Computing
https://doi.org/10.1007/s00607-020-00794-w

A system for effectively predicting flight delays based on IoT data

Abdulwahab Aljubairy[1,2] · Wei Emma Zhang[3] · Ali Shemshadi[4] · Adnan Mahmood[1] · Quan Z. Sheng[1]

Received: 17 February 2019 / Accepted: 27 January 2020

Abstract
Flight delay is a significant problem that negatively impacts the aviation industry and costs billion of dollars each year. Most existing studies investigated this issue using various methods based on historical data. However, due to the highly dynamic environments of the aviation industry, relying only on historical datasets of flight delays may not be sufficient and applicable to forecast the future of flights. The purpose of this research is to study the flight delays from a new angle by utilising data generated from the emerging Internet of Things (IoT) paradigm. Our primary goal is to improve the understanding of the roots and signs of flight delays as well as discovering related factors. In this paper, we present a framework that aims at improving the flight delay problem. We consider the IoT data generated from distributed sensors that have not been considered in existing works in the analysis of flight delays, and for that purpose, an automatic tool is developed to collect IoT data from various data sources including flight, weather, and air quality index. Based on the heterogeneous data, an algorithm is developed to merge different features from diverse data sources. We adopt predictive modelling to study the factors that contribute to flight delays and to predict the flight delays in the future. The results of our work show a high correlation among the developed features. In particular, the results clearly demonstrate the association between the flight delays and the air quality index factor. In particular, our current prediction model achieves 85.74% in accuracy.

Keywords Flight delay prediction system · Internet of Things (IoT) data · Real-time information retrieval

Mathematics Subject Classification 68T01 · 68U35

Extended author information available on the last page of the article

Published online: 06 February 2020

1 Introduction

With the rapid advances in the economy, air traffic has become one of the main modes of transportation in the aviation industry, which however is suffering from the flight delay problem. Indeed, flight delay is a prolonged and complex issue from which the aviation industry suffers for a long time [7]. A delay in this context is the difference between actual and scheduled times of departure or arrival of a flight. According to the Federal Aviation Administration (FAA), 15 min are the threshold of the judgement on flight delays. If the actual departure or arrival of any flight exceeds the 15 min from the scheduled time, the flight is considered delayed [5]. Flight delay has a massive impact on the productivity of the airlines and airports in terms of reputation, efficiency, and economy. In addition, the performance of the aviation networks is affected when flight delays occur, and this type of delay may propagate to other chains of airlines which subsequently also impact the airports' performance due to the limitation of the availability of resources.

Recent studies have examined the flight delay problem from different angles. Part of these studies aimed at discovering the factors that cause delays. Flight delays are sometimes caused by a number of sources of irregularities. For instance, weather plays a tremendous role since it is responsible for nearly 75% of delays [25,31]. Moreover, the recent changes in weather patterns as an effect of global warming may raise the number. The aforementioned fact shows the significance of this issue at a world wide level. Therefore, it is indispensable to look at this problem from various angles using different approaches. In previous studies, a variety of methods such as statistical analysis and machine learning are used, and these studies are mainly based only on historical data that can be obtained from the Bureau of Transportation or the FAA. Due to the highly dynamic environment of the aviation industry, relying only on historical datasets of flight delays as used in previous works such as [21] may not be sufficient and is not realistic to forecast the future status of flights.

The Internet of Things (IoT) is an emerging paradigm, which aims to connect everyday physical objects to the Internet via wireless and identification technologies [2,26,28]. IoT is revolutionizing numerous applications in a wide variety of real-world domains [27]. Novel IoT-based solutions can provide competitive advantages in several key sectors [12], in particular, predicting and controlling flight delays. Sensors have become generators of data on the Internet to enable a ubiquitous sensing of the environment. These sensors supply large stream of data, and such enormous amount of data can be utilized to study flight delays with the intent of understanding the roots of the problem and unearth other hidden factors that have not been considered before.

In contrast to the flight delay analysis in previous studies that rely mainly on historical data, this study is based on a new type of data obtained from the IoT data sources including flights, weather, and air quality sensors. However, due to privacy issues, the access to the real-world IoT data remains limited. We, therefore, tackle a number of technical challenges to enable more effective flight delay analysis based on the IoT data. To the best of our knowledge, this study is one of the first to examine the connection between contextual IoT data and the flight delays [1]. We leverage the Web Mapping such as Google Maps in order to collect the IoT data. We design and develop a platform for collecting, transforming, integrating, and analyzing data.

Our platform contains several components, and it is designed and developed for the purpose of collecting and integrating raw data from IoT data sources. Since the data comes from a wide variety of data sources, we develop an integration algorithm that can address the heterogeneity of these data sources by defining suitable criteria to integrate features from different data sources related to each flight. We identify features from the collected IoT data by investigating and demonstrating the correlation among features from various data sources. Subsequently, we adopt predictive modelling to study the factors that contribute to flight delays as well as predicting the future flight delays.

Two types of machine learning algorithms have been exploited in our study, namely *logistic regression* and *support vector machine*, in order to classify flights whether they are on-time or delayed. In addition to that, we use the multiple linear regression to predict the delay time. The results demonstrate the benefits of utilizing IoT data for tackling the flight delay problem. We summarize our main contributions as follows:

- Designing and developing a platform for collecting, transforming, integrating, and analysing IoT data. The existing studies rely on historical data obtained from relative agencies, but our research uses new type of data to study the flight delay problem. For that purpose, we design a system to collect IoT data, and also develop an integration algorithm to deal with heterogeneous data sources for the sake of merging features from different data sources.
- Identifying features related to the delay problem. We conduct correlation analysis among features extracted from various data sources to identify the most significant factors related the flight delay problem. We develop a list of features that are used in our model.
- Conducting extensive experiments to validate the effectiveness of our system. We conduct extensive experiments to demonstrate the impact of our approach. Results suggest that relying on real-time IoT data provides better understanding of flight delays.

The remaining sections of this paper are structured as follows. We discuss existing related state-of-the-art in Sect. 2. Section 3 illustrates our proposed IoT framework and technical details followed by an extensive experimental evaluation in Sect. 4. Conclusions are drawn in Sect. 5.

2 Related work

Predicting flight delays is not a new challenge, and overtime, has been considered by a number of researchers in both academia and industry. In this section, we would deliberate on the most recent and relevant research literature pertinent to our envisaged research-at-hand.

A brief glimpse of the research literature reveals a number of studies particularly devoted for highlighting and addressing various factors pertinent to flight delays. Some of these research studies [11,13,22,24,25] focused on highlighting the potential factors behind the flight delays. In [25], the authors studied major factors that contribute to flight delays, and also developed a model so as to predict flight delays using the

historical records of the Denver International Airport, Colorado, USA. In [11], the author highlighted the statistical characteristics and analysis of the factors causing flight delays, including but not limited to, airports, airlines, passengers, public safety, weather, fuel, departure control systems, and air force. In [22], the authors analyzed the influence of the time factor on the flight delays for twenty different airports in the United States, and observed the changes in the delay rate over an entire duration of a day. Both ANOVA and k-means clustering models have been employed to predict the delay at certain periods of time in a day. In [6,24], models to predict flight delays have been presented by taking into consideration both temporal and spatial delay states as explanatory variables. The underlying methodology was to predict the delays anticipated within 2–24 h in advance via Random Forest algorithm.

Other researchers have examined the delay propagation and its subsequent impact on the overall aviation ecosystem [18–20,25,30]. In [18], the authors studied the delay propagation for flight chains and estimated the delay in terms of where it has actually started so as to predict that how much time the flights in a given chain could be delayed. The said problem was modelled by utilizing the Bayesian networks. In a similar study [19], the delay propagation has been studied, wherein, the authors concluded that most of the fight delays occur in a specific time window of a day, i.e., primarily between 8 am and 9 pm. In [20], the authors deliberated at how the flights arrival delay could propagate and influence most of the other subsequent flights. They assumed that all of these types of delays only transpire in busy hubs (airports) and their influence could be propagated to other airports too. The authors proposed (a) a propagation model for investigating the relationships among flights, (b) arrival delay model using Bayesian networks, and (c) further discussed the propagation delay within an airport and argued that the arrival delay is the foremost reason behind the departure delays for the subsequent flights. Furthermore, propagation delays transpiring as a result of crew connection problems (i.e., from preceding flights) have been further studied in [25].

As-of-late, researchers have primarily shifted their focus on machine learning and deep learning schemes to come up with better *delay prediction models* in order to improve the overall efficiency of the aviation industry and enhance the satisfaction of its passengers. In [14], the authors predicted the occurrence of delays in arrival of flights at Hartsfield–Jackson Atlanta International Airport, USA. Special emphasis on predicting the occurrence of arrival delays has been made since it is closely interlinked with the satisfaction of the passengers and is regarded as one of the key reasons for triggering delays in the departure of subsequent flights. The notion of knowledge discovery along with a number of data mining techniques have been employed to achieve a reliable prediction. The historical data of flights, weather, information on the airlines, and propagation of delays were taken into consideration for training the respective model. In [16], a state-of-the-art predictive modeling approach has been employed on airlines' data sets to evaluate the feasibility of machine learning techniques for improvement in the service quality (flight delay prediction) and cost reduction (fuel consumption prediction). The predictive model was trained via different set of features available just a day before the flight, 1 week before the flight, and 5 months before the flight, and appropriate actions including assigning of a convenient gate, deployment of more supporting crew, and operation time for refueling were recommended. In

[8], a prediction model has been proposed in order to classify flight delays caused due to inclement weather conditions. Features such as historical weather and traffic data of origin–destination pairs have been taken into consideration to train the model by employing machine learning techniques (k-Nearest Neighbors, Random Forest, Decision Trees, and Adaptive Boosting) in order to enhance the predictive capability.

Moreover, an attempt to investigate the effectiveness of deep learning models for the prediction of flight delays has been undertaken in [17]. Accordingly, Recurrent Neural Networks (RNN) have been employed for predicting day-to-day flight delays since it can accurately ascertain the sequential and temporal relationships between the data. In [4], a predictor of the arrival delay of flight owing to weather conditions has been implemented which took into consideration the flight's information, i.e., its origin and destination airport, scheduled departure and arrival time, and the weather conditions at the origin and destination airport as per the flight's departing and arrival schedule. Furthermore, the authors demonstrated that the predictor's scalability can be ensured by implementing data preparation and mining tasks as MapReduce programs on a cloud infrastructure since a cloud infrastructure facilitates for a higher degree of reliability, elasticity, and scalability. A novel prediction system for predicting the estimated time of arrival of commercial flights has been presented in [3], wherein, historical trajectories (along with their 3D grid points) have been employed in order to collect key features, including but not limited to, weather parameters, air traffic, and airport data along the potential flight's path. The key features were subsequently fed into diverse regression models and a RNN, and subsequently, the best performing models with highly accurate prediction of the estimated time of arrival were compared with the estimated time of arrivals operated by the European Air Navigation Service Providers.

Nevertheless, previous works are mostly ad-hoc studies and have been conducted based on the historical data provided by some agencies, such as transportation departments. Some authors only recommended for their future work to combine the analysis of historical data with the real-time data. Different to the existing works, we perform flight delay prediction based on IoT data which has been automatically collected via various web platforms. Moreover, we develop novel methods to integrate features from multiple sources.

3 Our approach

Figure 1 illustrates our approach for targeting this issue, which includes four major steps, namely *finding IoT data sources*, *data acquisition and integration IoT platform*, *data processing and analysis*, and *prediction modelling*. Section 3.1 explains the process of finding the relevant IoT data sources. Section 3.2 introduces our IoT platform and discusses the tool we develop to collect and integrate data from the identified IoT data sources. Section 3.3 highlights our experience of preprocessing and analyzing the collected data. Section 3.4 discusses the prediction models we employ for predicting flight delays.

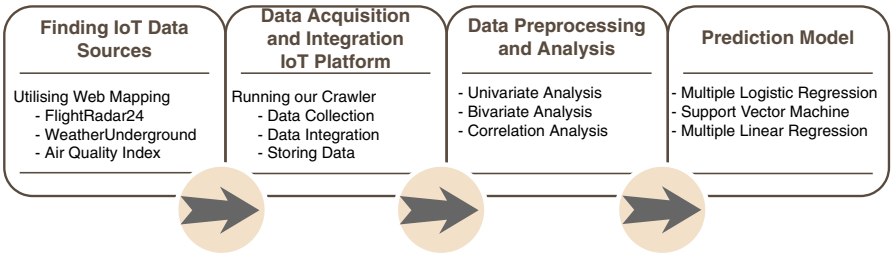


Fig. 1 Major steps of our approach for collecting the IoT data and modelling the flight delay problem

3.1 Finding and identifying IoT data sources

In this study, we rely on new type of data unlike the previous studies. This data is reported by a variety of distributed IoT devices and is typically accessed via Web services (e.g., RESTful Web services, also called RESTful APIs). The process of finding and identifying such data sources is by leveraging Web Mapping such as Google Maps. Google Maps are the most dominant way of visualizing spatial data on the Web. We limit our search scope to the sources which contain a map. To narrow our search results, we use a set of keywords in our search query in order to find such sources. Some of these keywords are "real-time map of [*application*]", "live map of [*application*]", and "tracker map of [*application*]". Also, the term *"application"* in the search query can be replaced by any IoT application that might be relevant to the flight delays problem. For this study, we use flight, weather, and air quality index as the application terms to look for the data sources that contains maps. We thus select *FlightRadar24*, *WeatherUnderground*, and *Air Quality Index* websites to retrieve IoT data. For instance, we can retrieve real-time flight data from the FlightRadar24 website. Figure 2 shows some of the features that each of these data sources include. It should be noted that additional IoT data sources could be identified and exploited if they are relevant to the flight delay problems.

3.1.1 FlightRadar24

FlightRadar24 [10] is a flight tracker that shows live air traffic around the world. The idea behind FlightRadar24 is that it operates a large network of receiver devices globally, and then it combines data that comes from several data sources including ADS-B, MLAT, Raspberry Pi, and FAA. In addition to that, schedule and flight status data such as the flight number, the origin, the destination, the scheduled and actual departure time, the scheduled and actual arrival time, the aircraft type, and many other flight details are retrieved from airlines and airports and aggregated together with the previous data sources.

3.1.2 WeatherUnderground

WeatherUnderground [29] is a well-known data source which provides live weather data. It has a large network that includes around 180,000 weather stations. It incor-

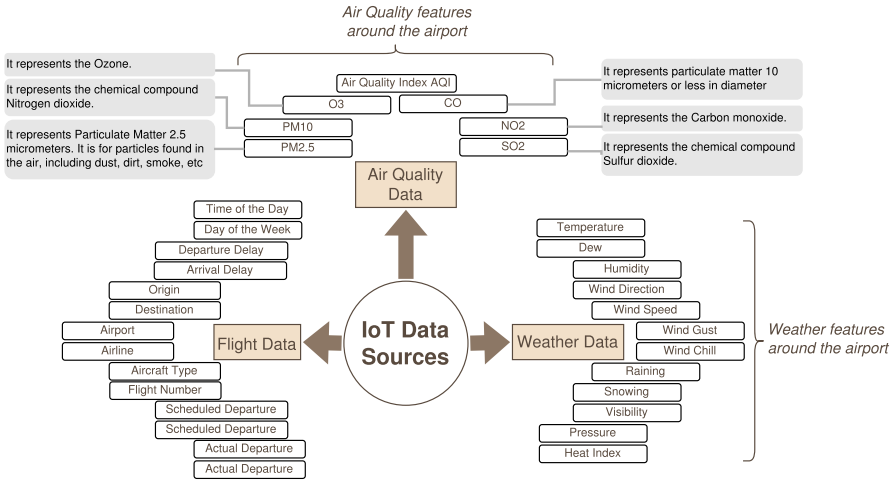


Fig. 2 The features of the three IoT data sources

porates real-time weather data obtained from various weather and climate agencies. These agencies receive the data from a huge number of distributed sensors connected to the Internet and offer a wide range of relevant weather information. They deliver the data in various formats such as XML or map format.

3.1.3 Air quality index

Air Quality Index [23] is a website which receives air quality data from distributed sensors that are connected to the Internet. It provides several indicators of air quality. One of these indicators is AQI which is an indicator that explains the quality of air in a place. It is measured by monitoring the air data followed by calculating the number of chemicals it carries.

3.2 Data acquisition and integration

We propose an IoT platform that collects data from a number of different data sources, transforms this data, and analyzes the data. Figure 3 illustrates our IoT platform. This platform includes a crawler engine that collects the data from IoT data sources and transforms the collected data before storing it in a database. The platform also contains a data analysis engine that predicts the flight status in the future.

3.2.1 The crawler engine

We develop a crawler engine for collecting real-time data from IoT data sources. This crawler contains six steps to retrieve data from an IoT data source as shown in Fig. 4. In the first step, a URL *generator* method initializes the queue of queries of a targeted IoT data source using the URL base of that data source. Each entry

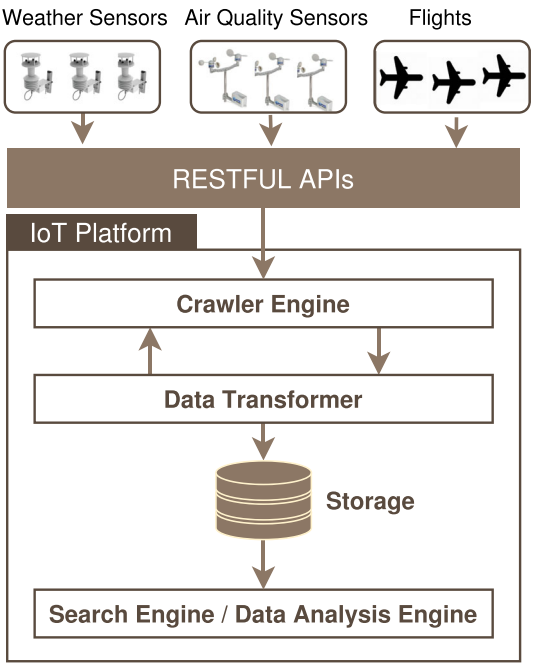


Fig. 3 The envisaged IoT platform

in the queue is supplied with certain parameters to construct a query to a page or a specific location. The parameters can be of several types such as: a flight number, time window, the boundaries of the querying region and/or other parameters. Providing these parameters depends on the data source from which we retrieve the data. In the second step, a *reader* method reads the content of each entity in the queue. In the third step, a *refiner* method converts and refines the read contents to a set of vectors. In the fourth step, the data for each subset is separately held until all subsets are refined where we merge all of the subsets of the resource's data using a *merger* method. In the fifth step, the collected data from different sources are integrated together using an *integrator* method and finally the data is stored in a distributed back-end storage. We develop our crawler using a set of tools to collect, process and visualize the dataset. Some of the tools are as follows: R programming language, SparkR, Apache Spark 1.4.1 and Rails framework.

3.2.2 Data collection

For each of the identified data sources in Sect. 3.1, a new crawling procedure is created. The new crawling procedure starts by initializing the URL link of the data source, and then goes through the steps of the crawling procedure illustrated in Fig. 4.

Algorithm 1 shows the details on the crawler engine. In line 1, a base URL (i.e., API access of the data source such as in Table 1 of the targeted data source is provided as an input. In line 2, a list of URLs are generated based on the parameters of the data source. For example, if we use the API access for the Flight Radar 24 and provide

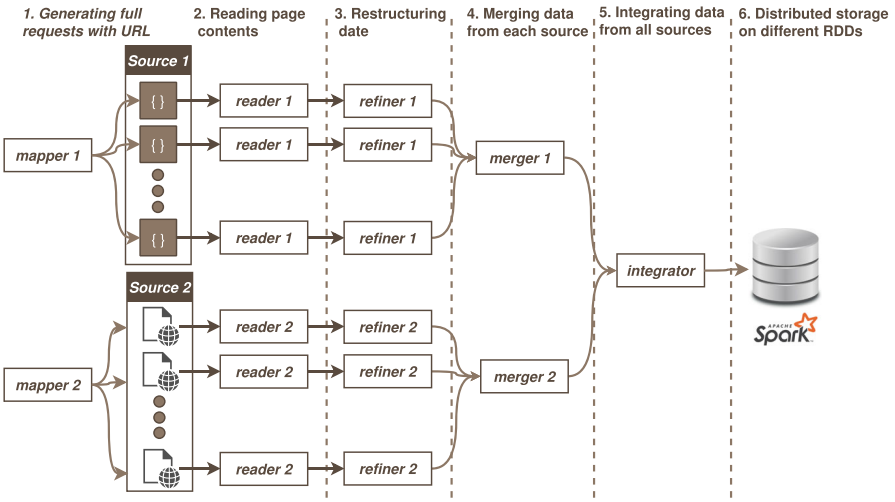


Fig. 4 The main steps of the crawler engine

100 flight IDs as parameters. Then, the generator will create 100 URLs. From line 3 to line 6, each URL from the list of the generated URLs will be fetched. Then the retrieved data will be refined and transformed into a matrix. In line 7, we merge the retrieved data from the list of URLs into one matrix. In line 8, based on the integration algorithm (Algorithm 1) we fuse the data sources together. In line 9, we store the data in a database. The time complexity of Algorithm 1 is dominated by the computation of fetching the data from each URL in the generated URL list. This computation is performed at most $O(n)$. The user can introduce her own modules by defining new ones. Finally, a strategic crawling queue for the new data source will be specified by the user. For example, users can choose to crawl the new data source hourly and only update certain parts of the data.

In this paper, we run our crawler on three IoT data sources using each data source's API to collect the data. We feed the API Access as a base URL to the crawler along with the required parameters to start the process of the automatic data collection. Table 1 shows each data source's API along with the required parameters that should be passed with the API. During the *merger* step, We store the collected data from each data source.

3.2.3 Data integration

The last step in the crawler is about integrating the collected data. After collecting the data from the the data sources, the crawler integrates this data into one dataset. We consider the flight dataset as the main container, and we bring the data from the other two data sources to it. The idea is to integrate a record i from the flight dataset with the matched record j from the weather dataset as well as the matched record k from the air quality index dataset. Each record in a dataset represents a tuple of values, $flight = \langle x_1, x_2, \ldots, x_i \rangle$; $weather = \langle y_1, y_2, \ldots, y_j \rangle$; $aqi = \langle z_1, z_2, \ldots, z_k \rangle$. We

Algorithm 1: Crawling Algorithm

```
Input: IoT Data Source Base URL
Output: Data Matrix Contains Data from the IoT Data Source
URLs = Generate a list of URLs for the targeted data sources to start reading the data;
for i = 0 to length(URLs) do
    Read the data from each URL;
    Refining the response from each URL by transforming the read data into data matrix;
end
Merge all the read data into one data matrix;
Integrate all data matrices from all data sources using the integrator;
Store the integrated data
```

Table 1 IoT data sources' APIs and the required parameters to feed each API

IoT data source	API access	Parameters
Flight radar 24	https://api.flightradar24.com/common/v1/flight/list.json?fetchBy=flight&query=	Flight number (ex: CA1855)
WeatherUnderground	http://stationdata.wunderground.com/cgi-bin/stationdata?maxstations=100000&format=json	Minimum and maximum of latitude and longitude
Real-time world air quality index	https://wind.waqi.info/mapq/bounds/?bounds=	Minimum and maximum of latitude and longitude

merge the three tuples together $\langle x_1, x_2, \ldots, x_i \rangle \cup \langle y_1, y_2, \ldots, y_j \rangle \cup \langle z_1, z_2, \ldots, z_k \rangle$ based on some criteria. The integration is not straightforward because there are no common columns among the three data sets. However, we observe that each of the three data sources include the latitude and longitude columns. These two columns represent location of airports in the flight dataset, weather sensor in the weather dataset, and AQI sensor in the air quality index dataset. The names of the columns are similar, but the values are different since they represent the locations of the entities. Therefore, our approach to overcome this issue is to utilize the two columns for the integration process by a particular means. We consider the three entities (an airport, a weather sensor, and an AQI sensor) as three points on a sphere (earth). Thus, we implement Algorithm 1 based on the Haversine formula [9,15]:

$$d = 2r \cdot \sin^{-1}\left(\sqrt{\sin^2\left(\frac{(\theta_2 - \theta_1)}{2}\right) + \cos(\theta_1) \cdot \cos(\theta_2) \cdot \sin^2\left(\frac{(\lambda_2 - \lambda_1)}{2}\right)}\right) \quad (1)$$

where d is the distance between the two points (an airport and a sensor), r is the radius of the sphere (earth), θ_1, θ_2 are the latitude of point 1 and latitude of point 2, in radians respectively, λ_1, λ_2 are longitude of point 1 and longitude of point 2, in radians respectively.[1]

[1] https://en.wikipedia.org/wiki/Haversine_formula.

Equation 1 calculates the distance between the airports and the IoT devices. We determine the proximity in terms of distance between an airport and these devices. We consider the nearest device to the airport in our work. We set the threshold distance between an airport and a sensor as 5km. If a smaller threshold is used, we may not be able to find enough sensors to retrieve data. On the other hand, if a larger threshold is used, we may retrieve information from devices that are far away from the airport, which might not be useful. We also consider the temporal information of the data from the devices. In particular, we retrieve the sensor data whose time stamps should be approximately close to the flight times of the airport.

Algorithm 2: Integrator

```
1  Input: datasets: flight, weather, and aqi
   Output: A new dataset that contains matched records from the input datasets
2  for i ← 0 to length(Sensor Data Records) do
       //  Get the longitude and latitude of each sensor
3      Long_1 ← Sensor.Longitude[i]
4      Lat_1 ← Sensor.Latitude[i]
5      Point_1 ← (Long_1, Lat_1)
6      for j ← 0 to length(flight dataset ) do
           //  Get the longitude and latitude of each airport
7          Long_2 ← Airport.Longitude[j]
8          Lat_2 ← Airport.Latitude[j]
9          Point_2 ← (Long_2, Lat_2 )
10         Distance ← Calculate the distance between (Point_1, Point_2)
11         if Distance <5 km then
12             if Flight.Date[i] == Sensor.Date[j] and Flight.Time[i] == Sensor.Time[j] then
13                 Flight.Record[i]⟨x_1, x_2, ..., x_i⟩ ∪ Sensor.Record[j]⟨y_1, y_2, ..., y_j⟩
14             end
15         end
16     end
17 end
```

For the integration algorithm, in line 1, we provide the data sources to be merged together. From line 2 to line 17, we compare the proximity of the IoT devices to the airports. In line 3 and 4, we get the spatial coordinate of each device and create as a point in line 4. We do the same for the spatial coordinate of the each airport as in line 7 and 8 and we create it as a point in line 9. In line 10, we calculate the distance between the device point and the airport point. If the distance in line 11 is less than 5 km, we check the date of the records in the data sources as in line 12. If the date is matched, we merge the records as in line 13. The time complexity of Algorithm 1 is dominated by assigning the latitude and longitude of sensors n times and airports m times. This computation is performed at most $O(n \cdot m + k)$. Here k is the number of times we merge records from the three data sources.

3.3 Data processing and analysis

One of the important steps is how to deal with the data after collection. In the following subsections, we explain our method on how to conduct the data processing and analysis.

3.3.1 Data processing

Investigating the nature of the collected data is an essential step to determine the quality of the collected data which leads to ascertaining of the validity of the modelling. We conduct a comprehensive univariant analysis on the dataset to examine any possible issues such as identifying outliers and missing values. We examine the distribution of each variable individually. We resolve the missing value issue by a particular means of imputing the missing values. The R language provides a good package called Multiple Imputation by Chained Equations (MICE). This package allows to perform imputations, and it offers several functions that identify the missing data patterns. We use md.pattern() and md.pairs() functions in order to get better understanding of the missing values pattern and the number of observations for each pattern for all pairs of variables. Then, we use the mice function which takes care of the imputing process. This function detects the variables that are missing information in the training dataset. By default, the mice function uses the Predictive Mean Matching (PMM) method, and we apply this method on numerical variables. Since the dataset includes various types of data, we use some other methods for each type. We employ two additional methods: logreg (Logistic Regression) which deals with Binary Variables and polyreg (Bayesian polytomous regression) which deals with factor variables that has more than two levels.

3.3.2 Data analysis

We perform the correlation analysis (i.e., Bivariate analysis) among all features in the dataset. This type of analysis is to explore the relationship between a dependant variable and an independent variable. It indicates the impact of the independent variable on the dependant variable. Thus, using this type of analysis helps us to have an initial vision of spotting good predictors of the phenomenon. In addition to that, applying this analysis aids to determine the independent variables that do not have a significant relationship with the dependent variable. Measuring the strength of the association between these variables ranges between $(+1)$ and (-1). The sign indicates to the direction of the correlation among features, and the measurement value indicates to the strength of the correlation. Whenever the value is close to $(+1)$ or (-1), it indicates the strength of the relation positively or negatively. The details can be found in Sect. 4.4.

3.4 Prediction modelling

After investigating the correlation among features in the dataset, we create the predictive models. The predictive models classify and predict the flight delays. That means, when we pass a given flight ID along with the time of the flight, the models should

classify the status of the flight whether delayed or on-time and should determine the delay time for that flight. Predicting the delay time can give customers more concrete information and help them make a decision on whether to accept that delay or not. We develop two types of models: *multiple logistic regression* and *multiple linear regression*. The first model classifies the flights as on-time or delayed. The second model predicts the delay time for each flight in minutes. Our regression models are detailed in Eqs. 2 and 3 where they can determine the status of a flight using set of features. We identify the features that are selected to be in the model based on the correlation with the Delay at Departure (DAD) to ascertain if these features are statically significant. We eliminate the features that have correlation with each other through the multicollinearity test.

$$DAD_{flight} = \sum_{X_i \in FEATURES}^{n} \beta_i X_i \tag{2}$$

$$DAD_{flight} = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \cdots + \beta_i * X_i \tag{3}$$

The flight delay problem can be modelled in many ways. Based on the objectives of our research, we need to study this phenomenon from a new angle using new types of data. We develop a multiple logistic regression model to predict and classify if a flight would be on-time or delayed. Logistic Regression is one of the most widely used methods when the response of the dependent variable is dichotomous. We build Multiple Logistic Regression with various set of features from flight, weather, and air quality index (AQI) datasets.

4 Experimental studies

This section evaluates the effectiveness and efficiency of our approach. We have extensively studied several countries as case studies and report the experimental results using the data from China due to its quickly-growing aviation industry and poor records of flight delays. First, we explain the case study setting in Sect. 4.1. Second, we discuss the data collection and preparation in Sect. 4.2. Then, we analyze the flight delay based on the collected data in Sect. 4.3 and the correlation among features in Sect. 4.4. Finally, we present the evaluation results of the prediction models in Sect. 4.5.

4.1 Case study setting

We select seven airports in China for this case study, which are the main airports located in seven large cities in China. The list of airports include: Beijing (PEK), Shanghai (SHA), Guangzhou (CAN), Wuhan (WUH), Chengdu (CTU), Harbin (HRB), and Dalian (DLC). The criteria of selecting these airports is based on the size of the city and its population.

After the selection of the cities for our case study, we start retrieving all operated flights among these cities. The process of finding the flights between an origin and a destination is as the follows: we first pass the list of the airports to the crawler, and

the crawler then selects the origin airport and the destination airport from the list and visits the relevant URL FlightRadar website.[2] A list of flights operated between the origin and the destination will be retrieved. Finally, the crawler stores the flights list in a .csv file to be used as parameters when we run the crawler to collect flights data.

For example, the crawler sets PEK airport as the origin, and takes SHA airport as the destination. It retrieves all operated flights between these two airports. The crawler repeats the process by changing the destination (e.g., CAN airport) until it goes over the other six airports as a destination. After that, the crawler changes the origin airport (e.g., SHA airport) and repeats the process. There are 800+ operated flights among these airports. The distribution of the number of flights varies according to the city size and the population. The airlines that operate among these cities include China Air (CA), Shanghai Airlines (FM), China Eastern Airlines (MU), China Southern Airlines (CZ), Juneyao Airlines (HO), Hainan Airlines (HU), Xiamen Airlines (MF), Sichuan Airlines (3U), Shandong Airlines (SC), Chongqing Airlines (OQ), Grand China Air (CN), Shenzhen Airlines (ZH), Spring Airlines (9C), Tibet Airlines (TV), Beijing Capital Airlines (JD), Chengdu Airlines (EU).

For the WeatherUnderground and Air Quality Index data sources, to limit the crawler searching space, we set the boundaries of China using geographic coordinates, i.e., the latitude (set as 14–55) and the longitude (set as 74–136).

4.2 Data collection and preparation

In the previous section, we explain the process of setting up the required parameters for our case study for enabling the crawler to collect the data from the three major data sources. After that, we run the crawler to start the data collection. To read data from flight data source, the crawler first reads the flight number from the stored .csv file and attaches it to the API Access of FlightRadar24 (Table 1). Then, the *generator* produces a URL for each flight number. The *reader* reads the content of each URL and the collected data is held until the reading task is completed. The resulted data are in a JSON format, so the *refiner* method restructures the retrieved data to a set of vectors. This refined data is merged into a single dataset called "flight dataset".

We also conduct the same process to retrieve data from the other two IoT data sources, WeatherUnderground and Real-time Air Quality Index. However, the process of collecting data from these data sources varies from FlightRadar24. For crawling the flights data, we need only to pass the flight number to the FlightRadar24's API Access. On the contrary, in the case of retrieving data from WeatherUnderground and Real-time Air Quality Index, we do not know the locations and the number of the IoT sensors around each airport. Therefore, our approach to find these sensors is by looking China's map and then retrieving data from these sensors. We perform this by passing the boundaries of China's latitude and longitude as parameters to our crawler in addition to the IoT data sources' APIs Access as shown in Table 1. Then, the crawler fetches all the distributed IoT sensors that belong to these data sources and bring the data from there. We store the collected data from WeatherUnderground and Real-time Air Quality Index in datasets named: "weather dataset" and "aqi dataset"

[2] https://www.flightradar24.com/data/flights.

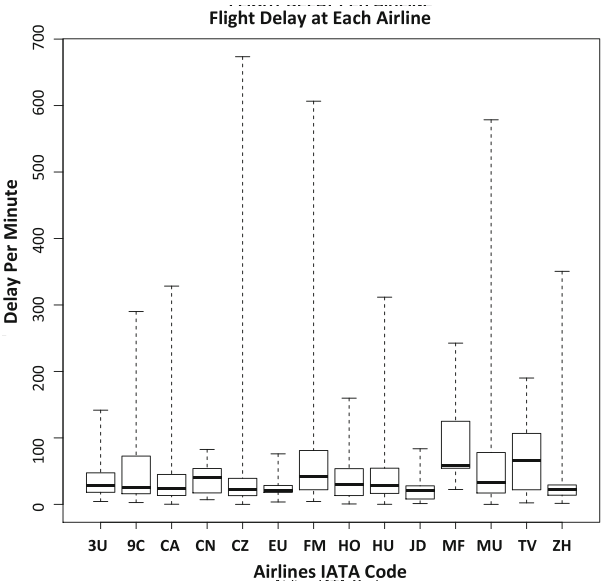


Fig. 5 Airlines' performance

respectively. Then, the crawler automatically integrates the three datasets using the integrator component.

We sort the collected data randomly and split the collected data into training and test datasets. The training data represent 70% and the test data represent 30%. We conduct a comprehensive univariate analysis on the training dataset and find out that most of the features in the collected data are categorized as category variables although some of these features contain numerical values. As a result, we rectify the data type of these features in order to perform the appropriated method for conducting the univariate analysis and deploying the statistical measures on these variables. We treat all outliers and missing values as explained in Sect. 3.3.

4.3 Flight delay analysis

We explore the airline performance data by using one of the statistical methods. Boxplots are used in order to visualize the overall performance of each individual airline. From the interesting plot (see Fig. 5), we can observe the airline impact on the flight delays problem. We believe that the airline factor may play a significant role on the flight delays. As we can see from Fig. 5, some airlines do not operate the majority of their flights on time. For example, the majority of the Xiamen Airlines (MF) flights are delayed and the same is true for the Spring Airlines (9C), China Air (CA), Shanghai Airlines (FM), China Eastern Airlines (MU), China Southern Airlines (CZ),and Sichuan Airlines (3U).

We also explore the airports' performance as shown in Fig. 6. Interestingly, we find that Shanghai airport (SHA) does not perform well since the majority of flights are

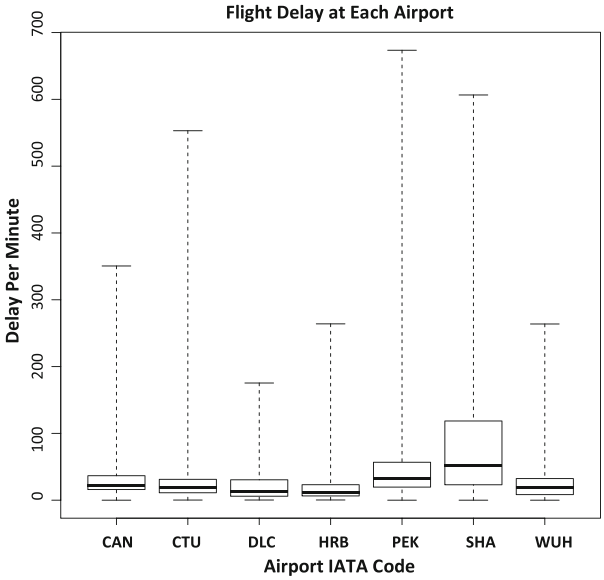


Fig. 6 Airports' performance

delayed, and the same situation pertains in Guangzhou (CAN), Dalian (DLC), and Beijing (PEK). This indicates that the airport factor should be taken into account as well.

To visualize the flight delay situation of each airport, we create a heat map that shows the delays of each airport. Figure 7 shows that most of the flight delays happen in large cities such as Shanghai, Beijing, and Guangzhou. This is due to the mega size of these cities and the large number of flights in their airports. Hence, this is a good indication that the airport factor may also play a significant role on the flight delays issue.

4.4 Correlation analysis

In this section, we study the correlations among features for the sake of determining the most important features related to the flight delay problem. Figure 8 presents the correlation among the features of the collected data. Our focus is the association between the Delay at Departure (DAD) (the dependent variable) and the remaining features. The figure shows a strong correlation between DAD and the Delay at Arrival (DAA), which is to some extend expected and not surprising. It also shows that there are positive and negative correlations between DAD and the weather factors such as temperature, heat index, dew, visibility, and elevation. In addition, the results show that the correlation among some weather features is almost perfect, whether positive or negative. The most interesting fact we can extract from this figure is that we find a considered correlation between DAD and air quality index (AQI). This refers that flights which depart from places suffering from higher AQI are most likely to get delayed.

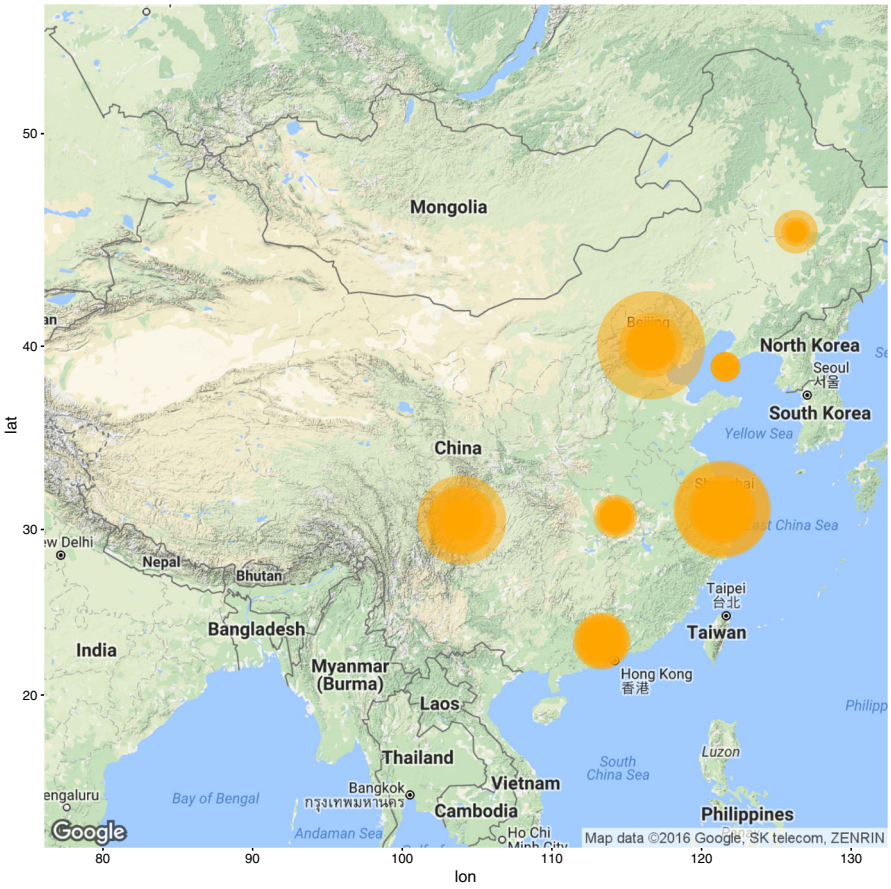


Fig. 7 Flight delays of the airports

After conducting the analysis, the following features are selected: *AQI, Heat.Index, Temperature.FEH, Dew.Point, DAA, Flight.ID, Origin.Airport, Humidity, Wind.speed, Wind.Direction, Wind.Gust, Elevation, and Destination.Airport* to be included in our model as shown in Fig. 9.

Since our approach differs from the start-of-the-art approaches in terms of the type of the data, we build the model using the aforementioned features. We create additional three versions of this model but with a subset of features as listed below. Subsequently, we compare the performance of our model using the full list of nominated features against the other three created versions. We aim to study the impact of using the new discovered factor AQI with other features from the flight and weather datasets. The original model (the first one) and the three versions of the model are as follows:

– The 1st model: Using features from all three datasets (i.e., flights, weather, aqi);
– The 2nd model: Using features only from the flight dataset;
– The 3rd model: Using features from the flight and the weather datasets;
– The 4th model: Using features from the flight and the aqi datasets.

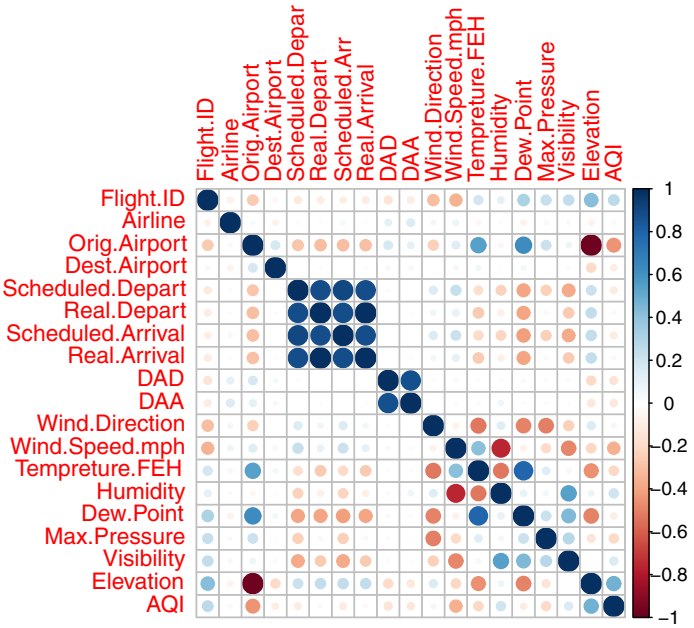


Fig. 8 Showing the correlations among all the attributes of the three IoT data sources

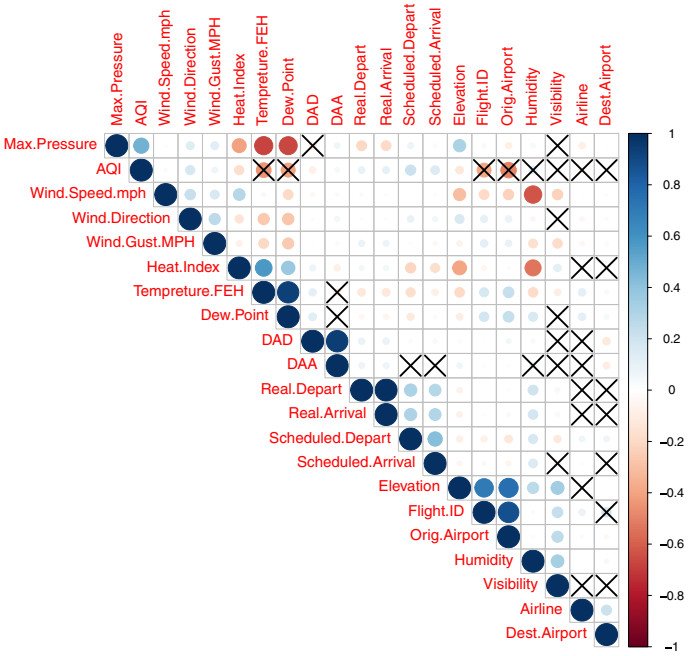


Fig. 9 The correlation analysis among heterogeneous features with the delay at departure

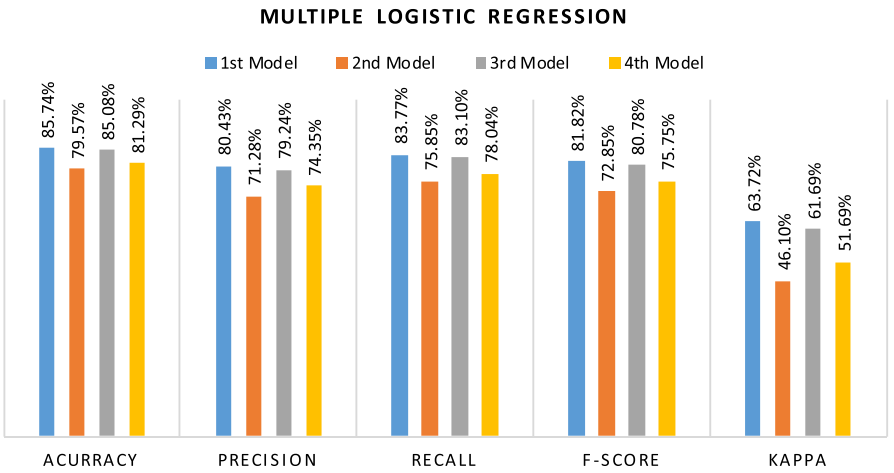


Fig. 10 The performance results of the multiple logistic regression models

4.5 The evaluation of prediction model

This study focuses on the evaluation of predicting the flight delays using IoT data. Figure 10 presents the performance metrics for each one of these models. The related performance metrics include: (1) accuracy, (2) precision, (3) recall, (4) F-score, and (5) kappa. *Precision* is the positive predictive value that shows how a model is correct to predict the positive class. *Recall* is a measurement on how complete the results are. *F-score* is the harmonic mean that combines precision and recall used for rates of change. *Kappa* adjusts the accuracy by accounting for the possibility of a correct prediction by chance alone. This metric is important in case if we have datasets with a severe class imbalance. We calculate these metrics for the four versions of our model in order to evaluate and compare their performance against each other (see Fig. 10).

The results indicate that using features from all the three datasets (flight, weather, and aqi) tends to provide better prediction. The results also illustrate that each one of the five-performance metrics of the 1st model outperform the other three models. The performance of the 1st and 3rd models is close to each other, but the 1st model includes the AQI feature that makes the difference, which suggests that AQI has an important impact on the flight delays.

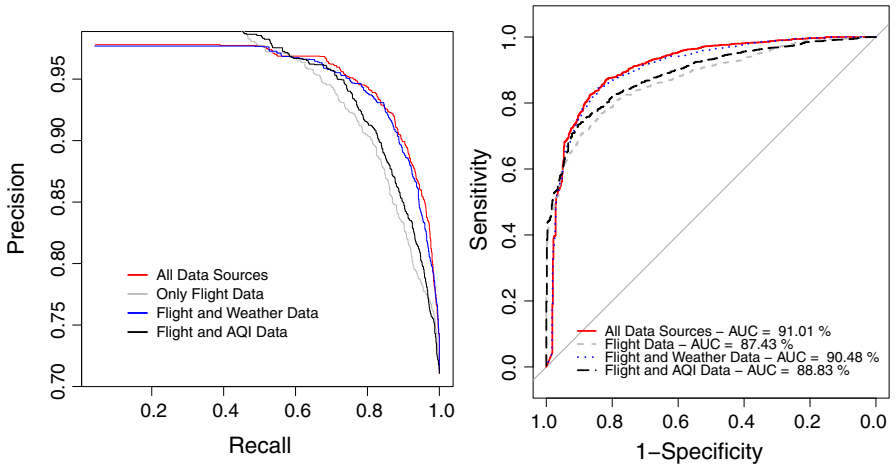
We also create the receiver operating characteristic (ROC) curves for all the models to illustrate the performance of these models. Each curve is created by plotting the *sensitivity* (probability of detection) against *1-specificity* (probability of false alarm). Figure 11b shows the ROC curves for the four models along with a diagonal line from the bottom-left to the top-right corner of the figure. This line represents a classifier that does not provide a predictive value, which means it detects true positives and false positives at exactly the same rate. Such type of classifiers cannot distinct between the two output values. In our study, we consider this line as the baseline where we can judge our models. ROC curves that are close to this baseline indicate the poor performance of its corresponding models. Clearly, a perfect classifier should have a

curve that passes through the point at a 100% sensitivity and 0% false positive rate. A model that has a ROC curve closer to the perfect classifier is considered the best since it can identify the positive values better than the other models. Area Under Cover (AUC) is a statistical measurement that provides such score. This score illustrates how a model close to the perfect model. AUC score ranges from 0.5 to 1 (AUC score = 1 for the perfect model). We plot the ROC curves for all the four models along with the AUC scores. In Fig. 11b the 1st model achieves the AUC score of 91.01% where the other models 2nd, 3rd, and 4th achieve 87.43%, 90.48%, and 88.83% respectively.

We also plot precision and recall curves for all models. The precision–recall curve is closely related to ROC curve. It indicates how models' results are relevant. The resulted curves in Fig. 11a show that the first model outperforms the other three models. Our primary goal of this research is to investigate the possibility of using new types of data and finding factors that contribute to the flight delay problem. We utilize real-time data provided by distributed and publicly available IoT sensors including flight, weather, and air quality index. We investigate features from the three data sources and study their impact on the delay at departure (DAD). To the best of our knowledge, we are one of the first to consider IoT data in addressing the flight delay problem. In particular, no other study has considered the AQI impact on the DAD. We create multiple logistic regression models to classify flights based on a set of features composed from the three IoT data sources. Our model achieves high accuracy of 85.74%.

In addition, the precision and the recall of the first model outperform other compared models in this study. We observe that the 2nd model which uses features from the data sources (flight and weather) accomplishes close results to the 1st model, with a slight difference. We believe that this slight difference transpired due to our newly discovered feature AQI from the air quality index data source. Calculating all performance metrics shows that the 1st model is better than the other models. The same trend of the results are observed when we use another classification model. Both machine learning algorithms illustrates the better performance when we use features from the three IoT data sources. Moreover, we observe that when the 4th model (AQI feature with the flight data is used), it improves the performance more than the 2nd model (only uses flight data). This shows the effect of AQI on the flight delay issue and gives an indication on the benefits of exploiting the Internet of Things paradigm in the investigation of the flight delays issue.

We also investigate another machine learning algorithm, Support Vector Machine algorithm (SVM), to see the effectiveness of our features. We perform the same process as we do with Multiple Logistic Regression algorithm and create four SVM models. Although we do not get higher results as we have obtained from the logistic regression, we observe the similar trend from the results. Using features from the three IoT data sources provides better results as shown in Fig. 12. In addition, Fig. 13a, b show the precision–recall and the ROC curves for all SVM models and illustrate better performance of the 1st model that uses features from the IoT data sources. It achieves AUC of 84.64%, whereas, the other 2nd, 3rd, and 4th models achieve 81.06%, 82.95%, and 81.63% respectively.



(a) Precision-Recall plot. **(b)** ROC curves plot.

Fig. 11 Precision–recall and ROC curves plot for the logistic regression models

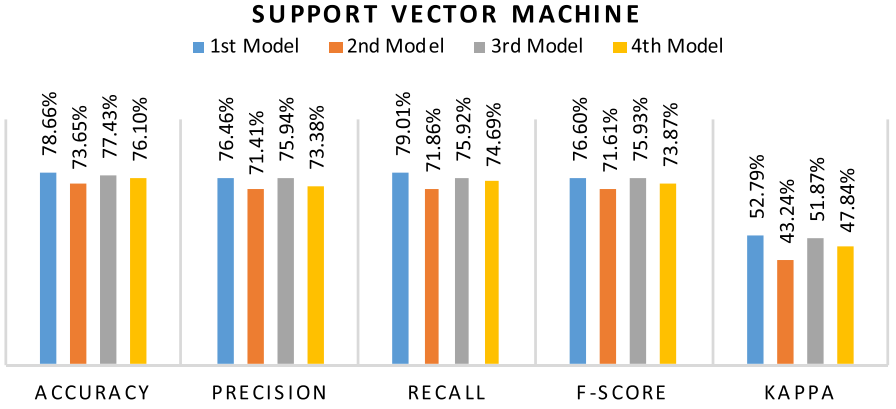
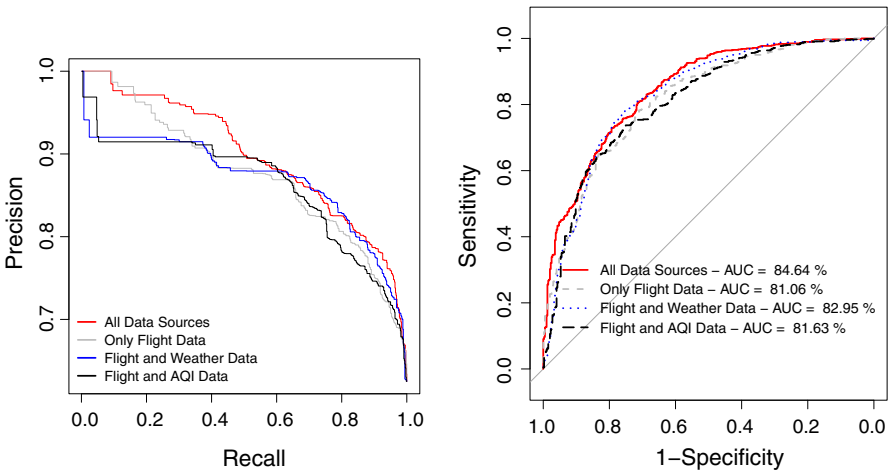


Fig. 12 The performance results of the support vector machine (SVM) models

5 Conclusion

Flight delays are a long-standing problem that has been studied previously mainly using historical records. In this work, we present a new approach for flight delay prediction by exploiting the Internet of Things (IoT) data. Our work leverages the Web Mapping such as Google Maps to find the real-world IoT data sources. We design and develop a system to collect, transform, and integrate the IoT data that come from different sources. We implement an algorithm to effectively discover and merge different features from diverse IoT data sources. The extensive experimental results demonstrate the benefits of utilizing IoT data for tackling the flight delay problem. For the future work, we plan to include more IoT data sources related to the airports



(a) Precision-Recall Plot. **(b)** ROC curves.

Fig. 13 Precision–recall and ROC curves plot for the support vector machine (SVM) models

to improve the effectiveness of the approach. We will also investigate other machine learning algorithms to improve the predication performance of the system.

References

1. Aljubairy A, Shemshadi A, Sheng QZ (2016) Real-time investigation of flight delays based on the Internet of Things data. In: Proceedings of international conference on advanced data mining and applications (ADMA), pp 788–800
2. Atzori L, Iera A, Morabito G (2010) The Internet of Things: a survey. Comput Netw 54(15):2787–2805
3. Ayhan S, Costas P, Samet H (2018) Predicting estimated time of arrival for commercial flights. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining, pp 33–42
4. Belcastro L, Marozzo F, Talia D, Trunfio P (2016) Using scalable data mining for predicting flight delays. ACM Trans Intell Syst Technol (TIST) 8(1):1–20
5. Bureau of Transportation Statistics (2018) Bureau of transportation statistics—airline on-time performance and causes of flight delays. https://www.bts.gov/topics/airlines-and-airports/airline-time-performance-and-causes-flight-delays. Accessed 01 Sept 2019
6. Chandramouleeswaran KR, Krzemien D, Burns K, Tran HT (2018) Machine learning prediction of airport delays in the US air transportation network. In: Proceedings of aviation technology, integration, and operations conference, pp 3672–3682
7. Chen Y, Yu J, Tsai SB, Zhu J (2018) An empirical study on the indirect impact of flight delay on China's economy. Sustainability 10(2):357
8. Choi S, Kim YJ, Briceno S, Mavris D (2016) Prediction of weather-induced airline delays based on machine learning algorithms. In: Proceedings of 35th digital avionics systems conference (DASC), pp 1–6
9. Chopde NR, Nichat M (2013) Landmark based shortest path detection by using A* and Haversine formula. Int J Innov Res Comput Commun Eng 1(2):298–302
10. Flightradar24 (2018) Flightradar24 live flight tracker. https://www.flightradar24.com. Accessed 01 Sept 2019
11. Geng X (2013) Analysis and countermeasures to flight delay based on statistical data. In: Proceedings of 5th international conference on intelligent human–machine systems and cybernetics (IHMSC), pp 535–537

12. Georgakopoulos D, Jayaraman PP (2016) Internet of Things: from internet scale sensing to smart services. Computing 98(10):1041–1058
13. Gopalakrishnan K, Balakrishnan H (2017) A comparative analysis of models for predicting delays in air traffic networks. In: Proceedings of 12th USA/Europe air traffic management research and development seminar (ATM2017), pp 1–10
14. Henriques R, Feiteira I (2018) Predictive modelling: flight delays and associated factors, Hartsfield–Jackson Atlanta International Airport. Proc Comput Sci 138(1):638–645
15. Hijmans RJ, Williams E, Vennes C (2012) Geosphere: spherical trigonometry. R package version 1.2–28. https://CRAN.R-project.org/package=geosphere
16. Horiguchi Y, Baba Y, Kashima H, Suzuki M, Kayahara H, Maeno J (2017) Predicting fuel consumption and flight delays for low-cost airlines. In: Proceedings of the 31st AAAI conference on artificial intelligence (AAAI), pp 4686–4693
17. Kim YJ, Choi S, Briceno S, Mavris D (2016) A deep learning approach to flight delay prediction. In: Proceedings of digital avionics systems conference (DASC), pp 1–6
18. Liu Y, Yang F (2009) Initial flight delay modeling and estimating based on an improved Bayesian network structure learning algorithm. In: Proceedings of 5th international conference on natural computation, vol 6, pp 72–76
19. Liu YJ, Ma S (2008) Flight delay and delay propagation analysis based on Bayesian network. In: Proceedings of knowledge acquisition and modeling, pp 318–322
20. Liu YJ, Cao WD, Ma S (2008) Estimation of arrival flight delay and delay propagation in a busy hub-airport. In: Proceedings of 4th international conference on natural computation, pp 500–505
21. Mueller ER, Chatterji GB (2002) Analysis of aircraft arrival and departure delay characteristics. In: Proceedings of AIAA's aircraft technology, integration, and operations (ATIO), pp 1–14
22. Qin Q, Yu H (2014) A statistical analysis on the periodicity of flight delay rate of the airports in the US. Adv Transp Stud (ATS) 3(1):231–241
23. Real-Time Air Quality Index (2018) Real-time air quality index for more than 60 countries in the world. https://aqicn.org. Accessed 01 Sept 2019
24. Rebollo JJ, Balakrishnan H (2014) Characterization and prediction of air traffic delays. Transp Res Part C Emerg Technol 44(1):231–241
25. Rosenberger JM, Schaefer AJ, Goldsman D, Johnson EL, Kleywegt AJ, Nemhauser GL (2002) A stochastic model of airline operations. Transp Sci 36(4):357–377
26. Sheng QZ, Qin Y, Yao L, Benatallah B (eds) (2017) Managing the web of things: linking the real world to the web. Morgan Kaufmann, Burlington
27. Tata S, Klai K, Jain R (2017) Formal model and method to decompose process-aware IoT applications. In: Proceedings of on the move to meaningful internet systems, pp 663–680
28. Tran NK, Sheng QZ, Babar MA, Yao L, Zhang WE, Dustdar S (2019) Internet of Things search engine. Commun ACM 62(7):66–73
29. Weather Underground (2018) Weather underground: weather forecast and reports—long range. https://www.wunderground.com. Accessed 01 Sept 2019
30. Wu W, Wu CL, Feng T, Zhang H, Qiu S (2018) Comparative analysis on propagation effects of flight delays: a case study of China airlines. J Adv Transp 2018(1):1–10
31. Zhu G, Matthews C, Wei P, Lorch M, Chakravarty S (2018) En route flight time prediction under convective weather events. In: Proceedings of aviation technology, integration, and operations, pp 2176–2191

Affiliations

Abdulwahab Aljubairy[1,2] · Wei Emma Zhang[3] · Ali Shemshadi[4] · Adnan Mahmood[1] · Quan Z. Sheng[1]

✉ Abdulwahab Aljubairy
abdulwahab.aljubairy@hdr.mq.edu.au

Wei Emma Zhang
wei.e.zhang@adelaide.edu.au

Ali Shemshadi
ali.shemshadi@gmail.com

Adnan Mahmood
adnan.mahmood@hdr.mq.edu.au

Quan Z. Sheng
michael.sheng@mq.edu.au

1 Macquarie University, Sydney, NSW 2109, Australia

2 Umm Al Qura University, Mecca, Makkah Province 21955, Saudi Arabia

3 The University of Adelaide, Adelaide, SA 5005, Australia

4 Complexica Pty Ltd, West Lake, SA 5021, Australia