Journal of Aviation/Aerospace Education & Research

Volume 32 | Number 2 Article 4

2023

A Deep BiLSTM Machine Learning Method for Flight Delay Prediction Classification

Desmond B. Bisandu PhD
Cranfield University, desmond.bisandu@cranfield.ac.uk

Irene Moulitsas PhD
Cranfield University, i.moulitsas@cranfield.ac.uk

Scholarly Commons Citation

Bisandu, D. B., & Moulitsas, I. (2023). A Deep BiLSTM Machine Learning Method for Flight Delay Prediction Classification. *Journal of Aviation/Aerospace Education & Research, 32*(2). DOI: https://doi.org/10.58940/2329-258X.1992

Abstract

This paper proposes a classification approach for flight delays using Bidirectional Long Short-Term Memory (BiLSTM) and Long Short-Term Memory (LSTM) models. Flight delays are a major issue in the airline industry, causing inconvenience to passengers and financial losses to airlines. The BiLSTM and LSTM models, powerful deep learning techniques, have shown promising results in a classification task. In this study, we collected a dataset from the United States (US) Bureau of Transportation Statistics (BTS) of flight on-time performance information and used it to train and test the BiLSTM and LSTM models. We set three criteria for selecting highly important features to train and test the models. The performance evaluation of the models and Confusion matrix shows that BiLSTM outperforms the LSTM model. In evaluating the models using the Mathews Correlation Coefficient (MCC), the BiLSTM model offers a better correlation of 0.99 between the original and predicted classes. Our experiment shows that for predicting flight delays, the BiLSTM model takes advantage of the forward and backward hidden sequences and the deep neural network for performance exploration and exploitation to achieve high accuracy, recall, and F1-Score. Our findings suggest that the BiLSTM model can effectively predict flight delays and provide valuable information for airlines, passengers, and airport managers.

*Keywords***:** analysis, BiLSTM, deep learning, flight delay, machine learning

Introduction

Recently, flight delays have increased due to the rapid growth of air transportation system demand and the influence of globalization in the 21st century (Cheevachaipimol et al., 2021; Wu et al., 2022). The International Civil Aviation Organisation (ICAO) report in 2019 shows constant growth in the aviation industry. There were 155 million commercial passengers between

2018 and 2019, resulting from an expansion year-on-year with a 6.07% increase (Bisandu et al., 2022; Cheevachaipimol et al., 2021). This growth in demand for air travel outpaces airport capacity expansions. Also, the Civil Aviation Administration (CAA) of China reported in 2020 that there was an 88.52% increase of the on-time rate of flights from civil aviation (CAA, 2021) and an 80.13% significant increase in 2018 (Fu et al., 2020). However, civil aviation still faces a major problem of flight delays (Carvalho et al., 2021; Liu et al., 2008; Maxson, 2018). The negative effects of flight delays on all the stakeholders cannot be overemphasized (Ball et al.,2010), leading to losses of huge amounts of money (Carvalho et al., 2021; Efthymiou et al., 2019), conflict between airlines and passengers (Gu et al.,2020), and reduced operational efficiency in the entire civil aviation sector (Cai et al., 2017; Wu, 2008). Flight delay characteristics research and prediction model establishment have improved the services of the decision support department of the aviation control and airlines (Etani, 2019; Gui et al., 2020), enhanced operational efficiency of civil aviation (Ding, 2017; Qu et al., 2020), and reduced loss (Dou, 2020).

Air transportation delay cost in 2007 was estimated to total $33 billion in the United States (U.S.), and $16.7 billion of the delays were associated with passengers (Baumgarten et al., 2014; Dou, 2020). Hence, this indicates how important the issue of flight delay is in the air transportation system, and management needs to pay close attention to understand and evaluate its occurrences effectively. Other, non-aviation businesses are also affected by the impact of flight delays, among other financial and operational aspects of civil aviation. The overall goal is to minimize unnecessary cost and enhance the performance of the flight delay prediction model to help avoid and mitigate risk for the stakeholders in the commercial aviation industry. There are internal and external factors that have been identified to cause flight delays. The controllable factors by the airline are referred to as the *internal factors*, such as the availability of the gate,

while the *external factors* depend on the uncontrollable factors, such as baggage handling, passenger handling, and bad weather. The complexity of understanding datasets related to flight delay prediction make it one of the aviation industry's most challenging problems. Also, internal, and external factors significantly increase the problem's complexity.

One of the most important performance indicators to effectively assess the service quality of airline and airport management is the punctuality of scheduled flights. In theory, the time required to design practices may not be possible because of uncontrollable factors, such as sudden pilot sickness and bad weather. Reports from Federal Aviation Administrator (FAA) and standards for operation show that any flight that landed 15 minutes after the originally scheduled time is considered *delayed.* There are many regulations by many airports requiring airlines to compensate passengers who experience delays exceeding a certain threshold. It is common to have congestion, which makes the delays common, too. Delay can lead to detrimental effects due to its consecutive propagation. Hence, it will be more beneficial to all involved stakeholders in the air transportation system to predict the occurrence as it will help them be prepared and make all the necessary responses to avoid further consequences.

The advances in data analytics and artificial intelligence have motivated research on the possible application of flight delay in predicting the use of commercial airlines. Several approaches are applied in the analysis and prediction of flight delays. These are machine learning, statistical analysis and deep learning (Bisandu et al., 2022). Therefore, the understanding and critical evaluation of the different techniques align with the application domain. Neural networks have been proven to perform better in state-of-the-art deep learning methods when applied to complex datasets with higher accuracy. However, the real-time video and image datasets were applied to feed the models with a complex structure considered unstructured or semi-structured datasets. But it is highly significant that the viability of applying

a bidirectional long short-term memory (LSTM) type of recurrent neural network (RNN) model with structure and offline dataset such as for flight delays to study the performance comparison similarity with others. Also, based on our findings, there are limited contributions in the research area of flight delay predictive modelling in which the bidirectional LSTM model is used to predict flight delays.

In this paper, we utilize a bidirectional deep LSTM architecture to perform flight delay analysis and prediction using flight on-time datasets obtained from the U.S. Department of Transportation's Bureau of Statistics (BTS). The primary objective is to investigate bidirectional deep LSTM architecture with the deep neural network and unidirectional LSTM architecture in the flight delay predictive task evaluated with well-known benchmark metrics.

The rest of the paper presents other past studies of flight delay analysis and predictive tasks, explains the material and methods of the proposed approach, discusses the results in detail, and concludes with future directions for research.

Related Work

Researchers and industry practitioners have faced challenges in accurately predicting flight delays for decades. However, the applicability of recent studies using state-of-the-art approaches such as machine learning, big data, and deep learning has demonstrated better prediction results on flight delays than statistical approaches (Kim et al., 2016; Lin et al., 2019). As a result, several research studies have been conducted to assess the factors and impact of non-meteorological and meteorological conditions on flight delays.

Bisandu et al. (2022), utilize a special type of deep recurrent neural network (RNN) known as deep long short-term memory (LSTM) and social ski driver conditional autoregressive-based deep learning to study non-weather impacted delays using datasets from the U.S. Bureau of Statistics. Their experiment shows that data pre-processing and improving the model learning

randomization of the deep learning algorithm with optimization algorithms improve the model's accuracy, error rate, and reduced computational requirements compared to other metaheuristic methods. In Carvalho et al. (2021), and Mueller and Chatterji (2002), the authors discuss the relevance of data in predicting flight delays and identifying major methods, such as machine learning, deep learning and statistical methods, as the currently applied methods in the research of flight delay predictive tasks. Kim et al. (2016) proposed different architectural designs and implementations of LSTM and RNN in predicting flight delays using sequences of thresholds. Lin et al. (2019) used convolutional LSTM (Conv-LSTM) to predict airport flight delays using temporal and spatial characteristics in China civil aviation.

Mueller and Chatterji (2002), explore the characteristics of an aircraft's arrival and departure delay to develop a time series model that determines the probability of the correlation of the features using poison and normal probability distribution with density function. Vandehzad and Holmgren (2020) combine a mathematical weight value with linear and non-linear kernels of regression algorithms to propose a model for predicting flight delays using data from a Swedish airline software and services provider, Aviolinx. Also, Gui et al. (2020), collected data using surveillance-broadcast aviation platforms and used random forests based on LSTM to explore factors influencing flight delays. Belcastro et al. (2016), incorporated parallel algorithms using MapReduce with weather datasets to predict arrival flight delays. Finally, Chen and Li (2019)proposed a multi-label random forest propagation and classification model using an optimal feature selection process that predicts chains of delays considering the initial departure.

Ye et al. (2020) proposed a method for predicting the flight departure delay in Nanjing Lukou International Airport by applying four different supervised machine learning algorithms known as support vector machine, Light Gradient Boosting Machine (LightGBM), multiple linear regression and extremely randomized trees using a comparative study to explain their

suitability base on their experiments. Bisandu et al. (2021) performed a comparative study between deep feedforward architecture with shallow architectures using a filter-based feature selection technique on non-weather impacted flight delays; the experiment results show how important parameter tuning is in making a model more reliable in predicting departure and arrival flight delays. Manna et al. (2018) analyzed air traffic data using a gradient boost decision tree, and the model produces higher accuracy based on their experiment. Chakrabarty (2019) used a grid search hyper-parameter tuning and gradient boosting classifier model for analyzing and predicting the arrival delay of American Airlines using the top five busiest airports with a binary classification technique. Kuhn and Jamadagni (2017) employed a single-layer neural network with logistic regression and a decision tree to detect whether an arrival flight will be delayed using only the top three features from the feature importance results as the model inputs. Takeichi et al. (2017) predicted arrival delays at Tokyo Airport using queue analysis artificial neural network (ANN) with Rectifier Linear Unit (ReLU).

Gopalakrishnan and Balakrishnan (2017) proposed a flight delay prediction model based on origin-destination pairs using a two-hour time horizon with an architecture of multiple artificial neural networks and the Markov Jump Linear System. Yu et al. (2019) proposed a model for flight delay with novel factors known as crowdedness in the Beijing International Airport route using a deep belief network and support vector regression (DBN_SVR) on the top layer of the proposed model. Khanmohammadi et al. (2016) offered a new method to predict incoming flights at John F. Kennedy (JFK) International airport using ANN and Defect of Modules Prediction (DPM). Lv et al. (2015), applied stacked autoencoders (SAE) for traffic flow prediction using a greedy deep learning architecture, which learns best from the features. Zhang et al. (2019) proposed an airport delay prediction model using BiLSTM sequence learning with spatiotemporal analysis. Their model shows better stability and accuracy than the other methods

in their experiment. Karim et al. (2017) developed a model for time series classification using fully connected convolutional networks and LSTM with a novel fine-tuning technique for the LSTM cell. Despite the remarkable results from the previous studies, there is a common conclusion among them, and it is the fact that there is a need to perform more analysis and propose more models based on new context and perspectives in checking the suitability and viability of the prediction models considering the change in every aspect of the transportation industry, most especially the air transport sector.

With the advent of neural networks and deep learning with LSTM architectures, there is a paradigm shift in the performance of predictive models, especially flight delay tasks. Previous revised studies have used complex data such as images and videos with models (Mueller & Chatterji, 2002; Manna et al., 2018). However, no study has compared bidirectional deep LSTM with unidirectional LSTM with structure data from flight on-time records. Therefore, this paper investigates the efficacy and viability of bidirectional deep LSTM on flight on-time records to perform flight delay predictive tasks. This study intends to contribute to the air transportation system by proposing an innovative, accurate flight delay prediction model using bidirectional deep LSTM architecture and the U.S. BTS dataset.

Materials and Methods

This study employed an airline performance on-time dataset downloaded from the U.S. Department of Transportation. Specifically, we used the October to December 2013 dataset for the experiment. There were 29 features initially in the dataset before pre-processing.

Data Description and Pre-Processing

The information in the dataset is from October to December 2013 flight on-time U.S. BTS. Table 1 shows the 29 features with their corresponding definitions (Yazdi et al., 2020) and the attribute types for each feature (Cios et al., 2007). They needed to be reduced further because

of constraints in computation, inconsistencies and empty cells or non-required features. Researchers divided the training and testing datasets into 70% and 30%, respectively, grouping the delay time into two for the presence of delay and no delay blocks instead of applying the exact delay time. The delay block and flight that arrived or departed 15 minutes later is considered *delayed*, known as Class 1, while the non-delay *on-time* block is any flight that arrived and departed in less than 15 minutes, called Class 0. From the 29 features, we selected 6 features for training the model, which are the most relevant features to the flight delay predictive task.

Table 1

Dataset Features and their Descriptions

S/No	Feature	Attribute Type	Details
1	Year	Numerical (Discrete)	Example, 2000
2	Month	Numerical (Discrete)	Example, 12
3	DayOfMonth	Numerical (Discrete)	Example, 01-31
4	DayOfWeek	Numerical (Discrete)	Example, 1 (Monday) - 7 (Sunday)
5	DepartureTime	Numerical (Continuous)	Example, 1456
6	ScheduledDepartureTime	Numerical (Continuous)	Example, 1456
7	ArrivalTime	Numerical (Continuous)	Example, 1456
8	ScheduledArrivalTime	Numerical (Continuous)	Example, 1456
9	UniqueCarrierCode	Categorical (Nomimal)	Example, PS
10	FlightNumber	Numerical (Discrete)	Example, 1454
11	PlaneTailNumber	Categorical (Nominal)	Example, N923XJ
12	ActualElapsedTime	Numerical (Continuous)	Example, 193
13	ScheduledElapsedTime	Numerical (Continuous)	Example, 200
14	FlightTime	Numerical (Continuous)	Example, 94
15	ArrivalTime	Numerical (Continuous)	Example, 1015
16	DepartureTime	Numerical (Continuous)	Example,1015
17	Origin	Categorical (Nominal)	Example, LHR
18	Destination	Categorical (Nominal)	Example, MAN
19	Distance	Numerical (Continuous)	Example, 1448
20	TaxiIn	Numerical (Continuous)	Example, 10
21	TaxiOut	Numerical (Continuous)	Example,10
22	Cancelled	Binary (Categorical)	Example, 0 or 1
23	ArrDelay	Numerical (Nominal)	Example, -11, 12, 0
24	DepDelay	Numerical (Nominal)	Example, -10, 13, 0
25	CarrierDelay	Binary (Categorical)	Example, 0 or 1
26	WeatherDelay	Binary (Categorical)	Example, 0 or 1
27	AviationSystemDelay	Binary (Categorical)	Example, 0 or 1
28	SecurityDelay	Binary (Categorical)	Example, 0 or 1
29	LateAircraftDelay	Binary (Categorical)	Example, 0 or 1

Features that do not add much to our model performance were removed because some had information after the delay had been confirmed, had complete zero in all the cells, or had the same value that did not impact the model's training performance. Summarily, our conditions and criteria for the feature selection process were to focus on delays arising in the departure airport. Therefore, features associated with post-departure information are not considered. For example, *TaxiIn* is omitted due to information about when the flight wheels arrive at the destination gate runaway; others are *ArrTime, AirTime, ActualElapsedTime,* and *ScheduledElapsedTime*. In addition, there are 5 other variables: *NASDelay, WeatherDelay, SecurityDelay, LateAircraftDelay,* and *CarrierDelay*). Features having no information about flight delays, such as *TailNum, Year, UniqueCarrier, FlightNum, DestCode, OriginCode, Diverted, CancellationCode, DayOfWeek, DayofMonth,* and *Cancelled*. We have calculated the correlation coefficient to estimate how a pair of variables influence one another and to detect collinearity between variables. Variables such as *Delay_Level*, *DepTime,* and *ScheduledArrivalTime* are omitted due to high correlation with *Month, DepDelay and ArrDelay*. We used the following six features in training our model: *Months, ScheduleDepTime, Distance, TaxiOut, DepDelay,* and *ArrvDelay.*

Flight delay is created by numerous factors, including carrier, weather conditions, air traffic congestion, security, and mechanical issues. However, some of the essential features that affect flight delays are*:* The month of the year (*Months*), as different months may experience different weather patterns or holiday travel patterns that can impact flight schedules. For instance, holiday seasons may experience more delays due to people travelling more spent time with their loved ones. *Distance:* The distance of the flight can also impact its delay, as longer flights may encounter more air traffic congestion, require more time for boarding and unloading, and may

have more complicated routing. *ScheduleDepTime:* The scheduled departure time can affect the delay of a flight as certain times of day, such as peak travel times, may experience more congestion on the runway or in the air. *TaxiOut:* The time it takes for an aircraft to taxi from the gate to the runway, known as taxi-out time, can significantly impact the delay of a flight. Longer taxi-out times can lead to a delayed departure, causing the flight to arrive late. *DepDelay:* The departure delay, which measures the time difference between the scheduled and actual departure times, is one of the most crucial features affecting flight delay. A longer departure delay may lead to more significant delays throughout the flight, impacting the arrival time. *ArrDelay:* Finally, the arrival delay, which measures the time difference between the scheduled arrival time and the actual arrival time, is also a critical factor in a flight delay. Longer arrival delays can disrupt passengers' schedules, cause missed connections, and impact airline performance.

Table 2 shows the features we utilized in training our model. The remaining features with incomplete information, such as N.A., were also removed to process the dataset further to be suitable for the proposed model.

Table 2

Features Used in Training the Model

S/No	Feature	Type	Description
1	Month	Numerical(Discrete)	Recorded flight months from October to December.
2	Distance	Numerical (Continuous)	Origin and destination distance in miles.
3	ScheduleDepTime	Numerical (Continuous)	Departure schedule.
4	TaxiOut	Numerical (Continuous)	Taxi-out time in minutes.
5	DepDelay	Numerical (Nominal)	Time difference between a scheduled and actual departure.
6	ArrDelay	Numerical (Nominal)	Time difference between a scheduled and actual arrival.

Note. The dataset contains 10 airports with their codes, as shown in Table 3.

Table 3

Airport Codes Used in the Study

S/No	Airport Codes	Name of Airport
1	ATL	Atlanta
2	ORD	Chicago
3	LAX	Los Angeles
4	SAN	San Diego
5	MSP	Minneapolis
6	PHX	Phoenix
7	PHL	Philadelphia
8	SFO	San Francisco
9	DEN	Denver
10	BOS	Logan

We used min-max normalization for our data because it is all in numerical form. We transformed the data to fit the training and testing set. This process is important because all

features may have various data types. The numerical differences are eliminated because of the different range of values when computing. A value *x* is converted into x' range [*max_new* – *min_new*] as follows

$$x' = \frac{x - x_{min}}{x - x_{max}} \text{ x } [max\_new - min\_new] + min\_new \quad (1)$$

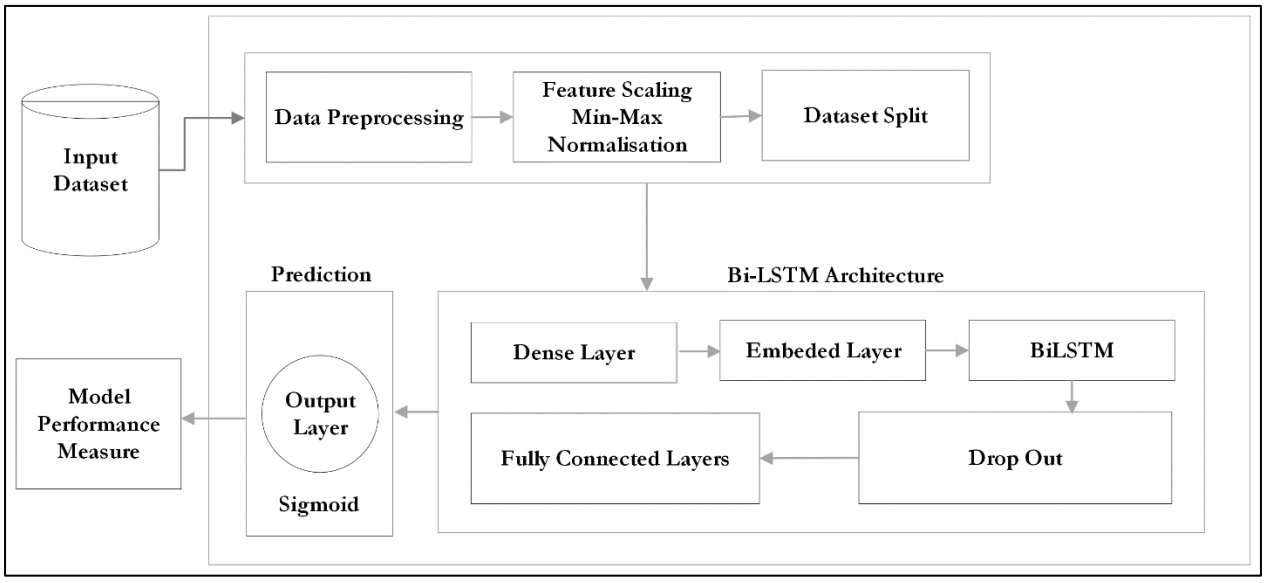
where the range of the transform values is denoted from *min_new* to *max_new*. We set *min_new* = 0 and *max_new* = 1. The transform values were then used as input into the BiLSTM architecture.

Proposed Methodology

The methodology proposed for this study is described in detail in this section, concentrating on all aspects that helped improve the tuning capabilities of the methods we compared. Figure 1 shows the clear steps taken in the proposed methodology. The proposed BiLSTM model pseudocode is presented in Algorithm 1.

Figure 1

Proposed Prediction Methodology



Algorithm 1

Proposed BiLSTM Approach Pseudocode

Input:	Dataset D_{train}, D_{test} of data
Output:	Flight classification prediction results *R* of test data
1	Load input dataset
2	Split dataset into D_{train} and D_{test}; 70:30 percentage ratio
3	Split D_{train} into train and validation set for training the algorithm
4	Dataset pre-processing of the D_{train}, D_{test} Min-Max normalization using Equation (1)
5	BiLSTM layers for enhancing the extraction of important prediction features
6	Preventing overfitting with Dropout layers
7	Batch normalization layer using Equations (2) to (5)
8	Dense layers
9	Fully connected layers
10	Sigmoid final layer
11	*For each training epoch*
12	Train the proposed model; *Train (P)*
13	*End for*
14	Evaluate the model D_{train}, D_{test} using performance measures

Batch Normalization Layer

The training dataset is acquired batch-by-batch. The distributions are consequently unstable and non-uniform, and the network parameters in each training cycle must be fitted, showing the model's significant convergence. Batch normalization is used for reparameterization. The mean α_D and variance $\propto_D^2$ is determined by the batch normalization technique for each training dataset batch and the scale adjusted for the original dataset to unity-variance and zero-mean. The shifted data $\hat{\alpha}_1$ are applied to weight and bias to help enhance expressive power. Equation 2 to Equation 5 shows the computation of the batch normalization. The batch normalization algorithm does the reparameterization update coordination across the neural network layers.

$$\alpha_D = \frac{1}{n}\sum_{i=1}^{n} a_i \quad (2)$$

$$\propto_D^2 = \frac{1}{n}\sum_{i=1}^{n}(a_i - \propto_D)^2 \quad (3)$$

$$\hat{a}_1 = \frac{a_i - \propto_D}{\sqrt{\propto_D^2 + \varepsilon}} \quad (4)$$

$$b_i = \gamma \hat{a}_1 + \beta \quad (5)$$

where b_i ε, γ and β is the final normalized value and parameters that make sure the batch normalization learns the identity function in a few cases.

Bidirectional Long Short-Term Memory (LSTM) Architecture

The most sophisticated type of machine learning available today is deep learning. It has brought about the growth in various models with neural networks for application in solving real-world problems. We utilized an effective deep learning method known as long short-term memory because of its memory-oriented features to model and analyse flight delays. BiLSTM, as a deep learning approach, is efficient in analysing and extracting important data features needed for a predictive task. It is an extension of the Recurrent Neural Network (RNN). The LSTM structure was designed to address the *vanishing gradient* problem of the RNN structure. The LSTM Cell structure contains an input gate, output-gate, forget gate and memory unit (Gers et al, 2000; Hochreiter & Schmidhuber, 1997). The memory block structure of a one-layer neural network controls the forget gate. Equation 6 can determine the activation of the gate.

$$f_a = \sigma\left(W\left[x_{a,}\, h_{a-1,}\, C_{a-1}\right] + b_v\right) \quad (6)$$

where x_a denotes the sequence of inputs; h_{a-1} is the output of the previous block; C_{a-1} denotes the block memory of the previous LSTM; and the bias vector is denoted by b_v. The individual

weight vectors for each input are denoted by W, while the logistic sigmoid activation function is denoted by σ.

The basic N.N. with the *tanh* activation function has an input gate as part of its structure with new memory created by the block effect of the prior memory. The operations can be computed using Equation 7 and Equation 8.

$$i_a = \sigma(W[x_{a,} h_{a-1,} C_{a-1}] + b_i) \quad (7)$$

$$C_a = f_a . C_{a-1} + i_a . tanh([x_{a,} h_{a-1,} C_{a-1}] + b_c) \quad (8)$$

LSTM with one-way movement relies on past data, but it is always inadequate. Dependencies are prevented by consciously remembering long-term information, which in practice is the behaviour of the LSTM by default. The BiLSTM only analyzes data in two directions. Two values are held by the hidden layer of the BiLSTM (Graves & Schmidhuber, 2005), which are employed for forward computation and reverse calculation. The final output prediction performance of the BiLSTM is enhanced by the two values which determine the output (Zhang et al., 2019). Figure 2 shows the architecture of a BiLSTM.

Figure 2

An Architecture of a BiLSTM

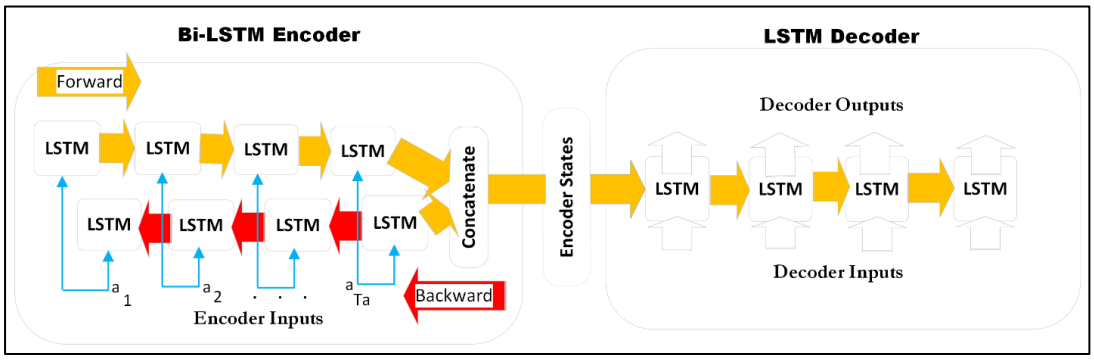


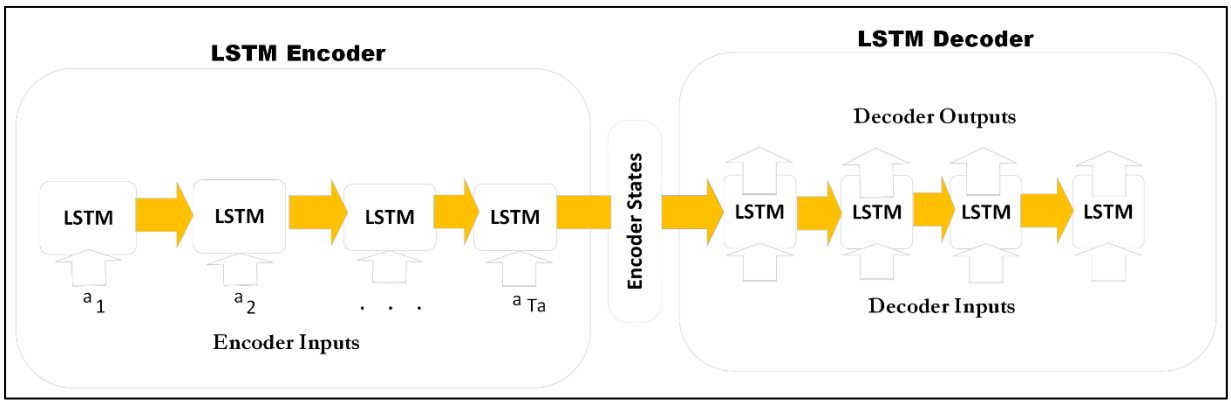
Figure 2 shows two recurrent components, forward and backward. The forward component computes the hidden and cell states, similar to a standard LSTM, while the backward component computes by a reverse-chronological input sequence. For example, it is taken from Ta to 1 timestamp. The backward component usage creates a way in which future data are captured by the network and tries to learn its weights, respectively. It helps capture some dependencies by the network that a standard unidirectional LSTM might not have been able to capture. BiLSTM is also good in Natural Language Processing because of its ability to capture input sequence dependencies quite well. The forward components' hidden and cell states differ from the backward components. Therefore, the hidden and cell states of the forward component are concatenated with the backward components to get encoded.

Unidirectional Long Short-Term Memory Architecture

The sequence of time series is taken as input by the LSTM encoder (each LSTM cell takes a one-time step), and an input sequence for encoding is created. A vector consisting of hidden and cell states is created by encoding all the LSTM cells. The LSTM decoder receives the encoding and other decoder inputs generated for the predictions (decoder outputs). We set our target output sequence during the model training as the decoder output for the model to train against the targeted output. Figure 3 shows the architecture of the unidirectional LSTM.

Figure 3

An Architecture of a Unidirectional LSTM



Model Evaluation Metrics

To measure the performance of our model in the experiment, we employed some well-known benchmark metrics for evaluating models for classification and predictive tasks. These benchmark evaluation metrics are accuracy, recall, precision, F1-score, and Mathew's correlation coefficient (MCC). The accuracy is the rate of the correctness of a classifier. We then use the sum of true-negative (T.N.), true-positive (T.P.), false-positive (F.P.) and false-negative (F.N.). Thus, the ratio of records correctly predicted to the total number of records is known as accuracy, as shown in Equation 9. The rate of correctly predicted values from the positive record is the recall, also known as sensitivity or true positive rate (TPR), thus calculated as Equation 10. The ratio of true positive records to the predicted positive records is the precision, as shown in Equation 11. The F1-score is the harmonic mean of the recall and precision, and Equation 12 shows how to calculate the F1-score. At the same time, Mathew's correlation coefficient measures the classification quality as compared with each of the classes' recall and precision relative to each other, as shown in Equation 13.

Accuracy can be represented by Equation 9:

$$\text{Accuracy} = \frac{\sum_{a=0}^{z}(TP_a + TN_a)}{\sum_{a=0}^{z}(TP_a + TN_a + FP_a + FN_a)} \quad (9)$$

Recall can be represented by Equation 10:

$$\text{Recall} = \frac{\sum_{a=0}^{z} TP_a}{\sum_{a=0}^{z}(TP_a + FN_a)} \quad (10)$$

Precision can be represented by Equation 11:

$$\text{Precision} = \frac{\sum_{a=0}^{z} TP_a}{\sum_{a=0}^{z}(TP_a + FP_a)} \quad (11)$$

F1-score can be represented by Equation 12:

$$\text{F1} = 2 \text{ X} \frac{(Precision \ X \ Recall)}{(Precision + Recall)} \quad (12)$$

Mathew's correlation coefficient (MCC) can be represented by Equation 13:

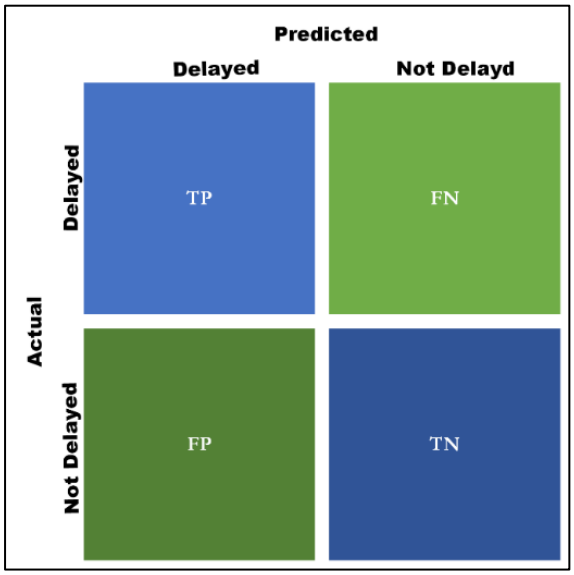
$$\text{MCC} = \frac{\sum_{a=0}^{z}(TP_a \text{ X } TN_a) - (FN_a \text{ X } FP_a)}{\sum_{a=0}^{z}(TP_a + FP_a) + (TP_a + TN_a) + (TN_a + FP_a) + (TN_a + FN_a)} \quad (13)$$

The model results can be visualized in Confusion Matrix format, as shown in Figure 4.

Figure 4 is the 2X2 matrix called the Confusion Matrix, where all the correctly classified results are diagonal. The sum of the diagonals is the number of all those correctly classified.

Figure 4

The Structure of a Confusion Matrix for a Binary Classification Task



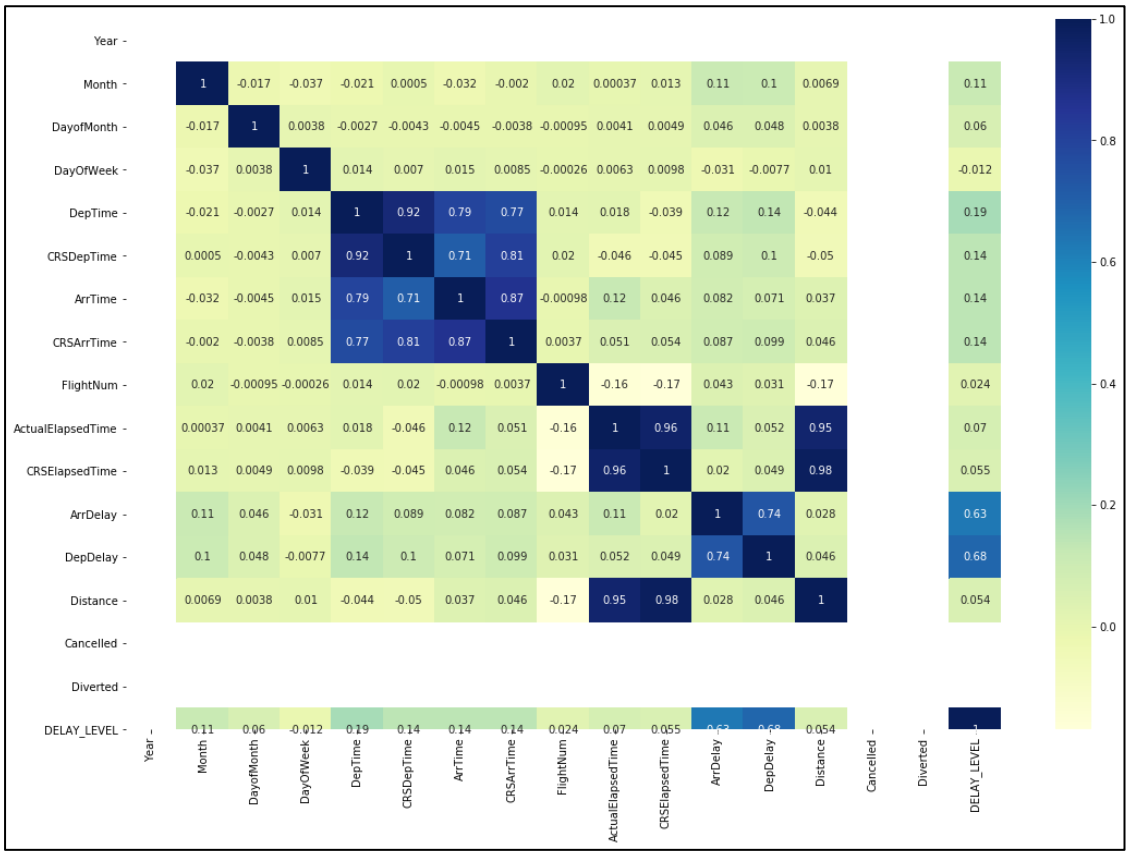
Results and Discussion

The proposed method results validated on three months public dataset. The performance of the proposed method with BiLSTM is then compared with unidirectional LSTM, and different merge mode performances were also evaluated to discover the most suitable.

We plotted a correlation matrix to show the collinearity of the multiple variables in the dataset. This helps to determine which variables have the same effect on the output variable: the flight delay. Figure 5 shows that Actual Elapsed Time and Distance are highly correlated, so dropping either from the dataset will not affect the model prediction. From the heap, we can see the importance of each feature in the flight delay class. The ScheduleDepTime has a high contribution base on the correlation values on the delay class, while DepDelay and ArrDelay features have a similar level of importance on the delay classes. The month is the next most important feature of the delayed classes. Finally, distance is the next important feature, and taxiOut has the minimum importance to the delay classes.

Figure 5

Flight Correlation Matrix



Experimentation Setup

We evaluated the proposed model efficacy by conducting our experiment on a U.S. BTS for 2013 with three months of records of real flights. All the computations were conducted on a Personal Computer (PC) with Intel(R) Core(T.M.) i7-9700 CPU with a processor speed of 3.00GHz and 32GHz RAM. We used the following libraries TensorFlow Core-2.4.1, TensorFlow GPU-2.4.1, Pytorch 1.9.1, NumPy-1.19.1, pandas-0.25.3, sci-kit learn-0.23.2, Scipy-1.5.2, PySimpleGUI-4.29.0, and Matplolib-3.3.1.

Results Comparative Discussions

In this section, we present the comparative analysis of the evaluated methods in this research. Table 4 contains the performance values of each of the methods.

Table 4

Performance Comparison of LSTM and BiLSTM Models

S/No	Methods	Classes	Precision	Recall	F1-Score	Support	MCC
1	LSTM	Class 0	0.8384	0.9443	0.8756	107892	
		Class 1	0.4469	0.0743	0.0896	101823	
		Accuracy	-	-	**0.7645**	209715	0.4644
		Macro Average	0.4532	0.4532	0.4356	209715	
		Weighted Average	0.7362	0.7453	0.7453	209715	
2	BiLSTM	Class 0	0.8324	0.9898	0.8945	107892	
		Class 1	0.5643	0.0989	0.4988	101823	
		Accuracy	-	-	**0.9756**	209715	0.9944
		Macro Average	0.4202	0.4332	0.4122	209715	
		Weighted Average	0.70023	0.7234	0.7213	209715	

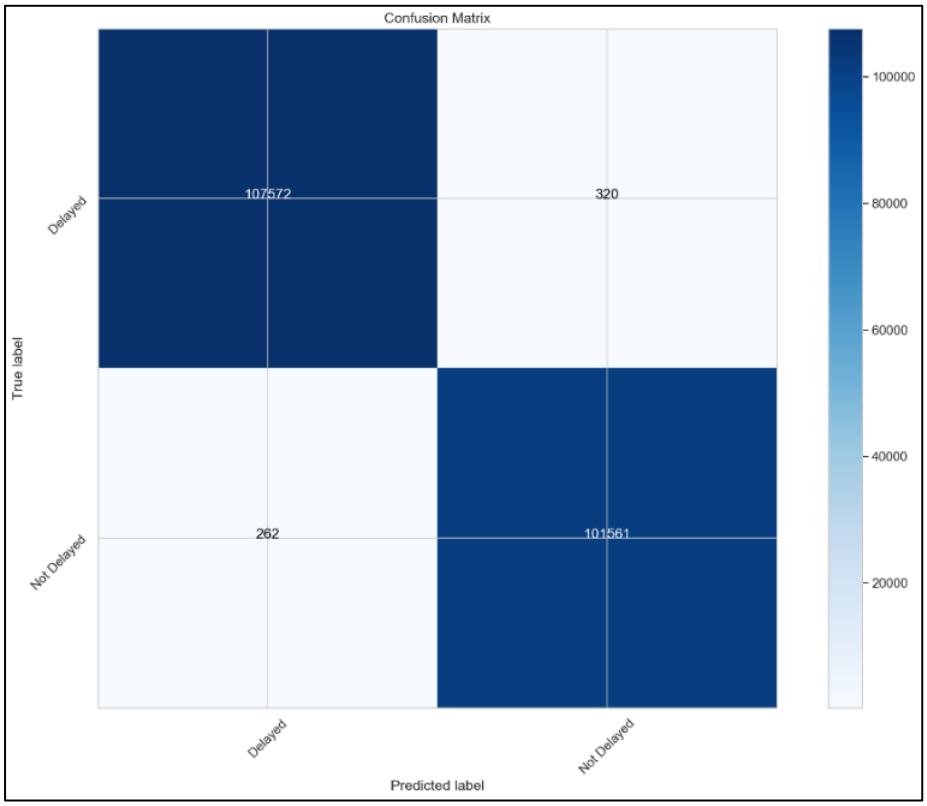
The LSTM model had an accuracy of 76.45%, which is comparatively lower than the BiLSTM model. The LSTM model has a precision, recall and F1-Score of 83.84, 94.43% and 87.56%, respectively, in predicting class 0, which is better than that of class 1, having a precision of 44.69%, recall of 7.43%, F1-Score of 8.96%, respectively. It can be seen in the confusion matrix classification report presented in Figure 6, where the model classified all the delay classes without misclassification while the not delayed class has some misclassification. When predicting classes of the delay, the LSTM model results show a lower performance when compared with the

BiLSTM; however, a better performance LSTM model shows on the precision of class 1, which indicates that the correctly retrieved instances of the model on the class. The mutual correlation of the LSTM predicted classes, an MCC score of 0.4644, is significantly lower than the BiLSTM because of their overall differences in the accuracy of predictions, which means forward and backward networks of BiLSTM help in retrieving the sequences of the previous layers during the training.

The results show that the BiLSTM model performs better than the LSTM in prediction instances with just a few misclassifications, as can be seen in the confusion matrix in Figure 7. The results of the BiLSTM model show an accuracy of 97.56%, an increase of 21.11% compared to the LSTM accuracy model. The model performance in class 0 is solid, with precision, recall, and F1-Score of 83.24%, 98.98% and 89.45%, respectively. Although the performance has a small margin compared to LSTM on class 0, the model correctly identifies a higher proportion of the not delay class. The precision of class 1 is 56.43%, but it is better than the baseline LSTM model in class 1. The models recall, and F1-Score in class 1 are 9.89% and 49.88%, respectively. The MCC score for the BiLSTM model is 0.9944, which shows a high correlation between the two predicted classes of the model and, thus, means a good model.

Figure 6

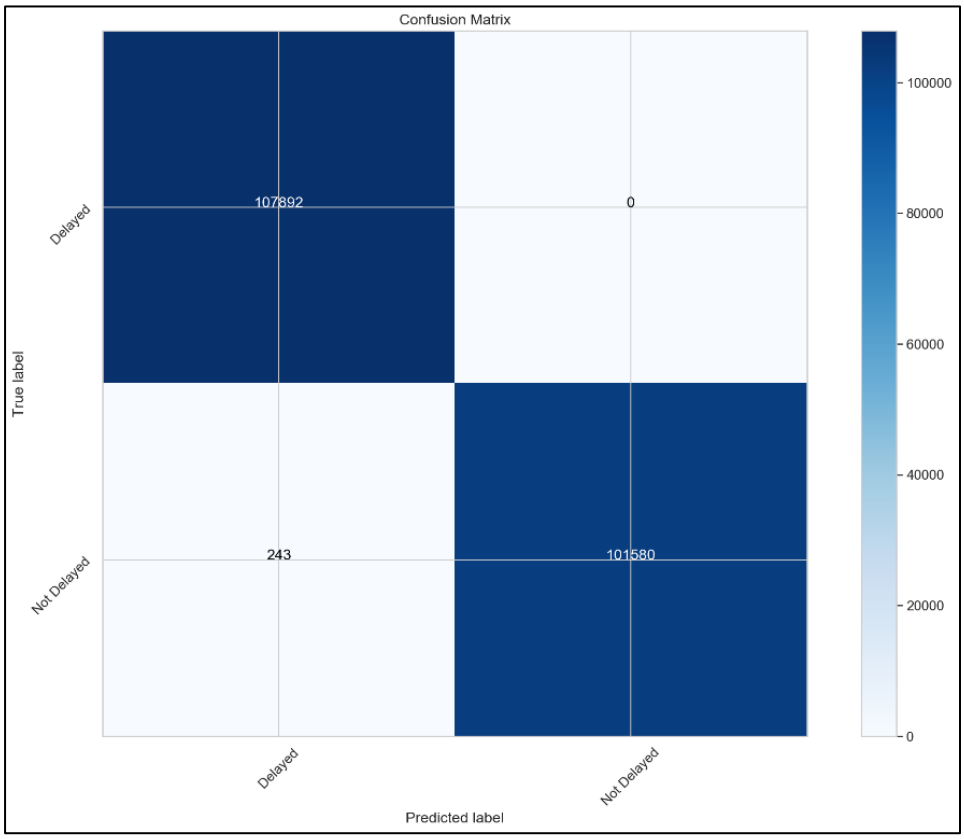
Confusion Matrix for LSTM Approach



The subsequent data points are more important than the previous ones, which is considered missing information by the LSTM model. The BiLSTM model solves this LSTM model problem by doing two forward and backward LSTM model training for the training dataset sequences. The confusion matrix in Figure 5 shows that the unidirectional LSTM model correctly predicts 107,572 and 101,561 of the delayed and not-delayed flight classes, respectively, while Figure 6 shows that BiLSTM correctly predicted 107, 892 and 101, 580 of the delayed and not-delayed flights classes. It clearly shows from the confusion matrix results that the BiLSTM model's performance in predicting the flight is better than that of the LSTM model. The overall result shows the advantage of using deep BiLSTM over an LSTM because the prediction error of an LSTM tends to increase significantly as the number of prediction steps.

Figure 7

Confusion Matrix for BiLSTM Approach



Conclusion and Future Direction

The delay propagation in the air traffic network propagates through the entire network with speed due to the complex nature of the system. This study compared two classification and prediction methods known as LSTM and BiLSTM by building a classification model to analyse flight delay on-time datasets to assess the efficacy and viability of BiLSTM to LSTM. The BiLSTM model has a higher accuracy of 97.56% than the unidirectional LSTM with an accuracy of 76.45%; the change in the accuracy of the BiLSTM model shows its potential in the prediction of flight delays. BiLSTM plausibly outperforms the LSTM model due to its bidirectional ability (forward and backward), leveraging any feature selection to handle missing sequences during the

model training. Another reason could be that BiLSTM takes additional time to fetch batches of data to have an equilibrium, though it is slow because of this process. It indicates that the additional features associated with the data are captured by BiLSTM but cannot be exposed by the unidirectional LSTM models because the training is only one way. Thus, the researchers recommend BiLSTM over LSTM for binary time series analysis, classification, or prediction.

This research contributes to air transportation and stakeholder in decision-making processes, thereby improving air passengers ' experience and increasing income from aviation and non-aviation services. More cross-validation methods and larger sample sizes across different regions is needed to develop models and further evaluate the regional performance of the model towards a universal central prediction of flights across nations. The architectural design should be improved to achieve better tuning and higher accuracy of the neural network. Training the model with the weather, aircraft age, aircraft model, and factors limiting airport infrastructure (i.e., available runways to flights)may help solve the low performance related to the unidirectional LSTM model because, from the dataset, the weather-related delay is highly imbalanced.

Acknowledgement and Funding

The Petroleum Trust Development Fund (PTDF) Nigeria partially funded this work through grant number PTDF/ED/OSS/PHD/DBB/1558/19 for one of the first author's Ph.D. studies. We also thank the UKRI for the Covid-19 recovery grant under the budget code SA077N.

Data Availability

All data used in this research is available at https://www.bts.gov/

References

Ball, M., Barnhart, C., Dresner, M., Hansen, M., Neels, K., Odoni, A., Peterson, E., Sherry, L., Trani, A., Zou, B., Britto, R., Fearing, D., Swaroop, P., Uman, N., Vaze, V., & Voltes, A. (2010). *Total delay impact study: A comprehensive assessment of the costs and impacts of flight delay in the United States*. NEXTOR. http://www.isr.umd.edu/NEXTOR/pubs/TDI_Report_Final_10_18_10_V3.pdf

Baumgarten, P., Malina, R., & Lange, A. (2014). The impact of hubbing concentration on flight delays within airline networks: An empirical analysis of the US domestic market. *Transportation Research Part E: Logistics and Transportation Review*, *66*, 103–114. https://doi.org/10.1016/j.tre.2014.03.007

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). https://doi.org/10.1145/2888402

Bisandu, D. B., Homaid, M. S., Moulitsas, I., & Filippone, S. (2021). A deep feedforward neural network and shallow architectures effectiveness comparison: Flight delays classification perspective. ICAAI '21: *Proceedings of the 5th International Conference on Advances in Artificial Intelligence*, 1–10. https://doi.org/10.1145/3505711.3505712

Bisandu, D. B., Moulitsas, I., & Filippone, S. (2022). Social ski driver conditional autoregressive-based deep learning classifier for flight delay prediction. *Neural Computing and Applications*, *34*(11), 8777–8802. https://doi.org/10.1007/s00521-022-06898-y

Cai, K. Q., Zhang, J., Xiao, M. M., Tang, K., & Du, W. B. (2017). Simultaneous optimization of airspace congestion and flight delay in air traffic network flow management. *IEEE Transactions on Intelligent Transportation Systems*, *18*(11), 3072–3082. https://doi.org/10.1109/TITS.2017.2673247

Carvalho, L., Sternberg, A., Maia Gonçalves, L., Beatriz Cruz, A., Soares, J. A., Brandão, D., Carvalho, D., & Ogasawara, E. (2021). On the relevance of data science for flight delay research: Asystematic review. *Transport Reviews*, *41*(4), 499–528. https://doi.org/10.1080/01441647.2020.1861123

Chakrabarty, N. (2019). A data mining approach to flight arrival delay prediction for American Airlines. *IEMECON 2019 - 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference*, 102–107. https://doi.org/10.1109/IEMECONX.2019.8876970

Cheevachaipimol, W., Teinwan, B., & Chutima, P. (2021). Flight delay prediction using a hybrid deep learning method. *Engineering Journal*, *25*(8), 99–112. https://doi.org/10.4186/ej.2021.25.8.99

Chen, J., & Li, M. (2019). Chained predictions of flight delay using machine learning. *AIAA Scitech 2019 Forum*, 1–25. https://doi.org/10.2514/6.2019-1661

Cios, K. J., Pedrycz, W., Roman, W. S., & Kurgan, L. A. (2007). Data mining: A knowledge discovery approach. *In Springer US*. 1-606. https://doi:10.1007/978-0-387-36795-8

Civil Aviation Administration of China. (2021). *2021 National Civil Aviation Work Conference and Safety Work Conference Held*. Civil Aviation Administration of China. http://www.gov.cn/xinwen/2021-01/12/content_5579282.htm

Ding, Y. (2017). Predicting flight delay based on multiple linear regression. *IOP Conference Series: Earth and Environmental Science*, *81*(1), 0–7. https://doi.org/10.1088/1755-1315/81/1/012198

Dou, X. (2020). Flight Arrival Delay Prediction And Analysis Using Ensemble Learning. *IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference*, 836–840. https://doi.org/10.1109/itnec48623.2020.9084929

Efthymiou, M., Njoya, E. T., Lo, P. L., Papatheodorou, A., & Randall, D. (2019). The impact of delays on customers' satisfaction: An empirical analysis of the British Airways on-time performance at Heathrow Airport. *Journal of Aerospace Technology and Management*, *11*, 1–13. https://doi.org/10.5028/jatm.v11.977

Etani, N. (2019). Development of a predictive model for on-time arrival flight of airliner by discovering correlation between flight and weather data. *Journal of Big Data*, *6*(1), 1–7. https://doi.org/10.1186/s40537-019-0251-y

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, *12*(10), 2451–2471. https://doi.org/10.1162/089976600300015015

Fu, X., Lei, Z., Liu, S., Wang, K., & Yan, J. (2020). On-time performance policy in the Chinese aviation market - An innovation or disruption? *Transport Policy*, *95*, A14–A23. https://doi.org/10.1016/j.tranpol.2020.06.008

Gopalakrishnan, K., & Balakrishnan, H. (2017). A comparative analysis of models for predicting delays in air traffic networks. *12th USA/Europe Air Traffic Management Research and Development Seminar*.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, *18*(5–6), 602–610. https://doi.org/10.1016/j.neunet.2005.06.042

Gu, Y., Yang, J., Wang, C., & Xie, G. (2020). Early warning model for passenger disturbance due to flight delays. *PLoS ONE*, *15*(9), 1–13. https://doi.org/10.1371/journal.pone.0239141

Gui, G., Liu, F., Sun, J., Yang, J., Zhou, Z., & Zhao, D. (2020). Flight delay prediction based on aviation big data and machine learning. *IEEE Transactions on Vehicular Technology*, *69*(1), 140–150. https://doi.org/10.1109/TVT.2019.2954094

Hochreiter, S., & Schmidhuber, J. (1997). Long Shortterm Memory. *Neural Computation, 9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Karim, F., Majumdar, S., Darabi, H., & Chen, S. (2017). LSTM fully convolutional networks for time series classification. *IEEE Access, 6*, 1662–1669. https://doi.org/10.1109/ACCESS.2017.2779939

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

Kim, Y. J., Choi, S., Briceno, S., & Mavris, D. (2016). A deep learning approach to flight delay prediction. *2016 IEEE/AIAA 35th Digital Avionics Systems Conference*. https://doi.org/10.1109/DASC.2016.7778092

Kuhn, N., & Jamadagni, N. (2017). Application of machine learning algorithms to predict flight arrival delays. *CS229*, 1–6.

Lin, Y., Zhang, J.-w., & Liu, H. (2019). Deep learning based short-term air traffic flow prediction considering temporal–spatial correlation. *Aerospace Science and Technology, 93*. https://doi.org/10.1016/j.ast.2019.04.021

Liu, Y.-J., Cao, W.-D., & Ma, S. (2008). Estimation of arrival flight delay and delay propagation in a busy hub-airport. *2008 Fourth International Conference on Natural Computation, 4*, 500–505. https://doi.org/10.1109/ICNC.2008.597

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2015). Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems, 16*(2), 865–873. https://doi.org/10.1109/TITS.2014.2345663

Manna, S., Biswas, S., Kundu, R., Rakshit, S., Gupta, P., & Barman, S. (2018). A statistical

approach to predict flight delay using gradient boosted decision tree. *ICCIDS 2017 - International Conference on Computational Intelligence in Data Science*, 1–5. https://doi.org/10.1109/ICCIDS.2017.8272656

Maxson, R. W. (2018). *Prediction of airport arrival rates using data mining methods* [Doctoral dissertation, Embry-Riddle Aeronautical University]. https://commons.erau.edu/edt/419/

Mueller, E. R., & Chatterji, G. B. (2002). Analysis of aircraft arrival and departure delay characteristics. *AIAA's Aircraft Technology, Integration, and Operations (ATIO) 2002 Technical Forum*, 1–14. https://doi.org/10.2514/6.2002-5866

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*(2), 1461–1484. https://doi.org/10.1007/s11063-020-10318-4

Takeichi, N., Kaida, R., Shimomura, A., & Yamauchi, T. (2017). Prediction of delay due to air traffic control by machine learning. *AIAA Modeling and Simulation Technologies Conference*, 1–8. https://doi.org/10.2514/6.2017-1323

United States Department of Transportation. (2020). *Bureau of Transportation Statistics.* https://www.bts.gov/

Vandehzad, M. (2020). *Efficient flight schedules with utilizing Machine Learning prediction algorithms* [Master's thesis, Malmö University].

Wu, C. L. (2008). Monitoring aircraft turnaround operations - Framework development, application and implications for airline operations. *Transportation Planning and Technology*, *31*(2), 215–228. https://doi.org/10.1080/03081060801948233

Wu, Y., Mei, G., & Shao, K. (2022). Revealing influence of meteorological conditions and flight factors on delays using XGBoost. *Journal of Computational Mathematics and Data Science*, *3*. https://doi.org/10.1016/j.jcmds.2022.100030

Yazdi, M. F., Kamel, S. R., Chabok, S. J. M., & Kheirabadi, M. (2020). Flight delay prediction based on deep learning and Levenberg-Marquart algorithm. *Journal of Big Data*, *7*(1). https://doi.org/10.1186/s40537-020-00380-z

Ye, B., Liu, B., Tian, Y., & Wan, L. (2020). A methodology for predicting aggregate flight departure delays in airports based on supervised learning. *Sustainability*, *12*(7). https://doi.org/10.3390/su12072749

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

Zhang, H., Song, C., Wang, H., Xu, C., & Guo, J. (2019). Airport delay prediction based onspatiotemporal analysis and Bi-LSTM sequence learning. *2019 Chinese Automation Congress*, 5080–5085. https://doi.org/10.1109/CAC48633.2019.8996754