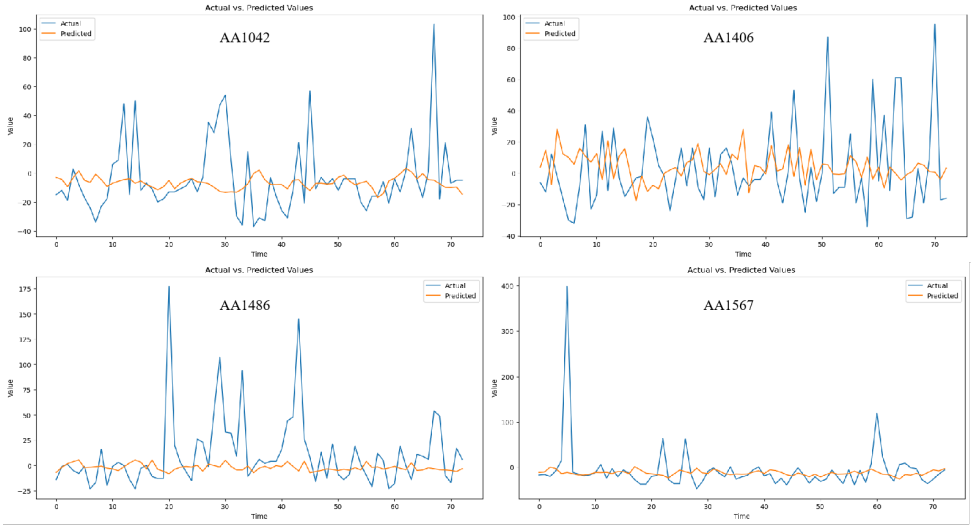

Figure 6: Forecast Results for Scheduled Flights

Table 6: ARIMA Model Forecast Performance

Circuit			Scheduled		
FlightID	RMSE	MAE	FlightID	RMSE	MAE
AS64	30.175	18.910	AA1042	25.504	16.741
AS65	33.297	19.821	AA1406	29.335	22.967
DL2452	15.304	10.404	AA1486	38.060	21.932
DL1270	33.304	13.659	AA1567	54.152	21.239

It can be observed that for both scheduled and circuit flights, the ARIMA model can capture the overall trend of the time series for forecasting but struggles to predict extreme outliers. This stems from the nature of the ARIMA model, which uses historical data to predict future values and thus finds it difficult to forecast data that significantly deviates from past trends. However, since the task at hand is to predict flight delays, the target of our prediction happens to be infrequent, irregular, and extremely large outlier values. This results in suboptimal forecasting performance of the ARIMA model for the core objective.

4.2 TFT Data Analysis and Result Evaluation

In this section, we will discuss the results and further analysis of the TFT model. In order to compare the prediction effect between different models, TFT model chooses same research objects as the previous ARIMA model, to be precise, AA1042, AA1406, AA1486, and AA1567.

Note: considering the time stamp (time_idx) of circuit flights is difficult to define, we will only focus on scheduled flights in this section.

4.2.1 Model Metrics

In the training process, the model learns the relationship between different types of input variables and the target variable (ARRIVAL_DELAY). The training set covers the first 300 days data of scheduled flights with more than 180 successful flights in 2015 (excluding flight cancellations or diversion to other airports for reasons) in terms of content. (reducing the possible disturbance caused by small samples)

The prediction performance of the trained TFT model on the four scheduled flights is shown in figure 7.

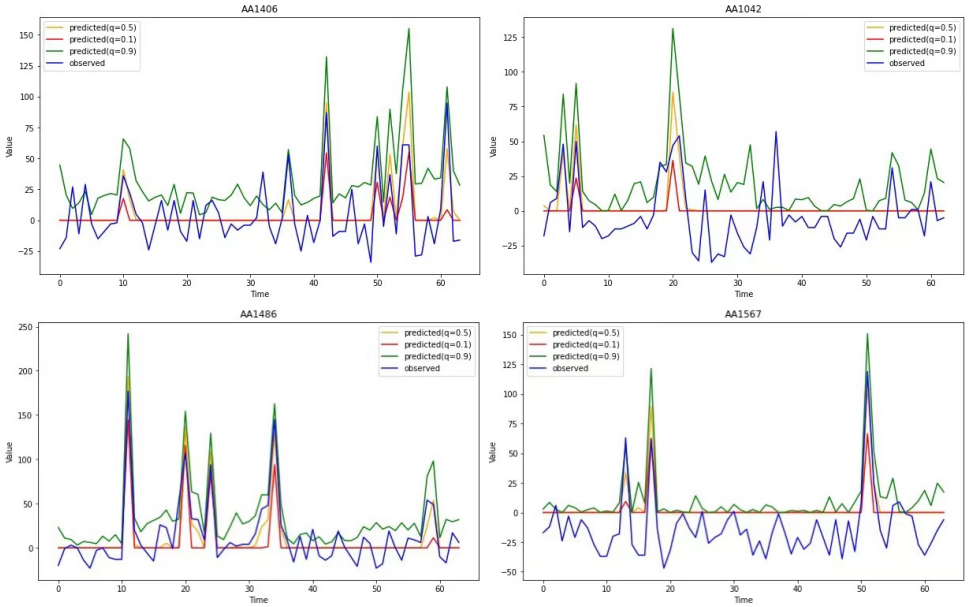

Figure 7: TFT Forecast Results

Intuitively, the predictions of the TFT model are generally in line with reality. Some long delays can also be reflected in the results. According to the model introduction, the TFT model can output quantile prediction results and we can find that the 90% quantile forecast results could cover most of the delays. Quantitatively speaking, the RMSE and MAE metrics are also improved compared with the ARIMA model above, as shown in table 7.

Table 7: Comparison of Forecast Performance between ARIMA and TFT

	RMSE		MAE	
FlightID	ARIMA	TFT	ARIMA	TFT
AA1042	25.504	**19.250**	16.741	**15.608**
AA1406	29.335	**17.241**	22.967	**13.515**
AA1486	38.060	**15.198**	21.932	**12.234**
AA1567	54.152	**22.869**	21.239	**19.280**

Note: The actual delay time of the flight can be negative, but strictly speaking, a negative delay is not actually a delay but should be counted as early arrival. Therefore, flights with negative delay time in the data set are not decomposed into each cause, but directly recorded as 0. Thus, when the model learns the relationship between variables, it will learn that there is no delay, so the model does not give a negative delay time prediction in the final result. The performance of the model might be better if we record the negative delays as 0.

4.2.2 Attention Weight Analysis

Due to the introduction of the self-attention layer, the TFT model has a good interpretability. Figure 8 shows the trained model's attention to the past delays of four flights.

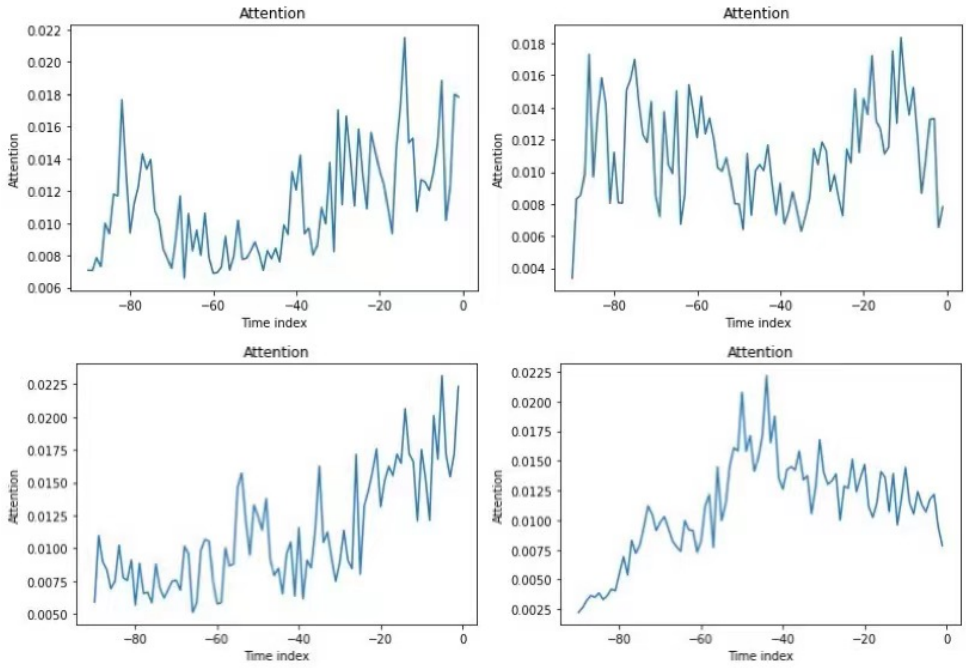

Figure 8: TFT Attention

The attention weights here are based on the trained model and reflect the overall situation for a flight (the situation at a particular time may differ from this). The top left, top right, bottom left and bottom right are AA1042, AA1406, AA1486 and AA1567, respectively. We can find that the attention to the past delays varies between different flights. For example, for AA1586, the delay 40 days before is relatively more important. To some extent, this also reflects the trend and seasonal information.

Moreover, the model can output the importance of each input variable in predicting the future delays (shown in Appendix B, order of the results is the same as figure 8). Overall, the importance of the flight id and tail number is relatively high, reflecting that the aircraft and the route itself are important factors affecting arrival delays. This is also consistent with the reality. Among the flights with positive delays, the proportion of flights whose delays caused by the aircraft and the route itself accounted for more than 90% of the total delay time reached about 55%. Some other variables also count, such as the origin and destination airport, suggesting that airport scheduling also plays a somewhat important role in flight delays.

4.2.3 Multifaceted Improvement with TFT

Compared with ARIMA model, TFT model has a comprehensive improvement in forecasting effect and model interpretability. TFT model can not only capture long and short term information in time series, but can also accept various types of variable data input. Compared with other machine learning models that can be used for time series prediction, TFT model has a higher interpretability due to the introduction of components such as the self-attention layer, which partially alleviates the "black box" common problem of machine learning models.

In our practical application, the TFT model has achieved relatively good results. Compared with the ARIMA model as a benchmark, the TFT model has made significant progress in the evaluation metrics such as RMSE and MAE. Moreover, the TFT model also performs better in predicting the occasional long delays.

References

Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting, 37(4), 1748-1764.

A ACF and PACF

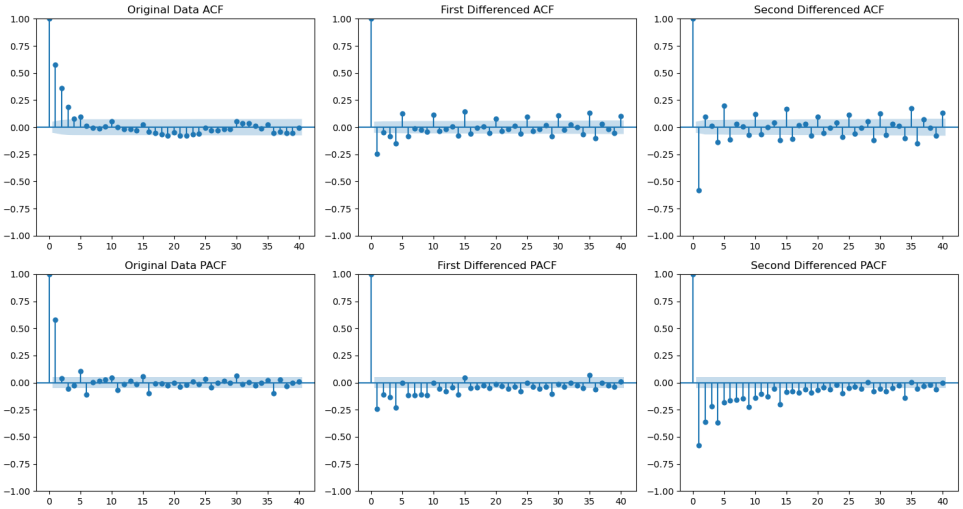

Figure 9: ACF and PACF of AS64

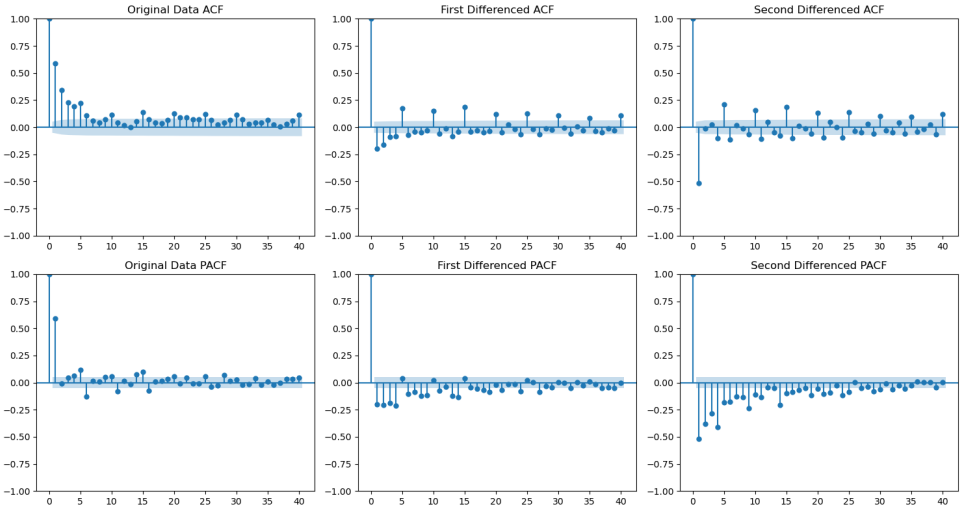

Figure 10: ACF and PACF of AS65

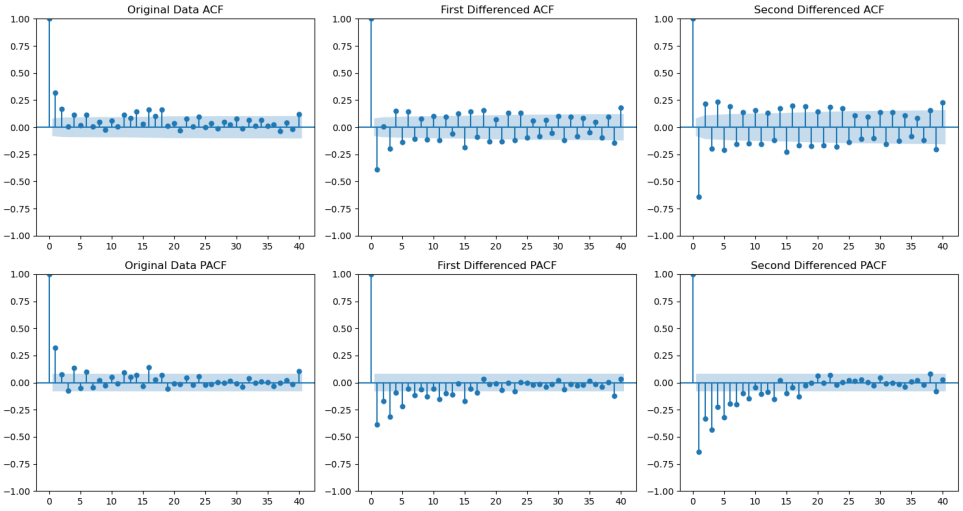

Figure 11: ACF and PACF of DL1270

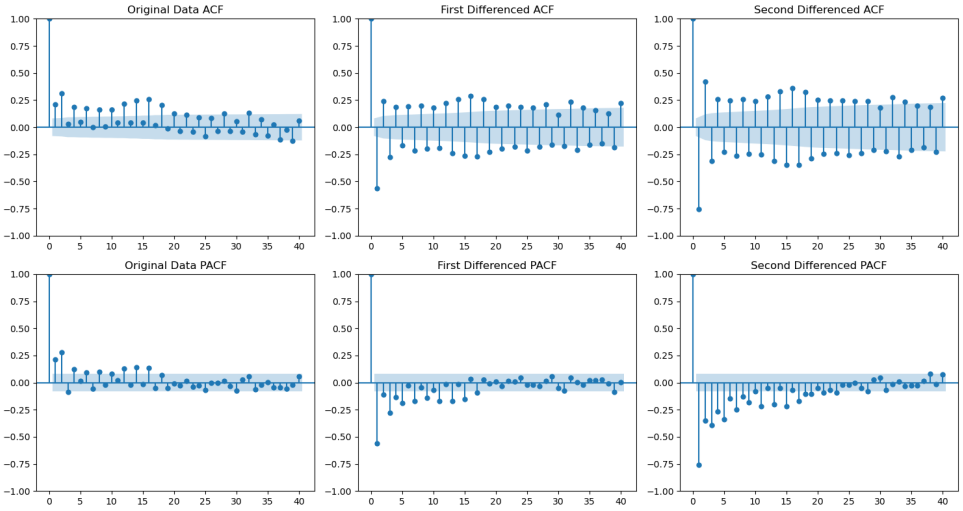

Figure 12: ACF and PACF of DL2452

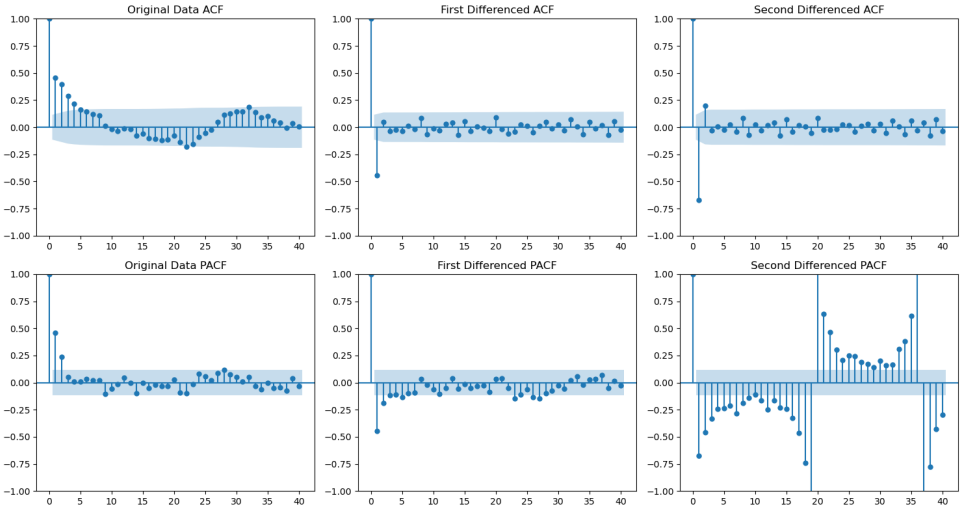

Figure 13: ACF and PACF of AA1042

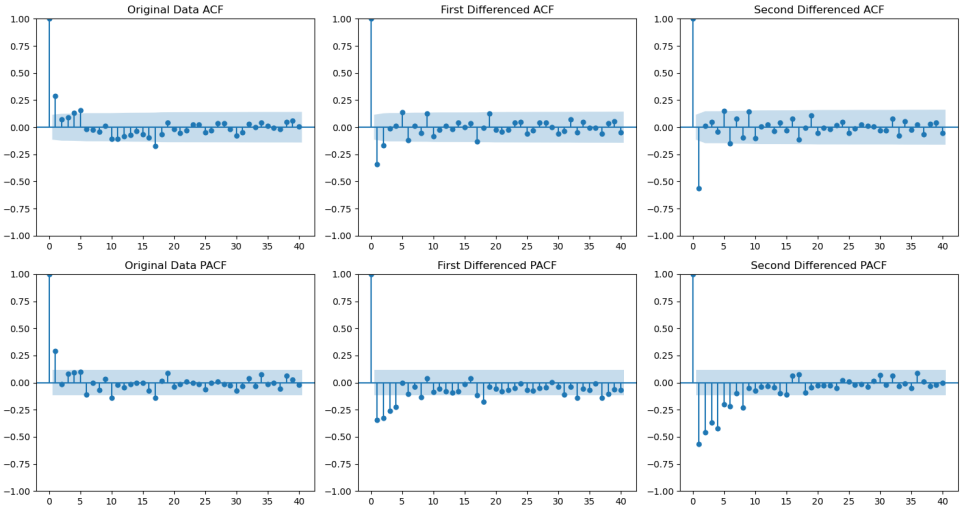

Figure 14: ACF and PACF of AA1406

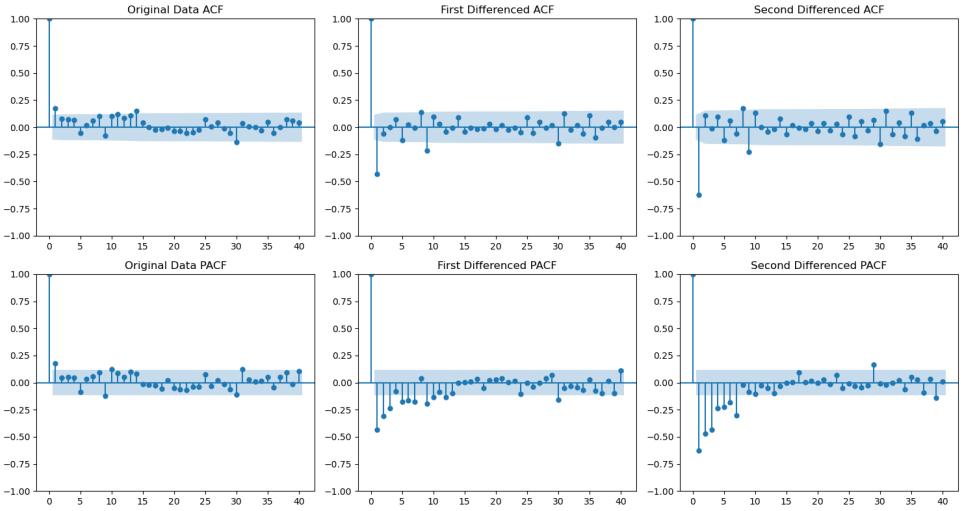

Figure 15: ACF and PACF of AA1486

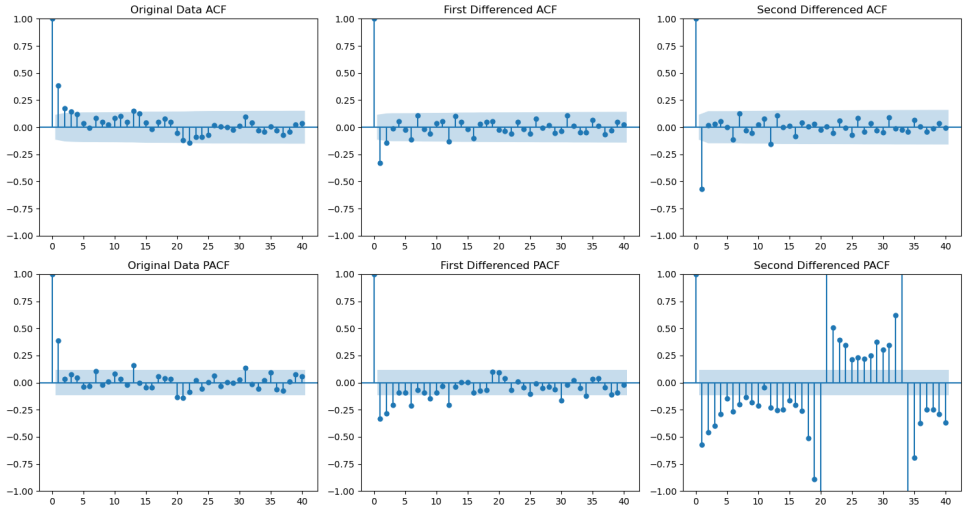

Figure 16: ACF and PACF of AA1567

B Variable Importance

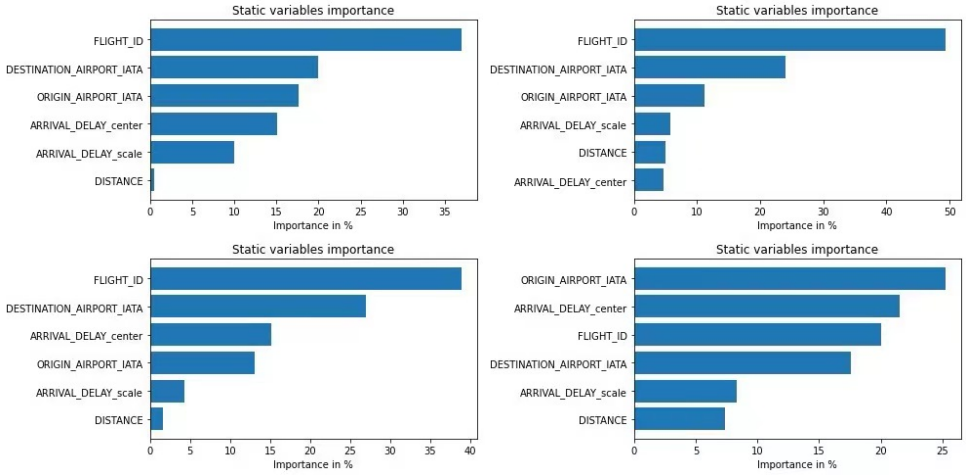

Figure 17: Static Variables Importance

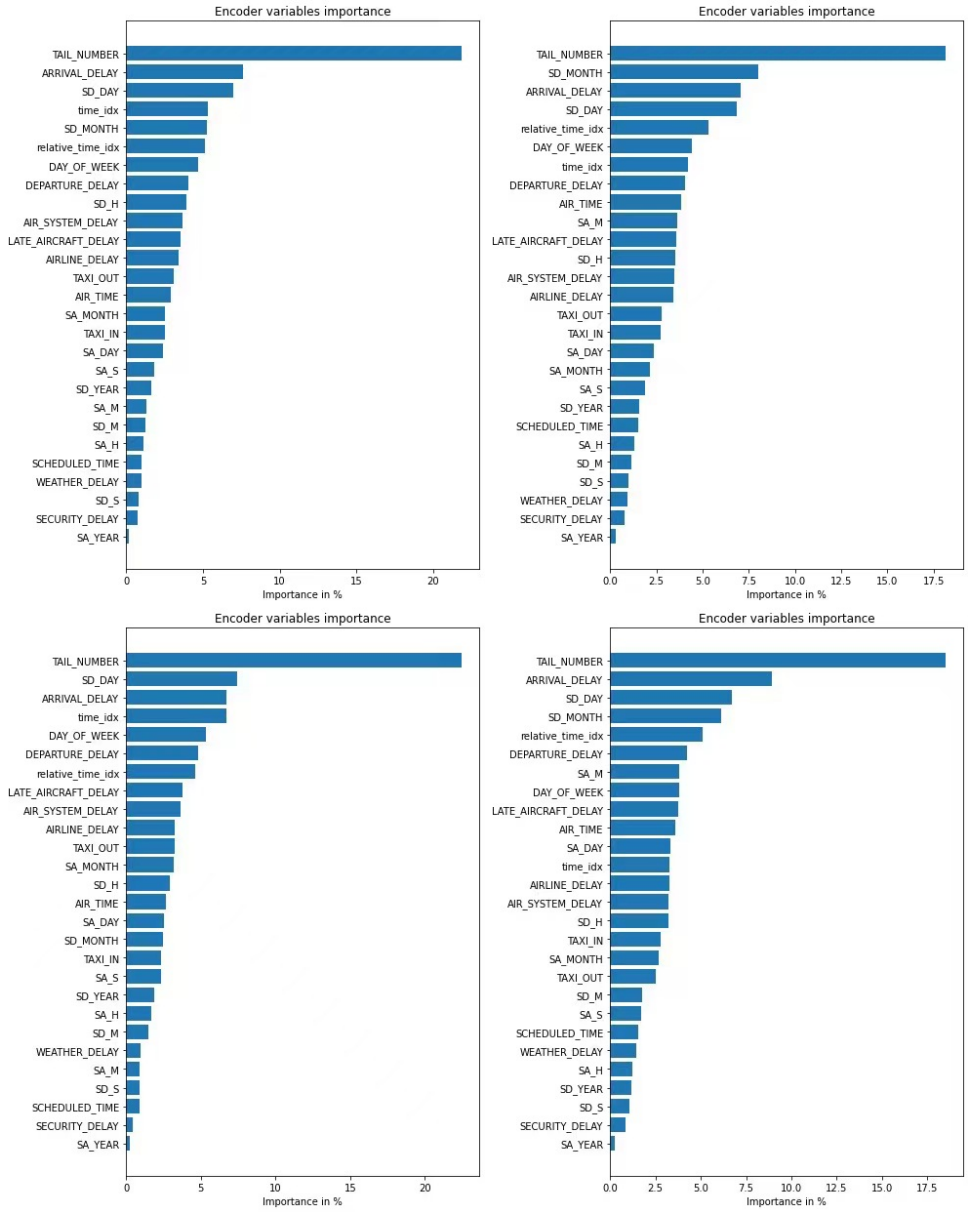

Figure 18: Encoder Variables Importance

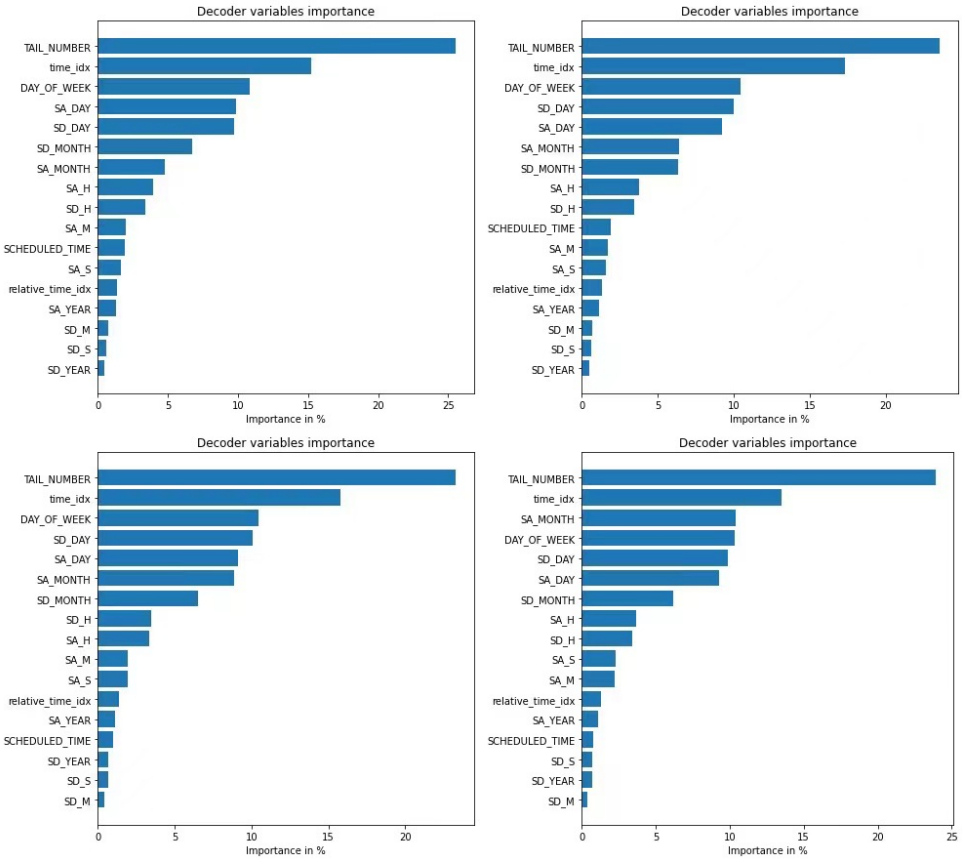

Figure 19: Decoder Variables Importance