Flight delay prediction based on deep learning and Levenberg-Marquart algorithm

Maryam Farshchian Yazdi[1], Seyed Reza Kamel[2*] , Seyyed Javad Mahdavi Chabok[2] and Maryam Kheirabadi[1]

*Correspondence:
Drkamel@mshdiau.ac.ir
[2] Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran
Full list of author information is available at the end of the article

Abstract

Flight delay is inevitable and it plays an important role in both profits and loss of the airlines. An accurate estimation of flight delay is critical for airlines because the results can be applied to increase customer satisfaction and incomes of airline agencies. There have been many researches on modeling and predicting flight delays, where most of them have been trying to predict the delay through extracting important characteristics and most related features. However, most of the proposed methods are not accurate enough because of massive volume data, dependencies and extreme number of parameters. This paper proposes a model for predicting flight delay based on Deep Learning (DL). DL is one of the newest methods employed in solving problems with high level of complexity and massive amount of data. Moreover, DL is capable to automatically extract the important features from data. Furthermore, due to the fact that most of flight delay data are noisy, a technique based on stack denoising autoencoder is designed and added to the proposed model. Also, Levenberg-Marquart algorithm is applied to find weight and bias proper values, and finally the output has been optimized to produce high accurate results. In order to study effect of stack denoising autoencoder and LM algorithm on the model structure, two other structures are also designed. First structure is based on autoencoder and LM algorithm (SAE-LM), and the second structure is based on denoising autoencoder only (SDA). To investigate the three models, we apply the proposed model on U.S flight dataset that it is imbalanced dataset. In order to create balance dataset, undersampling method are used. We measured precision, accuracy, sensitivity, recall and F-measure of the three models on two cases. Accuracy of the proposed prediction model analyzed and compared to previous prediction method. results of three models on both imbalanced and balanced datasets shows that precision, accuracy, sensitivity, recall and F-measure of SDA-LM model with imbalanced and balanced dataset is improvement than SAE-LM and SDA models. The results also show that accuracy of the proposed model in forecasting flight delay on imbalanced and balanced dataset respectively has greater than previous model called RNN.

Keywords: Deep learning, Stacked denoising autoencoders, Flight delay prediction, Big data

Yazdi *et al. J Big Data*   (2020) 7:106

Page 2 of 28

Introduction

As the air travels have a significant role in economy of agencies and airports, it is necessary for them to increase quality of their services. One of the important modern life challenges of airports and airline agencies is flight delay. In addition, delay in flight makes passengers concerned and this matter causes extra expenses for the agency and the airport itself. In 2007, U.S government had endured 31–40 billion dollar downsides due to flight delays [1]. In 2017, 76% of the flights arrived on time. Where, in comparison to 2016, the percentage of on time flights decreased by 8.5% [2]. As some of the reasons of flight delays the following can be mentioned: security, weather conditions, shortage of parts and technical and airplane equipment issues and flight crew delays [3–5]. Delay in flight is inevitable [6], which has too much negative economic effects on passengers, agencies and airport [7–11]. Furthermore, delay can damage the environment through fuel consumption increment and also leads to emission of pollutant gases [1, 12–16]. In addition, the delay affects the trade, because goods' transport is highly dependent to customer trust, which can increase or decrease the ticket sales, so that on time flight leads to customer confidence [17, 18]. So that, flight prediction can cause a skillful decision and operation for agencies and airports, and also a good passenger information system can relatively satisfy the customer [19].

According to abundant and diversity of reasons for flight delays, We are faced with a massive amount of data which is not possible to be processed through previous methods of data [17] analysis like classification [1], or the decision tree [8] and machine learning based methods [1, 2, 17, 20, 21] to process this volume of data are not proper, because characteristics of older intelligent system has been designed by human and usually were personalized, also people rarely perceive some features and usually neglect these matters. On the other hand, in older learning process, as the number of categories available for classification increases, the level of difficulty increases [8] and extraction of important and effective features becomes relatively impossible. Due to complexity and effect of parameters on each other, the problem of flight delay prediction is considered as NP-Complete [22]. Furthermore, the problem essentially is accompanied by oscillation and also these are considered as non-linear problems [23]. On the other hand, applied data includes noise and error that should be handled to cope with the problem [24, 25].

There have been too many studies in this area. For example, older Regression method [26] has been used to compute delay propagation. For this model, the destination delay is highly dependent to arrival flights and the effective factors include; day, time, airport capacity and some factors are related to passenger loads. In addition, as the problem neglects the weather conditions, this model shows inefficiency in U.S.A but it is suitable for Europe. Where, only 1–4% of the Europe flights delayed due to weather condition, this value for U.S.A is between 70 and 75% in [27] an intelligent neural network has been designed which estimated the destination delay for actual applications in controlling traffic progress. This model employs factors of airport type, airplane type, date, time, flight path, flight frequency for network training and non-linear and linear for data analysis. As it is difficult to interpret neural network parameters, the way factor behavior and most important verification of the most important factors in flight is extremely difficult. Furthermore, older intelligent algorithm usually uses shadow learning models to solve conditions with a big data in complicated classifications. However, results of this

analysis are very different with respect to ideal condition. Although model design can have a good or bad situation, response is highly dependent to experience and even happenstance and this procedure requires too much time. Therefore, traditional simulation and modelling techniques is not suitable or even efficient for such problems. There is an ongoing subject of study which solves this problem and this paper also has tried to use that subject in modelling.

One of the newest modern methods in solving such extended and complicated accompanied by bulky data that has been concerned by many scientists is deep neural networks [21, 24, 25, 28]. The design of learning technology is taken from human neural network learning is a branch of machine learning and collection of algorithms that trying to model such high-level abstract contents through application learning in different layers and levels. Therefore, this subject enables the deep learning to process a bulky data volume in complicated data classification [29]. Moreover, this structure is proper for extracting some the characteristics, so that learning is capable to extract maximum number of possible characteristics [29]. Layered network structure and capability of computation for each data scale has led to progressing application of techniques. This networks have different types including convolutional Neural Network [30], Autoencoders [31], Restricted Boltzmann Machine [32] and Sparse coding based method [33] that each of them is applicated for specified problem.

One of the recently presented works in solving problem employs the recurrent Deep Neural Network and its results has a high accuracy in flight delay prediction [24]. However, this model has drawbacks of overfitting, that researchers have solved that through typical data dropout technique for each step of repeated training procedure. Moreover, application of this method decreases the computation time and memory space during the training.

The next drawback is the noise of input data. However, the researcher neglects the noise during prediction.

This paper tries to represent a model based on deep learning, which considers the effective factors in the delay. Moreover, noisy data [24, 25] requires utility of stack denoising autoencoder (SDA) in designing the model. Afterwards, optimized structure of the flight delay forecasting model with Levenberg-Marquart (LM) algorithm. In addition, in this paper by developing a deep learning-based model, the accuracy of flight delay predictions can be increased.

Finally, we review previous work related to our topic in "Literature review" section, a complete description of research process and also the holistic structure of the designed model is represented in the third section. Fourth section evaluates the determined results from the previous methods. Fifth section presents a conclusion and an overall view about the study.

Literature review

Nowadays, service quality plays an important role in attracting customers. Among these, air travels have their special customers and the most important matter in these travels is the flight time, on time arrival at destination for passengers such those who have an important meeting, that has been leading to high expenses for the passenger until get to their destination on time [34, 35]. Flight delay has negative economic effects on

Yazdi *et al. J Big Data*   (2020) 7:106

Page 4 of 28

the passenger, agencies and airports. Therefore, any reduction of these effect requires decreasing postponed flight price, so that prediction or estimation has a great significance and numerous studies has been to dedicated this subject. Correspondingly, all the scientists have tried to design a model that understands effective factors and computes effect of each factor and their relation. Overall, the prediction methods are classified into five groups including Statistical Methods [3, 36–56], Probability methods [7, 9, 50, 57–69], network-based methods [5, 70–85], operational methods [86–102] and machine learning methods [1, 2, 8, 17, 20, 27, 29, 34, 103–124].

In one of the best studies [56] that has been performed based on statistics delay time has been considered to be reduced. Their study has investigated important factors before fly and those which occur on the ground. In the next step, it has predicted the delay at destination based on factors that occur in the vicinity of arrival time at destination. Eventually, results have shown that whenever, the delay is correctly predicted, passenger disaffection and fuel consumption decrease and consequently number of flight increases. Moreover, it is possible to increase the agencies' benefits through reducing number of passengers who wrongly selected their routs or specifying the probabilities for some flights and optimizing delay time prediction.

Another prominent investigation based on Probability [57] has been done and the author believes that huge storm in U.S.A has highly affected the flight delay. This study has been devoted to predict delay based on mathematical calculations and through considering delay time duration of the flights that had been engaged to storm in the same day. Metrological reports have shown the effect of storm one hour before and after event cause ephemeral climate at the region. In the next step, Monte-Carlo simulation has been used to estimate the airport runway capacity, so that traffic of each runway would have been estimated. As the research has employed only one factor, the model has not enough accuracy, but it is possible to increase region air capacity path structure [57].

A model has been presented in [82], which is one of the best network-based models. The researchers have presented a model based on Bayesian and Gaussian mixture model- expectation maximization (GMM-EM) algorithm to predict and analyze the factors affecting the flight delay in Brazil for several point along the path. At the first stage of model, the degree of effectiveness for each factor is specified and then it has specified investigated whether the delay had happened in a greater domain or no. the next delay probability is computed using GMM-EM [82] and EM algorithm which are specified based on similarity. The result has shown that it is possible to predict the probability of delay in higher levels through specifying low level factors. Moreover, GMM-EM [82] similarity function has more values rather than EM algorithm [82] in each step, so that the results would have been converged sooner. In addition, the model accuracy is increased, so that the prediction is more trustable.

One of the best studies [93] in the area of operating method has been presented. Studied the effects of capacity and damage on different levels of delay in American airports.

Other simulations focus on stability and reliability during the delay and its propagation. For instance, in [90] the problems of congestion were studied. Then, a queue-based model was presented for analyzing delay propagation in consecutive flights in the Los Angeles airport.

One of the best studies [119] in the area of machine learning method has been presented by a model which applicate machine learning techniques to investigate delay in arrival flights. This research firstly has extracted important characteristics and then has been used for both neural networks and deep believe network through arbitrary samples to train the model. The model utilizes Memento [119] and Resilient Back Optimized Propagation [119] that the Resilient back propagations quicker than back propagation [119] and as a result the model training and consequently has been increased. Deep believe networks [119] is based on a few Boltzmann machine [119] that each communication layer receives data from the previous layer and in each step a Boltzmann machine [119] is added to Believe Network overall, training time reduced using parameter adjustment operation and learning rate, false classified error rate. As each layer has convergence at the output, training speed is reduced and the gradient approaches zero. In addition, a relatively small data base is used for the model because of limited system capacity. So that this problem leads to a noticeable reduction in prediction precision whenever it is not at database.

A model has been presented [125] which was one of the machine learning method. the researcher has presented a model based on support vector regressor (SVR) algorithm to predict flight delay in U.S.A airports. Due to the large amount of data, the data was grouped and sampled by month. At the first stage for categorical variables, cat-boost used the ordered boosting method. Because cat-boost itself had the effect of scoring features, it was possible to select parameters that were more important to the model when the threshold was unknown, so cat-boost was used to evaluate the features of each feature to select features, and finally 15 features were selected to build a training model.

Then has been used several common regression prediction algorithms to predict the delay at the same time for the round-trip flight between John F. Kennedy International Airport and O'Hare International Airport.

Finally, the specific delay time was predicted. The results have shown SVR has the best prediction result for the flight delay time with the best accuracy value was 80.44%. Also, the time characteristics had a large impact on the mode performance.

The air time and flight distance would also have a greater impact on on-time performance of specific flight; Different carriers and specific aircraft would also have a slight influence of on time performance. Accuracy of this model is low because detailed weather and aircraft data could not be collected.

A research [126] analyzes flight information of U.S domestic flight operated by American Airlines, covering top 5 busiest airports of US and predicting possible arrival delay of the flight using Data Mining and Machine Learning Approaches. Due to the imbalanced data, Over-Sampling technique, Randomized SMOTE was applied for Data Balancing. The Gradient Boosting Classifier Model was deployed by training and then Grid Search on Gradient Boosting Classifier Model on flight data, caused hyper-parameter tuned and achieving a maximum accuracy of 85.73%. Result showed that deleting some features affected the value of accuracy and reduced it.

A group of researchers [127] have designed 5 models to predict flight delay based on machine learning models such as Logistic Regression, Decision Tree Regression, Bayesian Ridge, Regression and Gradient Boosting Regression. They collected data

from Bureau of Transportation, U.S. Statistics of all the domestic flights taken in 2015 and predicted whether the arrival of a particular flight would be delayed or not.

The metrics to evaluate the performance of the models were: Mean squared error (MSE), Mean Absolute Error (MAE), Explained Variance Score, Median Absolute Error and R2 Score. Due to the used of imbalanced data sets, the amount of calculated error was high. Based on the results, Random Forest Regressor was observed as the best model in prediction of arrival and departure delay.

One of the newest studies in the area of machine learning method has been presented by a model which applicate supervised learning methods to aggregate flight departure delays in china airports [128]. The expected departure delays in airports was selected as the prediction target while four popular supervised learning methods: multiple linear regression, support vector machine, extremely randomized trees and LightGBM were investigated to improve the predictability and accuracy of the model. Of special note was that the model performances with local weather characteristics was not as good as those without Sustainability meteorological data.

They measured accuracy, MSE and MAE for evaluating 4 methods and result has shown LightGBM model could provide the best result, giving 0.86 accuracy.

A group of Researchers [129] designed a framework to integrate multiple data sources to predict the departure delay of a scheduled flight and discuss the details of the data pipeline. They were the first group, to take advantage of airport situational awareness map, which was defined as airport traffic complexity (ATC), and combined the proposed ATC factors with weather conditions and flight information.

In the first stage, historical data, weather condition data, and tarmac aircraft and vehicles GPS data were collected from different data sources. After that the feature extraction stage, was applied principal component analysis to weather data, and were extracted ATC features from tarmac aircraft and vehicle trajectory data, also utilize the historical scheduling table data. It seems that except for the extracted features more potentially useful features can be explored from the airport situational awareness map. Then in the modelling stage, multiple datasets were combined and various data combinations were used to train a regressor model that could be used for predicting departure delay time.

Authors selected four popular regressors from different families (linear regression, SVR, ANN, and regression trees) to show the robustness of their proposed approach to different regressors. Finally, has been evaluated the prediction results using Root Mean Square Error (RMSE) to measure the performance of flight delay time prediction using different models and different combinations of data sources. Result has shown LightGBM regressor outperforms other conventional regressors with extensive experiments on a large real-world dataset.

Although Other works which have been done in recent years is not in the scope of this article, it is still related to the topic in a way that contributes to the progress of this article, so here we have included studies [130] that employed a support vector machine (SVM) model to explore the non-linear relationship between flight delay outcomes and another model that [131] explored a broader spectrum of factors. This model could potentially affect the flight delay and proposed a gradient boosting decision tree (GBDT) based models for generalized flight delay prediction.

The presented techniques are faced to limitations, because these techniques cannot resist against the massive data volume and complicated computations. For example, in some of these studies, the model is designed based on the specifications and conditions of a special country [43–45, 73, 75, 100, 104]. Some other consider weather conditions in their prediction [38, 132], next group has considered the special situation like en-route [5, 82] or destination [61, 88].

Deep neural networks

Deep neural networks are composed of several hidden layers that each layer has an important role in learning the model [133, 134]. Therefore, actual learning process is repeatedly performed through theses layers [133, 134]. Therefore, it can be inferred that the difference of deep learning techniques from older method is the learning part and lack of limitation in amount of data and also finding the best solution for NP-Complete problems [22]. Deep learning is employed in different areas including speech recognition [135–139], machine vision [30, 140–142], language processing [143–145], recommender systems [146], urban traffic forecasting [147] and air traffic [70, 71, 78, 96]. It is clear that raising the number of variables in forecasting, modeling and simulation results in more precise final model that is achieved using deep neural networks. The remaining part of the section investigates previous studies in flight delay forecasting.

One of the newest studies [24, 25] has been presented which solves problems with massive data volume. This research has designed high precision model for forecasting U.S.A flight delays, which employs Recurrent Deep Neural Network. The research aimed to firstly compute daily delay for each airport and then estimate the delay for a special flight using results of the primary step. This study has used recurrent deep neural network, which stores information of each hidden layer and this matter increases the model performance. Although model has high precision but high model complicacy has led to depth increment and finally takes the model to overfitting state that has been solved using dropout techniques. Moreover, employing this technic, can reduce the computation time and memory space during training. Next challenge is the extremely noisy input data, that the author has neglected in the data during forecasting, which is highly effective in forecasting.

Some research [148] have designed a model to forecast which is based on Bayesian networks and long- and short-term memory (LSTM)that uses discretizing variables like water and air, crowd and airport parameters to compute daily delay for some airports in USA.

This model is composed of three memory layers in network and also uses earlier four days to compute average delay for the destination. Moreover, non-specify or properties are extracted through Mont Carlo Dropout techniques. Although the research has determined a stabilized state between complicacy and overfitting using variable dimension reduction, although it cannot forecast some unique event that highly affect the delay.

Some researchers [132] have investigated the weather conditions and its effect on origin–destination delay and used one of weather underground (WU) protocols related to some variables of wind, that temperature and morning dew. In addition, the following tools including Apache spark, the Analysis service, Elastic tools are used to analyze the data, which Apache spark is a processor for parallel computations and libraries for

machine learning. Statistical findings showed that 89% delays were due to wind. In the next step, they specified the correspondence between variables through dependencies decision tree and associate laws and then they have computed probability of delay occurrence using linear Regression. Moreover, using associate laws, they have proved that next factor in the delay is the humidity. As their researches only have investigated the weather conditions of some airport, so that only 10% of the flights have been postponed. Moreover, their research has not investigated the weather condition during the flight. Therefore, the model can be used only for some specified airports in specific states.

A group of researchers [149] have designed 2 models to forecast delay, one of which was based on long- and short-term memory (LSTM) [148] and the other model was based on Random Forest. In this study to create a dataset, the ground station continuously received automatic dependent surveillance broadcast (ADS-B) messages and then uploaded to central cloud server. After that the weather information of airports, scheduled flight time, departure airport, and destination airport were collected and then were integrated them. The random forest classification architecture was constructed in this model, then the ensemble classifier used the most voted result of the N sub-classifiers as its prediction. The ability of each sub-classifier and the independence of the sub-classifiers jointly improved the model accuracy. Experimental results have shown that Random Forest based method could obtain good performance with the best accuracy was of 90.2 and the LSTM-based architecture can obtain relatively higher training accuracy, but overfitting problem occurred in limited dataset.

One of the newest studies has been presented [150] which could solve problems with high-dimensional data and considered its relationship with space and time. this research was designed high accuracy model for forecasting U.S.A flight delays, which employed Stacked autoencoder. A stacked autoencoder was adopted to train networks and optimizing all the networks' parameters with back propagation method. The model revealed the evolution rule of flight delay in space–time variation and superior after being compared with the performance of traditional neural network. Results from plenty of experiments had implicated that the prediction accuracy with deep stacked autoencoders was above 90%.

In one of the best studies that has been performed based on deep learning a framework was designed which has three parts: command executive, data structure, and utilities [151]. The command executive was described to provide the communication channel between the user and the functions. The information such as flight plan and airport parameters via the data structures were defined as the inputs of the functions. The utilities were known to contain common operations and tools to facilitate the commend executive and data structures. This platform supporting the FAA's Collaborative Decision-Making (CDM) process with the intent of reducing flight delays in the NAS Based on deep learning algorithms and used LTSM to predict accurate arrival and departure delays using time series data. This system at first could integrate various databases to the NextGen's SWIM program framework and then it could predict flight delays. Finally, in this study assessments of risks and sustainability of the proposed platform were presented. Based on the results they demonstrated that this platform can save billions of dollars and millions of hours, respectively but it is not possible to use this framework for everyone.

Some Research [152] combined a deep belief network with a support vector machine to create a prediction model (DBN–SVR), in which DBN extracted the main factors with tangible impacts on flight delays, reduced the dimension of inputs, and eliminated redundant information. The output of DBN was then used as the input of the SVR model to capture the key influential factors (leading to flight delays) and generated the prediction value of delays. They employed a grid-search method to identify the key parameters in SVR and selected the optimal parameter values. After training the DBN–SVR model with proper parameter tuning, have been tried to detect and characterized the key influential factors using the observed DBN. Finally, the prediction performance was described by MAE and RSME. The MSE was finally employed to measure the importance of input factors and detected the key influential factors. Results have shown that air traffic control was one of the key influential factors. Also, there was a strong relationship between the average delay of current and previous flights during 16:00–22:59, so that delays occurring in the afternoon and evening flights have a higher possibility of propagating and affecting the subsequent flights.

Some research [153] was carried out by employing quantitative research method. Author focused mainly on predicting airlines flight delays by analyzing flight data, especially, for the domestic Airlines that moves around the United States of America. The main aim of the study was to reduce the number of data dimension before feeding it to the deep learning network. The primary dataset was filtered first from more than 100 feature to one third of it. According to this study, before deep learning model implementing, dataset need to divide into train and test sets. Train set was divided randomly 80%, while the test set contained 20% of the whole data. Train set was used to train the deep learning model. Where test set was used to check the accuracy by using confusion matrix performance measures.

Author used mainstream classification machine learning and deep neural networks to classify whether a flight would be delayed or not. For the machine learning algorithm, Decision Tree was used while for deep neural network as the name stands Deep Artificial Neural network (DANN) was used. They showed that the accuracy of DANN was slightly higher than the Decision Tree, however, even a tiny difference in accuracy was believed to be of tremendous valuable since the dataset was enormous and number of flights per day is numerous.

Based on the results of this study, with the reduced number of features the accuracy did not change. Also, the best accuracy was 82.10%. Therefore, several experiments had been carried out with the same setup with different number of neurons and hidden layers. Surprisingly, there was no clear differences in accuracy rate. But when the number of hidden layers increased then the accuracy was 81.80%. So, it can be concluded that number of increased hidden layer did not ensure with higher accuracy.

According to the recommended structure [24, 25], one of the recent studies in this area still has some problems such as overfitting or memory space shortage. Moreover, data noise is neglected. These problems are effective in model forecasting precision.

Methodology

In this section, we issues to represent a our technique in which we tried to solve the problems related to massive data, processing complications [21, 24, 25, 28], lack of computational space, overfitting and existing noise in data [24, 25]. Figure 1 gives an

illustration of the development of proposed model. As can be seen from the figure the proposed technique contains three phases. We descript most important notation in the Table 1.

First phase: data collecting and pre-processing

Firstly, and at the beginning of the phase, it is necessary that model inputs be determined so that based on them, model learn and result in final structure. The dataset used for evaluating the model was obtained from historical data which contains flight schedules data for 5 years. Variables which are used as inputs are shown in Table 2. It is applied to real-world data collected from the airports in the U.S and is compared with existing flight delay predictors.
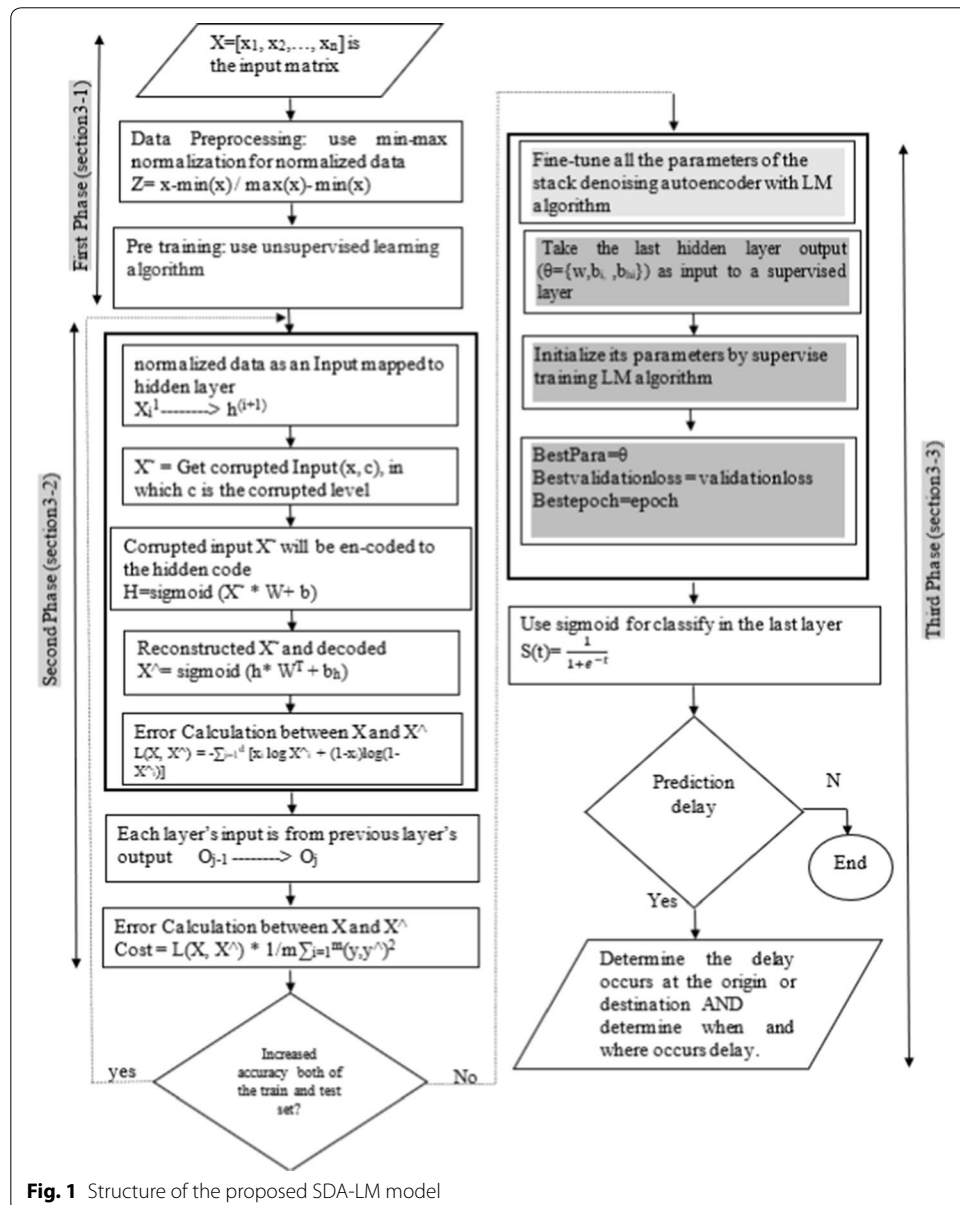

Fig. 1 Structure of the proposed SDA-LM model

Table1 Description of the most important notations

Notation	Description	Notation	Description
X	Input matrix	$\hat{X}$	Reconstruct-ed $\tilde{X}$
Min(x)	Lowest value in series and its number is zero	W^T	Transposition of the weight matrix W
Max(x)	Highest number in series and has value of 1	b_h	Show the bias associated with each hidden code
X_i^1	Normalized data in first layer	$L(X, \tilde{X})$	Reconstruct-ion error rate
h	Hidden layer	cost	Error rate
$\tilde{X}$	Corrupted input	W	Weighted matrix
c	Corruption level	y	Output per x
H	Activation function	$\hat{y}$	Output per $\hat{X}$
$O_{j-1} \rightarrow O_j$	Each layer's input is from previous layer's output	θ	Is the parameters of the denoising autoen-coder
b	Bias vector		

Table 2 Variables names

Variables related to delay time	Variables related to delay reasons	Variables related to flight information	Variables related to flight timing
DepDel15	CarrierDelay	Flight Number	Year
DepTime	WeatherDelay	Origin	Month
DepDelay	NASDelay	Destination	Dayofweak
ArrDelay	SecurityDelay		DayofMonth
ArrDel15	LateAircraft		FlightDate
ArrTime			

After collecting data, characteristics enters system as X vectors which contains all variables in form of $X = \{X_1 \cdot X_2 \cdot \ldots \cdot X_{19}\}$. In this model each X_i represents a single characteristic. Since these characteristic's adjustment range has lots of oscillation and no accordance to each other, pre-processing must be operated on the database. Thus, we look for normalization techniques and among them, we use 'min max' normalization one. This technique is mostly known as Feature Scaling in which Eq. 1 is used for each variable normalization.

$$Z_i = X_i - \min(x_i) / \max(x_i) - \min(x_i) \tag{1}$$

In (1) X_i represents each variable and min(x) shows the lowest value in series and its number is zero. Max(x) represents highest number in series and has value of 1. Of all variables, ones related to time and flight information is normalized based on Eq. 1 and Fig. 2 shows min–max algorithm. Delay is calculated using timing difference of fields ArrTime and ArrDelay in beginning and DepTime and DepDelay in destination and if flight delay is more than fifteen minutes, values of DepDel15 and ArrDel15 fields in that flight turns 1, otherwise it turns 0. Also, flights delay due to various reasons which in database are divided into five general categories: CarrierDelay, WeatherDelay, NAS-Delay, SecurityDelay and LateAircraft which value of 1 in these fields determines flight delay cause or causes. WeatherDelay field is weather conditions related where weather

```
Algorithm 1 min- max normalization
1:    Import numpy as np
2:    def normalize(X) :
3:        min = np.min(x)
4:        max = np.max(x)
5:        range = max – min
6:        returne  [(a-min) / range  for  a  in  x]
```

Fig. 2 The step of min–max algorithm

```
Algorithm 2 Denoising Autoencoder Training
1:  procedure DA TRAINING(e,x,c,I,θ)
2:      x = [x₁, x₂, …, x₁₉]∈ Rⁿ·ᵐ is the input matrix, in which xᵢ∈ [0,1]ᵐ (1≤i≤m) is a single input data
3:      e is the number of epochs to be iterated
4:      I is the learning rate
5:      c is the corruption level
6:      θ= {W , b , bₕ} where W∈ Rⁿ·ᵈ , b ∈ Rᵈ , bₕ ∈ Rᵈ , θ is the parameters of a DA
7:      for  0 to e do
8:      X⁻ = Get corrupted Input(x,c), in which c is the corrupted level
9:      H=sigmoid (X⁻ * W+ b)
10:     X^= sigmoid (h* Wᵀ + bₕ)
11:     L(X, X^) = -∑ⱼ₌₁ᵈ [xᵢ log x^ᵢ + (1-xᵢ)log(1- x^ᵢ)]
12:     Cost = L(X, X^) * 1/m∑ⱼ₌₁ᵐ(y,y^)²
13:     if  the accuracy both of the train and testing set is increasing then
14:     θi= update θi
15:     end for
16:     end procedure
```

Fig. 3 Training algorithm of a denoising autoencoder

information is provided by National Oceanic and based on database reports, flight time could change or face delay. If delay occurs due to bad weather, field value changes to 1. Generally, if a flight is delayed due to any reasons, value of CarrierDelay, WeatherDelay, NASDelay, SecurityDelay and LateAircraft field or fields related to delay cause changes to 1.

Second phase: pre-training model- building stack denoising auto encoder (SDA)

After pre-processing phase, second phase initiates, in which model enters pre-training phase that the training algorithm of a denoising autoencoder is summarized in Fig. 3. Normalized variables enter first denoising autoencoder as inputs and is mapped to first hidden layer in form of $X_i^1 \rightarrow h^{(i+1)}$ [154]. Then, some of characteristics of X vectors inputs will be decayed randomly by rate of c. There are different methods to decay data and, in this study, used zero mask, meaning we change the value of those variables to zero and organize vector $\tilde{X}$. Therefore, encrypting phase begins and $\tilde{X}$ vector is encrypted in hidden code **H** and its value is calculated based on Eq. 2 [154].

$$H = \text{sigmoid} \left(\tilde{X} * W + b \right) \tag{2}$$

W represents variable's Weight and **b** represents its Bias. When an input enters a neuron, it's categorized by a Weight. In addition to Weight, another linear component that

affects input is called Bias and its value is added to Weight multiplier in input in order to change the range of resulted value from Weight multiplier in input. Bias is the last linear component which assists input conversion. The initial Weight and Bias is randomly assigned and are updated during training process. After training process beginning, Neural network assign more Weight to inputs that it considers more important. Having Weight of 0, show Ineffective variable. After encrypting phase, $\tilde{X}$ vector is reconstructed based on Eq. 3 and using hidden code H, resulting in $\hat{X}$ vector. This phase is well known as decoding phase. $\hat{X}$ vector is transferred into output [154].

$$\hat{X} = \text{sigmoid}(h * W^T + b_h) \tag{3}$$

$\mathbf{W^T}$ show transposition of the weight matrix w and $\mathbf{b_h}$ show the bias associated with each hidden code, After the decryption process is completed, the reconstruction error rate [155] of $\hat{X}$ vector is calculated based on Eq. 4.

$$L\left(X, \tilde{X}\right) = -\sum_{i=1}^{d} [x_i \log \tilde{x}_i + (1 - x_i) \log (1 - \tilde{x}_i)] \tag{4}$$

One denoising autoencoder is formed in this phase. Therefore, using Cost Function, error rate could be estimated based on Eq. 5, which means measuring difference between real inputs and reconstructed inputs. Precision rate of each coded unit is determined by Cost Function. Minimizing the amount of difference between real input and reconstructed input, is the goal here. Next, model parameters are randomly initialized and then optimized using gradient descend algorithm. The best value of $\hat{X}$ vector, is the one that costs the least. $\tilde{X}$ vector is forced to have smarter mapped than X vector, so that in situation where there's lots of noise, this method is able to extract useful characteristics and remove their noise while reconstructing.

$$\text{Cost} = L(X, \tilde{X}) * 1/m \sum_{i=1}^{m} (y, \hat{y}) \tag{5}$$

Cost Function tries to penalize the network whenever it makes a mistake. After establishing network, foresight precision must increase while error rate decreases. The most optimized output, is the one that costs the least. In order to increase network's learning ability and decrease its error, numbers of denoising autoencoder must be increased. Figure 4 shows training algorithm of stack denoising autoencoder.

In fact, each autoencoder represents a hidden layer containing a few hidden units in which encryption, decoding, weight determination and bias operations takes place and finally, $\hat{X}$ vector is output of each hidden unit. After adding a denoising autoencoder to the network, previous hidden layer information is transferred as an input to this layer and non-linear transmission among consecutive hidden layers cause learning of structure and next, the resulted network could foresee flight delay. Therefore, with computing cost.

Function, rate of error between real output and predicted output can be computed. Finally, assigning the network to the two sets of training and testing, in case forecast accuracy are increasing in both series, a denoising auto encoder will be added to network again. Otherwise, if training accuracy increase while test set decreases, it shows

Algorithm 3 Unsupervised PreTrain SDA

1: **procedure** PRETRAINING(e,x,c,I,θ)
2:    $X = [x_1, x_2, ..., x_n] \in R^{n*m}$ is the input matrix, in which $x_i \in [0,1]^m (1 \le i \le m)$ is a single input data
3:    I *is* the learning rate
4:    $h = [h_1, h_2, ..., h_z] \in z^I$, where h_i is the number of hidden units in layer i and I is the number of hidden layers
5:    $c = [c_1, c_2, ..., c_z] \in (0,1)^z$, in which c_i is the corruption level of the input of the hidden layer i.
6:    $\Theta = [\theta_1, \theta_2, ..., \theta_z]$ where $\theta_i = \{W_i , b_i , b_{hi}\}$
7:    $O = [O_1, O_2, ..., O_I]$ is the output of each hidden layer, where $O_i = [o_{i,1}, o_{i,2}, ..., o_{i,n}] \in R^{n*hi}$ $(0 < i < 1)$
8:    θ_1 = DA TRAINING (e,x,c,I, θ_1)
9:    **for** i from 1 to n do
10:    $o_{i,1}$ = sigma (x_i* W_1 + b_i)
11:    end for
12:    **for** j from 2 to I do
13:    θ_j = DA TRAINING (e, $O_{j-1,i}$,c,I, θ_j)
14:     **for** i from 0 to n do
15:    $o_{j,i} = \sigma(o_{j-1,i}$* W_j+ b_j)
16:    end for
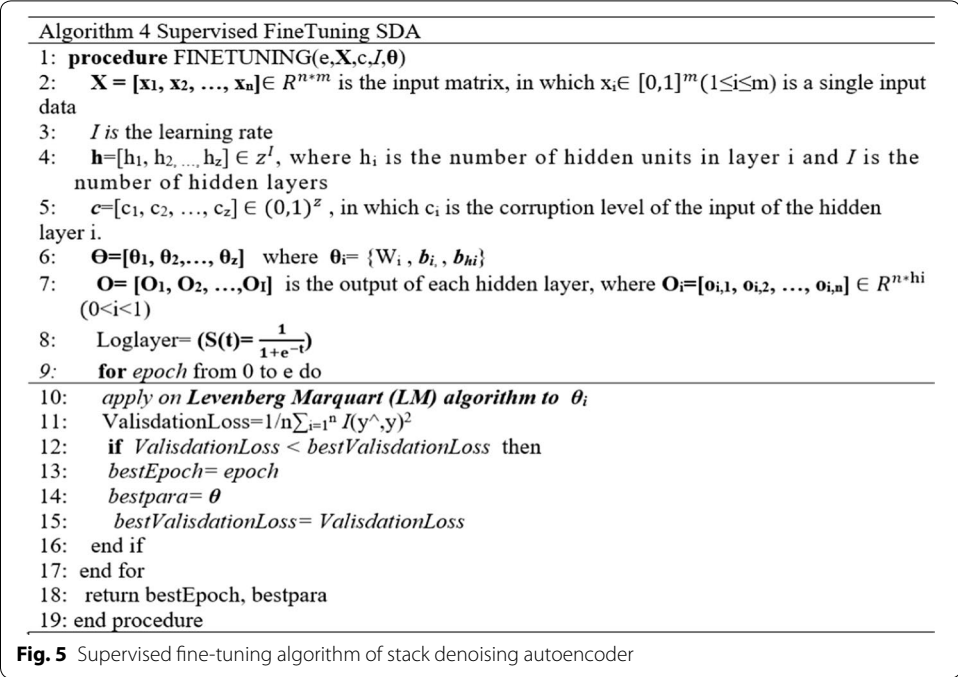17:   end for
18:    end procedure

Fig. 4 Training algorithm of a stack denoising autoencoder

that training series has estimated the noise in data and learned the noise-related behavior. Therefore, denoising autoencoder addition operation is ended and stack of denoising auto encoder structure [154] is finally formed against noise.

Third phase: model optimization with Levenberg-Marquart (LM) algorithm

The third phase's goal is model optimization. Figure 5 show supervised fine-tuning algorithm of stack denoising autoencoder. When a network is formed, Weight and Bias values are distributed randomly among the nodes. After determining the output, with its help, network error could be computed and then return the value along with Cost Function chart back to the network to update network's Weights. These Weights are updated in the way that decreases similar errors. This action is called back-propagation. In back-propagation, network's movement is backwards, errors and charts return to the hidden layer so that Weights are updated based on them.

The last hidden layer's output is taken as input to a supervised learning algorithm to fine-tune all the parameters of this deep architecture with respect to the supervised criterion [156]. In this phase parameters are fine-tuned, we use the LM [157] on top of the whole network to train the input generated by the last autoencoder. The LM Algorithm can provide a numerical solution to the nonlinear problem minimizing a function over a space of the function parameters [158] and also it is stable and can generate good convergence [157, 159]. LM algorithm has benefits of gradient descent and Gauss–Newton methods at the same time and is created from linear combination gradient descent and Gauss–Newton based on adaptive rules [160]. This algorithm interpolates between gradient descent and Gauss–Newton and in most cases, it finds an answer, even if it started off from farthest final minimum. This algorithm is stronger than Gauss–Newton but in some occasions where initial parameters are logical and function behavior is compatible, it's a little slower than Gauss–Newton. It's also one of the most popular curve

```
Algorithm 4 Supervised FineTuning SDA
 1: procedure FINETUNING(e,X,c,I,θ)
 2:    X = [x₁, x₂, …, xₙ]∈ Rⁿ*ᵐ is the input matrix, in which xᵢ∈ [0,1]ᵐ (1≤i≤m) is a single input
       data
 3:    I is the learning rate
 4:    h=[h₁, h₂, …, h_z] ∈ z^I, where hᵢ is the number of hidden units in layer i and I is the
       number of hidden layers
 5:    c=[c₁, c₂, …, c_z] ∈ (0,1)^z , in which cᵢ is the corruption level of the input of the hidden
       layer i.
 6:    Θ=[θ₁, θ₂,…, θ_z]   where  θᵢ= {Wᵢ , bᵢ, , b_hi}
 7:    O= [O₁, O₂, …,O_I]  is the output of each hidden layer, where Oᵢ=[oᵢ,₁, oᵢ,₂, …, oᵢ,ₙ] ∈ Rⁿ*ʰⁱ
       (0<i<1)
 8:    Loglayer= (S(t)= 1/(1+e⁻ᵗ))
 9:    for epoch from 0 to e do
10:      apply on Levenberg Marquart (LM) algorithm to θᵢ
11:      ValisdationLoss=1/n∑ᵢ₌₁ⁿ I(y^,y)²
12:      if ValisdationLoss < bestValisdationLoss then
13:        bestEpoch= epoch
14:        bestpara= θ
15:        bestValisdationLoss= ValisdationLoss
16:      end if
17:   end for
18:   return bestEpoch, bestpara
19: end procedure
```

Fig. 5 Supervised fine-tuning algorithm of stack denoising autoencoder

fitting algorithms that its main usage is in least squares [161]. This algorithm has two main phases [162]; Computing the Jacobian Matrix that is the most complicated part of this algorithm, and calculate the Hessian matrix and updating Weights which Network's error is computed in this phase. According to the update rule, if the error goes down, which means it is smaller than the last error, it implies that the quadratic approximation on total error function is working and the combination coefficient μ could be changed smaller to reduce the influence of gradient descent part (ready to speed up). On the other hand, if the error goes up, which means it's larger than the last error, it shows that it's necessary to follow the gradient more to look for a proper curvature for quadratic approximation and the combination coefficient μ is increased.

After learning is finished by LM algorithm, it's time for choosing activation Function for the last layer so that it could foresee precisely. At the last layer, a logistic sigmoid function is used because the final output should be a binary class which is 0 and 1. After determine optimized values for weight and bias, we expect network's foresight improve and the proposed model get as close as possible to reality. After prediction delay for an airport, the delay cause and whether it was a delay in the source or the destination will be determined.

Results and discussions

The model is designed using Python in Tensor flow and is installed on a system of 40 core CPU at a frequency of 2.6 hz, 80 G RAM and 250 G Hard. The flight info data is an open dataset collected by the Bureau of Transportation Statistics of United State Department of Transportation [163] where, the reason for delay is due to canceled or flight delay, and time duration of each flight. Model testing and training employs these data that include 18 million records.

Model, uses 80% of data for training and the remaining 20% for testing [164]. Finally, the model evaluation considers two analysis which are studied in the following section.

First analysis

In order to evaluate the model, the number of denoising autoencoders and neurons must be determined based on the values for precision, accuracy and time consuming. In order to do this, at first, the model is trained using one stack and 64 neurons, and the precision and accuracy values are calculated. By adding another denoising autoencoder, the values for precision and accuracy are increased; therefore, another stack was added to the model's structure. On the other hand, by adding each stack denoising autoencoder to the structure, the processing time is also risen. Therefore, denoising autoencoder increment process should consider excellence between processing time and number of denoising autoencoder. As a result, adding denoising autoencoder addition is continued until differences of precision and accuracy for previous and newer structure exceed the threshold limit. Figure 6 shows the amount of accuracy based on number of denoising autoencoders and computation time.

After determining the number of stacks denoising autoencoders, it is time to determine the number of neurons. By increasing the number of neurons in each hidden layer, the values for precision and accuracy for both the training and testing sets are evaluated. When the number of neurons increases from 16 to 32 and from 32 to 64, the values for precision and accuracy increased for both datasets; however, by increasing the number of neurons from 64 to 128, the precision and accuracy of the model increased in the training set while they decreased in the testing set. Therefore, increasing the number of neurons was also stopped. The final structure is created with 3 stack denoising autoencoders, 64 neurons and 4 hidden layers.

Second analysis

The data classified in two classes of 0 and 1. The data in Class 0 include 15 million records for non-delayed flights and the data in Class 1 include 3 million records of delayed flights. Due to the imbalance of the datasets, the model was trained by the imbalanced and balanced datasets separately, and then the effects of each mode on the evaluation parameters were evaluated separately. In order to create balance in the dataset, we
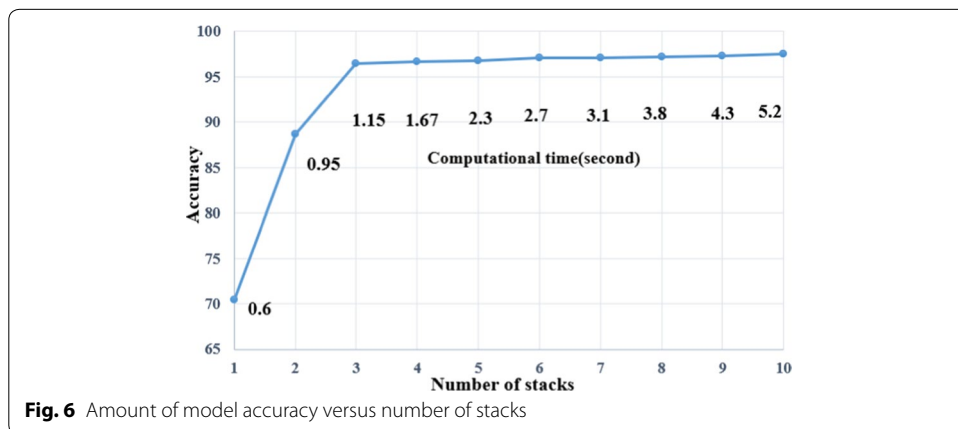

Fig. 6 Amount of model accuracy versus number of stacks

Table 3 Presents some parameters for evaluation of model which are studied in the evaluation

	Predicted delay	Predicted non-delay
Actual Delay	True Positive (TP) = number of behaviors that represent delay and is forecasted by model	False Positive (FN) = number of behaviors that represent delay and the model has incorrectly predicted no delay
Actual Non-Delay	False Negative (FP) = Number of behaviors that show no delay and model incorrectly predicted as a delay	True Negative (TN) = Number of behaviors that show no delay and is forecasted by model

Table 4 Computation method precision, recall, specificity, F1 measure and accuracy

Parameter	Precision	Recall	Specificity	F1 measure	Accuracy
Formula	$\frac{TP}{TP+Fp}$	$\frac{TP}{TP+FN}$	$\frac{TN}{TN+FN}$	$\frac{2*Recall*Precision}{Recall+precision}$	$\frac{TP+TN}{TP+TN+FN+Fp}$

have to use sampling methods; undersampling and upsampling are two famous sampling methods [165]. In undersampling method, it is required to class zero data to 3 million and increase the classes to 15 million in upsampling. Whenever the upsampling method is used it is required to create 12 million chaos data that cause increment in processing time, reduction in process velocity model overfitting and finally leads to lower confidence of the model, so that it is required to use undersampling method. The proposed operation is measured by confusion Table 3 [166]. Each column of the table shows predefined samples.

These four criterions in the confusion table show the essence for quality of algorithms that perform forecasting. Table 4 shows how to solve evaluation problems such as precision, accuracy, sensitivity, recall and F-measure [166].

Moreover, in each measurement there are several micro and macro averages that are slightly different. Macro Average measurement is computed for each class and then their average is equally computed by considering all classes, while in computing average measurement for micro average adds share of all categories and finally weight average performs averaging according to amount of data in each class.

Table 5 shows how to solve evaluation problems such as micro avg, macro averages and weighted avg [167]. In addition, it is assumed that the delay means true Possible delay, and it is expected the proposed method has greater precision and accuracy in comparison to previous methods.

In order to study effect of stack denoising autoencoder and LM algorithm on the model structure, two other structures are also designed. First structure is based on autoencoder and LM algorithm (SAE-LM), and the second structure is based on denoising autoencoder only (SDA). The first stage, trains three imbalanced model and the results of comparison is represented in Table 6.

Afterwards, in order to study the effect of balanced dataset on evaluation parameters, trains three balanced model and the results of comparison is represented in Table 7.

As it is shown in Table 7, balanced dataset has increased all values for evaluation. Moreover, all the evaluation parameters of the proposed model have increased over models. Therefore, effect of stack denoising autoencoder on noisy data is positive and

Table 5 Computation method micro avg, macro avg and weighted avg

Parameter	Micro avg	Macro avg	Weighted avg
Precision	$\dfrac{\sum_{c=1}^{4} TP_c}{\sum_{c=1}^{4} TP_c + \sum_{c=1}^{4} FP_c}$	$\dfrac{1}{4}\sum_{c=1}^{4}\dfrac{TP_c}{TP_c + FP_c}$	$\dfrac{\sum_{c=1}^{4} T_c P_c}{\sum_{c=1}^{4} T_c}$
Recall	$\dfrac{\sum_{c=1}^{4} TP_c}{\sum_{c=1}^{4} TP_c + \sum_{c=1}^{4} FN_c}$	$\dfrac{1}{4}\sum_{c=1}^{4}\dfrac{TP_c}{TP_c + FN_c}$	$\dfrac{\sum_{c=1}^{4} T_c R_c}{\sum_{c=1}^{4} T_c}$
Specificity	$\dfrac{\sum_{c=1}^{4} TN_c}{\sum_{c=1}^{4} TN_c + \sum_{c=1}^{4} FN_c}$	$\dfrac{1}{4}\sum_{c=1}^{4}\dfrac{TN_c}{TN_c + FN_c}$	$\dfrac{\sum_{c=1}^{4} T_c S_c}{\sum_{c=1}^{4} T_c}$
Accuracy	$\dfrac{\sum_{c=1}^{4} TP_c + \sum_{c=1}^{4} TN_c}{\sum_{c=1}^{4} TP_c + \sum_{c=1}^{4} FP_c + \sum_{c=1}^{4} TN_c + \sum_{c=1}^{4} FN_c}$	$\dfrac{1}{4}\sum_{c=1}^{4}\dfrac{TP_c + TN_c}{TP_c + FP_c + TN_c + FN_c}$	$\dfrac{\sum_{c=1}^{4} T_c A_c}{\sum_{c=1}^{4} T_c}$
F1 measure	$\dfrac{2*Recall*Precision}{Recall + precision}$	$\dfrac{2*Recall*Precision}{Recall + precision}$	$\dfrac{\sum_{c=1}^{4} T_c F1_c}{\sum_{c=1}^{4} T_c}$

Table 6 Comparison between SDA-LM, SAE-LM and SDA structures on imbalanced dataset

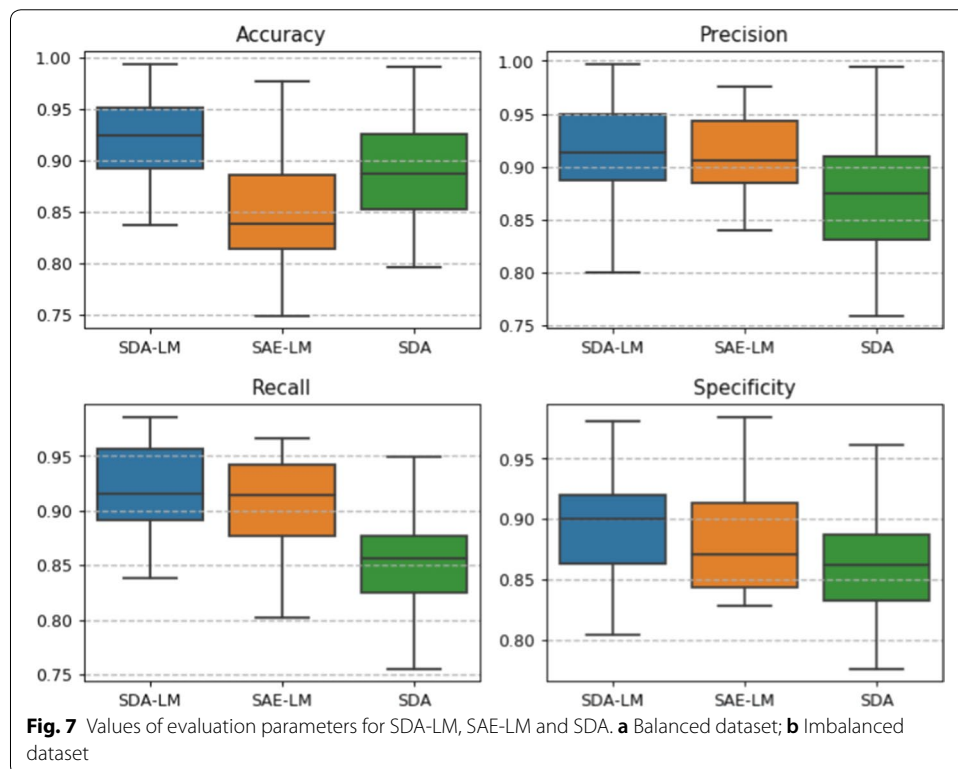|	Accuracy	Precision	Recall	Specificity	F1 measure	Support
SDA-LM model						
Class 0	**0.92**	0.98	0.88	0.89	0.92	2888640
Class1	**0.92**	0.74	0.89	0.88	0.84	713039
Micavg	**0.92**	0.88	0.88	0.89	0.88	3601679
Macavg	**0.92**	0.89	0.88	0.88	0.83	3601679
Weighted Avg	**0.92**	0.90	0.88	0.88	0.89	3601679
SAE-LM model						
Class 0	0.83	0.92	0.86	0.68	0.89	2888640
Class1	0.83	0.55	0.68	0.86	0.61	713039
Micavg	0.83	0.83	0.83	0.74	0.83	3601679
Macavg	0.83	0.73	0.77	0.77	0.75	3601679
Weighted Avg	0.83	0.84	0.83	0.75	0.83	3601679
SAE-LM model						
Class 0	0.80	0.93	0.82	0.73	0.87	2888640
Class1	0.80	0.50	0.74	0.82	0.60	713039
Micavg	0.80	0.80	0.80	0.78	0.80	3601679
Macavg	0.80	0.72	0.78	0.77	0.73	3601679
Weighted Avg	0.80	0.84	0.80	0.78	0.82	3601679

increase in precision and accuracy in the structure. On the other hand, SDA model shows that optimization through LM algorithm is suitable for solving non-linear problems and achieving a stable model with good degree of convergence. Figure 7 shows the evaluation parameters for three structures; SDA-LM, SAE-LM and SDA.

Moreover, proposed model accuracy with imbalanced dataset had increased 4.1% compared to maximum accuracy in the previous model which is based on RNN [24]. This value also is approached to 8% after balancing the dataset. In Fig. 8, the accuracy of the SDA-LM, SAE-LM and SDA structures is compared with the structure of RNN [24, 25]. As shown in Fig. 8, the accuracy of the proposed model is increased relative to the accuracy of the previous model which is based on RNN [24].

Finally, accuracy of the proposed prediction model is compared to other previous prediction methods. As you can see in Table 8, the accuracy of the proposed model is higher than other methods.

Yazdi *et al. J Big Data*   (2020) 7:106

Page 19 of 28

Table 7 Comparison between SDA-LM, SAE-LM and SDA structures on balanced dataset

	Accuracy	Precision	Recall	Specificity	F1 measure	Support
SDA-LM model						
Class 0	**0.96**	0.96	0.93	0.84	0.94	2888640
Class1	**0.96**	0.84	0.84	0.93	0.89	713039
Micavg	**0.96**	0.95	0.91	0.88	0.93	3601679
Macavg	**0.96**	0.95	0.90	0.88	0.92	3601679
Weighted Avg	**0.96**	0.94	0.92	0.88	0.93	3601679
SAE-LM model						
Class 0	0.86	0.96	0.90	0.89	0.93	2888640
Class1	0.86	068	0.85	0.83	0.76	713039
Micavg	0.86	0.89	0.89	0.86	0.89	3601679
Macavg	0.86	0.82	0.88	0.87	0.85	3601679
Weighted Avg	0.86	0.91	0.89	0.87	0.90	3601679
SDA model						
Class 0	0.89	0.96	0.83	0.85	0.89	2888640
Class1	0.89	0.57	0.89	0.89	0.69	713039
Micavg	0.89	0.84	0.84	0.87	0.84	3601679
Macavg	0.89	0.77	0.86	0.87	0.79	3601679
Weighted Avg	0.89	0.89	0.84	0.87	0.85	3601679

Fig. 7 Values of evaluation parameters for SDA-LM, SAE-LM and SDA. **a** Balanced dataset; **b** Imbalanced dataset

At the end, for evaluate the validity of the proposed model and the results from training, we evaluate the standard deviation of all the parameters after the 30 times repetition. The smaller the standard deviation of the data That is, the data are closer

Fig. 8 Comparison of the accuracy value between SDA-LM, SAE-LM, SDA and RNN methods. **a** Balanced dataset, **b** imbalanced dataset
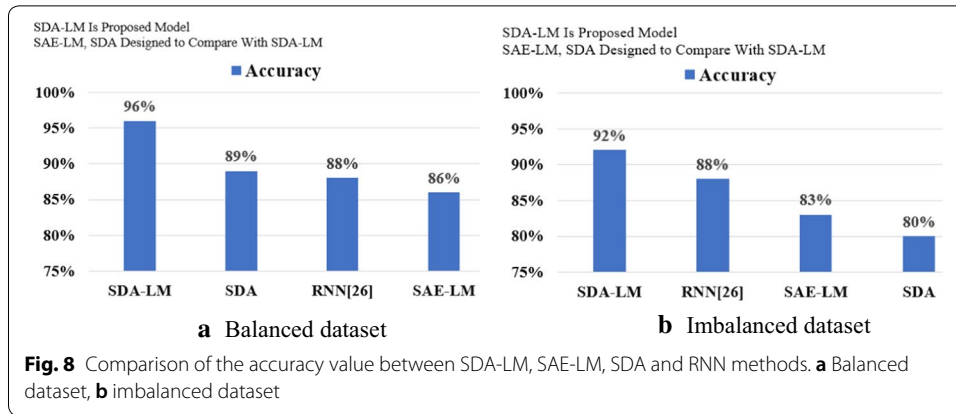
Table 8 Comparison of the accuracy value between of SDA-LM, SAE-LM, SDA and other methods

Method	Accuracy	
Stack Denoising Autoencoder- Levenberg Marquart	**Balanced dataset**	**Imbalanced dataset**
	96%	**92%**
Random Forest [149]	90.2%	–
Stack Autoencoder [151]	–	90%
Stack Denoising Autoencoder	89%	80%
Recurrent Neural Network [24]	–	88%
Stack Autoencoder- Levenberg Marquart	86%	83%
Supervised Learning Methods [128]	–	86%
Data mining + Machine Learning [126]	85.73%	–
Deep Belief Neural Network + Decision Tree [154]	82.10%	–
Cat-boost Model [125]	80.44%	–

Table 9 Standard deviation calculated for all evaluation parameters for the three structures using the imbalanced dataset

model	Accuracy	Precision	Recall	Specificity	F1 measure
SDA-LM	**0.045**	0.042	0.036	0.055	0.048
SAE-LM	**0.051**	0.078	0.062	0.063	0.064
SDA	**0.053**	0.084	0.071	0.067	0.082

to the average and the results are less scattered and therefore more reliable Tables 9 and 10 show the standard deviation of the evaluation parameters using imbalanced and balanced datasets for the three structures of the current study, respectively.

As can be seen from Table 9, the standard deviation for all the evaluation parameters is a small number, and using the balanced dataset, this value is reduced further. Therefore, the balanced dataset has a positive impact on the standard deviation and reduces it, as shown in Table 10.

Table 10 Standard deviation calculated for all evaluation parameters for the three structures using the balanced dataset

Model	Accuracy	Precision	Recall	Specificity	F1 measure
SDA-LM	**0.021**	0.013	0.021	0.021	0.016
SAE-LM	**0.065**	0.032	0.069	0.052	0.048
SDA	**0.058**	0.067	0.046	0.053	0.054

Conclusion

Predicting flight delays is on interesting research topic and required many attentions these years. Majority of research have tried to develop and expand their models in order to increase the precision and accuracy of predicting flight delays. Since the issue of flights being on-time is very important, flight delay prediction models must have high precision and accuracy. In this study, we proposed a novel optimized forecasting model based on deep learning which engages LM algorithm. Afterwards, two other structures are created to study and validate the positive effect of denoising autoencoder and LM algorithm, which one has deleted denoising autoencoder and the other has omitted LM algorithm. Moreover, we have imbalanced dataset which should be balanced. We used undersampling and upsampling technique to balance the data. However, results show that upsampling leads to overfitting. Therefore, under sampling is used for balancing.

Comparing the three models for two of imbalanced and balanced datasets shows that accuracy of SDA-LM model with imbalanced dataset respectively is greater by 8.2 and 11.3% Than SAE-LM and SDA models. On the other hand, these values for balanced datasets are respectively as 10.4 and 7.3%. Therefore, using stack denoising autoencoder and LM algorithm in optimizing the results, and also balancing the dataset, has positive effect on delay forecasting and leads to increment in accuracy and precision of SDA-LM model with imbalanced dataset is greater by 6.1 and 5.4% than SAE-LM and SDA models. Whereas, the accuracy of the SDA-LM model with balanced dataset is greater by 10% than SAE-LM and SDA models and the amount of precision is the same for all three models with balance dataset.

At the next stage, the model has been evaluated and computed for subjects of discarding with a standard deviation for all evaluation parameters during 30 times of model run. The results, shows that standard deviation for all balanced evaluation parameters is lower than the imbalanced form. Therefore, data balance leads to lower standard deviation. amount of model standard deviation for imbalanced dataset is 0.045 while this value is reported 0.21 for balanced dataset which is a small value and means that scattering results are low and close to average.

Finally, we compared the accuracy of the proposed Model against SAE-LM, SDA and RNN [24, 25] models. Using our experimental results, we show that accuracy of the model on imbalanced dataset is 92.1% and for balanced dataset is 96.2%, which is respectively greater by 4.1 and 8.2% respectively. Therefore by proposed model has greater accuracy in forecasting flight delay compared to previous model called RNN [24, 25]. The next step would be to apply this technique on other data sets or on other sampling data and investigate the accuracy.

Author details
[1] Department of Computer Engineering, Neyshabur Branch, Islamic Azad University, Neyshabur, Iran. [2] Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran.

Appendix

Runway: a paved strip of ground on a landing field for the landing and takeoff of aircraft.

En-route: In aviation, an en-route chart is an aeronautical chart that guides pilots flying under instrument flight rules (IFR) during the en-route phase of flight.

Random Forest: It is an ensemble learning method, which uses decision tree as sub-classifiers, and introduces random attributes selection into the decision tree.

LSTM: LSTM network is one of most powerful RNNs with more complex cell structure, and overcomes the gradient vanishing problem in RNNs.

Autoencoder: An autoencoder is a type of artificial neural network used to learn efficient data coding in an unsupervised manner. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal "noise".

Denoising autoencoder: Denoising autoencoders are an extension of the basic autoencoder, and represent a stochastic version of it. Denoising autoencoders attempt to address identity-function risk by randomly corrupting input (i.e. introducing noise) that the autoencoder must then reconstruct, or denoise.

Weight: Weights in an ANN are the most important factor in converting an input to impact the output. This is similar to slope in linear regression, where a weight is multiplied to the input to add up to form the output. Weights are numerical parameters which determine how strongly each of the neurons affects the other.

Bias: is the conflict in trying to simultaneously minimize these two sources of error that prevent supervised learning algorithms from generalizing beyond their training.

Cost function: A cost function is a measure of "how good" a neural network did with respect to it is given training sample and the expected output. It also may depend on variables such as weights and biases.

Activation function: An activation function determines the output behavior of each node, or "neuron" in an artificial neural network.

Overfitting: A model overfits the training data when it describes features that arise from noise or variance in the data, rather than the underlying distribution from which the data were drawn. Overfitting usually leads to loss of accuracy on out-of-sample data.

Dropout: Dropout changed the concept of learning all the weights together to learning a fraction of the weights in the network in each training iteration.

Epoch: in neural networks generally, an epoch is a single pass through the full training set.

Supervised learning: Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input–output pairs.

Unsupervised learning: Unsupervised learning is a type of machine learning that looks for previously undetected patterns in a data set with no pre-existing labels and with a minimum of human supervision.

Fine-tune: Fine tuning is a process to take a network model that has already been trained for a given task, and make it perform a second similar task.

Precision: precision is the ration of system generated results the correctly predicted positive observations (True Positive) to the system's total predicted positive observations, both correct (True positive) and incorrect (False Positives).

Recall: Recall is the ratio of system generated results that correctly predicted positive observations (True positives) to all observations in the actual malignant class (Actual positives).

Accuracy: Accuracy is the most intuitive performance measure and is simply a ratio of the correctly predicted classifications (both True Positives + True Negatives) to the total Test Dataset.

Fi measure: the F1 Score is the weighted average (or harmonic mean) of Precision and Recall. Therefore, this score takes both False Positives and False Negatives into account to strike a balance between precision and Recall.

Specificity: Specificity (also called the true negative rate) measures the proportion of actual negatives that are correctly identified as such (e.g., the percentage of healthy people who are correctly identified as not having the condition).

References
1. Rebollo JJ, Balakrishnan H. Characterization and prediction of air traffic delays. Transportation Res Part C Emerg Technol. 2014;44:231–41.
2. Thiagarajan B, et al. A machine learning approach for prediction of on-time performance of flights. In 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC). New York: IEEE. 2017.
3. Reynolds-Feighan AJ, Button KJ. An assessment of the capacity and congestion levels at European airports. J Air Transp Manag. 1999;5(3):113–34.
4. Hunter G, Boisvert B, Ramamoorthy K. Advanced national airspace traffic flow management simulation experiments and vlidation. In 2007 Winter Simulation Conference. New York: IEEE. 2007.
5. AhmadBeygi S, et al. Analysis of the potential for delay propagation in passenger airline networks. J Air Transp Manag. 2008;14(5):221–36.
6. Liu YJ, Cao WD, Ma S. Estimation of arrival flight delay and delay propagation in a busy hub-airport. In 2008 Fourth International Conference on Natural Computation. New York: IEEE. 2008.

7.   Tu Y, Ball MO, Jank WS. Estimating flight departure delay distributions—a statistical approach with long-term trend and short-term pattern. J Am Stat Assoc. 2008;103(481):112–25.
8.   Oza S, et al. Flight delay prediction system using weighted multiple linear regression. Int J Eng Comp Sci. 2015;4(05):11765.
9.   Evans JE, Allan S, Robinson M. Quantifying delay reduction benefits for aviation convective weather decision support systems. In Proceedings of the 11th Conference on Aviation, Range, and Aerospace Meteorology, Hyannis. 2004.
10. Hsiao C-Y, Hansen M. Air transportation network flows: equilibrium model. Transp Res Rec. 2005;1915(1):12–9.
11. Britto R, Dresner M, Voltes A. The impact of flight delays on passenger demand and societal welfare. Transp Res Part E Logist Transp Rev. 2012;48(2):460–9.
12. Pejovic T, et al. A tentative analysis of the impacts of an airport closure. J Air Transp Manag. 2009a;15(5):241–8.
13. Ryerson MS, Hansen M, Bonn J. Time to burn: flight delay, terminal efficiency, and fuel consumption in the National Airspace System. Transp Res Part A Policy Pract. 2014;69:286–98.
14. Simić TK, Babić O. Airport traffic complexity and environment efficiency metrics for evaluation of ATM measures. J Air Transp Manag. 2015;42:260–71.
15. Balaban E, et al. Dynamic routing of aircraft in the presence of adverse weather using a POMDP framework. In 17th AIAA aviation technology, integration, and operations conference. 2017.
16. Xu Y, Dalmau R, Prats X. Maximizing airborne delay at no extra fuel cost by means of linear holding. Transp Res Part C Emerg Technol. 2017;81:137–52.
17. Choi S, et al. Prediction of weather-induced airline delays based on machine learning algorithms. In 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC). 2016. New York: IEEE.
18. Sternberg A, et al. A review on flight delay prediction. arXiv preprint. 2017.
19. D'Ariano A, Pistelli M, Pacciarelli D. Aircraft retiming and rerouting in vicinity of airports. IET Intel Transp Syst. 2012;6(4):433–43.
20. Al-Tabbakh SM, El-Zahed H. Machine learning techniques for analysis of Egyptian Flight delay. J Sci Res Sci. 2018;35(1):390–9.
21. Bishop CM. Pattern recognition and machine learning. Berlin: Springer; 2016.
22. Hämäläinen W. Class NP, NP-complete, and NP-hard problems. 2006.
23. Vlahogianni EI, Karlaftis MG, Golias JC. Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. Transp Res Part C Emerg Technol. 2005;13(3):211–34.
24. Kim YJ, et al. A deep learning approach to flight delay prediction. In 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC). New York: IEEE. 2016
25. Kim YJ. A deep learning and parallel simulation methodology for air traffic management. 2017, Georgia Institute of Technology.
26. No EN. Flight Delay Propagation Synthesis of the Study.
27. Liou JS. Delay prediction models for departure flights. 2006.
28. Demuth HB, et al. Neural network design. Martin Hagan. 2014.
29. Gopalakrishnan K, Balakrishnan H. A comparative analysis of models for predicting delays in air traffic networks. ATM Seminar. 2017.
30. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Adv Neural Inform Process Syst. 2012. https://doi.org/10.1145/3065386.
31. Liou C-Y, et al. Autoencoder for words. Neurocomputing. 2014;139:84–96.
32. Hinton GE, Sejnowski TJ. Learning and relearning in Boltzmann machines. Parallel distributed processing: explorations in the microstructure of cognition. 1986;1(282-317): 2.
33. Wang J, et al. Locality-constrained linear coding for image classification. In 2010 IEEE computer society conference on computer vision and pattern recognition. New York: IEEE. 2010.
34. Balakrishna P, Ganesan R, Sherry L. Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: a case-study of Tampa Bay departures. Transp Res Part C Emerg Technol. 2010;18(6):950–62.
35. Fleurquin P, et al. Trees of reactionary delay: addressing the dynamical robustness of the us air transportation network transportation. 2014;11:12.
36. Zou B, Hansen M. Flight delay impact on airfare and flight frequency: a comprehensive assessment. Transp Res Part E Logist Transp Rev. 2014;69:54–74.
37. Zou B, Hansen M. Flight delays, capacity investment and social welfare under air transport supply-demand equilibrium. Transp Res Part A Pol Pract. 2012a;46(6):965–80.
38. Klein A, Craun C, Lee RS. Airport delay prediction using weather-impacted traffic index (WITI) model. In 29th Digital Avionics Systems Conference. New York: IEEE. 2010.
39. Glover CN, Ball MO. Stochastic optimization models for ground delay program planning with equity–efficiency tradeoffs. Transp Res Part C Emerg Technol. 2013;33:196–202.
40. Pathomsiri S, et al. Impact of undesirable outputs on the productivity of US airports. Transp Res Part E Logist Transp Rev. 2008;44(2):235–59.
41. Xiong J, Hansen M. Modelling airline flight cancellation decisions. Transp Res Part E Logist Transp Rev. 2013;56:64–80.
42. Beatty R, et al. Preliminary evaluation of flight delay propagation through an airline schedule. Air Traffic Control Q. 1999;7(4):259–70.
43. Wong J-T, Tsai S-C. A survival model for flight delay propagation. J Air Transp Manag. 2012;23:5–11.
44. Ionescu L, Kliewer N. Examining delay propagation mechanisms for aircraft rotations.
45. Shao Q, Xu C. Air transportation delay propagation analysis with uncertainty in coloured–timed Petri nets. In Proceedings of the Institution of Civil Engineers-Transport. London: Thomas Telford Ltd. 2018.
46. Pejovic T, et al. Factors affecting the frequency and severity of airport weather delays and the implications of climate change for future delays. Transp Res Rec. 2009b;2139(1):97–106.

47. Montlaur A, Delgado L. Flight and passenger delay assignment optimization strategies. Transp Res Part C Emerg Technol. 2017;81:99–117.
48. Markovic D, et al. A statistical study of the weather impact on punctuality at Frankfurt Airport. Meteorol Appl. 2008;15(2):293–303.
49. Evans A, Schäfer A. The impact of airport capacity constraints on future growth in the US air transportation system. J Air Transp Manag. 2011;17(5):288–95.
50. Abdel-Aty M, et al. Detecting periodic patterns of arrival delay. J Air Transp Manag. 2007;13(6):355–61.
51. Azadian F, Murat AE, Chinnam RB. Dynamic routing of time-sensitive air cargo using real-time information. Transp Res Part E Logist Transp Rev. 2012;48(1):355–72.
52. Mofokeng TJ, Marnewick A. Factors contributing to delays regarding aircraft during A-check maintenance. In 2017 IEEE technology & engineering management conference (TEMSCON). New York: IEEE. 2017.
53. Morrison SA, Winston C. The effect of FAA expenditures on air travel delays. J Urban Econ. 2008;63(2):669–78.
54. Baumgarten P, Malina R, Lange A. The impact of hubbing concentration on flight delays within airline networks: an empirical analysis of the US domestic market. Transp Res Part E Logist Transp Rev. 2014;66:103–14.
55. Pérez-Rodríguez JV, Pérez-Sánchez JM, Gómez-Déniz E. Modelling the asymmetric probabilistic delay of aircraft arrival. J Air Transp Manag. 2017;62:90–8.
56. Wu C-L. Inherent delays and operational reliability of airline schedules. J Air Transp Manag. 2005;11(4):273–82.
57. Zhong Z, Varun D, Lin Y. Studies for air traffic management R&D in the ASEAN-region context. J Air Transp Manag. 2017;64:15–20.
58. Biesiada M, Piórkowska A. Gamma-ray burst neutrinos, Lorenz invariance violation and the influence of back-ground cosmology. J Cosmol Astropart Phys. 2007;2007(05):011.
59. Evans A, Schafer A, Dray L. Modelling airline network routing and scheduling under airport capacity constraints. In The 26th Congress of ICAS and 8th AIAA ATIO. 2008.
60. Cai K, et al. A novel biobjective risk-based model for stochastic air traffic network flow optimization problem. Sci World J. 2015; 2015.
61. Mueller E, Chatterji G. Analysis of aircraft arrival and departure delay characteristics. In AIAA's Aircraft Technology, Integration, and Operations (ATIO) 2002 Technical Forum. 2002.
62. Fleurquin P, Ramasco JJ, Eguiluz VM. Systemic delay propagation in the US airport network. Sci Rep. 2013;3:1159.
63. Sim KL, Koh HC, Shetty S. Some potential issues of service quality reporting for airlines. J Air Transp Manag. 2006;12(6):293–9.
64. Boswell SB, Evans JE. Analysis of downstream impacts of air traffic delay. Lexington: Lincoln Laboratory, Massachusetts Institute of Technology; 1997.
65. Kotegawa T, et al. Impact of commercial airline network evolution on the US air transportation system. In Proceedings of the 9th USA/Europe Air Traffic Management Research and Development Seminar (ATM'11). 2011.
66. Evans A, Schäfer A. The rebound effect in the aviation sector. Energy Econ. 2013;36:158–65.
67. Lin L, Wang Q, Sadek AW. Border crossing delay prediction using transient multi-server queueing models. Transp Res Part A Policy Pract. 2014;64:65–91.
68. Ciruelos C, et al. Modelling delay propagation trees for scheduled flights. In Proceedings of the 11th USA/EUROPE Air Traffic Management R&D Seminar, Lisbon, Portugal. 2015.
69. Hansen M, Zhang Y. Operational consequences of alternative airport demand management policies: case of LaGuardia airport, New York. Transp Res Rec. 2005;1915(1):95–104.
70. McCrea MV, Sherali HD, Trani AA. A probabilistic framework for weather-based rerouting and delay estimations within an airspace planning model. Transp Res Part C Emerg Technol. 2008;16(4):410–31.
71. Wan Y, Roy S. A scalable methodology for evaluating and designing coordinated air-traffic flow management strategies under uncertainty. IEEE Trans Intell Transp Syst. 2008;9(4):644–56.
72. Cong W, et al. Empirical analysis of airport network and critical airports. Chin J Aeronaut. 2016;29(2):512–9.
73. Liu YJ, Ma S. Flight delay and delay propagation analysis based on Bayesian network. In 2008 International Symposium on Knowledge Acquisition and Modeling. New York: IEEE. 2008.
74. Xu N, et al. Estimation of delay propagation in the national aviation system using Bayesian networks. In 6th USA/Europe Air Traffic Management Research and Development Seminar. 2005. FAA and Eurocontrol Baltimore, MD.
75. Cao W, Ding J, Wang H. Analysis of sequence flight delay and propagation based on the bayesian networks. In 2008 Fourth International Conference on Natural Computation. New York: IEEE. 2008.
76. Yao R, Jiandong W, Tao X. A flight delay prediction model with consideration of cross-flight plan awaiting resources. In 2010 2nd International Conference on Advanced Computer Control. 2010.
77. Baspinar B, Koyuncu E. A data-driven air transportation delay propagation model using epidemic process models. Int J Aerospace Eng. 2016; 2016.
78. Campanelli B, et al. Modeling reactionary delays in the European air transport network. Proceedings of the Fourth SESAR Innovation Days, Schaefer D (Ed.), Madrid, 2014. 1.
79. Belkoura S, Peña JM, Zanin M. Beyond linear delay multipliers in air transport. J Adv Transp. 2017; 2017.
80. Xu N, et al. Bayesian network analysis of flight delays. In Transportation Research Board 86th Annual Meeting, Washington, DC. 2007.
81. Laskey KB, Xu N, Chen CH. Propagation of delays in the national airspace system. arXiv preprint; 2012.
82. Rong F, et al. The prediction of flight delays based the analysis of Random flight points. In 2015 34th Chinese Control Conference (CCC). New York: IEEE; 2015.
83. Wieland F. Parallel simulation for aviation applications. In Proceedings on 1998 Winter Simulation Conference. (Cat. No. 98CH36274). New York: IEEE; 1998.
84. Robinson M, Evans JE, Hancock T. Assessment of air traffic control productivity enhancements from the Corridor Integrated Weather System (CIWS). Lexington: Lincoln Laboratory Massachusetts Institute of Technology; 2006.
85. Bertsimas D, Frankovich M. Unified optimization of traffic flows through airports. Transp Sci. 2016;50(1):77–93.
86. Yao R, Jiandong W, Jianli D. RIA-based visualization platform of flight delay intelligent prediction. In 2009 ISECS International Colloquium on Computing, Communication, Control, and Management. New York: IEEE; 2009.

Yazdi *et al. J Big Data*   (2020) 7:106

Page 26 of 28

87. Chandran BG. Predicting airspace congestion using approximate queueing models. 2002.

88. Wieland F. Limits to growth: results from the detailed policy assessment tool [air traffic congestion]. In 16th DASC. Proceedings on AIAA/IEEE Digital Avionics Systems Conference. Reflections to the Future. New York: IEEE; 1997.

89. Zou B, Hansen M. Impact of operational performance on air carrier cost structure: evidence from US airlines. Transp Res Part E Logist Transp Rev. 2012b;48(5):1032–48.

90. Hansen M. Micro-level analysis of airport delay externalities using deterministic queuing models: a case study. J Air Transp Manag. 2002;8(2):73–87.

91. Jayam H, Nozick LK. Understanding the trade-off between maximum passenger throughput and airline equity in allocating capacity under severe weather conditions. Transp Res Rec. 2017;2626(1):18–24.

92. Abdelghany KF, et al. A model for projecting flight delays during irregular operation conditions. J Air Transp Manag. 2004;10(6):385–94.

93. Kim A, Hansen M. Deconstructing delay: A non-parametric approach to analyzing delay changes in single server queuing systems. Transp Res Part B Methodol. 2013;58:119–33.

94. Pyrgiotis N, Malone KM, Odoni A. Modelling delay propagation within an airport network. Transp Res Part C Emerg Technol. 2013;27:60–75.

95. Dück V, et al. Increasing stability of crew and aircraft schedules. Transp Res Part C Emerg Technol. 2012;20(1):47–61.

96. Schaefer L, Millner D. Flight delay propagation analysis with the detailed policy assessment tool. In 2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat. No. 01CH37236). New York: IEEE; 2001.

97. Lan S, Clarke J-P, Barnhart C. Planning for robust airline operations: optimizing aircraft routings and flight departure times to minimize passenger disruptions. Transp Sci. 2006;40(1):15–28.

98. Castaing J, et al. Reducing airport gate blockage in passenger aviation: models and analysis. Comput Oper Res. 2016;65:189–99.

99. Soomer MJ, Franx GJ. Scheduling aircraft landings using airlines' preferences. Eur J Oper Res. 2008;190(1):277–91.

100. Que Z, Yao H, Yue W. Simulation analysis of the effect of initial delay on flight delay diffusion. In IOP Conference Series: Earth and Environmental Science. Bristol: IOP Publishing; 2018.

101. Lapp M, et al. A recursion-based approach to simulating airline schedule robustness. In 2008 Winter Simulation Conference. New York: IEEE; 2008.

102. Ganesan R, Balakrishna P, Sherry L. Improving quality of prediction in highly dynamic environments using approximate dynamic programming. Qual Reliab Eng Int. 2010;26(7):717–32.

103. Musaddi R, et al. Flight delay prediction using binary classification.

104. Sternberg A, et al. An analysis of Brazilian flight delays based on frequent patterns. Transp Res Part E Logist Transp Rev. 2016;95:282–98.

105. Khaksar H, Sheikholeslami A. Airline delay prediction by machine learning algorithms. Scientiairanica. 2019;26(5):2689–702.

106. Khanmohammadi S, et al. A systems approach for scheduling aircraft landings in JFK airport. In 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). New York: IEEE; 2014.

107. Zonglei L, Jiandong W, Guansheng Z. A new method to alarm large scale of flights delay based on machine learning. In 2008 International Symposium on Knowledge Acquisition and Modeling. New York: IEEE; 2008.

108. Wang Y. Prediction of weather impacted airport capacity using RUC-2 forecast. In 2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC). New York: IEEE; 2012.

109. Murça MCR, Hansman RJ. Identification, characterization, and prediction of traffic flow patterns in multi-airport systems. IEEE Trans Intell Transp Syst. 2018;20(5):1683–96.

110. Zhu G, et al. En route flight time prediction under convective weather events. In 2018 Aviation Technology, Integration, and Operations Conference. 2018.

111. Bloem M, Bambos N. Ground Delay Program analytics with behavioral cloning and inverse reinforcement learning. J Aerosp Inform Syst. 2015;12(3):299–313.

112. Wang Y. Analysis and prediction of weather impacted ground stop operations. In 2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC). New York: IEEE; 2014.

113. Ball MO, Lulli G. Ground delay programs: optimizing over the included flight set based on distance. Air Traffic Control Q. 2004;12(1):1–25.

114. Balakrishna P, et al. Estimating taxi-out times with a reinforcement learning algorithm. In 2008 IEEE/AIAA 27th Digital Avionics Systems Conference. New York: IEEE; 2008.

115. Pfeil DM, Balakrishnan H. Identification of robust terminal-area routes in convective weather. Transp Sci. 2012;46(1):56–73.

116. Takeichi N, et al. Prediction of delay due to air traffic control by machine learning. In AIAA Modeling and Simulation Technologies Conference. 2017.

117. Kulkarni D, Wang Y, Sridhar B. Data mining for understanding and improving decision-making affecting ground delay programs. In 2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC). New York: IEEE; 2013.

118. Khanmohammadi S, Tutun S, Kucuk Y. A new multilevel input layer artificial neural network for predicting flight delays at JFK airport. Procedia Comp Sci. 2016;95:237–44.

119. Venkatesh V, et al. Iterative machine and deep learning approach for aviation delay prediction. In 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON). New York: IEEE; 2017.

120. Ding Y. Predicting flight delay based on multiple linear regression. In IOP Conference Series: Earth and Environmental Science. Bristol: IOP Publishing; 2017.

121. Kuhn N, Jamadagni N. Application of machine learning algorithms to predict flight arrival delays. CS229; 2017.

122. Ganesan R, Sherry L. Taxi-out Prediction using Approximate Dynamic Programming.

123. Hoffman J. Demand dependence of throughput and delay at New York LaGuardia Airport. The MITRE Corporation; 2001.

124. Gürbüz F, Özbakir L, Yapici H. Data mining and preprocessing application on component reports of an airline company in Turkey. Expert Syst Appl. 2011;38(6):6618–26.
125. Dou X. Flight arrival delay prediction and analysis using ensemble learning. In 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). New York: IEEE; 2020.
126. Chakrabarty N. A data mining approach to flight arrival delay prediction for american airlines. In 2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON). New York: IEEE; 2019.
127. Meel P, et al. Predicting flight delays with error calculation using machine learned classifiers. In 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN). New York: IEEE; 2020.
128. Ye B, et al. A methodology for predicting aggregate flight departure delays in airports based on supervised learning. Sustainability. 2020;12(7):2749.
129. Shao W, et al. Flight delay prediction using airport situational awareness map. In Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. 2019.
130. Esmaeilzadeh E, Mokhtarimousavi S. Machine learning approach for flight departure delay prediction and analysis. Transp Res Rec, 2020.
131. Liu F, et al. Generalized flight delay prediction method using gradient boosting decision tree. In 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring). New York: IEEE; 2020.
132. Woda M, Wątrucka A. A case study-analysis of weather data impact on the disruptions/delays in passenger air traffic. Studia Informatica. 2018;39(1):89–101.
133. Najafabadi MM, et al. Deep learning applications and challenges in big data analytics. J Big Data. 2015;2(1):1.
134. Chen X-W, Lin X. Big data deep learning: challenges and perspectives. IEEE Access. 2014;2:514–25.
135. Dahl G, et al. Phone recognition with the mean-covariance restricted Boltzmann machine. Adv Neural Inform Process Syst; 2010.
136. Hinton G, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Process Mag. 2012;29(6):82–97.
137. Seide F, Li G, Yu D. Conversational speech transcription using context-dependent deep neural networks. In Twelfth annual conference of the international speech communication association. 2011.
138. Mohamed A-R, Dahl GE, Hinton G. Acoustic modeling using deep belief networks. IEEE Trans Audio Speech Lang Process. 2011;20(1):14–22.
139. Dahl GE, et al. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans Audio Speech Lang Process. 2011;20(1):30–42.
140. Bengio Y, et al. Greedy layer-wise training of deep networks. Adv Neural Inform Process Syst. 2007.
141. Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. Neural Comput. 2006;18(7):1527–54.
142. Zeiler MD, Taylor GW, Fergus R. Adaptive deconvolutional networks for mid and high level feature learning. In 2011 International Conference on Computer Vision. New York: IEEE; 2011.
143. Mikolov T, et al. Empirical evaluation and combination of advanced language modeling techniques. In Twelfth Annual Conference of the International Speech Communication Association. 2011.
144. Socher R, et al. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. Adv Neural Inform Process Syst. 2011.
145. Bordes A, et al. Joint learning of words and meaning representations for open-text semantic parsing. Art Intell Stat. 2012.
146. Crego E, Munoz G, Islam F, Big data and deep learning: big deals or big delusions? Business. 2013.
147. Yang H-F, Dillon TS, Chen Y-PP. Optimized structure of the traffic flow forecasting model with a deep learning approach. IEEE Trans Neural Netw Learn Syst. 2016;28(10):2371–81.
148. Vandal T, et al. Prediction and uncertainty quantification of daily airport flight delays. In International Conference on Predictive Applications and APIs. 2018.
149. Gui G, et al. Flight delay prediction based on aviation big data and machine learning. IEEE Trans Veh Technol. 2019;69(1):140–50.
150. Chen M, et al. Delay prediction based on deep stacked autoencoder networks. In Proceedings of the Asia-Pacific Conference on Intelligent Medical 2018 & International Conference on Transportation and Traffic Engineering 2018. 2018.
151. Yang C, Marshall ZA, Mott JH. A novel integration platform to reduce flight delays in the National Airspace System. In 2020 Systems and Information Engineering Design Symposium (SIEDS). New York: IEEE; 2020.
152. Yu B, et al. Flight delay prediction for commercial air transport: A deep learning approach. Transp Res Part E Logist Transp Rev. 2019;125:203–21.
153. Saadat MN, Moniruzzaman M. Enhancing airlines delay prediction by implementing classification based deep learning algorithms. In International Conference on Ubiquitous Information Management and Communication. Berlin: Springer; 2019.
154. Goodfellow I, Bengio Y. Courville A, Deep learning. Cambridge: MIT press; 2016.
155. Bengio Y. Learning deep architectures for AI. Found Trends Mach Learn. 2009;2(1):1–127.
156. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science. 2006;313(5786):504–7.
157. Wilamowski BM, Yu H. Improved computation for Levenberg–Marquardt training. IEEE Trans Neural Netw. 2010;21(6):930–7.
158. Van Tol, J., The programme Simfit is a MSDOS compatible routine for non-linear regression analysis as described by Marquardt. J Soc Industr Appl Math. 1963; 11.
159. Wilamowski BM. Neural network architectures and learning algorithms. IEEE Ind Electron Mag. 2009;3(4):56–63.
160. Shukla PK. Levenberg-Marquardt algorithms for nonlinear equations, multi-objective optimization, and complementarity problems. 2010.
161. Mittelmann H. The Least Squares Problem. http://plato.asu.edu/topics/problems/nlolsq.html. 2004.

Yazdi *et al. J Big Data*   (2020) 7:106

Page 28 of 28

162. Yu H, Wilamowski BM. Levenberg-marquardt training. In: Industrial electronics handbook. London: Routledge; 2011. p 1.
163. U.S. Department of Transportation Bureau of Transportation Statistics .Airline On-Time Statistics. https://www.transtats.bts.gov/ONTIME/. 2019.
164. Zhang A, et al. Dive into deep learning. Unpublished Draft. Retrieved, 2019. 19:2019.
165. Chawla NV. Data mining for imbalanced datasets: an overview, In Data mining and knowledge discovery handbook. Berlin: Springer. 2009. p. 875–886.
166. Powers DM. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. 2011.
167. Maimon O, Rokach L. Data mining and knowledge discovery handbook. Berlin: Springer; 2005.

Publisher's Note