11-6-2022

A Machine Learning Approach Towards Analyzing Impact of Surface Weather on Expect Departure Clearance Times in Aviation

Dothag Truong
Embry-Riddle Aeronautical University, truongd@erau.edu

Shlok Misra
Embry-Riddle Aeronautical University, misras@my.erau.edu

Godfrey V. D'souza
Embry-Riddle Aeronautical University, GODFREYDSOUZA7@GMAIL.COM

Collegiate Aviation Review International

Volume 40 | Issue 2 Peer Reviewed Article #6

11-6-2022

A Machine Learning Approach Towards Analyzing Impact of Surface Weather on Expect Departure Clearance Times in Aviation

Shlok Misra
Embry-Riddle Aeronautical University

Godfrey Dsouza
Embry-Riddle Aeronautical University

Dothang Truong
Embry-Riddle Aeronautical University

Commercial air travel in the United States has grown significantly in the past decade. While the reasons for air traffic delays can vary, the weather is the largest cause of flight cancellations and delays in the United States. Air Traffic Control centers utilize Traffic Management Initiatives such as Ground Stops and Expect Departure Clearance Times (EDCT) to manage traffic into and out of affected airports. Airline dispatchers and pilots monitor EDCTs to adjust flight blocks and flight schedules to reduce the impact on the airline's operating network. The use of time-series data mining can be used to assess and quantify the impact of surface weather variables on EDCTs. A major hub airport in the United States, Charlotte Douglas International Airport, was chosen for the model development and assessment, and Vector Autoregression and Recurrent Neural Network models were developed. While both models were assessed to have demonstrated acceptable performance for the assessment, the Vector Autoregression outperformed the Recurrent Neural Network model. Weather variables up to six hours before the prediction time period were used to develop the proposed lasso regularized Vector Autoregression equation. Precipitation values were assessed to be the most significant predictors for EDCT values by the Vector Autoregression and Recurrent Neural Network models.

Recommended Citation:
Misra, S., Dsouza, G. & Truong, D. (2022). A machine learning approach towards analyzing impact of surface weather on expect departure clearance times in aviation. *Collegiate Aviation Review International,* 40(2), 79-102. Retrieved from http://ojs.library.okstate.edu/osu/index.php/CARI/article/view/9356/8433

Commercial air travel in the United States (US) has grown significantly in the past decade (2010-2019) (Department of Transportation, 2020). An increase in air traffic in the National Airspace System (NAS) leads to delays and higher operating costs for airlines (Federal Aviation Administration [FAA], 2018). As per the FAA, flight delays are documented under five causes which are carrier delay, late arrival delay, NAS delay, security delay, and weather delay. Weather delays account for the largest cause of flight delays in the US and are the factor for nearly 70% of flight delays in the US (FAA, 2021). Airlines operate with constrained resources and schedule flights based on fixed block times (Sohoni et al., 2017). Delays lead to block time deviations, which can significantly affect the operating network and dispatch operations of an airline. The cost of an hour of flight delay is estimated to be about $1,400 to $4,500 per flight for an airline with the value of passenger time estimated to be in a range of $35 to $63 per hour (FAA, 2021). Airline dispatchers rely on updated air traffic information such as Expect Departure Clearance Times (EDCT) to plan and manage flights and mitigate disruptions to the overall airline network.

A large number of airlines operate in a hub-and-spoke network where the airline's operating network is characterized by single hub or multiple hub airports that are connected to several spokes or connecting airports (Parsa et al., 2019). Airlines develop schedules to ensure passengers traveling within the network can connect to different flights through different hubs (Abdelghany & Abdelghany, 2019). Airlines schedule flights to minimize connection times for passengers and ensure efficiency in the hub airports. For airlines operating in a hub-and-spoke network, EDCT for flights arriving into a hub airport can significantly affect the operations of the entire network due to passenger misconnections, lack of ground equipment, and delays to subsequent flights for the delayed aircraft. EDCTs usually affect flights at specific time banks, which are affected by factors including but not limited to weather, airport capacity constraints, or runway closures (FAA, 2009). Extended EDCTs can lead to extensive delays, ground stops, flight crew limiting on flight duty periods, and flight cancellations. As a possible mitigation tool, delay forecasting is used by airline management to predict the impact of independent factors such as weather events on whether a flight will be delayed (Etani, 2019; Goodman & Griswold, 2019). Airlines invest considerable resources in improving the efficiency of their operational network. An accurate delay forecasting model, such as the model developed in this study, can aid an airline in forecasting EDCTs and planning.

Literature Review

Traffic Management Initiatives (TMI) and Expect Departure Clearance Time (EDCT)

Traffic Management Initiatives (TMIs) are used by Air Traffic Control (ATC) to manage air traffic based on excess demand or a lowered acceptance rate at a particular airport (FAA, 2009). Terminal TMIs are airport-specific initiatives that impact arrivals into a particular airport (FAA, 2009). Some of the common TMIs are Ground Delay Programs (GDPs), Airspace Flow Programs (AFPs), EDCT, and Ground Stop (GS) (FAA, 2009). Non-compliance with a TMI can

lead to holding and diverting for aircraft as well as extensions of GDPs, AFPs, and GSs, which leads to further delays due to the overabundance of airplanes, unused slots at destination airports, and increased volume in the airspace (FAA, 2009).

EDCT, a type of TMI, is a runway release time assigned to an aircraft by ATC due to applicable TMIs, which require the aircraft to hold on the ground at their departure airport (FAA, 2009). When EDCTs are assigned to aircraft, the flight crew is given a time window within which the flight is expected to depart (FAA, 2009). EDCTs can be changed based on the conditions at the affected airports, such as changing weather conditions and airport acceptance rate (FAA, 2009). Airline dispatchers monitoring their respective flights provide updated EDCTs directly to company personnel, while pilots can also receive their modified EDCT times from the ATC at the departure airport (FAA, 2009). Like all TMIs, EDCTs are highly influenced by the weather conditions at the airport the flights are scheduled to arrive at (Swot et al., 2018).

Flight Delays Forecasting in Aviation

Flight delay forecasting can be operationalized through different statistical techniques. However, due to the advancements in machine learning algorithms, various studies have focused on forecasting flight delays utilizing machine learning techniques. Machine learning techniques have been demonstrated to be effective for flight delays prediction (Belcastro et al., 2016; Khan et al., 2021; Khanmohammadi et al., 2016; Rebollo & Balakrishnan, 2014; Yu et al., 2019). Khan et al. (2021) utilized a hierarchical integrated machine learning model to predict the flight delays and flight durations for an airline based in Hong Kong. The authors utilized a dataset provided by an airline that consisted of flight data for 19,105 flights and contained data on the runway configuration for the departure airports, weather variables such as atmospheric pressure, air temperature, altitude for flight, speed of the flight, ramp weight of the flight, and type of aircraft. The dataset was regarded as a cross-sectional dataset, and the delays were predicted as a classification problem. The authors developed a Convolutional Neural Network which was named a hyperparameter-free cascade principal component least-squares neural network (hyp-free CPCLS). The hyp-free CPCLS was capable of determining the hyperparameters, such as the number of neurons and layers, without the need for manual hyperparameter tuning. The model was designed due to the highly unbalanced, high dimensional, and highly skewed dataset that was used for the modeling. The authors determined that “categories such as passenger and baggage handling, aircraft and ramp handling, air traffic flow restriction, and government authority, and reactionary and miscellaneous are the main reason for airline departure delay" (p.21). While the study by Khan et al. (2021) contributed in literature to modeling using skewed, high dimensional, and unbalanced datasets through re-sampling and feature engineering techniques, Rebollo & Balakrishnan (2014) focused on capturing the spatial and temporal dependency of departure delays data. While departure delays research focuses highly on local spatial variables for the departure airports, the authors focused on new network delays variables that could impact the entire NAS. The authors defined spatial variables such as NAS Delay State and Type of Delays Day along with temporal variables such as Time of the Day, Month of the Year, and Day of the Week. The authors' work was considered novel due to the focus on including variables that not only impacted the airports for analysis but also impacted the NAS at large. The final model created was a Random Forest model for a 2-hour forecasting period with an average test error of 19%.

Yu et al. (2019) utilized a combination of Deep Belief Networks and Support Vector Repressors to predict flight delays on city-pair routes in China. Yu et al. (2019) explained different feature selection techniques that can be used to develop robust prediction models from high dimensional data. The authors replaced macro-level factors that are commonly seen in flight delay prediction models with specific micro-level influential factors such as aircraft capacity, boarding options, number of passengers in the flight, airline properties, and delay of previous flight for the aircraft. Yu et al. (2019) emphasized the need for feature selection techniques to reduce the dimensionality of datasets by using two conventional filter methods like the Correlation Coefficient Method and the Standard Deviation Selection Method. The final Deep Belief Network-Support Vector Repressor model was able to predict flight delays with a Mean Absolute Error of 8.41, Root Mean Squared Error of 12.65, and Coefficient of Determination of 0.93. The authors determined that air traffic control, delay of the previous flight, and air route situation were the most significant independent variables for the model.

Belcastro et al. (2016) utilized data mining to predict arrival delays due to weather conditions. The authors of the study utilized flight information such as origin airport, destination airport, scheduled departure and arrival times, and weather observations at the departure and arrival airports. The arrival delays prediction was processed as a classification task. The authors developed Decision Tree, Support Vector Machine, Random Forest, Stochastic Gradient Descent, and Naïve Bayes classifiers. The authors evaluated the scalable parallel version of the Random Forest to be the best predictor that could predict arrival delays at a threshold of 60 minutes with an accuracy of 85.6% and a recall of 86.9%. The authors also tested the model with only flight information as predictors and removed the weather conditions predictors, which reduced the model accuracy to 69.1%. In another similar study, Khanmohammadi et al. (2016) examined literature in the field of machine learning models to predict flight delays and examined the role of nominal independent variables in skewing model performance. Khanmohammadi et al. (2016) proposed an Artificial Neural Network that utilized a new type of multi-level input layer to capture the relationship of nominal independent variables. The authors designed a Neural Network model with a multi-level input layer designed for defect of module prediction. The model was deployed to predict flight delays at New York-John F. Kennedy International Airport and was compared to a Gradient Descent Backpropagation model with the same dataset. Some of the nominal independent variables used for the model included the day of the month, day of the week, origin airport, delay at departure at the origin airport, and scheduled departure time. The authors evaluated that model developed was robust to nominal independent variables and was able to predict the flight delays with a Root Mean Squared Error of 0.1366 as compared to 0.1603 for the Gradient Descent Backpropagation model.

Temporal Nature of Flight Delays

As reviewed, machine learning techniques have been successfully utilized for flight delay prediction. However, flight delay data has been modeled differently by scholars. Flight delay data can be treated as cross-sectional data, time-series data, or even spatial data. Determining the data type and format is crucial while deciding the modeling strategy for a machine learning model. While Khan et al. (2021), Belcastro et al. (2016), Khanmohammadi et al. (2016), and Yu et al. (2019) modeled the data as cross-sectional, Rebollo & Balakrishnan (2014) modeled the flight delays utilizing the temporal and spatial dependencies of the variables. Time series

forecasting utilizing the temporal dependency of variables has been demonstrated to be an effective method for delay forecasting in aviation (Guvercin et al., 2021; Lan & Shangheng, 2020). Guvercin et al. (2021) used a combination of time series clustering and time series forecasting techniques to build a prediction model to predict flight delays at 305 airports in the US. For the time series forecasting, the authors utilized "a combination of a regression and an Autoregressive Integrated Moving Average (ARIMA) model" (Guvercin et al., 2021, p. 1). As the study was not based on the data for a single airport, the authors needed to utilize a Clustered Airport Model approach to improve forecasting accuracy for the 305 airports. The authors evaluated that the ARIMA approach in combination with the Clustered Airport Model provided forecasting results comparable to forecasting results expected from a complex Long Short Term Memory (LSTM) neural network model. While Guvercin et al. (2021) utilized a clustered airport model to develop a prediction model that could be used for a large number of airports, Lan & Shangheng (2020) collected data from a single "large airport" for four years to develop a model to predict hourly departure delays (p.1). While the hourly departure delays variable contained continuous values, the authors utilized K-means clustering to cluster the delay variable into five categories or bins. For the prediction, the authors determined that Vector Autoregression (VAR) in comparison to Autoregressive Conditional Heteroskedasticity (ARCH) was an effective time series forecasting technique for delay forecasting. While Guvercin et al. (2021) and Lan & Shangheng (2020) were successful in utilizing autoregression models, Zen et al. (2021) utilized a deep graph-embedded LSTM neural network approach for airport delay prediction. A deep graph-embedded LSTM approach was preferable because the authors aimed to develop a model that was based on the data from 325 airports in the US. The authors described the use of the graph-embedded network as a "directed graph network with an airport as a node, a spatial distance weighted adjacency matrix and a demand weighted adjacency matrix are constructed, and the two are integrated to obtain a combined weighted adjacency matrix" (Zeng et al., 2021, p. 13).

Machine Learning Approach for Delay Prediction

The advancement of machine learning techniques has allowed their usage and deployment in tasks across different fields, including aviation. Carvalho et al. (2020) aimed to review the different approaches used by scholars for flight delay predictions from a data science perspective. The authors explored the use of machine learning techniques for flight delay prediction and concluded that the most popular machine learning techniques included k-Nearest Neighbors, Neural Networks, Support Vector Machine, Fuzzy Logics, and Random Forest. The choice of model depends on the prediction, purpose of the project, and data structure. For aviation delay prediction datasets, it is important to preserve the temporal dependencies of variables. Qu et al. (2020) demonstrated the use of Convolutional Neural Networks for time series flight delay prediction. For the modeling process, the authors fused meteorological data and concluded that flight delay prediction accuracy could be improved by up to 1% when using weather data in comparison to predictions by only using flight information. The authors utilized the Airline On-time Performance Database provided by the Bureau of Transportation Statistics in the US for the flight information and Local Climatological Data provided by the National Climate Data Center in the US. While Recurrent Neural Networks are mostly associated with temporal data, Convolutional Neural Networks, as standalone models or in conjunction with any other model, are common for time-series predictions due to their ability to extract the most

significant features. The authors utilized a Dual-channel Convolutional Neural Network and Squeeze and Excitation-Densely Connected Convolutional Network for the study. The Dual-channel Convolutional Neural Network and Squeeze and Excitation-Densely Connected Convolutional Network were able to achieve accuracies of 92.1% and 93.19%, respectively.

Research Questions

This study aimed to answer the following research questions:

RQ1: Can EDCT values be predicted for a large hub airport in the US using surface weather observations?

RQ2: What variables are the most significant predictors of EDCT values?

Significance of the Study

The literature reviewed highlighted the viability and success of machine learning models in predicting different types of flight delays. The effect of surface weather on delays, including EDCTs, has been studied by scholars in the past (Belcastro et al., 2016; Qu et al., 2020). EDCT, just like other TMIs, is severely affected by weather and can disrupt traffic flow for an airport. The studies by Guvercin et al. (2021) and Lan & Shangheng (2020) demonstrated the effectiveness of time series autoregressive models in predicting flight delays. However, there is a significant gap in research in predicting EDCTs utilizing any type of statistical modeling, even though there is domain importance and need for such a prediction model. This study is an attempt to bridge the research gap by utilizing surface weather variables to develop time series models to predict EDCT values for a major hub airport. Time series models will allow the model to retain the temporal dependency of the endogenous variables, which has been demonstrated to be an important concept in published literature.

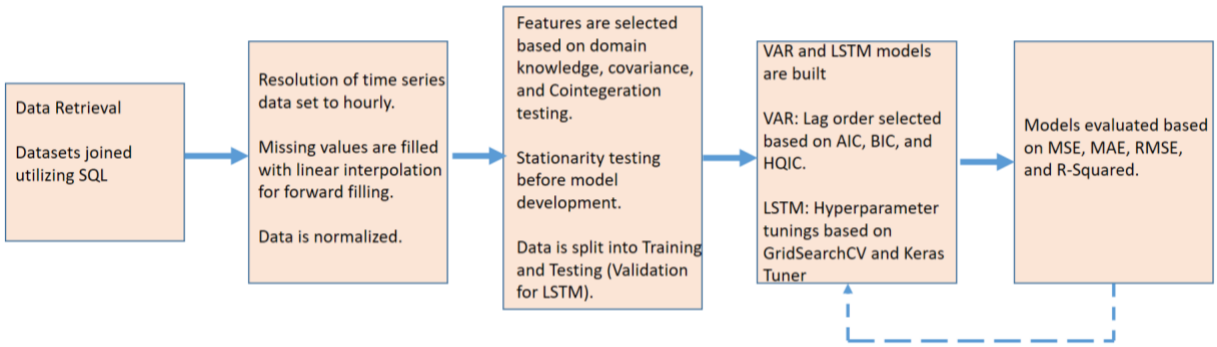
An EDCT prediction model will allow airline management to make better informed short-term operations decisions such as contingency fuel and resource and gate allocations. EDCT prediction will also help airline management with customer service as longer EDCT predictions can be treated as a direct indication of higher arrival delays at a hub airport which can lead to passengers' missed connections. Additionally, unlike delay parameters such as arrival delays, block delays, and departure delays, EDCTs are issued and enforced by ATC with little to no control by airline management. Based on domain expertise, the scope of EDCT prediction for enhanced airline management and planning is immense, and this study is aimed at adding literature to the subject.

Methodology

The purpose of the study was to develop a time series model to predict EDCTs based on surface weather observations for a large hub airport in the US. Based on the reviewed literature, the researchers adopted Vector Autoregression and Recurrent Neural Network, specifically Long Short Term Memory, modeling approaches for the study. The researchers aimed to develop a VAR model and an LSTM model and compare model performance to predict the EDCTs. For the

modeling, the researchers used Charlotte Douglas International Airport (Charlotte). Charlotte is the largest hub for American Airlines in the US, with 397,983 departures and arrivals in 2020 (Charlotte Airport Media, 2021). While the model was built based on the data for Charlotte Douglas International Airport, the researchers expect the results of the study to be transferable for prediction and analysis at other large hub airports as well. Figure 1 depicts the overall model development pipeline adopted for the study.

Figure 1
Overall Model Development Pipeline for the Study



Data Collection and Preprocessing

The researchers acquired historical hourly surface weather observations and hourly traffic data, including EDCT data for Charlotte Douglas International Airport. Two databases were provided by the National Oceanic and Atmospheric Administration (NOAA) and FAA for the weather and traffic information, respectively (FAA, n.d.; NOAA, n.d.). The hourly weather and traffic data for Charlotte Douglas International Airport from 2014-2019 was used in this study. The data for 2020 was included due to the effects of the COVID-19 pandemic on air travel. Once the data was downloaded in comma separate values (CSV) formatted files, the researchers formed a dataset from the different data files using a Structured Query Language (SQL) application with the date/time column as the foreign key. Since the data was structured with data points corresponding to every hour, it could be treated as a time series dataset for the data preprocessing and data analysis stages. The dataset required significant preprocessing due to missing values for some data points. The researchers utilized the Pandas library for the Python programming language for the preprocessing tasks and a forward-filling method to handle missing values. Once the data was preprocessed, it was used to build the VAR and LSTM models.

Vector Autoregression Architecture

Vector Autoregression is a statistical technique used to capture the dependencies of multiple time series variables and the temporal dependencies over time. VARs have been extensively developed and deployed for multivariate time-series predictions. VARs can be used to develop multiple simultaneous equations with the time-lagged values of all the variables, called endogenous variables, used to model and analyze the relationship between the different variables. The VAR model for the study was built utilizing the Statsmodels library in the Python 3.0 Programming Language. Figure 2 depicts the model development strategy developed by the

VAR model. The VAR model can be represented by Equation 1.

$$Yt = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} + \varepsilon_t \quad (1)$$

$$\alpha = constant$$

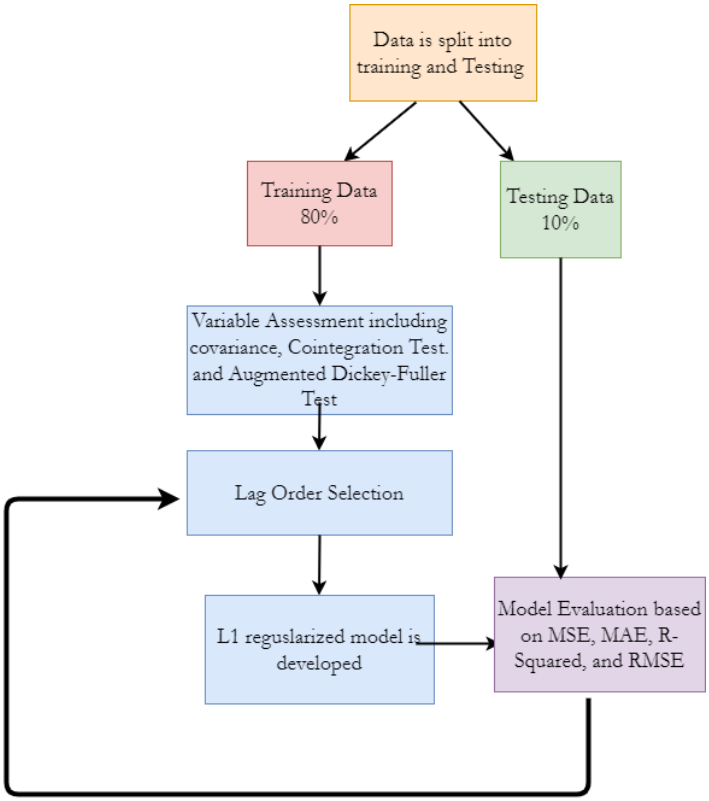
$$\beta_1, \beta_2 \ \ldots \beta_p = Coefficients\ of\ the\ lags\ from\ t\ to\ t-p$$

$$Y_t, Y_{t-1} \ldots Y_{t-p} = Endogenous\ Variables$$

$$\varepsilon_t = Error\ Term$$

Figure 2

Model Development Strategy for the VAR model



Stationarity Testing for VAR

Autoregressive models perform most effectively when the time series variables exhibit stationarity (Abdulnasser, 2004). The researchers utilized the Augmented Dickey-Fuller Test to test the stationarity of the time series at a significance level of 0.05 (Kulaksizoglu, 2005). The Augmented-Dickey Fuller Test tests the null hypothesis that a unit root is not present in the time series analyzed. Based on the test statistic of the test, which is a negative number, the null hypothesis can be rejected and determined that the unit root is present.

Table 1 depicts the results of the Augmented Dickey-Fuller Test. Based on the results of the test, all the time series variables were determined to be stationary and could be used for the model development without any further adjustments. Figure 3 is a heatmap of the covariance matrix of the variables utilized for the VAR model. The heatmap depicts the covariance between

each pair of variables for a given random vector. The covariance matrix can be used to analyze the interrelation of all the individual random variables in the matrix and used in conjunction with the Augmented Dickey-Fuller Test to evaluate any data processing and variable selection needs.

Table 1
Augmented Dickey-Fuller Test

Variable	Test Statistic	Critical Value (0.05)	Number of Lags Chosen	Stationarity
Hourly Arrivals	-20.2629	-2.862	54	Stationary
Hourly Gate Delays	-21.0193	-2.862	55	Stationary
Altimeter	-18.4218	-2.862	55	Stationary
Temperature	-7.7599	-2.862	54	Stationary
Precipitation	-30.8831	-2.862	30	Stationary
Hourly Relative Humidity	-19.8074	-2.862	51	Stationary
Hourly Visibility	-22.411	-2.862	52	Stationary
Average EDCT	-24.7523	-2.862	50	Stationary

Order Selection

Order selection is a crucial aspect of developing a time series model. While tools such as autocorrelation function (ACF) or partial correlation functions (PACF) can be used to determine the appropriate order, the researchers used a combination of the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and Hannan-Quinn Information Criterion (HQIC) to determine the appropriate lag order. Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and Hannan-Quinn Information Criterion (HQIC) are estimators of prediction errors for a statistical model. AIC (Equation 2), BIC (Equation 3), and HQIC (Equation 4) can be used as indicators of the qualities of a model in comparison to other models and can be used for model selection.

$$AIC = 2k - 2\ln(\hat{L}) \quad (2)$$
$$k = number\ of\ estimated\ parameters\ in\ the\ model$$
$$\hat{L} = Maximum\ value\ of\ the\ likelihood\ of\ the\ function$$

$$BIC = k\ln(n) - 2\ln(\hat{L}) \quad (3)$$
$$\hat{L} = Maximum\ value\ of\ the\ likelihood\ of\ the\ function$$
$$n = number\ of\ data\ points$$
$$k = number\ of\ estimated\ parameters\ in\ the\ model$$

$$HQIC = -2L_{max} + 2k\ln(\ln(n)) \quad (4)$$
$$L_{max} = Log - Likelihood$$
$$n = number\ of\ data\ points$$
$$k = number\ of\ estimated\ parameters\ in\ the\ model$$

The researchers used a loop algorithm in the Python Programming Language to determine the AIC, BIC, and HQIC for VAR models with lag orders ranging from 1 to 50. Based

on the AIC, BIC, and HQIC evaluation, a lag order of 13 was determined to be the optimal lag order. Once the lag order was determined, the researchers developed the VAR model based on the parameters selected.

Data Preparation

Once the initial statistical testing was completed, the researchers split the data for the training and testing of the model. The total dataset consisted of 52,582 instances or rows, with each row representing an hourly interval. Sci-Kit Learn library on Python was used to split the data with 80% of the data used for training and 20% of the data used for testing. Finally, the researchers set the Shuffle to False to ensure that the temporal order was maintained during the splitting operation. The training data had 47,323 instances, and the testing data had 5,259 instances.

Regression Equation and L1 Regularization

The VAR model developed to predict EDCT would consist of 104 independent variables (for EDCT prediction) due to eight time series variables and a lag order of 13. Such a complex model would increase model cost, complexity, sensitivity to noise or outliers, and the possibility of overfitting (Tan et al., 2019). The researchers utilized the L1 regularization (Lasso) technique to regularize the model and reduce the number of independent variables for the model. Such a regression model is expected to exhibit high performance with lower cost, complexity, and low possibility of overfitting. L1 regularization computation can be illustrated by Equation 5.

$$Cost = \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\,\beta_j)^2 + \lambda \sum_{j=1}^{p} I\beta_j I \quad (5)$$

where $\lambda \sum_{j=1}^{p} I\beta_j I$ is regarded as the penalty term, which is the absolute value of the magnitude of the coefficients.

Recurrent Neural Network

Recurrent Neural Networks are a type of neural network commonly used to model sequential or time series data. Applications of Recurrent Neural Networks include Natural Language Processing, Time Series prediction, Signal Processing, speech recognition, and language translation (Geron, 2019). Long Short Term Memory models are a type of Recurrent Neural Network with the presence of 'gates' that are useful for combatting issues such as vanishing and exploding gradients and short-term memory commonly seen in normal Recurrent Neural Networks. With the presence of a Forget Gate, Input Gate, and Output Gate in every LSTM neuron in the network, the model is able to retain long-term memory and dependencies for sequential or temporal data (Geron, 2019). The LSTM model for the study was built using the Tensorflow library in the Python 3.0 Programming Language.

Data Preparation

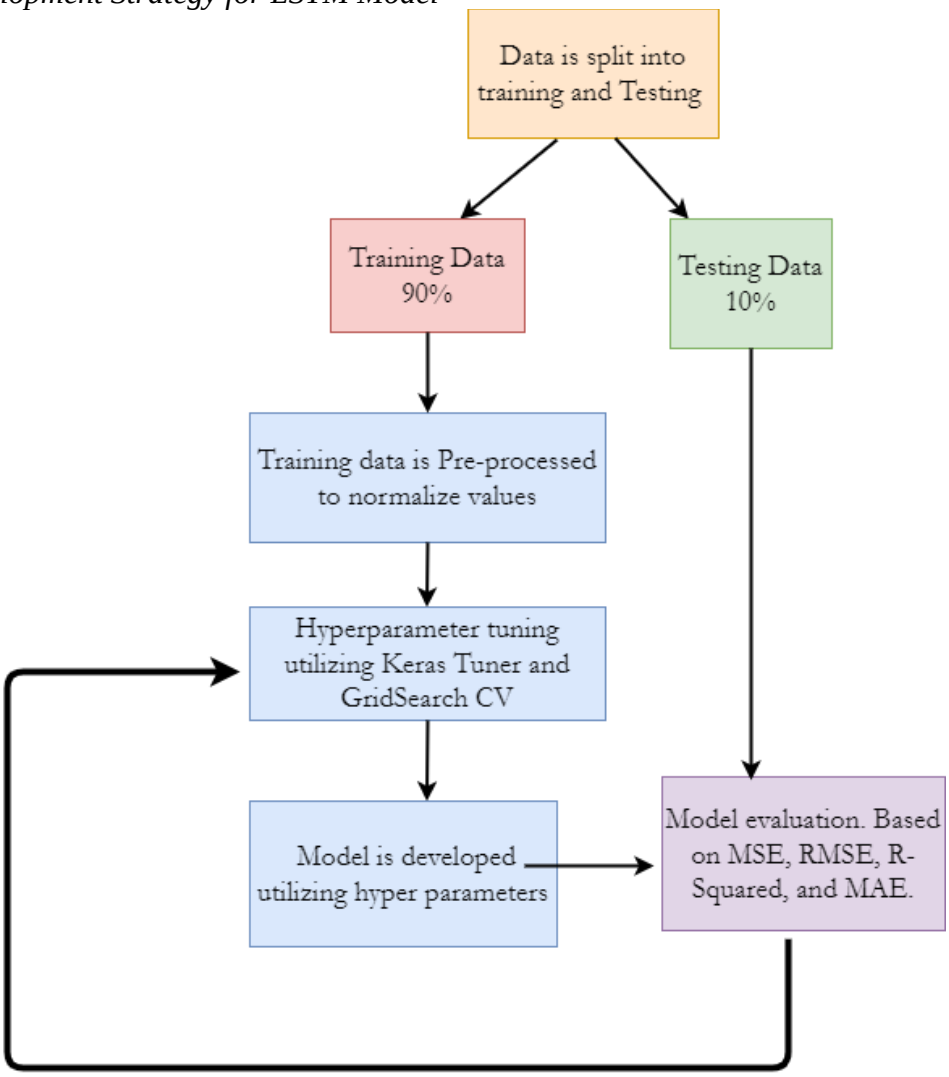
To build the LSTM model using the TensorFlow library, the researchers needed to preprocess the data into a special array format utilizing the TimeSeriesGenerator library on Python. The researchers utilized the Sci-Kit Learn library to conduct the Train-Test Split

operation and set the Shuffle to False to maintain the temporal order of the dataset. For the LSTM model, a validation set was used for the hyperparameter tuning. The data was split with 80% of the data used for training, 10% of the data used for validation, and 10% of the data used for testing. With a total of 52,582 instances, the training dataset had 42,066 instances, the validation dataset had 5258 instances, and the testing dataset had 5,259. The testing dataset for the LSTM and VAR models was the same.

The researchers intended to create a sliding LSTM model and train the model in batches. The window length for the LSTM was set to four, batch size to 32, and sliding to 1. This could be seen as each batch consisting of 32 data points, with each data point containing 4 hours of data with a sliding operation of 1 step. Figure 3 depicts the model development strategy used for the LSTM model.

Figure 3

Model Development Strategy for LSTM Model



LSTM Architecture and Hyperparameter Tuning

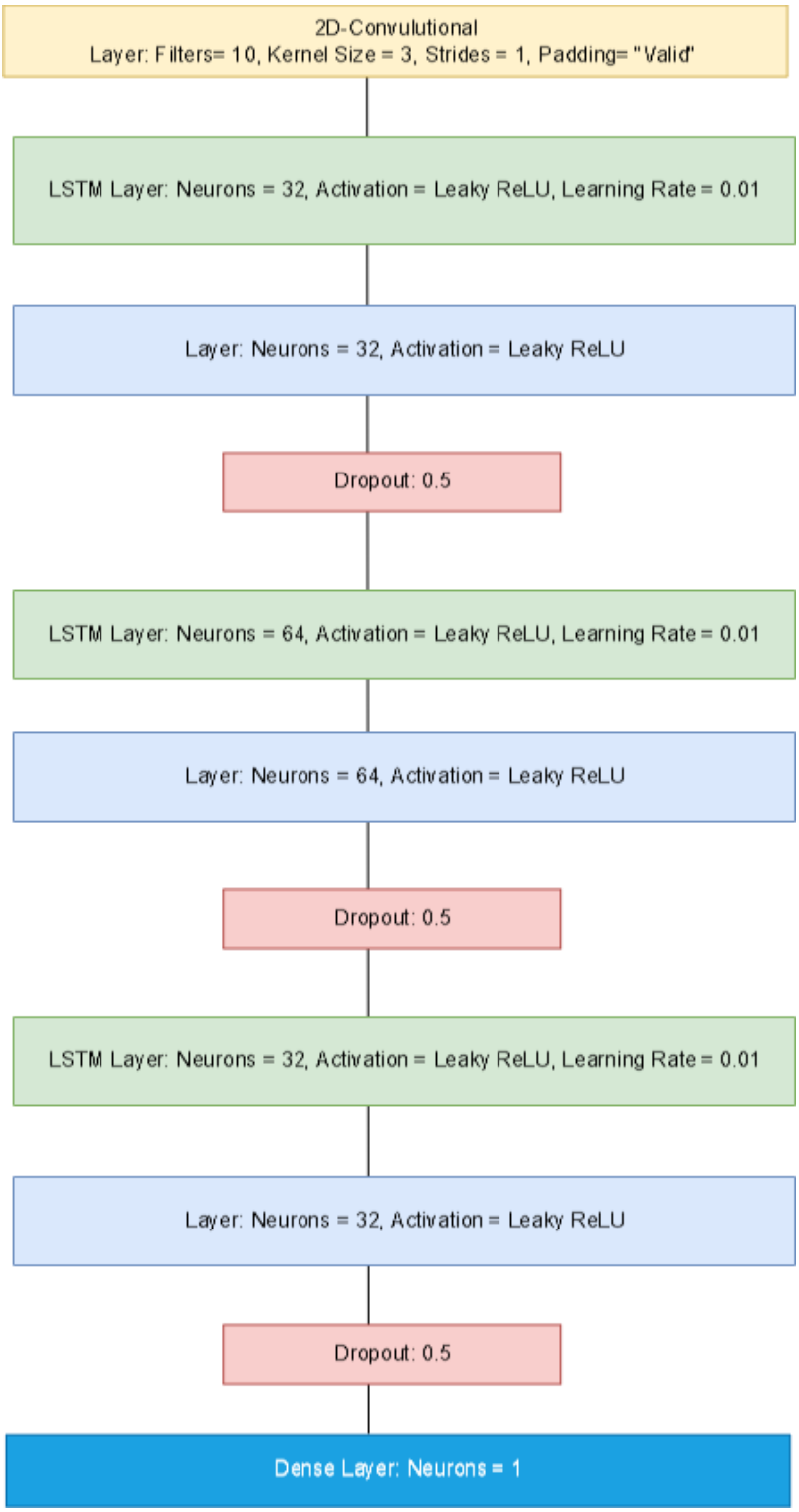
The LSTM model architecture was severely inspired by previous literature on similar prediction tasks. Once the initial model architecture was tuned, the researchers tuned the hyperparameters of the model using the Keras Tuner. A 2D-Convolutional layer was used as the first layer, followed by three LSTM layers with the Leaky Rectified Linear Unit (ReLU) as the activation function. Each LSTM layer was followed by a 50% dropout layer as a regularizer. Additionally, the optimizer was set to Adaptive Momentum (Adam), and the loss function was the mean squared error. Early stopping of the training was added as an additional regularizer. Figure 4 is the model summary output from the Tensorflow library that describes the layer type, activation function, output shape, and parameters for each layer. Figure 5 is an illustration of the LSTM model developed for the study.

Figure 4
LSTM Model Parameters

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 1, 10)	220
lstm (LSTM)	(None, 1, 32)	5504
leaky_re_lu (LeakyReLU)	(None, 1, 32)	0
dropout (Dropout)	(None, 1, 32)	0
lstm_1 (LSTM)	(None, 1, 64)	24832
leaky_re_lu_1 (LeakyReLU)	(None, 1, 64)	0
dropout_1 (Dropout)	(None, 1, 64)	0
lstm_2 (LSTM)	(None, 32)	12416
dropout_2 (Dropout)	(None, 32)	0
dense (Dense)	(None, 1)	33

Total params: 43,005
Trainable params: 43,005
Non-trainable params: 0

Figure 5
LSTM Model Parameters



Model Interpretability

The LSTM model was a deep neural network model intended for the regression prediction of the EDCT values. However, for any machine learning model, model interpretability is an important aspect rather than treating the developed model as a *black box*. Shapley Additive Explanations (SHAP) can be used to assess the features utilized to develop a machine learning model, especially a neural network model (Molnar, 2021). Derived from a game theory approach to explain the output of models, SHAP computes Shapley Values utilizing coalitional game theory by treating each feature as a *player* in the game. The SHAP computation can be illustrated by Equation 6.

$$g(\acute{z}) = \emptyset_0 \acute{+} \sum_{j=1}^{M} \emptyset_j \acute{z}_j \quad (6)$$

Where g is the explanation model, $\acute{z} \in \{0,1\}^M$ is the coalition vector, M is the maximum coalition size, and $\emptyset_j$ is the feature attribution of a feature j. A significant advantage of utilizing SHAP to interpret a model is the robustness of SHAP to attribute dependency. As feature importance and permutation importance methods are poor in capturing attribute dependency among the attributes or features used for the model development, they might over-emphasize or under-emphasize some features depending on how those features correlate with other features, which is commonly referred to as the high-correlation variable problem (Hooker et al., 2019). Utilizing Shapely Value Imputation, SHAP is robust to the multicollinearity among the features (Lipovetsky & Conklin, 2001; Lundberg & Lee, 2017). The mean magnitude of SHAP values will be derived utilizing the SHAP library in Python. While the TimeSeriesGenerator library was used to develop the LSTM models on TensorFlow, the training set had to be formatted to a 3D-Array format utilizing the NumPy library due to the limitations of the SHAP library.

Results

Vector Autoregression Model

Based on the data preprocessing, variable selection, stationarity testing, and lag order selection procedures, a VAR model with an order of 13 was developed. Table 2 summarizes the results of the VAR model.

Table 2
Vector Autoregression Model Results

Parameter	Value
Number of Equations	8
Akaike Information Criterion	7.22924
Bayesian Information Criterion	7.57313
Hannan-Quinn Information Criterion	7.33783
Final Prediction Error	1379.17
Log-Likelihood	-627489

The original VAR model built using the Statsmodel library on Python does not involve any sort of regularization. The original VAR model was modified with L1 regularization to remove non-significant endogenous variables for predicting the EDCTs. Finally, a regression equation was developed to predict the EDCT. Table 3 depicts the coefficients, standard error, T-statistic, and probability value associated with each of the endogenous variables used for the regression equation.

Table 3
EDCT Regression Equation Analysis

Variable	Coefficient	Standard Error	T-Statistic	p-value
Lag: 4 Precipitation	25.27544	1.446	17.480	<0.001
Lag: 3 Precipitation	19.0414	1.443	13.194	<0.001
Lag 2: Precipitation	15.2144	1.439	10.566	<0.001
Lag 5: Precipitation	4.1695	1.451	2.873	0.004
Lag 1: Precipitation	2.8705	1.3760	2.086	0.007
Lag 1: EDCT	0.3204	0.0045	70.872	<0.001
Lag 3: Hourly Visibility	0.216	0.0532	4.059	<0.001
Lag 2: EDCT	0.112	0.0047	23.789	<0.001
Lag 1: Hourly Visibility	0.1029	0.0456	2.254	0.004
Lag 6: Temperature	0.0856	0.0265	3.253	0.001
Lag 1: Relative Humidity	0.039	0.0089	4.392	<0.001
Lag 2: Gate Delay	0.006	0.0018	3.295	0.001
Lag 3: EDCT	0.014	0.0046	3.032	0.002
Lag 4: EDCT	0.015	0.0046	3.145	0.002
Lag 5: Precipitation	4.24	1.414	3.003	0.003
Lag 6: Temperature	0.014	0.025	3.147	0.004
Lag 3: Temperature	-0.076	0.0255	-2.679	0.007
Lag 1: Precipitation	3.31	1.342	2.466	0.008
Lag 1: Hourly visibility	0.102	0.044	2.315	0.009
Lag 3: Gate Delays	3.004	1.498	-2.066	0.04
Constant	14.979	10.5503	1.420	0.156

Figure 6
Correlation Matrix of Residuals from the VAR Model

	Hourly Arrivals	Hourly Gate Delays	Altimeter	Temperature	Precipitation	Hourly Relative Humidity	Hourly Visibility	EDCT
Hourly Arrivals	1	0.061448	-0.002337	-0.000043	-0.00274	-0.000108	0.002058	-0.010119
Hourly Gate Delays	0.061448	1	0.01008	0.00863	-0.001987	-0.004312	0.004738	0.11692
Altimeter	-0.002337	0.01008	1	-0.273899	-0.015984	-0.24822	0.074359	-0.006302
Temperature	-0.000043	0.00863	-0.273899	1	-0.129173	-0.456269	0.139224	0.001294
Precipitation	-0.00274	-0.001987	-0.015984	-0.129173	1	0.149912	-0.287004	0.03455
Hourly Relative Humidity	-0.000108	-0.004312	-0.24822	-0.456269	0.149912	1	-0.268077	0.017797
Hourly Visibility	0.002058	0.004738	0.074359	0.139224	-0.287004	-0.268077	1	0.005882
EDCT	-0.010119	0.11692	-0.006302	0.001294	0.03455	0.017797	0.005882	1

Once the VAR model was developed and significant endogenous variables were determined, there was a need to inspect the serial correlation of the residuals to ensure there was a minimal correlation in the residuals and that any patterns in the time series were not left unexplained by the VAR model. Figure 7 depicts the correlation matrix for the model residuals. We can see that there is no endogenous variable that exhibits a high correlation of residuals with EDCTs. We can see a negative correlation between Arrivals and Altimeter with EDCT. The strongest correlation of residuals is exhibited by Hourly Relative Humidity and Temperature. Additionally, the researchers utilized the Durbin Watson Test to check for the serial correlation of the residuals and ensure that the model had sufficiently explained the patterns and variances in the time series dataset used. The value of the Durbin Watson Test can vary between 0 and 4, where a value close to 2.00 implies there is no significant serial correlation (Durbin & Watson, 1971). The Durbin Watson Test is utilized to detect autocorrelation at lag 1 for the prediction errors of an autoregressive model. Table 4 depicts the results of the Durbin Watson Test. The Durbin Watson test results in Table 4 and correlation matrix results in Figure 6 ensured that there was no serial correlation of the residuals and that the model had adequately explained the variance in the data. The Durbin-Watson Statistic used in the Durbin-Watson Test can be represented by Equation 7.

$$d = \frac{\sum_{t=2}^{T}(e_t - e_{t-1})^2}{\sum_{t=1}^{T} e_t^2} \tag{7}$$

$$T = Number\ of\ total\ observations$$

$$e_t, e_{t-1} = Residual\ of\ the\ autoregression$$

Table 4
Durbin Watson Test

Attribute	Durbin Watson Statistic
Hourly Arrivals	1.99
Hourly Gate Delays	1.98
Altimeter	1.99
Temperature	2.01
Precipitation	2.0
Hourly Relative Humidity	1.99
Hourly Visibility	1.98
Average EDCT	2.0

Model Evaluation

The VAR model was evaluated on the testing set on evaluation parameters such as mean squared error, mean absolute error, and root mean squared error. Table 5 illustrates the model results.

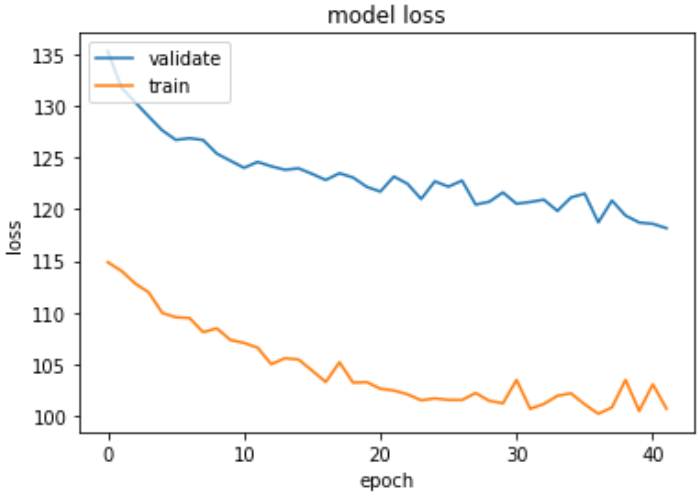
Table 5
Model Evaluation of the VAR model

Evaluation Parameter	VAR model
Mean Squared Error	91.126
Root Mean Squared Error	9.55
Mean Absolute Error	1.99
R-Squared	0.6812

Long Short-Term Memory

An LSTM model was developed on the training test and a validation test. Figure 7 depicts the training and validation model loss with the different epochs. As early stopping was used as a regularizer, the training ceased after epoch 42.

Figure 7
Training and Validation Loss for the LSTM Model with Epochs



The model was evaluated on the evaluation parameters for the training, validation, and testing sets. Table 6 illustrates the model results.

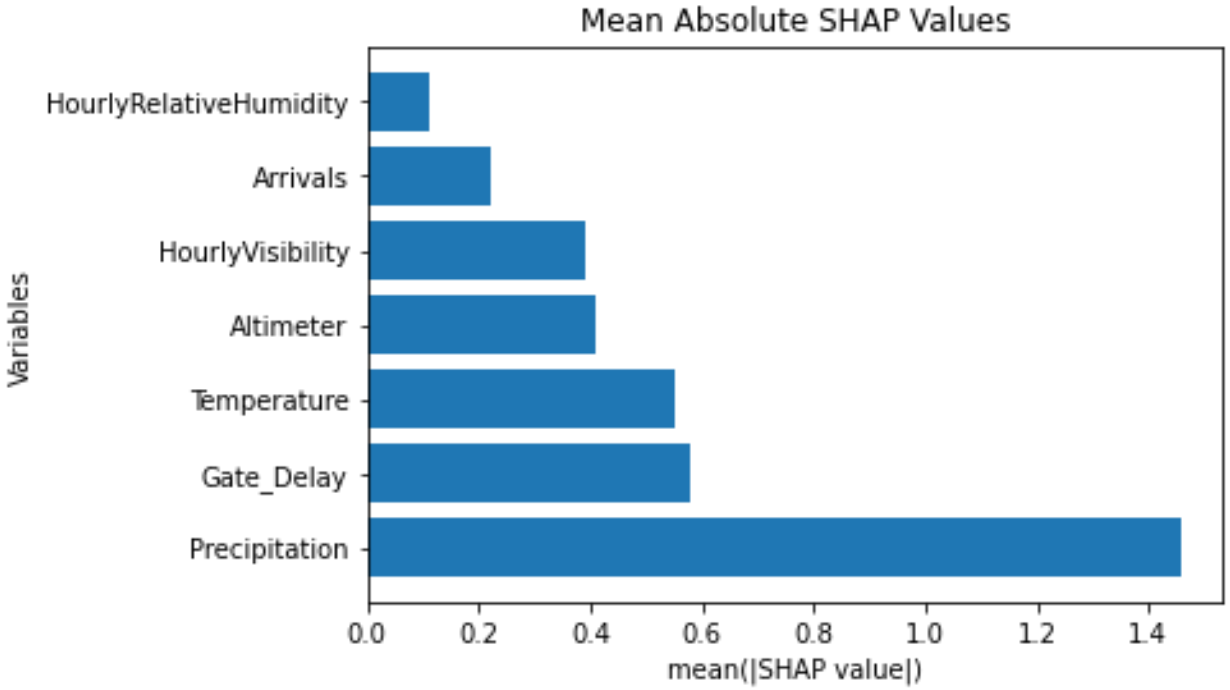
Table 6
LSTM Model Results

Evaluation Parameter	Training	Validation	Testing
Mean Squared Error	102.01	121.11	168.14
Root Mean Squared Error	10.01	11.004	12.96
Mean Absolute Error	2.0073	2.3443	2.85
R-squared	0.681	0.677	0.643

Model Interpretation

SHAP was used to assess the most significant features utilized by the LSTM model for the prediction. The mean absolute SHAP values for the seven features were used to assess their impact on the predictions of the model. Figure 8 depicts the mean absolute SHAP values of the features. Precipitation had the highest mean absolute SHAP value and the highest impact on the target variable followed by Gate Delays.

Figure 8
Mean Absolute SHAP Values



Discussion and Conclusion

The purpose of the study was to develop VAR and LSTM models to predict EDCT for a large hub airport based on surface weather observations. The study was developed based on the significant research gap identified to utilize machine learning techniques to predict EDCTs for an airport, given the importance of EDCTs for dispatch operations of an airline. While there are several demonstrated machine learning algorithms demonstrated by scholars, VAR and LSTM were selected based on previous literature on other domain-related studies. The VAR and LSTM model predictions were primarily evaluated on Mean Absolute Error, Mean Squared Error, Root Mean Squared Error and R-Squared values. While the VAR outperformed the LSTM model on all three evaluation parameters, the performance of both models is considered acceptable. While the LSTM model had lower performance, the researchers believe the results of the LSTM established the viability of utilizing RNNs such as LSTM or Gated Recurrent Units for EDCT predictions. While LSTMs are commonly regarded as more robust time series modeling algorithms due to the non-linear activation and optimization functions involved as compared to VARs, VARs have been demonstrated to outperform LSTMs in previous studies on related subjects (Goel et al., 2016).

While the prediction power of both models was deemed acceptable, it is important to critically analyze the model coefficients or feature importance to understand the most significant predictors. The VAR model can be analyzed based on the endogenous variables coefficients, and the LSTM model can be analyzed based on the SHAP values. The VAR model was regularized with L1 regularization to reduce the number of endogenous variables in the final regression

equation and reduce the scope for overfitting the model. It is distinctly clear that precipitation values until Lag 6 had the strongest influence on the EDCT predictions. Additionally, the final regression equation only consisted of endogenous variables up to Lag 6. While the VAR model was built utilizing Lag 13, the L1 regularization reduced the number of endogenous variables because of their low coefficients and, in turn, insignificant impact on the EDCT prediction. The feature assessment for the LSTM was conducted using SHAP. The SHAP analysis is consistent with the coefficients of the VAR model as precipitation was distinctly the highest influencer of EDCT prediction. The model assessments do match intuition as heavy precipitation can be directly associated with convective activity, such as thunderstorms that are a significant cause for delays and in turn issuance of EDCTs.

The model proposed in the study is expected to be a dynamic model in which the input variables are updated hourly for EDCT predictions. Such a model is expected to aid airline dispatchers and airline managers with short-term forecasts and predictions to improve planning and resource allocations. Estimations of EDCT a few hours before the flight can be useful in contingency planning, customer service, and resource allocations at the hub airport. An important utilization for EDCT prediction would be to make necessary adjustments to the airline's contingency fuel policy. In the event of long EDCTs, aircraft return to the gate to obtain more fuel should they go below their minimum fuel required on the dispatch release. Using these predictions, airlines can develop dynamic contingency fuel requirements based on the EDCT estimations. Lastly, the results from the model can help airlines develop and optimize a flight schedule to reduce the number of heavy arrival banks into hubs. Spacing out arrivals among different time banks will reduce the EDCTs and airspace flow constraints.

References

Abdelghany, A., & Abdelghany, K. (2019). *Airline network planning and scheduling*. Wiley Publishing.

Abdulnasser, H. (2004). Multivariate tests for autocorrelation in the stable and unstable VAR models. *Economic Modelling, 21*(4), 661-683. https://doi.org/10.1016/j.econmod.2003.09.005

Belcastro, L., Marozzo, F., Talia, D., & Trunfio, P. (2016). Using scalable data mining for predicting flight delays. *ACM Transactions on Intelligent Systems and Technology, 8*(1), 1-20. https://doi.org/10.1145/2888402

Carvalho, L., Sternberg, A., Gonçalves, L., Cruz, A., Soares, J., & Brandao, D. (2020). On the relevance of data science for flight delay research: a systematic review. *Transport Reviews, 41*(4). https://doi.org/10.1080/01441647.2020.1861123

Charlotte Airport Media. (2021). *CLT welcomes 27.2 million passengers in 2020*. https://cltairport.mediaroom.com/2020-Passenger-Numbers#:~:text=Overall%2C%20aircraft%20operations%20logged%20397%2C983,added%20last%20year%20from%20Charlotte

Cirium. (2015, November 9). *What is "Block Time" in airline schedules? Why does it matter?* https://www.cirium.com/thoughtcloud/block-time-airline-schedules/#:~:text=Block%20time%20includes%20the%20time,t%20break%20these%20elements%20apart

Durbin, J, & Watson, G. (1971). Testing for serial correlation in least squares regression. III. *Biometrika, 58*(1), 1-19. https://doi.org/10.2307/2334313

Etani, N. (2019). Development of a predictive model for on-time arrival flight of airliner by discovering the correlation between flight and weather data. *Journal of Big Data, 6*. https://doi.org/10.1186/s40537-019-0251-y

Federal Aviation Administration. (2009). *Traffic Flow Management in the National Airspace System*. https://www.fly.faa.gov/Products/Training/Traffic_Management_for_Pilots/TFM_in_the_NAS_Booklet_ca10.pdf

Federal Aviation Administration. (2015). *FAQ: Weather delay*. https://www.faa.gov/nextgen/programs/weather/faq/#:~:text=By%20far%2C%20the%20largest%20cause,%22Delay%20by%20Cause%22%20Reports.

Federal Aviation Administration. (2018). NextGen implementation plan 2018-19. *Office of NextGen*. https://www.faa.gov/nextgen/media/ NextGen_Implementation_Plan-2018-19.pdf

Federal Aviation Administration. (2009). *Traffic flow management in the National Airspace System.* https://www.fly.faa.gov/Products/Training/Traffic_Management_for_Pilots/TFM_in_the_NAS_Booklet_ca10.pdf

Federal Aviation Administration. (2021). *Inclement weather*. https://www.faa.gov/newsroom/inclement-weather-0?newsId=23074

Federal Aviation Administration. (n.d.). *FAA operations & performance data.* https://aspm.faa.gov/

Geron, A. (2019). Hands-On machine learning with Sci-kit Learn, Keras, and Tensorflow: Concepts, Tools, and Techniques to build intelligent systems. O'Reilly Publishing. ISBN: 978-1492032649

Goel, H., Melnyk, I., Oza, N., Matthews, B., & Banerjee, A. (2016). *Multivariate aviation time series modeling: VARs vs. LSTMs*. https://goelhardik.github.io/images/Multivariate_Aviation_Time_Series_Modeling_VARs_vs_LSTMs.pdf

Goodman, C., & Griswold, J. (2019). Meteorological impacts on commercial aviation delays and cancellations in the continental United States. *Journal of Applied Meteorology and Climatology*, *58*(3), 479–494.

Guvercin, M., Ferhatosmanoglu, N., & Gedik, B. (2021). Forecasting flight delays using clustered models based on airport networks. *IEEE Transactions on Intelligent Transportation Systems*. 22(5). https://doi.org/10.1109/TITS.2020.2990960

Hooker, G., Mentch, L., & Zhou, S. (2019). *Unrestricted permutation forces extrapolation: Variable importance requires at least one more model, or there is no free variable importance*. https://arxiv.org/pdf/1905.03151.pdf

Johansen, S. (1991). Estimation and hypothesis testing of cointegration vectors in Gaussian vector autoregressive models. *Econometrica, 59*(6), 1551-1580. https://doi.org/10.2307/2938278

Khan, W., Ma, H., Chung, S., & Wen, X. (2021). Hierarchical integrated machine learning model for predicting flight departure delays and duration in series. *Transportation Research Part C: Emerging Technologies, 129*. https://doi.org/10.1016/j.trc.2021.103225

Khanmohammadi, S., Tutun, S. & Kucuk, Y. (2016). A New Multilevel Input Layer Artificial Neural Network for Predicting Flight Delays at JFK Airport. *Procedia Computer Science, 95*, 237-244. https://doi.org/10.1016/j.procs.2016.09.321

Kirchgässner, G., & Wolters, J. (2008). *Introduction to modern time series analysis*. Springer. https://doi.org/10.1007/978-3-540-73291-4

Kulaksizoglu, T. (2015). Lag order and critical values of the augmented dickey-fuller test: A replication. *Journal of Applied Econometrics, 30*(6), 1010-1010. https://doi.org/10.1002/jae.2458

Lan, M, & Shangheng, O. (2020). Characteristic analysis of flight delayed time series. *Journal of Intelligent Systems, 30*(1), 361-375. https://doi.org/10.1515/jisys-2020-0045

Lipovetsky, S., & Conklin, M. (2001). Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry, 17*, 319-330. https://doi.org/10.1002/asmb.446

Lundberg, S., & Le,,, S. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. https://papers.nips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

Molnar, C. (2021). *Interpretable machine learning: A guide for making black box models explainable*. https://christophm.github.io/interpretable-ml-book/

National Oceanic and Atmospheric Administration. (n.d.). *Data tools: Local climatological data (LCD)*. https://www.ncdc.noaa.gov/cdo-web/datatools/lcd

Parsa, M., Nookabadi, A., Flapper, S., & Atan, Z. (2019). Green hub-and-spoke network design for aviation industry. *Journal of Cleaner Production, 229*, 1377-1396. https://doi.org/10.1016/j.jclepro.2019.04.188

Qu, J., Zhao, T., Ye, M., Li, J., & Liu, C. (2020). Flight delay prediction using deep Convolutional Neural Network based on fusion of meteorological data. *Neural Processing Letters, 52*, 1461-1484. https://doi.org/10.1007/s11063-020-10318-4

Rebollo, J., & Balakrishnan, H. (2014). Characterization and prediction of air traffic delays. *Transportation Research Part C: Emerging Technologies, 44*, 231-241. https://doi.org/10.1016/j.trc.2014.04.007

Sohoni, M., Lee, Y., & Klabjan, D. (2011). Robust airline scheduling under block time uncertainty. *Transportation Science, 45*(4), 451– 464. http://doi.org/10.1287/trsc.1100.0361

Swot, C., Stalnaker, S., & Coats, P. (2018). Simulation-based analysis of early scheduling in the time-based flow management (TBFM) system for flights with expect departure clearance times (EDCT). *AIAA Aviation Forum 2018*. https://doi.org/10.2514/6.2018-3355

Tan, P., Steinbach, S., & Kumar, V. (2019). *Introduction to data mining*. Addison-Wesley Publishing.

U.S Department of Transportation. (2020). Airlines and airports: Traffic. *Bureau of Transportation Statistics.* https://www.transtats.bts.gov /Data_Elements.aspx?Data=2

Yu, B., Guo, Z., Asian, S., Wang, H., & Chen, G. (2019). Flight delay prediction for commercial air transport: A deep learning approach. *Transportation Research Part E: Logistics and Transportation Review, 125*, 203-221. https://doi.org/10.1016/j.tre.2019.03.013

Zeng, W., Li, J., Quan, Z., & Lu, X. (2021). A deep-graph-embedded LSTM Neural Network approach for airport delay prediction. *Journal of Advanced Transportation, 2021*. https://doi.org/10.1155/2021/6638130