Journal of Big Data

RESEARCH **Open Access**

A novel intelligent approach for flight delay prediction

Maged Mamdouh[1*], Mostafa Ezzat[1] and Hesham A.Hefny[1]

*Correspondence:
magedmamdouh@pg.cu.edu.eg

[1] Department of Computer Science, Faculty of Graduate Studies for Statistical Researches (FGSSR), Cairo University, Cairo, Egypt

Abstract

Flight delay prediction is one of the most significant components of intelligent aviation systems that may spread throughout the whole aviation network and cause multi-billion-dollar losses faced by airlines and airports, it is quickly becoming an important research issue to improve airport and airline performance. Thus this paper proposed an effective algorithm called Flight Delay Path Previous-based Machine Learning (FDPP-ML) capable of improved prediction of individual flight delay minutes using regression models to an up level of accuracy. As aviation system connectivity presents complex spatial–temporal correlations, machine learning approaches have addressed flight delay prediction by using complex flight or weather features, or private information for specific airports and airlines that are hard to obtain, In contrast, the proposed FDPP-ML improved prediction based only on basic flight schedule features even with wide flight networks. The FDPP-ML consists of a novel algorithm with a supervised learning model, which works on reshaping datasets and creates two new features the main feature is previous flight delay (PFD) for flight paths, there is a strong relationship between departure and arrival delay, and vice versa for the same flight path, which increases the strength of the training model based on historical data. For target future flights, the algorithm works on inheriting the predicted flight delay to the next flight on the same flight path and repeats this process to end the prediction forecast horizon. The proving of approach effectiveness by using a wide network of US flight arrival and departure flights containing 366 airports and 10 airlines with various metrics accuracies of regression, and explanatory the impacts on various forecast horizons 2, 6, and 12 h for future flights. The FDPP-ML outperforms traditional training models by using machine and deep learning models and improving model accuracy in 10 models with an average of up to 39% in MAE, and 42% in MSE in a forecast horizon of 2 h. Finally, providing airport and airline analysis further reveals that can improve prediction than traditional training models for the individual busiest airports "Core 30" with an average of 35% in MAE and 42% in MSE respectively, and for the busiest 10 airlines with an average of 36% in MAE and 47% in MSE respectively. The findings of this study may offer informative recommendations to airport regulators and aviation authorities for developing successful air traffic control systems for enhanced flight delay prediction to flight operational effectiveness, not only over the US flight network but with wide worldwide flight networks if a dataset of flights existed.

Keywords: Flight delay prediction, Regression, Machine learning, Deep learning, Air traffic

Introduction

Flight performance on time is an important measure of an airport's and airline's service excellence, the prediction of flight delay duration over a predetermined horizon can help airlines adopt contingency plans as soon as possible and eliminate missed revenue and penalty costs [1]. Flight delays whether arrival or departure had a great impact on airlines, passengers, and airports. Flight arrival delays are one of the leading reasons for commercial airlines' losses and passenger complaints [2]. In 2019, the Federal Aviation Administration FAA anticipated that delays would cost $ 33 billion per year [3]. It also has a negative influence on the environment since it raises petrol emissions through fuel usage, to save fuel, airlines are also always looking for new technology and improving flight procedures [4]. On the other hand, flight departure delay prediction benefits the airport by allocating unused airport capacity and airspace to alternative airlines, providing customers with reliable travel plans, and improving airline service performance by altering schedules ahead of time [5]. Overall the arrival and departure delay prediction provides airport planners effective staff workload curve because flight time changes in actual time while they depend on flight schedule time when planned such as in [6], this enables them to make more reliable decisions regarding the strategic workforce required. Furthermore, the prediction helps Airport operation control centers (AOCC) for the monitoring and adjustment of airline schedules on the day of operations to help air traffic control, airport, and ground handling service providers, instead of elaborating manually based on expertise [7]. As a result, precise flight delay forecasts will remain crucial in supporting airports and airlines to provide high-quality service, there are several attempts to forecast flight delays, in parallel with increased competition in the commercial aviation sectors. Despite the unsupervised techniques being successful classification [8], this paper uses supervised techniques to predict individual flight delay minutes.

A flight delay indicates a delay of more than 15 min, according to the FAA [9], most flight delay prediction studies are divided into two main categories regression and classification prediction, the classification prediction of delay known as binary classification has two variables (on time, delay 15 min), while regression prediction can do that and be more robust for air transportation systems by predict specific delay times approached to actual delay minutes, providing more granular guidance for practical application in the relevant sectors. The regression model examines significant associations of the strength of the effects of multiple different independent variables on a dependent variable, and the flight delay problem is the result of the interaction of multiple flight features in the data [10]. In this context, this paper proposed FDPP-ML contains ML regression models hybrid with a novel algorithm and demonstrates its improvement on these advanced regression models for flight delay prediction. in the flight features, the temporal variables constitute the fundamental components features [11], called "Flight schedule" which contains basic features of flights that airports have in advance [12]. No study can rely on flight schedules only to predict flight delays so far, owing to the major causes of flight delays being different according to the numerous stochastic features involved and the flight network domain, it might be 30 features in the default flight dataset. An initial flight delay can be attributed to extreme weather, air carriers, security concerns, flight network congestion [13], and other factors [14]. According to the Bureau of

Transportation Statistics (BTS) report summarized from June to November 2022 [15], Fig. 1 shows the factors causing flight arrival delays. It should be noted the weather has a great portion represents 52.43% of causes of delays, volume 35.47% %, closed runways 5.78%, other factors account for 5.47%, and equipment 0.86%, respectively, due to airline company problems or technological issues. Accordingly, the first objective of this paper avoid these features to a high degree at the same time provide effective flight delay prediction.

For example in weather features several studies demonstrated that it is a major cause of flight delays [16], the weather is an important aspect in investigating aircraft delays since it impacts other delay factors, but due to data gathering challenges, assessing the influence of meteorological conditions along the airway is particularly difficult [17]. Flight delay requires further particularly confidential data for the specific airport such as Automatic dependent surveillance-broadcast messages (ADS-B), special airport information [18, 19], and one airline [2], the flight delay can need specific flying features such as altitude, ramp weight, and runway direction [1], that means these approach implemented at one their case study of an airport or one airline. Consequently, the airports, airlines, and aviation authorities desperately need an effective flight delay prediction model that relies on only basic flight features "Flight schedule", and capable implemented for a wide airports network, in the proposed approach FDPP-ML provides a solution for effective flight delay prediction based on only features flight schedule and can be used over all wide world flight networks if a flight data set existed in real-time implemented, this paper's actual experiments are already based on US flights from 366 airports and 10 airlines.

Through investigation, we found that departure delay was commonly chosen as a successful element to improve arrival flight delay prediction because arrival and departure delays are inextricably linked, a flight that is delayed on departure will almost certainly be delayed on arrival. According to [20] overcrowding at the destination arrival airport was mostly caused by the origin departure airport. Figure 2 shows the correlation between US flight departure and arrival delay within the available dataset, the same flight that has a delay in departure has caused an arrival delay with a percent correlation of 91.82%.

As a result, according to the close relationship between departure and arrival delays for the same flight, the studies depended on the feature we called the "Previous Flight

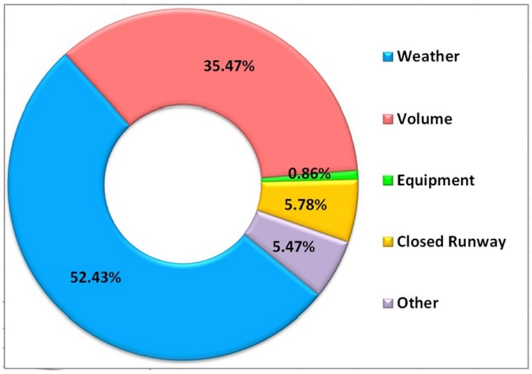


Fig. 1 Factors causing flight arrival delays

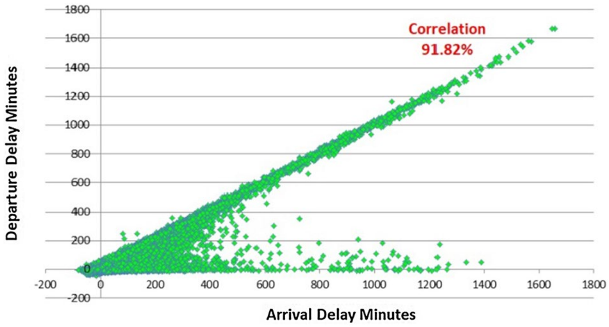


Fig. 2 Correlation between US flight departure and arrival delay

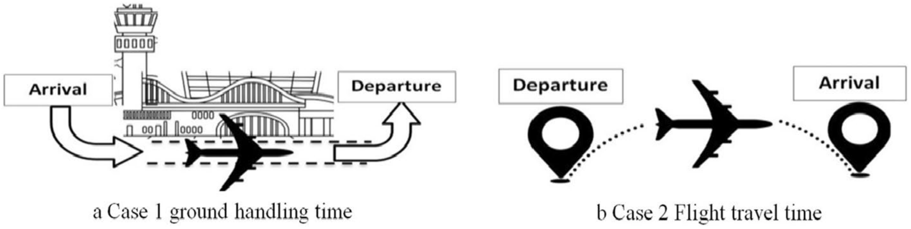
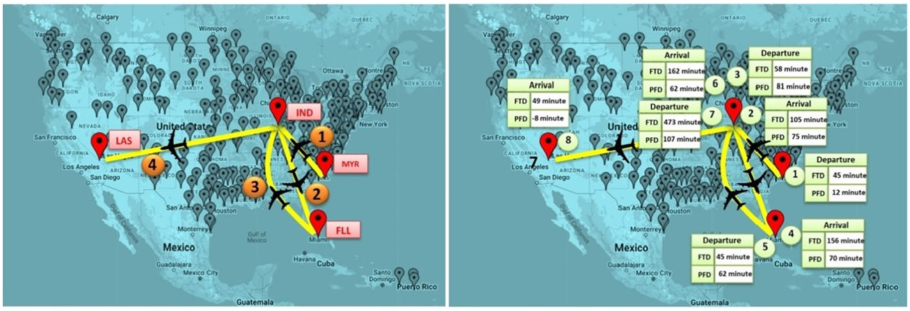


Fig. 3 Flight duration time (FDT)

Delay" feature (PFD) to enhance prediction, for example, the previous arrival delay feature was used to enhance the prediction next of departure delay in studies [20, 21]. The reverse is also, using the previous departure delay feature to enhance the prediction of the next arrival delay [22]. And others used a delay of arrival and departure feature to predict the corresponding delay of departure and arrival, respectively [23]. However, in common practice, this action PFD gives a forecast horizon for delay just a few minutes before flight time equal to the distance between the two airports, which lessens taking advantage of prediction. For example, on US domestic flights the shortest distance from Wrangell International Airport (WRG) to Petersburg International Airport (PSG) is 31 miles, with 33 min flight duration [24]. That means when we rely on the opposite delay feature to obtain the departure delay feature in this flight to forecast flight arrival delay, that is not available only 33 min before the flight arrival time, leading to a weak flight delay forecast horizon. As a result, the studies have a critical issue in that it is difficult to interpret the variables or features that influence the enhanced delay prediction, so resorted to using the PFD opposite delay feature, which leads to reduce the forecast horizon to a few minutes. In this context, we proposed an FDPP-ML algorithm for a flight delay prediction solution by utilizing the previous flight delay feature PFD advantages, but at the same time providing a longer forecast horizon effectively depending on only basic flight schedule features. On another hand, we established a second new feature called "Flight Duration Time" (FDT) which is possible to be one of two cases. Figure 3 shows the cases of flight duration time FDT Fig. 3a, shows case 1 ground handling time, represented flight duration time between arrival and departure means the aircraft case turnaround to provide resources ground handling time until flight departure and this turnaround process performance relates to the amount of flight delay [7]. Figure 3b shows case 2 flight travel time, representing flight duration time between departure airport origin and arrival destination airport, as much as traveling time increases, the flight has an opportunity to reduce delays caused by the previous point. as given in these two cases there is a relationship between FDT and flight delay, we established this feature as one of the proposed features created to support ML models and play their role for discover relations and improving delay prediction.

An overview of proposed FDPP-ML

The proposed FDPP-ML contains an algorithm to create new flight features side-by-side to support machine learning models to capture the impact of delay propagation over the flight network and their impacts on future individual flights on the same path, which contains three phases, the first is an algorithm for a data-driven approach capable of using and organizing flight features to catch the paths in historical data, then re-structured flight data and transform flights from default to points stops in the flight path. Figure 4 explains the flight paths, Fig. 4a represents default flights Fig. 4b represents reshaped flight path to points by algorithm, the algorithm dismantles the default flight's record and transforms it to flight points, thus each flight is converted to two points as departure and arrival and converted to an integrated flight path, then the algorithm creates each point two new features the first feature is flight time duration (FTD) represented the difference time between the current point and the previous point, the second feature is the previous flight delay feature (PFD) that mentioned before of how to utilize the worth of using opposite previous flight delay feature for effective current flight prediction. The second phase is the ML model role for training on a recently formed data set from phase one which contains flight schedule features with new two features created (FDT, PFD), then phase three contains the algorithm with a model trained to predict flight delay for new future flights, the algorithm takes the model output of flight delay predicted to be a PFD feature for the next flight point in the same path, and the algorithm looping continues to finish the end of all flight paths. In this regard, we utilize



a default flights

b reshaped flight path to points

Fig. 4 Flight paths

the worth of the PFD feature and reach the best solution to improve flight delay prediction with a longer time forecast horizon than a few minutes before flight time equal to the distance between the two airports, we take into consideration implementing three different forecast horizons 2, 6, and 12 h to measure how far FDPP-ML outperformed based the previous flight delay prediction than traditional training models.

The major findings of the study represented to improve the literature on flight delay modeling summarized as follows: (1) It is the first attempt to develop flight delay prediction individually based only on basic flight features. (2) Support ML regression models to improve prediction by the proposed algorithm FDPP-ML to be implemented into 10 regression Models to prove outperformed. (3) The power of FDPP-ML prediction results based on US flight networks' real-world data provide analysis insights into 366 airports and 10 airlines involved in the estimation of their future flight delay minutes per flight. The remaining paper is organized as follows: "Literature reviews" Section illustrates the literature reviews on flight delay prediction. "Methodology" Section illustrates the methodology and basic concepts. "Implement" Section illustrates the implementation of the proposed FDPP-ML model. "Result and discussion" Section illustrates our model results and discusses the comparison of proposed state-of-the-art and basic models.

Literature reviews

Lately, because of the massive data involved in flight features, the development of machine learning is a perfectly viable technique for forecasting flight delays. Previous studies have been researched, analyzed, and benefitted from by focusing on the primary aspects to better enhance prediction in this suggested strategy. Flight delay prediction has tended to learn a single model for flight delay prediction, while an individual model can't make a sensible explanation of complicated flight features. The studies resorted to ensemble learning which alternate strategy for overcoming this bottleneck, developed to enhance flight delay binary classification using ensemble voting classification and [18], ensemble stacking [25], despite these studies' interest in binary classification and neglecting the estimated duration time of flight delay, we will already concern it in our approach by predicting delay duration time in minutes, but flight delay prediction needs to evolve not narrowly by a new strategy that has a group of models only such as ensemble strategy, but by a mechanism capable of moving all models performance to an up level, that is provided by proposed our approach. In interesting regression models to predict flight delay [26], estimate average delay for airlines, our proposed approach is more robust by using flight features for individual flight delay prediction.

From the perspective scope of airport application [13], proposed a model for capturing flight delay effects of en-route traffic congestion in an air traffic network, including a data-driven technique and a cluster model to quantify delays in China's 56 airport air traffic network, this study suitable for analyzing and developing improvement methods for airport traffic management. Also [27] proposed graph architecture-based learning with an attention mechanism (AG2S-Net) to multi-step-ahead hourly predicted for arrival and departure delay of the traffic network with 75 airports, providing the lowest RMSE and MAE values when estimated delay. On another hand, implemented in a wider traffic network of 366 airports and 10 airlines for US flights, the proposed is capable to implement over a worldwide traffic network if the dataset existed. In addition, the proposal gives accurate flight delay minutes for individual flight, it's widening to provide benefits, and be effective for airports and airlines. While the studies on comparative Ml algorithms revealed flight delay prediction has constraints like the number of airlines and airports involved [2]. Proposed flight data and weather variables created using data correlation by big data analytics, the proposed prediction model of on-time arrival flight explores the relationship between pressure patterns and flight data for just one airline Peach-Aviation [25]. Suggested comparing ML models for predicting arrival delay using flying and weather parameters, using 5 years of US flights but just with 45 airports, the RF outperformed other models with an accuracy of 80.36% [28]. Analysis of arrival delay prediction for the top five busiest airports in US flights, and proposed a comparison using four ML algorithms, (GBC) outperformed accuracy was 79.7%, the default in improvement to the variables has reflected a low level of accuracy. However, the same author in the study [29] used the SMOTE method for balanced data in binary classification, the accuracy increased from 80.89 to 85.73%. The author advocated predicting flight delays by taking into account air traffic flow to improve flight individual delay prediction. However, the domain of implementation is only in five airports in the US flights. According to previous studies besides having limitations in the scope of airports, the (GBC) is outperforming other models, so will be nominated for using gradient-boosted regression (GBR) as one of the benchmark models in our approach. In addition, highlight the advantage of the regression model used in the proposed approach because not needs a method for balancing binary value to be equivalent such as (SMOTE), and the regression model predicts continuous minute delay and provides more near-delay near-delay minute values [25]. Presented stacking binary classification of delays using six algorithms, the RF accuracy was 0.822 However; the implementation considers binary categorization and just two airports in the US flights [30]. Proposed stacking regression with proposed two novel variables, arrival/departure pressure, and cruise pressure, and proposed sine and cosine functions to convert data to (x, y) including hours of the day, days of the week, and months in a year. This approach could be effective in creating these features when using one airport as the author used Beijing Capital International Airport and one route from PEK to HGH was utilized to conduct a case study. It's a huge extreme and impossible to create when applied to a wide flight network connected to each other such as our approach to US 366 airports involved [31]. Constructing a machine learning model RF to predict delay time, the implementation was on US flights for one air carrier, it's minimal effective for airport systems because has a shortage of carriers in the dataset [32]. Proposed a gated recurrent unit (GRU) model to predict flight departure delays using the flight of (ZSNJ) airport in China, outperforming other models. In our proposal taking into consideration the GRU model as one of the set ML models to ensure our proposed FDPP-ML is robust. In addition, predict flight departure and arrival delays given its importance [18]. Proposed RF and LSTM models with create a dataset from automatic dependent surveillance-broadcast (ADS-B) messages and integrated weather conditions, flight schedules, and airport information [16]. Proposed RF classification and regression with special factors such as previous flight delays PFD, airport crowdedness, wind direction, the extent of weather conditions, wind speed, and so on. When using all special features, has an accuracy of 96.48%, whereas with basic flight a feature was 89.46%. The special features improved accuracy, but accumulating particular airport data makes it more difficult to obtain these features especially when using a wide airport network.

The selection of flight features and special information of airports have a significant impact on the accuracy of flight delay prediction, Which encourages our approach to improve flight delay based on only basic flight schedules minimize specific information elements used for airports, and improve accuracies based on appropriate data represented with just flight schedule features. In addition, using the RF model according to contributed in the last studies, we nominated using RFR regression in the benchmark model approach.

The time of forecast horizon before flight time in studies to indicate the beginning of the forecast horizon to be useful to apply in our proposed approach, as the following studies resorted to the prediction of just a two-hour forecast horizon before flight time to enhance flight delay prediction [9], proposed a framework considered an agent-based delay prediction model, containing two RFR models based on a conditional probability model, using US flight delay on the Ground Delay Programmes (GDPs) or carrier-related reasons, The prediction model was 89.5% accurate using a 15-min threshold for only a 2-h forecast horizon [17]. Proposed hybrid Deep belief network to mine the inner patterns, and support vector regression (DBN-SVR) to perform supervised fine-tuning inside the current predictive architecture, other ML models were outperformed by the suggested DBN-SVR. Studies take into consideration the forecast prediction horizon two hours before the intended flight time and use specific airport information such as (GDPs). Accordingly, the proposed approach takes into consideration the forecast horizon of two hours to prove results improved. The usage of the departure delay feature to predict arrival delay called PFD action caused to limits the prediction horizon to a few minutes, however studies taking into consideration this feature to enhance prediction, for example [33] presented the XGBoost model, and LR model to determine the link between independent and dependent factors, use of the departure delay feature to predict arrival delay aided in achieving an accuracy of 94.2% in their dataset, as [23] used the departure delay feature to enhance predict arrival delay using a framework that combines the Social Ski Driver algorithm (SSDCA based LSTM), using US flights, the SSDCA enhances the network with an accuracy of 92.68% and was less error than other state-of-art models. In another hand, we utilize PFD action in our FDPP-ML model to handle to make the forecast horizon longer than flight travel time [26]. Proposed a framework for flight delay using a set of models regression, the experiments used four dataset kinds based on four sorts of flight characteristics (flight schedule—weather—airport GPS trajectories awareness map air traffic). LightGBM has the best results with the lowest error, which makes us consider using the LightGBM model as one of the benchmark models. proposes deep Learning based on an autoencoder hybrid with the Levenberg-Marquart algorithm (SAE-LM), to identify the right weights and biases of great complexity while using enormous quantities of data, and has outperformed other two state-of-the-art models, it's a useful approach to improving prediction within individual models, on the other hand, we seek to improve the flight delay prediction by finding a new approach to support all ML models to prediction enhance [34].

However, most existing studies on individual flight delay modeling do not consider alternative solutions rather than developed machine learning or deep learning models to enhance flight delay prediction, which led to using private information hard to obtain to implement in other airports, studies have enhanced prediction airports and airlines just within limited scope implementation or relied on flight delay feature for departure to enhance arrival delay prediction (PFD) action which leads to minimized forecast horizon of time in advance. Therefore, we propose a novel flight delay model, to fill in the research gap, (FDPP-ML) utilizes PFD action to make the forecast horizon longer than flight travel time, capable of improving flight delay prediction models to an up level of accuracy based only on basic flight schedule features even with a huge flight traffic network.

Methodology

Proposed approach (FDPP-ML)

The proposed approach FDPP-ML works on disassembly flights and re-structure them to catch flight paths in historical data, and organizing each path to be a sequence of points of flights, starting from the first beginning point whether arrival or departure, and following the end of the path. Then the FDPP-ML framework creates for each point in track two new features called (flight time duration, and previous flight delay). The FDPP-ML uses ML to train historical data with these new features, then works on future flights to organize and put flights on their paths to inherit their previous delay, the algorithm makes the ML predict flight delay for each flight in the path, and the algorithm continues side by ML to inherit this predicted delay to the next flight on the same path, and continues to be all paths finished and complete prediction of future flights delay. The proposed FDPP-ML algorithm was developed to help provide the proposed approach represented with pseudo-code in Algorithms 1–4 showing the proposed FDPP-ML model. Figure 5 shows the architecture of the proposed FDPP-ML, which contains 8 phases explained as follows:

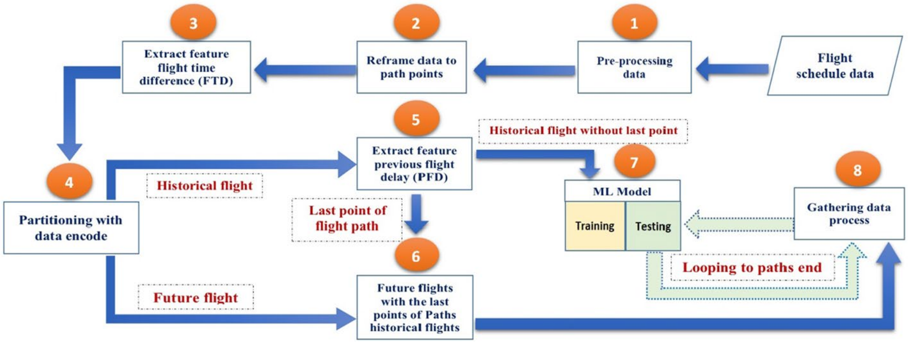

3
Extract feature flight time difference (FTD)
2
Reframe data to path points
1
Pre-processing data
Flight schedule data
5
Historical flight without last point
Extract feature previous flight delay (PFD)
4
Historical flight
7
ML Model
Training
Testing
8
Gathering data process
Partitioning with data encode
Last point of flight path
6
Looping to paths end
Future flight
Future flights with the last points of Paths historical flights

Fig. 5 Architecture of proposed FDPP-ML

Pre-processing data

The pre-processing step eliminates irrelevant data and saves just crucial data to ensure coherence before FDPP-ML work, the data is organized and prepared using removing duplicates and missing values with, null values removed, and removed flights canceled which represents the percentage (2%) of total data.

Reframe flight schedule to path points

The flight schedule data frame has each flight feature details from the departure (origin) airport to the arrival (destination) airport, in this stage each flight is restructured into two points containing departure and arrival, which means each point inherits the same flight features to prepare for creating new two features. Algorithm 1 represents steps 1 and 2.

Flight time duration (FTD)

Before start creating new features (FTD, PFD), we use sort values based on these features "Tail number" then "Schedule date time" and the algorithm works on divides the flights to paths. The worth of the FTD feature was explained in the introduction section as how may impact and has a relation with flight delay time, the FDPP-ML establishes the calculation of the flight time duration in minutes between the current point (departure, arrival) and the previous point which on the same path. The FTD feature is extracted based on (the flight schedule date and times feature) which airports obtain in advance, when we restructured flights and transformed them into points, we unified these (flight schedule date and times features) of departure or arrival into one feature renamed "Schedule date time". Then FDPP-ML calculates the FTD feature for all flights including historical with future flights.

Partition with data encode

This step converts sting and categorical data to numerical with encoded flight features to prepare for the subsequent training and testing using ML models. In addition, flight data is partitioned based on the current time into two partitions the first is historical flights before the current time and future flights after the current time. In other words, the time of implementing the proposed algorithm for forecast horizon future flights. The historical flight moved to step 5, and the future flight moved to step 6. Algorithm 2 represents steps 3 and 4.

Extract previous flight delay feature (PFD)

The action of using the PFD feature in studies was explained in the introduction and literature sections how it's more improves flight delay prediction accuracy but is useless because this feature is obtained only before flight time equal to flight duration time whether arrival to departure or vice versa. We used this action and improved by FDPP-ML and handled the forecast horizon to be longer. Thus, in the proposed FDPP-ML the ML model predicts the flight delay for the first flights on the paths, and the algorithm passes the predicted flight delay for the next flight that is on the same paths. Each flight record has schedules and actual times for both departure and arrival, the difference between the scheduled time and actual time represents the delay of departure or arrival. When we restructured flights and transformed them into points, we unified these delays of departure or arrival into one feature renamed "Flight delay" which represents the value of minutes. The PFD feature is extracted only for historical flights based on (the previous flight delay feature) that are on the same path. The FDPP-ML extracting of the PFD feature is only for historical flights, on other meaning that flights before the current time become with PFD feature to be training data for Ml models represented a value in minutes. At this step, the algorithm cuts the last flight only for all paths and moves these flights to step 6 future flights.

Future flights with the last points of Paths historical flights

In this step, after partitioning and handling the data is ready for training and testing using ML models. The historical flight's features before the current time become with PFD feature. The last point for each path of historical flights will be cut and merged with future flights, to be the first point for each path of future flights, to guarantee each first flight on paths has a PFD feature which machine learning trained on it and the remaining future flights did not have PFD value, this is the future flights which need to predict a flight delay with complete their paths points, this mission will be executed at another step 8. Algorithm 3 represents steps 5 and 6.

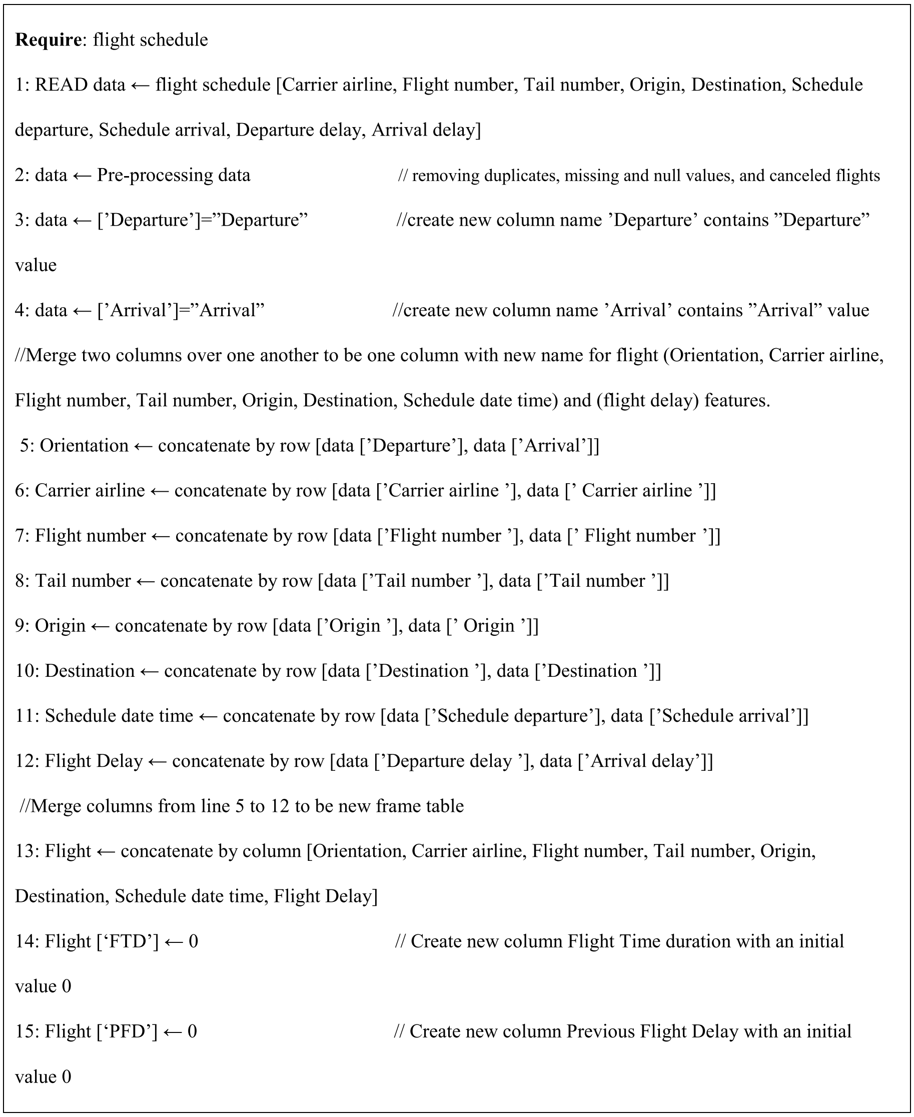
Machine learning

The FDPP-ML passes historical flights to the ML model for training with support to improve prediction by new features created (FTD, and PFD), the ML model considers a built-in FDPP-ML and is ready to predict flight delay for future flights. The FDPP-ML framework depends upon the primary model but in experiments trained 10 regression models that would later explain to illustrate the algorithm contribution to improve ML models for flight delay prediction.

Gathering data

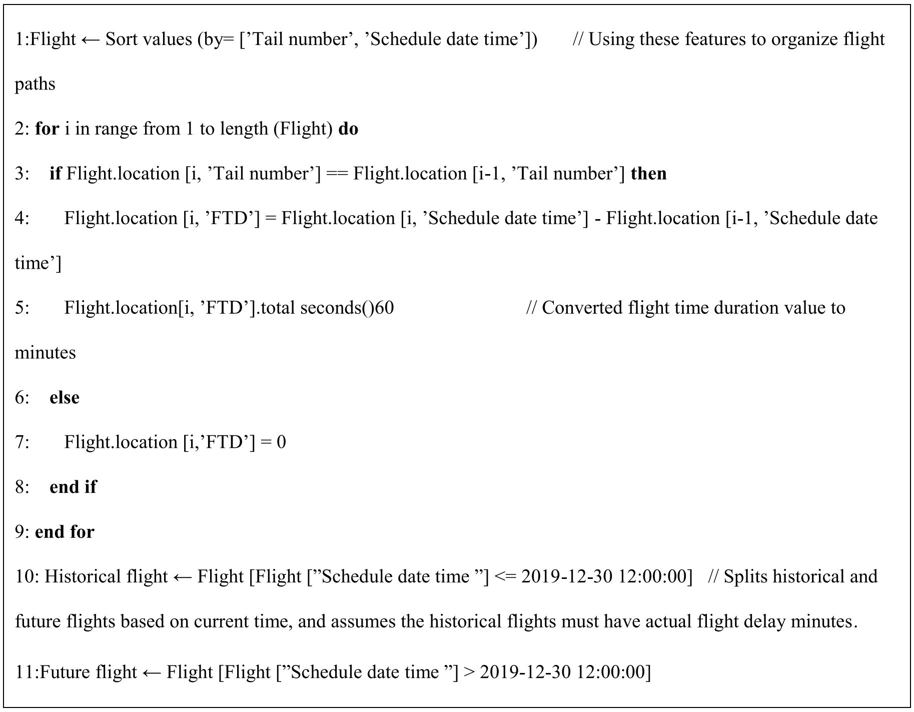
In this step, the FDPP-ML will pass each first point in the paths of future flights which has a PFD feature to the model for predicting the delay of this point or flight. Then the predicted flight delay merged to the next point to be the PFD feature to the next flight on the same flight path, and the algorithm resends the new flight (point) to the model for predicting the delay with the PFD predicted feature. The FDPP-ML will repeat these steps to complete this current path and move to the next path to complete their points and so on. Finally, the FDPP-ML algorithm completes predicting flight delay to all of the specific horizons time of future flights. Algorithm 4 represents steps 7 and 8.

Algorithm 1 FDPP-ML steps 1 & 2



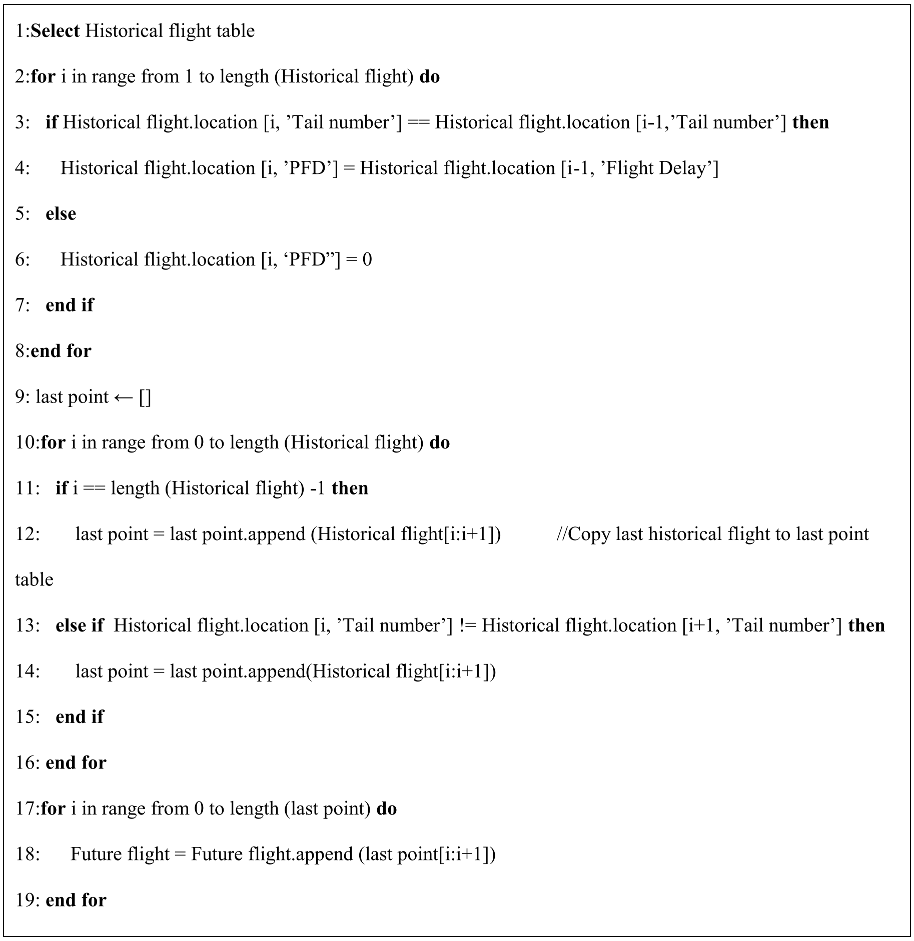
```
Require: flight schedule
1: READ data ← flight schedule [Carrier airline, Flight number, Tail number, Origin, Destination, Schedule departure, Schedule arrival, Departure delay, Arrival delay]
2: data ← Pre-processing data                // removing duplicates, missing and null values, and canceled flights
3: data ← [’Departure’]=”Departure”          //create new column name ’Departure’ contains ”Departure” value
4: data ← [’Arrival’]=”Arrival”              //create new column name ’Arrival’ contains ”Arrival” value
//Merge two columns over one another to be one column with new name for flight (Orientation, Carrier airline, Flight number, Tail number, Origin, Destination, Schedule date time) and (flight delay) features.
 5: Orientation ← concatenate by row [data [’Departure’], data [’Arrival’]]
6: Carrier airline ← concatenate by row [data [’Carrier airline ’], data [’ Carrier airline ’]]
7: Flight number ← concatenate by row [data [’Flight number ’], data [’ Flight number ’]]
8: Tail number ← concatenate by row [data [’Tail number ’], data [’Tail number ’]]
9: Origin ← concatenate by row [data [’Origin ’], data [’ Origin ’]]
10: Destination ← concatenate by row [data [’Destination ’], data [’Destination ’]]
11: Schedule date time ← concatenate by row [data [’Schedule departure’], data [’Schedule arrival’]]
12: Flight Delay ← concatenate by row [data [’Departure delay ’], data [’Arrival delay’]]
 //Merge columns from line 5 to 12 to be new frame table
13: Flight ← concatenate by column [Orientation, Carrier airline, Flight number, Tail number, Origin, Destination, Schedule date time, Flight Delay]
14: Flight [‘FTD’] ← 0                       // Create new column Flight Time duration with an initial value 0
15: Flight [‘PFD’] ← 0                       // Create new column Previous Flight Delay with an initial value 0
```

Algorithm 2 FDPP-ML steps 3 & 4



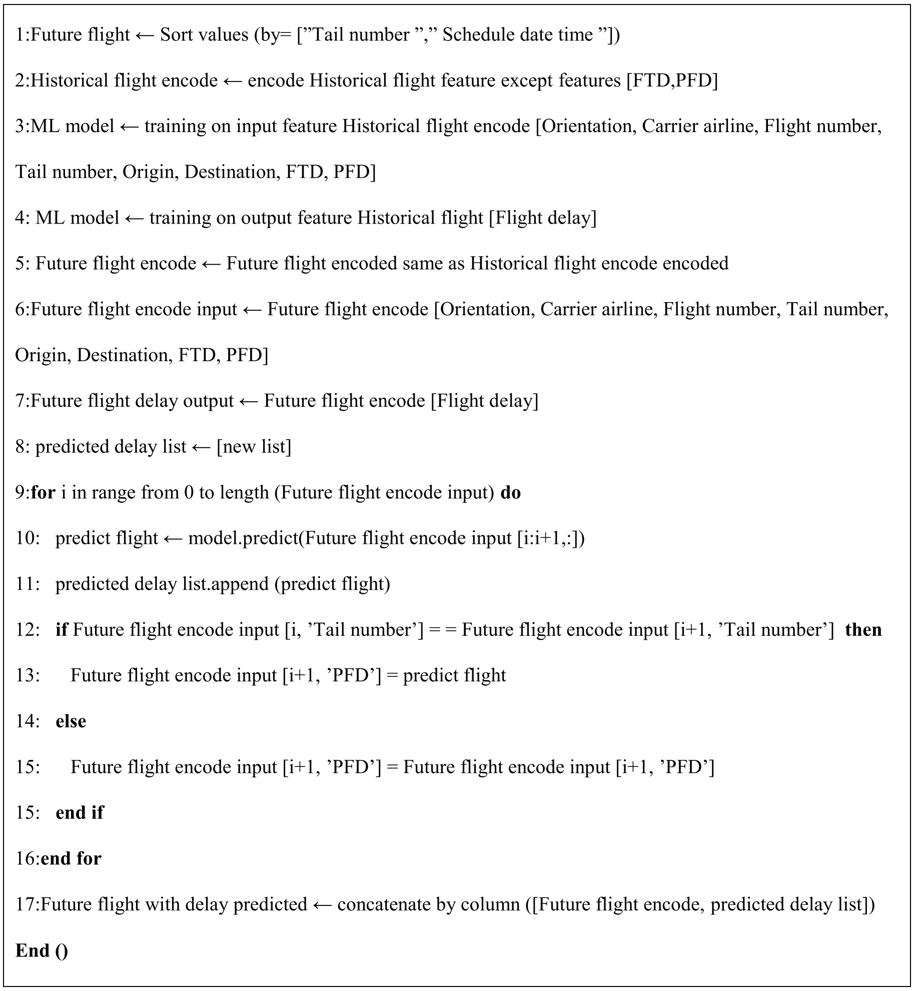
```
Flight ← Sort values (by= [’Tail number’, ’Schedule date time’])       // Using these features to organize flight paths
for i in range from 1 to length (Flight) do
    if Flight.location [i, ’Tail number’] == Flight.location [i-1, ’Tail number’] then
        Flight.location [i, ’FTD’] = Flight.location [i, ’Schedule date time’] - Flight.location [i-1, ’Schedule date time’]
        Flight.location[i, ’FTD’].total seconds()60                            // Converted flight time duration value to minutes
    else
        Flight.location [i,’FTD’] = 0
    end if
end for
Historical flight ← Flight [Flight [”Schedule date time ”] <= 2019-12-30 12:00:00]   // Splits historical and future flights based on current time, and assumes the historical flights must have actual flight delay minutes.
Future flight ← Flight [Flight [”Schedule date time ”] > 2019-12-30 12:00:00]
```

Algorithm 3 FDPP-ML steps 5 & 6



```
Select Historical flight table
for i in range from 1 to length (Historical flight) do
  if Historical flight.location [i, ’Tail number’] == Historical flight.location [i-1,’Tail number’] then
    Historical flight.location [i, ’PFD’] = Historical flight.location [i-1, ’Flight Delay’]
  else
    Historical flight.location [i, ‘PFD”] = 0
  end if
end for
last point ← []
for i in range from 0 to length (Historical flight) do
  if i == length (Historical flight) -1 then
    last point = last point.append (Historical flight[i:i+1])      //Copy last historical flight to last point table
  else if  Historical flight.location [i, ’Tail number’] != Historical flight.location [i+1, ’Tail number’] then
    last point = last point.append(Historical flight[i:i+1])
  end if
end for
for i in range from 0 to length (last point) do
    Future flight = Future flight.append (last point[i:i+1])
end for
```

Algorithm 4 FDPP-ML steps 7 $ 8



```
Future flight ← Sort values (by= [”Tail number ”,” Schedule date time ”])
Historical flight encode ← encode Historical flight feature except features [FTD,PFD]
ML model ← training on input feature Historical flight encode [Orientation, Carrier airline, Flight number, Tail number, Origin, Destination, FTD, PFD]
ML model ← training on output feature Historical flight [Flight delay]
Future flight encode ← Future flight encoded same as Historical flight encode encoded
Future flight encode input ← Future flight encode [Orientation, Carrier airline, Flight number, Tail number, Origin, Destination, FTD, PFD]
Future flight delay output ← Future flight encode [Flight delay]
predicted delay list ← [new list]
for i in range from 0 to length (Future flight encode input) do
    predict flight ← model.predict(Future flight encode input [i:i+1,:])
    predicted delay list.append (predict flight)
    if Future flight encode input [i, ’Tail number’] = = Future flight encode input [i+1, ’Tail number’]  then
        Future flight encode input [i+1, ’PFD’] = predict flight
    else
        Future flight encode input [i+1, ’PFD’] = Future flight encode input [i+1, ’PFD’]
    end if
end for
Future flight with delay predicted ← concatenate by column ([Future flight encode, predicted delay list])
End ()
```

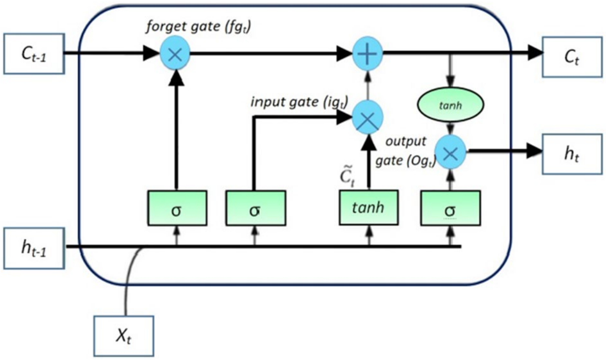


Fig. 6 Basic architecture of LSTM model

Machine learning regression models

In this section, we overview models implemented with machine learning benchmarks and advanced models to highlight how to improve performance with the proposed FDPP-ML.

RNN and LSTM

RNNs are robust models that perform classification and sequential inputs such as text and audio. Due to the exploding gradient and vanishing gradient resulting from the primary issue of long-term reliance when the distance between the pertinent information and the point at which it is needed widens, the RNN cannot learn to link the information. The Lstm is a kind of RNN that has the capacity to selectively store and retrieve information and can avoid this problem of long-term dependency by using certain hidden units. Figure 6 shows the architecture of LSTM, the Lstm's repeating units each have four neural components, the forget gate layer, the input gate layer, the output gate layer, and all memory cells are responsible for remembering values over time [35], the Lstm unit at the current time step t, the input gate represented by (igt), while the forget gate represented by (fgt), and (ogt) represents the output gate. The internal cell state is represented by (Ct), while the hidden state known as the unit's output represented by (ht). Figure 6 demonstrates the Lstm's fundamental architecture, the calculated controls how the Lstm processes data in the following ways:

$$ig_t = \sigma\left(W_{ig}x_t + U_{ig}h_{t-1} + B_{ig}\right) \tag{1}$$

$$fg_t = \sigma\left(W_{fg}x_t + U_{fg}h_{t-1} + B_{fg}\right) \tag{2}$$

$$og_t = \sigma\left(W_{og}x_t + U_{og}h_{t-1} + B_{og}\right) \tag{3}$$

$$C_t^{\sim} = tanh(W_c x_t + U_c h_{t-1} + B_c) \tag{4}$$

$$C_t = fg_t \odot c_{t-1} + ig_t \odot C_t^{\sim} \tag{5}$$

$$h_t = og_t \odot \tanh\left(c_t\right) \tag{6}$$

The activation function is represented by σ, which is a sigmoid function here, and ⊙ conducts multiple elements to elements. The responsible weight vectors are represented by (W and U), respectively, while the responsible bias vectors are by (B), The memory cell state's potential candidate values vector is managed by $\left(\mathrm{C}_{\mathrm{t}}^{\sim}\right)$, to obtain the cell state, which is coupled with the input gate by$(\mathrm{C}_{\mathrm{t}})$, Eq. (1) shows the input gate, Eq. (2) represents the forget gate. The output gate, represented by Eq. (3), and Eq. (4) shows the instantaneous information $\left(\mathrm{C}_{\mathrm{t}}^{\sim}\right)$ is obtained by putting the previous current input and output through a tanh function. The outputs of Eq. (3) and (5) are multiplied by the information about the current memory and input gates in Eq. (5). Finally, The LSTM output (Ot) is created by multiplying this data by the output gate result from Eq. (6) after it has passed through a tanh layer [36].

GRU

In [37] 2014 proposed Gated Recurrent Neural Networks GRU similar to LSTM, but simpler to calculate and apply. The hidden state output at time t is calculated using the input time series value at time t and the hidden state at time t-1. GRU is introduced but with attention to LSTM's inherently complicated and unpredictable learning process, and combines hidden states and neurons, combines the input gate and forget gate into a single update gate, and reduces the number of network parameters to hasten training. The four following formulae are used by the GRU neural network to extract features from the original data.

$$rg_t = \sigma(W_r[h_{t-1}, x_t]) \tag{7}$$

$$\overline{h_t} = \tan\big(W_h[rg_t \times h_{t-1}, x_t]\big) \tag{8}$$

$$z_t = \sigma(W_z[h_{t-1}, x_t]) \tag{9}$$

$$h_t = (1 - z_t)h_{t-1} + z_t \times \overline{h_t} \tag{10}$$

We consider the current time input to be $\mathrm{x_t}$, and the hidden state of the earlier moment is represented in $\mathrm{h_{t-1}}$, and $\overline{\mathrm{h_t}}$ represents the hidden state calculated at the current moment. Equation (7) represents the calculation of the reset gate $\mathrm{rg_t}$ value, while σ is the sigmoid activation function, and $\mathrm{W_r}$ is the hidden state responsible for the reset gate layer. Equation (8) represents $\overline{\mathrm{h_t}}$ hidden state calculation. Equation (9) is a function to calculate the update gate value, Eq. (10) calculates the current hidden state at the end, and the hidden state is supplied to the next layer of neurons [32]. The function of reset gates is similar to forget gates of LSTM, as GRU NNs have several similarities to LSTM NNs, we will not go deep into the detailed formula. the regression part and optimization method we use in this paper for GRU NNs are the same as the LSTM NNs. On the side deep learning models used in this paper, we used RNN, LSTM, and GRU models.

Gradient boosting

Freidman [38] Explains the Gradient Boosting GB approach in his work, because it is an upgraded form of the gradient-boosting decision tree technique that has been expressly designed to incorporate categorical features. It employs binary decision trees as its primary predictors, the data during training is randomly mixed and reshuffled numerous times to determine the mean for each item based only on its previous data [39]. We begin with considering the dataset $\mathrm{D} = \{(\mathrm{x_i}, \mathrm{y_i})\}_{\mathrm{i}=1......,\mathrm{n}}$ the number of iterations to be n, and a differentiable loss function $\mathrm{L} = (\mathrm{y}, \mathrm{F}(\mathrm{x}))$. The goal of the boosting strategy is to find a function that minimizes the loss function as closely as possible. The mapping function is termed $\mathrm{F_0}(\mathrm{x})$ is following describes how to minimize the loss function using a constant value in Eq. (11):

$$\mathrm{F_0(x) = arg\,min_f} \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{L\big(y, F(x)\big)} \tag{11}$$

The pseudo-residuals ($\mathrm{r_{im}}$) are calculated from iteration 1 to *n* to resolve the optimization problem, as shown in Eq. (12).

$$r_{im} = -\left[\frac{\partial L\big(y_i, F(x_i)\big)}{\partial F(x_i)}\right]_{F(x)=F_{m-1(x)}} \quad for\ i = 1, \ldots .n \tag{12}$$

Moreover, the base learner $\mathrm{\beta_m(x)}$ is carried to $\mathrm{r_{im}}$ by using the training set $\mathrm{\{(x_i, r_{im})\}_{i=1}^{n}}$ the multiplier represented in $\mathrm{y_m}$ is calculated next step by the following one-dimensional solving optimization problem represented in Eq. (13).

$$y_m = \mathrm{arg}\, min_f \sum_{i=1}^{n} L\big(y, F_{m-1}(x_i) + y\beta_m(x_i)\big) \tag{13}$$

Consequently, by utilizing the following Eq. (14), the model may be updated to generate the output

$$F_m(x) = F_{m-1}(x) + y_m\beta_m(x) \tag{14}$$

An Eq. (12)-(14) demonstrates the categorical gradient boozsting method's significant steps, where splitting parameters, cut points and individual tree nodes are included in the parameterized function.

Categorical boost

Prokhorenkova et al., propose and compare the CatBoost CAT algorithm to XGBoost and LightGBM. In their explanation of the CatBoost learner, they discuss their improvements to the GBDT method described by Friedman [40]. CatBoost is a part of the GBDT machine-learning ensemble approach family. CatBoost has been utilized effectively by academics for machine learning projects including prediction since its launch in late 2018 [41], we use this chance to include CatBoost in our prediction model and gain best practices from research that promotes CatBoost. CatBoost encrypts categorical variables to address the issue of target leakage. CatBoost is also well suited to machine learning applications involving category, heterogeneous data because it is a Decision Tree-based technique. Recent works from a variety of areas demonstrate CatBoost's usefulness and limitations in classification and regression challenges. CatBoost is its sensitivity to hyperparameters and the importance of hyperparameter tuning [42], which is a perfect model we are interested in implementing in our experiments approach.

Random forest (RF)

The RF technique used in this work is considered a successful model, despite the major of ML research in the field of Autism Spectrum Disorder ASD [43], RF is one of the important models used to diagnose it [44]. RF is a supervised machine learning approach that employs an average of many decision trees to reduce uncertainty and overfitting [45]. It outperforms a basic regression tree, especially when the training data is minimal, hence it is sensitive to the training data used to create the tree. Furthermore, can handle huge datasets, is more interpretable, and incorporates many characteristics, based on the architecture of the tree structure's expectation, which is all assessed using the same distribution, the random forest approach may rank the relative significance of each predictor variable. The bootstrap technique is used to build a forest of decision trees, with each tree created individually using a subset of predictor data. Without pruning, the trees grow to their maximum size, and the ultimate output means from all of the multiple decision trees are the conclusion, RF providing efficient regression performance [46]. Because of its improved stability and generalization, it has a wide range of applications. These characteristics motivated us to use the random forest regression RFR model.

LightGBR

LightGBM is another GBDT algorithm that supports the automatic encoding of categorical features. We used it in the first level because many of the works proved it has high accuracy and always compares the performance of CatBoost. Lightgbm is a highly efficient gradient boosting decision tree. LightGBM's support for categorical features, has an advantage in general it uses a fast, distributed, high-performance gradient lifting decision tree technique based on the Histogram algorithm [41]. The main distinction between it and other approaches is that it breaks the tree based on leaves, allowing it to find crucial spots and cease computations (other lifting algorithms are level-based or depth-based). When the number of leaf nodes in a tree is increased, the leaf-wise approach minimizes more loss than the levels technique, resulting in higher accuracy. The downside of leaf-wise is that it can create overfitting by growing a somewhat deep decision tree. As a result, LightGBM has a maximum depth restriction leaf-wise, assuring great efficiency while limiting overfitting. Faster training times, more efficacy, improved accuracy, reduced memory utilization, and support for parallel and distributed computing are some of the benefits of the LightGBM. The maximum tree leaves for base learners (num_leaves) and the maximum tree depth for base learners (max_depth) are the major hyperparameters to modify when creating a LightGBM model [47].

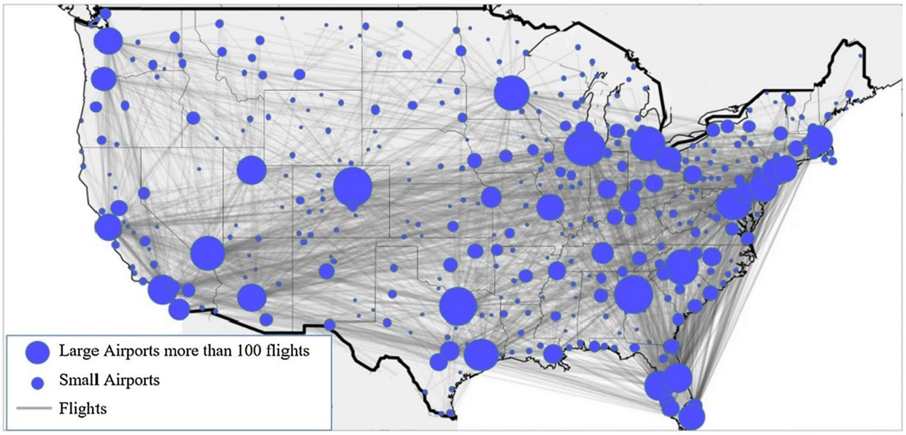


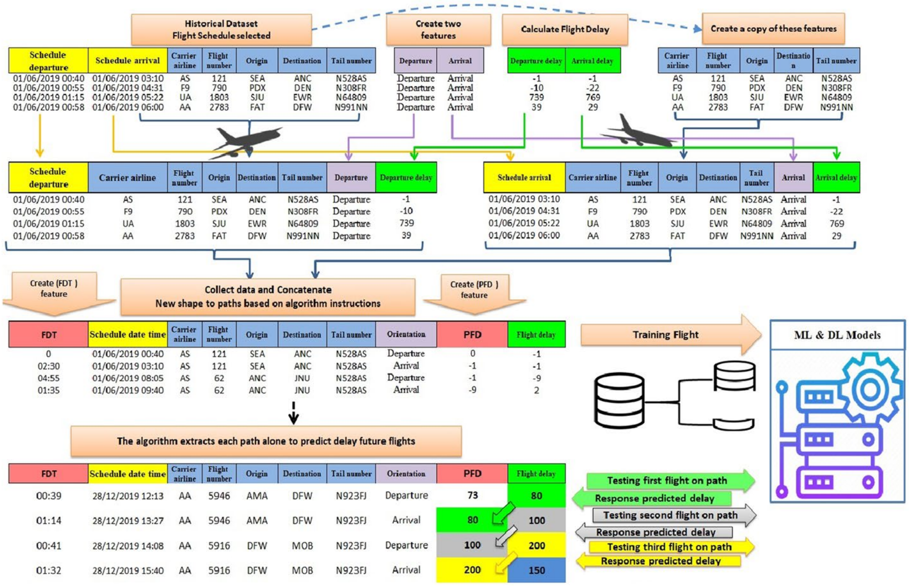
Fig. 7 Sample of US flight network visualization

Ensemble learning

By combining a number of weak learners into a single strong learner, ensemble algorithms have been successful in machine learning. By combining the predictions from several classifiers into a single, reliable prediction, ensemble learning can increase the effectiveness of classification [48]. The primary goal of ensemble learning is to decrease the likelihood of picking a single learning algorithm that performs badly and to improve the performance of one algorithm by employing an intelligent ensemble of several distinct algorithms [49]. The value of ensemble learning has been well proved in extensively demonstrated. One of the states-of-art in ensemble transform learning (TL) [50], outperformed rice disease detection over ML [51], also ensemble averaging of transfer learning models [52], outperformed the advanced learning model like Convolution-XGBoost [53]. We used the Voting model as one of the benchmark models, a prediction model is created by fusing the skills of several basic base learners. Despite ML models utilized for identifying the depression diagnosis [54], the majority voting enhanced by carrying out in all possible ML selection techniques [55]. The same dataset is used to train many separate models, each of which produces independent predictions and has a unique capacity for learning. These predictions are subsequently used as input for ensemble learning to produce a final prediction that is more reliable and error-free. Voting regression is one ensemble learning technique that integrates the predictions from several individual models as base learners. Because VR is flexible and can be used with any base learner that represents a variation bias trade-off, it is more effective at reducing reducible error. The suggested approach's fundamental component is Stacking, which often takes into account heterogeneous weak learners, trains them concurrently and then combines them to provide a new prediction based on the predictions of the multiple weak models [56]. The meta-learner receives its input from the first-level learners' output. Although it is feasible to build stacked ensembles using the same learning

Table 1 Flight schedule features used

Type	No	Flight features used	Model input and output	Data type	Attribute type	Variables format
Airport	1	Origin	Input	Object	Categorical	3 digits letters code eg. SEA
	2	Destination	Input	Object	Categorical	3 digits letters code eg. JFK
Carriers	3	Carrier code	Input	Object	Categorical	2 digits letter code eg. F9
Flight date and time	4	Tail number	Input	Object	Categorical	6 digits code eg. N170SY
	5	Flight number	Input	Object	Categorical	4 digits number eg. 2434
	6	Schedule departure	Input	Datetime64	Continuous	16 digits
	7	Schedule arrival	Input	Datetime64	Continuous	16 digits
Flight delay	8	Departure delay minutes	Output	int64	Continuous	5 digits number eg. -28
	9	Arrival delay minutes	Output	int64	Continuous	5 digits number eg. 16


Historical Dataset
Flight Schedule selected
Create two features
Calculate Flight Delay
Create a copy of these features
Create (FDT) feature
Collect data and Concatenate
New shape to paths based on algorithm instructions
Create (PFD) feature
Training Flight
ML & DL Models
The algorithm extracts each path alone to predict delay future flights
Testing first flight on path
Response predicted delay
Testing second flight on path
Response predicted delay
Testing third flight on path
Response predicted delay

Schedule departure	Schedule arrival	Carrier airline	Flight number	Origin	Destination	Tail number
01/06/2019 00:40	01/06/2019 03:10	AS	121	SEA	ANC	N528AS
01/06/2019 00:55	01/06/2019 04:31	F9	790	PDX	DEN	N308FR
01/06/2019 01:15	01/06/2019 05:22	UA	1803	SJU	EWR	N64809
01/06/2019 00:58	01/06/2019 06:00	AA	2783	FAT	DFW	N991NN

Departure	Arrival
Departure	Arrival
Departure	Arrival
Departure	Arrival
Departure	Arrival

Departure delay	Arrival delay
-1	-1
-10	-22
739	769
39	29

Carrier airline	Flight number	Origin	Destinatio n	Tail number
AS	121	SEA	ANC	N528AS
F9	790	PDX	DEN	N308FR
UA	1803	SJU	EWR	N64809
AA	2783	FAT	DFW	N991NN

Schedule departure	Carrier airline	Flight number	Origin	Destination	Tail number	Departure	Departure delay
01/06/2019 00:40	AS	121	SEA	ANC	N528AS	Departure	-1
01/06/2019 00:55	F9	790	PDX	DEN	N308FR	Departure	-10
01/06/2019 01:15	UA	1803	SJU	EWR	N64809	Departure	739
01/06/2019 00:58	AA	2783	FAT	DFW	N991NN	Departure	39

Schedule arrival	Carrier airline	Flight number	Origin	Destination	Tail number	Arrival	Arrival delay
01/06/2019 03:10	AS	121	SEA	ANC	N528AS	Arrival	-1
01/06/2019 04:31	F9	790	PDX	DEN	N308FR	Arrival	-22
01/06/2019 05:22	UA	1803	SJU	EWR	N64809	Arrival	769
01/06/2019 06:00	AA	2783	FAT	DFW	N991NN	Arrival	29

FDT	Schedule date time	Carrier airline	Flight number	Origin	Destination	Tail number	Orientation	PFD	Flight delay
0	01/06/2019 00:40	AS	121	SEA	ANC	N528AS	Departure	0	-1
02:30	01/06/2019 03:10	AS	121	SEA	ANC	N528AS	Arrival	-1	-1
04:55	01/06/2019 08:05	AS	62	ANC	JNU	N528AS	Departure	-1	-9
01:35	01/06/2019 09:40	AS	62	ANC	JNU	N528AS	Arrival	-9	2

FDT	Schedule date time	Carrier airline	Flight number	Origin	Destination	Tail number	Orientation	PFD	Flight delay
00:39	28/12/2019 12:13	AA	5946	AMA	DFW	N923FJ	Departure	73	80
01:14	28/12/2019 13:27	AA	5946	AMA	DFW	N923FJ	Arrival	80	100
00:41	28/12/2019 14:08	AA	5916	DFW	MOB	N923FJ	Departure	100	200
01:32	28/12/2019 15:40	AA	5916	DFW	MOB	N923FJ	Arrival	200	150

Fig. 8 Simulation of FDPP-ML with flight data

algorithms, first-level learners are frequently composed of unique and distinct learning algorithms [57], studies have demonstrated that a stacked ensemble performs admirably, generally outperforming a single best classifier.

Implement

The proposed FDPP-ML makes the flight delay prediction a close representation effective and easy to implement with a wide flight traffic network.

Overview dataset

The dataset used in the proposed approach includes flights from the Bureau of Transportation Statistics (BTS) archive of historical US flights [15], the datasets are available in [58]. We selected 8 months from May to December 2019, the dataset contains 5,512,904 individual flights containing 366 international airports and 10 airlines in the United States. Figure 7 shows US domestic flights to represent airports and flights recreated code from [59] introduces visualization based on Basemap library code for 60,000 flights approximately one day of operation, enough to illustrate flight paths with airports that have more flights according to the blue point size with their flights (paths) represented by gray lines.

Table 1 shows flight schedule features used, the input data contains a variety of numerical, categories, and chronological variables. Some features, such as the actual arrival and departure times and the Flight delay causes group, and weather features, were rationally eliminated and omitted from the list of essential flight schedule features.

Simulation of FDPP-ML with flight data

The proposed FDPP-ML is based on utilizing the major feature of a previous flight delay (PFD), this feature enhances flight delay prediction if exists, but at the same time, this feature is known before flight time a few minutes. Accordingly, the FDPP-ML will provide for ML and DL models this feature (PFD) with an increase in the forecast horizon to hours rather than minutes. Figure 8 shows the simulation of FDPP-ML with flight data, to reshape the flight data to be a flight path containing the sequence of points of flights to catch flight paths in historical data and organize each path to be a sequence of points of flights, starting from the first beginning point whether arrival or departure and following the end of the path. Then the FDPP-LM creates proposed features (FTD, PFD) according to paths, and separates the dataset into historical flights to train the model and future flights that need to predict delay. The FDPP-LM created an FTD feature based on the schedule time feature for both historical and future data. Regarding the PFD feature the FDPP-LM was created based on previous flight delays in historical flights, but in future flights (testing) the FDPP-LM will pass each first point in the paths of future flights which has a PFD feature to the model, and transfer the predicted delay from model to be a PFD feature for next flight on the path, and so on to the paths finished.

Handling dataset

During pre-processing, we eliminate irrelevant data and save just crucial ones to ensure coherence. Before training, the data is organized and prepared using removing missing values and duplicates, and removing canceled flights represents the percentage of total data (2%). Null values and errors are removed, converting categorical data to numerical.

Train ML model

Firstly, the prediction FDPP-ML is trained using a dataset of US historical flights after reshaping and extracting new features process. In general, linear models perform prediction by training and developing the computing of the parameters of the model weighted sum of input variables and bias. Finding parameter values that reduce the sum of squared errors is the training aim, which is the cost function applied to the models' evaluation. Essentially the training of the proposed approach FDPP-ML is based on one model nominated for implementation and completes the cycle of an algorithm, but this paper will be experimenting with 10 models to evaluate the prediction development results of FDPP-ML, we explained these models in detail in the methodology section. Accordingly, we use machine learning models with default parameters, because the tuning of parameters is another issue, and the main what's behind these experiments is to measure the size of flight delay prediction improvement when using FDPP-ML. So we selected models (CATR, GBR, RFR, LGR, LR) with default hyper parameters, and (Stacking, Voting) models basically contain three models (GBM, RFR, LGBM) with default parameter, except the Stacking has LR model at the end level of stacking with default parameter. On the other hand, the deep learning selected models are RNN, LSTM, and GRU have a normal parameter network containing two layers with 32 units of nods for each layer, the activation function is "relu" for each layer, and the activation function for the dense layer is "linear", the training was with epoch 50 and batch size 100.

The ML and DL models were trained as an initial hidden experiment on 1 month of flights, we found the results could be enhanced if using more flights. The number of flights should not be overlooked especially when training with basic flight schedule

Table 2 Accuracy of models training

Models	Traditional training model			Proposed FDPP-ML			Error reduction percent by FDPP-ML		
	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE (%)	MSE (%)	RMSE (%)
CATR	23.37	2593	50.92	13.36	1321	36.34	43	49	29
GBR	23.74	2641	51.39	14.47	1485	38.54	39	44	25
GRU	23.68	2647	51.45	14.49	1493	38.64	39	44	25
LGR	23.52	2616	51.14	13.65	1385	37.21	42	47	27
LR	24.03	2672	51.69	17.76	1863	43.17	26	30	16
LSTM	24.40	2660	51.58	15.49	1568	39.60	37	41	23
RFR	18.28	1669	40.86	9.21	716	26.76	50	57	34
RNN	23.94	2667	51.64	14.99	1488	38.57	37	44	25
Stacking	21.85	2429	49.28	10.71	1026	32.04	51	58	35
Voting	21.57	2244	47.37	12.18	1140	33.77	44	49	29
Average error reduction percent by FDPP-ML							41	46	27

Table 3 Accuracy of forecast horizons 12 h

Models	Traditional training model			Proposed FDPP-ML			Error reduction percent by FDPP-ML		
	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE (%)	MSE (%)	RMSE (%)
CATR	30.62	4341	65.88	25.87	3483	59.01	16	20	10
GBR	31.02	4422	66.50	26.48	3777	61.45	15	15	8
GRU	31.04	4441	66.64	26.46	3966	62.98	15	11	5
LGR	30.87	4366	66.08	25.96	3574	59.79	16	18	10
LR	31.25	4485	66.97	29.78	4329	65.80	5	3	2
LSTM	31.25	4467	66.84	27.02	4180	64.66	14	6	3
RFR	31.76	4477	67.27	26.30	3501	59.17	17	22	12
RNN	31.17	4488	67.00	26.78	3999	63.24	14	11	6
Stacking	30.34	4385	66.22	24.82	3541	59.50	18	19	10
Voting	30.87	4392	66.28	25.76	3549	59.57	17	19	10
Average error reduction percent by FDPP-ML							15	14	8

Table 4 Accuracy of forecast horizons 6 h

Models	Traditional training model			Proposed FDPP-ML			Error reduction percent by FDPP-ML		
	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE (%)	MSE (%)	RMSE (%)
CATR	29.21	3943	62.79	22.63	2841	53.30	23	28	15
GBR	29.43	4013	63.35	23.31	3115	55.81	21	22	12
GRU	29.44	4034	63.52	23.41	3266	57.15	21	19	10
LGR	29.35	3963	62.95	22.67	2913	53.97	23	26	14
LR	29.52	4082	63.89	26.78	3735	61.11	9	9	4
LSTM	29.61	4072	63.81	24.14	3448	58.72	18	15	8
RFR	30.29	4098	64.02	22.76	2830	53.20	25	31	17
RNN	29.58	4078	63.86	23.57	3184	56.43	20	22	12
Stacking	28.83	3977	63.07	21.52	2855	53.43	25	28	15
Voting	29.38	3981	63.10	22.50	2897	53.83	23	27	15
Average error reduction percent by FDPP-ML							21	23	12

Table 5 Accuracy of forecast horizons 2 h

Models	Traditional training model			Proposed FDPP-ML			Error reduction percent by FDPP-ML		
	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE (%)	MSE (%)	RMSE (%)
CATR	26.71	3544	59.53	17.34	2088	45.70	35	41	23
GBR	26.68	3616	60.14	18.24	2262	47.56	32	37	21
GRU	26.83	3620	60.17	18.57	2355	48.53	31	35	19
LGR	26.71	3570	59.75	17.59	2131	46.16	34	40	23
LR	27.22	3655	60.46	21.56	2733	52.28	21	25	14
LSTM	27.21	3671	60.59	19.64	2451	49.51	28	33	18
RFR	28.16	3685	60.71	17.29	2032	45.08	39	45	26
RNN	26.91	3665	60.54	19.08	2339	48.36	29	36	20
Stacking	26.29	3579	59.83	16.69	2071	45.50	37	42	24
Voting	26.87	3583	59.86	17.47	2111	45.95	35	41	23
Average error reduction percent by FDPP-ML							32	38	21

features, we observed that when the dataset increases the model is trained well. Accordingly, ML and DL models were trained on 8 months of flights, which maximum data can be obtained and available online for the US a wide traffic network. The testing was on three scenarios represented on forecast horizons 2, 6, and 12 h, containing flights 5286, 15954, and 26006 respectively. The experimental computer's hardware is set up with an Intel core i7 10th processor clocked at 2.60 GHz and 16 GB of RAM with a maximum speed of 5.3 GHz. NVIDIA GeForce RTX 2080 SUPER graphics processor-accelerated graphics card. The Python 3.7-based Keras framework serves as the development environment [60].

Result and discussion

The evaluation of the proposed approach FDPP-ML to improve flight delay prediction was done with different analyses. The main way is to demonstrate the results of flight delay prediction based using basic flight schedule features by using 10 benchmarks and state-of-the-art regression models and comparing when using traditional training models, and when using the FDPP-ML algorithm with new features created, and demonstrating the measure of proposed FDPP-ML reduction errors with three forecast horizons 12, 6, and 2 h. These interval times to measure what extent of the strength of the FDPP-ML forecast horizon when relying on (PFD) action because when the forecast horizon is longer will be weaker, owing to FDPP-ML relying on the prediction inheritance of the "Previous Flight Delay" (PFD) as additional feature supported to the next flight that on the same flight path to predict their delay.

The second way is to demonstrate the improved reduction of error by FDPP-ML to Traditional ML, by select an outperformed model with a deep analysis of FDPP-ML impacts on 366 airports and demonstrate how FDPP-ML enhances accuracies using a sample of airports represented in the busiest US OPE (Operational Evolution Partnership), which is the commercial US airports with a significant activity that is employed in the streamlined National Airspace System (NAS), More than 70 percent of passengers

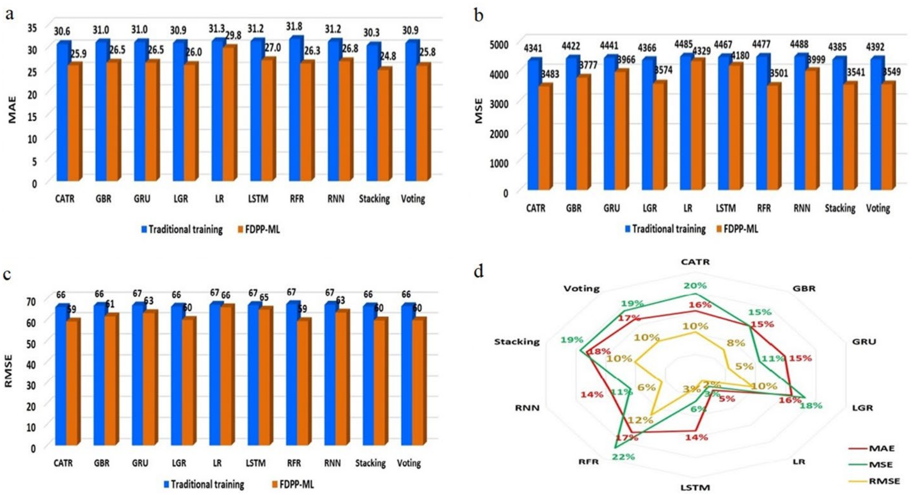


Fig. 9 Accuracy of forecast horizon 12 h, **a** mean absolute error MAE, **b** mean absolute error MSE, **c** root mean absolute error RMSE, and **d** error reduction by FDPP-ML

move through these airports, this airport list contains 30 airports [61]. In addition, provides an analysis of how to improve prediction on all airlines in the dataset used contains 10 airlines.

Accuracy of proposed FDPP-ML

Firstly, estimated the accuracy of flight delay prediction of both traditional models and FDPP-ML by using basic flight schedule features (BFS). Tables 2, 3, and 4, and 5 show measure error tools MAE, MSE, and RMSE to explain flight delay prediction accuracy for 10 regression models implemented, each table is divided into three groups containing three measure error tools, the first group is the traditional training model, the second is the proposed FDPP-ML, and the third is Error reduction percent by FDPP-ML. Table 2 shows errors measure for training models, despite the testing considered the main core of the proposed approach contribution, the training accuracies are a strong indicator success of FDPP-ML. in the group name "Error reduction percent by FDPP-ML" in the table demonstrates the percent of enhanced prediction by FDPP-ML. Overall we can say the average error reduction percent by FDPP-ML in training 10 models were 41%, 46%, and 27% in MAE, MSE, and RMSE respectively.

Let highlighted the RFR model in details, it can be seen in the traditional training model was less error MAE of 18.28 min, on the other side the proposed FDPP-ML also had less MAE error of 9.21 min, which indicates FDPP-ML improved prediction than the traditional training model by 50%. As well MSE in the traditional training model was 1669, and in the FDPP-ML the MSE was 716, the FDPP-ML improved prediction than the traditional training model by 57%, and so on in RMSE the FDPP-ML improved prediction by 34%. let highlighted the Stacking model, in the traditional

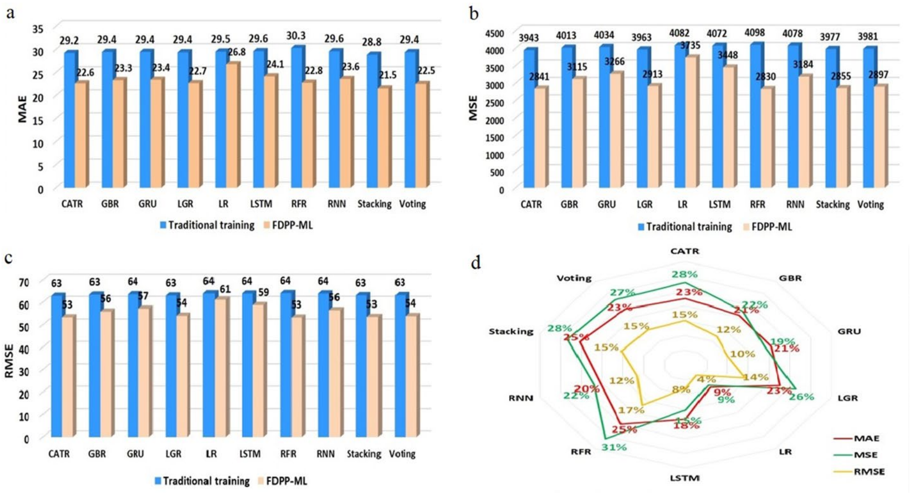


Fig. 10 Accuracy of forecast horizon 6 h, **a** mean absolute error MAE, **b** mean absolute error MSE, **c** root mean absolute error RMSE, and **d** error reduction by FDPP-ML

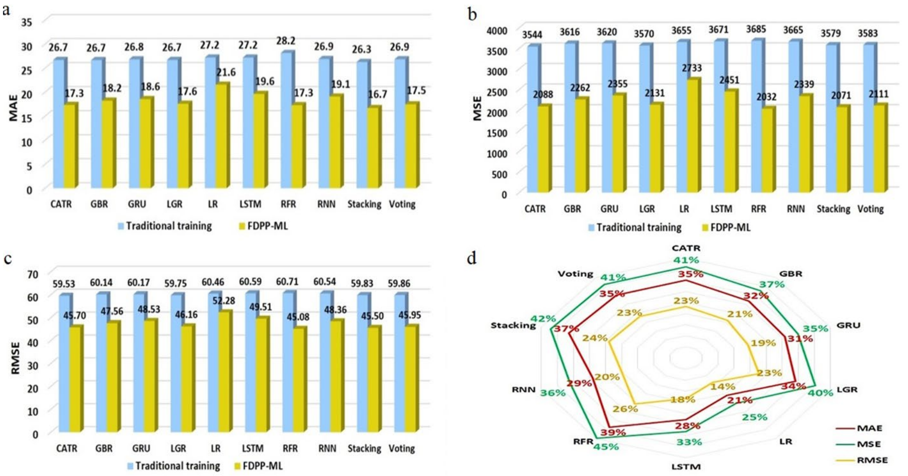


Fig. 11 Accuracy of forecast horizon 2 h, **a** mean absolute error MAE, **b** mean absolute error MSE, **c** root mean absolute error RMSE, and **d** error reduction by FDPP-ML

model was a second less MAE error with 21.85 min on the other hand in the proposed FDPP-ML the Stacking model was also a second less MAE error with 10.71 min, which indicates the percent of enhanced and improved FDPP-ML than traditional training model archive to 51%, and improved MSE and RMSE with 58% and 35% respectively. Thus proposed FDPP-ML improves prediction by the massive reduction of error than the traditional training models within similar proportions.

The proposed FDPP-ML will become clear with the testing represented in various forecast horizons of flight delay predicted, illustrating how FDPP-ML is much better than traditional training models. Firstly, Table 3 shows the measured error tools within the long forecast horizon 12 h, in spite of this containing 26006 flights relying on flight delay prediction inheriting to the next flight to be as FDP feature to predict the next delay that is on the same path for 12 h, should be noted that the average error reduction percent by FDPP-ML in testing 10 models in forecast horizon 12 h were 15%, 14%, and 8% in MAE, MSE, and RMSE respectively. let highlighted in details on Stacking model it can be seen the outperform model in experiments the FDPP-ML improves accuracy with a reduced error by 18%, becoming from an MAE of 30.34 error in traditional training models to 24.82 in FDPP-ML, also reduced error in Stacking by 19% and 10% in MSE and RMSE respectively. As well as FDPP-ML improved accuracy at all models from 14 to 18% in MAE, except the LR model with only 5%. Figure 9 shows the accuracy of the forecast horizon 12 h, Fig. 9a, b, and c represent MAE, MSE, and RSME respectively for all model errors with traditional trained models and proposed FDPP-ML. Figure 9d represents the error reduction percent by FDPP-ML for all models.

The experiments are still ongoing there is plenty of time before flight time represented in the forecast horizon. As much as the forecast horizon is short the FDPP-ML will be more outperform. Accordingly, the proposed FDPP-ML will implement within the forecast horizon of 6 h which contains 15,954 flights monitoring the results. Table 4 shows the model's result in this case, should be noted that the average error reduction percent by FDPP-ML in testing 10 models in forecast horizon 6 h were 21%, 23%, and 12% in MAE, MSE, and RMSE respectively. To show the result model in details let highlighted the Stacking model can be seen that is the best model in both the traditional training model and FDPP-ML, was MAE of 28.83, and become 21.52, as well in MSE was 3977 and become 2855, the FDPP-ML improve accuracy in the stacking model by 25% and 28% in MAE and MSE respectively. We found the FDPP-ML with a shortage of forecast horizon will start to improve prediction, it should be noted that the FDPP-ML improves accuracy in the forecast horizon of 12 h in MAE from 14 to 18%, and was in the forecast horizon of 6 h increased all models from 18 to 25% except LR with 9%. There is a reverse relation between the proposed FDPP-ML and the forecast horizon, as much as the shortage of forecast horizon the FDPP-ML is capable of improving accuracy and increasing the gap than the traditional training model. Figure 10 shows the accuracy of the forecast horizon 6 h, Fig. 10a, b, and c represent MAE, MSE, and RSME respectively for all model errors with traditional trained models and proposed FDPP-ML. Figure 10.d represents the error reduction percent by FDPP-ML for all models, it can be seen that error reduction was in MAE from 18 to 25%, and in MSE from 19 to 31%, and so on error reduction in RMSE.

The experiments were established based forecast horizon of 2 h containing 5286 flights, we found studies interested in the forecast horizon of two hours before the intended flight time even using special airport information. Table 5 shows the accuracy of forecast horizons 2 h, Fig. 11 shows the accuracy of the forecast horizon, and Fig. 11a. shows the MAE for models, it should be noted that the gap between traditional training and FDPP-ML becomes more than other forecast horizons, should be noted that the average error reduction percent by FDPP-ML in testing 10 models in

Table 6 Accuracy of core 30 airports

NO	Airports IATA Code	Traditional training model			Proposed FDPP-ML			Error reduction percent by FDPP-ML		
		MAE	MSE	RMSE	MAE	MSE	RMSE	MAE (%)	MSE (%)	RMSE (%)
1	ATL	17	1074	33	10	502	22	44	53	32
2	BOS	32	3485	59	22	2503	50	31	28	15
3	BWI	25	2528	50	16	919	30	38	64	40
4	CLT	17	841	29	11	638	25	34	24	13
5	DCA	24	1636	40	13	734	27	48	55	33
6	DEN	28	2586	51	17	1079	33	40	58	35
7	DFW	24	3441	59	15	3055	55	40	11	6
8	DTW	27	3317	58	17	2640	51	36	20	11
9	EWR	57	21885	148	43	12873	113	25	41	23
10	FLL	15	394	20	12	394	20	21	0	0
11	HNL	17	689	26	14	449	21	18	35	19
12	IAD	32	4800	69	21	3620	60	34	25	13
13	IAH	23	1466	38	14	652	26	41	55	33
14	JFK	21	1157	34	16	932	31	22	19	10
15	LAS	18	1263	36	13	965	31	30	24	13
16	LAX	19	1300	36	14	487	22	29	63	39
17	LGA	22	956	31	12	453	21	47	53	31
18	MCO	18	707	27	12	406	20	33	43	24
19	MDW	17	922	30	10	221	15	40	76	51
20	MEM	29	3087	56	13	553	24	56	82	58
21	MIA	26	7798	88	21	7568	87	19	3	1
22	MSP	66	15660	125	40	7413	86	39	53	31
23	ORD	42	7405	86	29	4536	67	33	39	22
24	PHL	31	2760	53	24	2314	48	22	16	8
25	PHX	18	836	29	11	529	23	39	37	20
26	SAN	18	1294	36	10	228	15	45	82	58
27	SEA	20	834	29	13	361	19	36	57	34
28	SFO	21	774	28	13	390	20	39	50	29
29	SLC	14	338	18	9	124	11	37	63	39
30	TPA	20	1580	40	15	1192	35	25	25	13
Average error reduction percent by FDPP-ML								35	42	25

forecast horizon 2 h were 32%, 38%, and 21% in MAE, MSE, and RMSE respectively. It could be higher in individual model it can be seen in the stacking model has the lowest error in traditional training in MAE 26.3 min and becomes 16.7 min in FDPP-ML. Followed by the RFR model in MAE was 28.2 became 17.3, and the model CATR in MAE was 26.7 became 17.3, and so on in all models. Figure 11b shows the MSE for models, it's relatively similar to MAE, the first model has low error RFR in MSE was 3685 and becomes 2023, followed by stacking in MSE was 3679 and becomes 207, and so on. Figure 11C shows the RMSE for models, it's relatively similar to MSE, overall the Stacking outperformed in the proposed FDPP-ML has MAE 16.7 while in traditional training was MAE 26.3. Figure 11d shows the error reduction percent by

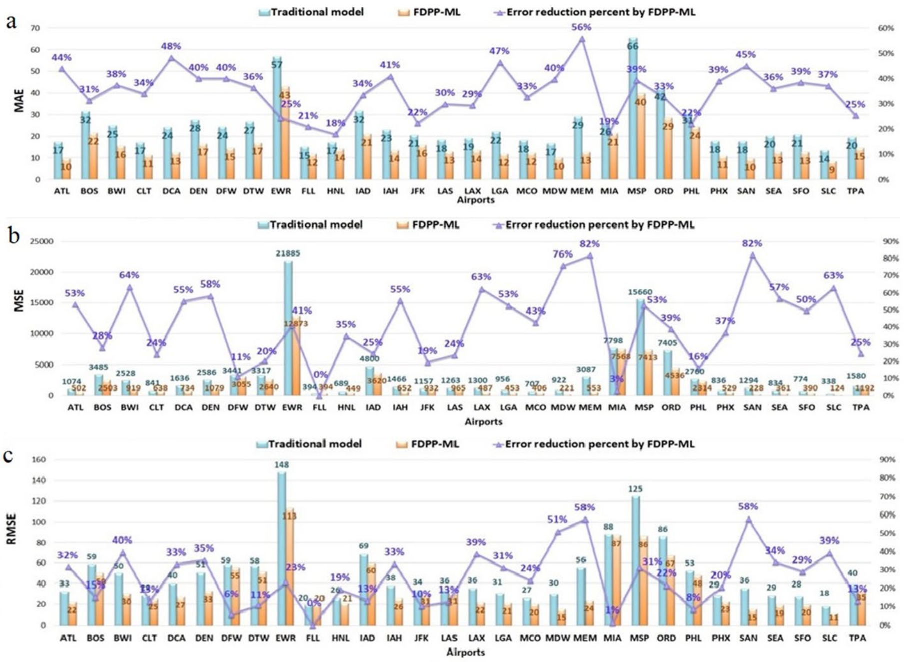


Fig. 12 Accuracy of Core 30 airports, **a** mean absolute error MAE, **b** mean absolute error MSE, and **c** root mean absolute error RMSE

Table 7 Accuracy of US airlines

NO	Airlines IATA Code	Traditional training model			FDPP-ML			Error reduction percent by FDPP-ML		
		MAE	MSE	RMSE	MAE	MSE	RMSE	MAE (%)	MSE (%)	RMSE (%)
1	AA	26	2788	53	16	1931	44	37	31	17
2	AS	18	779	28	11	314	18	38	60	37
3	B6	24	1080	33	17	641	25	30	41	23
4	DL	28	4323	66	17	2226	47	40	48	28
5	F9	19	497	22	11	196	14	45	61	37
6	G4	25	2438	49	15	847	29	40	65	41
7	HA	12	321	18	10	198	14	19	39	22
8	NK	22	1806	42	13	1164	34	38	36	20
9	UA	37	7762	88	24	4554	67	34	41	23
10	WN	16	1023	32	11	498	22	35	51	30
Average error reduction percent by FDPP-ML								36	47	28

FDPP-ML, the proposed FDPP-ML is capable of significantly improving accuracy for example in MAE with the red line the enhancement accuracy amounts to 39% in the RFR model, and enhancement accuracy by 37% in the stacking model, and relatively similar for remain models.

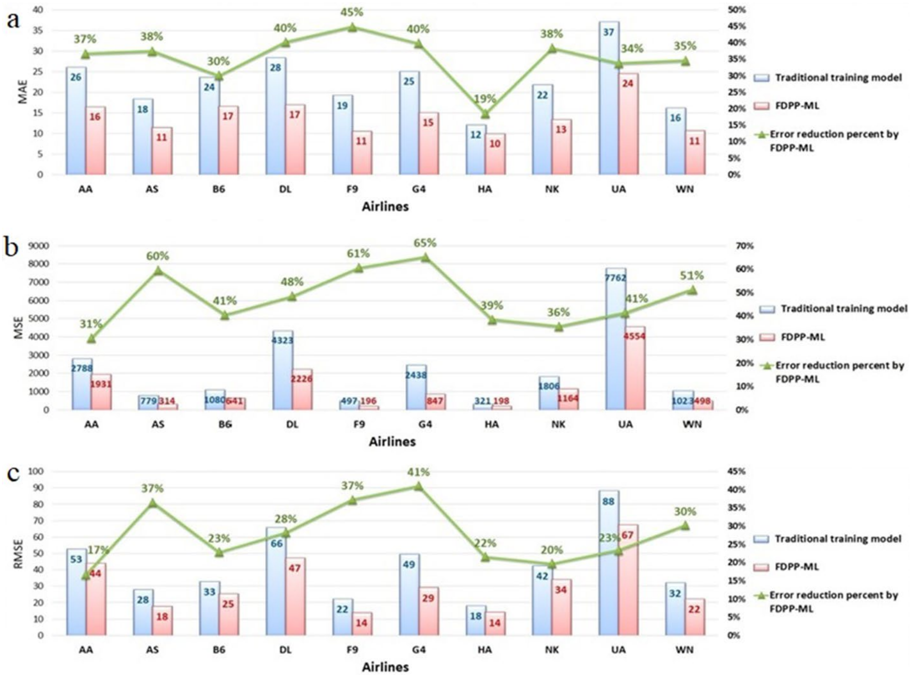


Fig. 13 Accuracy of US airlines, **a** mean absolute error MAE, **b** mean absolute error MSE, and **c** root mean absolute error RMSE

Overall we deduced from a summary of the results, that must be taken into consideration that the success of FDPP-ML training accuracies was due to the created new features, especially (PFD) feature, we found considerable differences in improving prediction at training and testing model accuracies, that because we use training models dataset containing the actual (PFD) feature for historical flights. While in the testing model, the flight delay prediction is considered as (PFD) feature for the next flight on the same path, during this inherited looping of course the model will be weaker as much as long of forecast horizon. Thus the testing of proposed FDPP-ML has successful prediction with the short-term forecast horizons, especially in 2 h and the improvement decreases when increasing the forecast horizon for future flights.

Accuracy improvement of airports and airlines

This subsection provides a deep analysis of FDPP-ML impacts on 366 airports involved in the dataset by showing the busiest 30 airports called "Core 30". The FDPP-ML enhances the accuracies of the traditional trained models for these airports separately during flight delay prediction experiments based on a forecast horizon of 2 h within stacking model. Table 6 shows the core 30 airports' flight delay accuracies between the traditional training models compared to the proposed FDPP-ML and illustrates the error reduction percent by FDPP-ML in these airports. Overall, it should be noted that the average error reduction percent by FDPP-ML in the core 30 airports in forecast horizon 2 h were 35%, 42%, and 25% in MAE, MSE, and RMSE respectively.

Figure 12 shows in details the three measure error tools for core 30 airports, Fig. 12a,b, and c represent MAE, MSE, and RSME respectively. It should be noted that FDPP-ML contributes to improving the accuracy of airports differently according to the size of flights operation and times. In Fig. 12a represents MAE error note that in ATL airport which is considered the busiest airport at US airport, the FDPP-ML error reduction was 44% the percentage represents cuts MAE from 17 to 10 min. In Fig. 12b represents MSE in ATL airport the FDPP-ML error reduction was 53% the percentage represents cuts MSE from 1074 to 502. In Fig. 12c represents RMSE in ATL airport the FDPP-ML error reduction was 32% the percentage represents cuts RMSE from 33 to 22, and so on in the remaining airports. Overall, the percent of FDPP-ML error reduction in the three measure tools was relatively symmetric and illustrates the improved accuracy in all airports significantly, with the knowledge that we sufficiently represent the improvement at 30 airports, on another hand, we have implicit results that can draw by of 366 airports and it is difficult to their represented in this paper.

We also provide an analysis of improved prediction on all airlines involved in the dataset used containing 10 airlines, the proposed FDPP-ML contributed to enhancing accuracy over traditional training models. Table 7 shows the flight delay accuracies of US airlines compared to the traditional training model and proposed FDPP-ML and illustrates the error reduction percent by FDPP-ML in these airlines.

Overall, it should be noted that the average error reduction percent by FDPP-ML in the involved 10 airports in forecast horizon 2 hour were 36%, 47%, and 28% in MAE, MSE, and RMSE respectively. Figure 13 shows in details the three measure error tools, Fig. 13a, b, and c represent MAE, MSE, and RSME respectively. It should be noted that FDPP-ML contributes to improving the accuracy of airlines differently according to the size of flights airlines. In Fig. 13a represent MAE error, it can be seen that the error reduction percent by FDPP-ML for example reached to 45% in F9 airline the percentage represents cuts from 19 to 11 minutes, In Fig. 13b represent MSE error can be seen that the percent of FDPP-ML was 61% in F9 the percentage representing cuts from 497 to 196, In Fig. 13C represents RMSE error, it can be seen that the percent of FDPP-ML improve accuracy was 37% in F9 the percentage representing cuts from 22 to 14. The FDPP-ML improved accuracy in all airlines and airports significantly, with the knowledge that there are studies that seek to provide enhanced flight delay prediction for one specific airline or specific airport. Appendix A and B represent the flight operation for Core_30 airports and 10 airlines respectively for each airport and airline with their names, each Figure represents 80 flights as a sample of flight delay predicted from a forecast horizon of 2 hours, with the nominated model Stacking from models implemented, containing three curves in chart actual flight delay represent with a red dashed line, the predict traditional training model represent with a light blue line, and predicted FDPP-ML represent with a green line.

Improvement insights

Airport flight delays continue to be a major problem that has an impact on both airport and airline operations. If aviation system decision-makers want to stay competitive, they should prioritize the inclusion of flight delay estimates in their insights at the multi-level. The suggested method for integrating IOT and cloud computing apps with passenger re-communication capabilities for the smart airport [62, 63] that enable crew members, flexible counters, gates, and airlines to interact to notify passengers of updates and emphasize flight time issues in order to win their loyalty. On the other hand side, when it comes to workforce planning and workload in airports [6, 64], aviation authorities and airline suppliers ought to consider the duration of aircraft delays when rescheduling workers. This may include making a fast decision, such as extending staff shifts to cover busy hours.

Limitations and future research

While the proposed approach technique improves the ability to anticipate flight delays, there are certain drawbacks that point to possible directions for further study.

1- The results show outperformed FDPP-ML flight delay prediction using basic flight schedule features. Future studies may take into account extracting relevant characteristics to improve prediction from the existing flight information, such as an index for each airport containing the average of the number of flights that are delayed and the number of flights that are planned for a certain time period, to enable training and testing.

2- The proposed implemented 10 benchmark and state-of-art regression models in FDPP-ML but with their default parameters, the parameters play an important role in enhancing prediction, Future work could be considered implementing stronger parameters of models hybrid with FDPP-ML, consequently the prediction enhances.

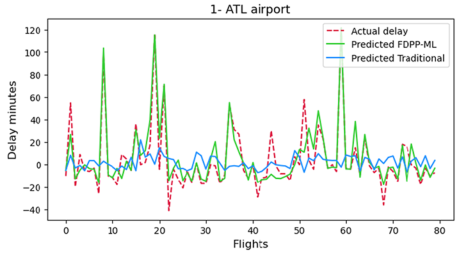
3- on the part of the training model, in real-time when future flight delay prediction is required, we don't need at every time to retrain the model it's a waste of time, It is better to save the training model in advance, and FDPP-ML all steps execution without lines 3 and 4 in (Algorithm 4 FDPP-ML steps 7and8).

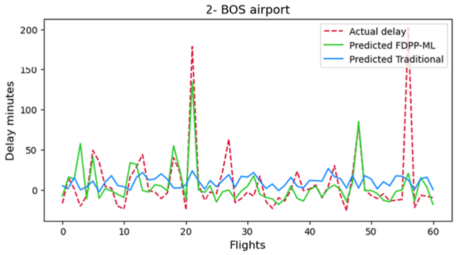
Conclusion

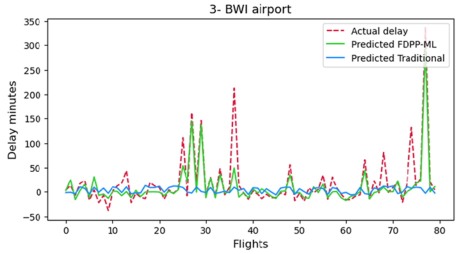
In an intelligent aviation system, it is essential to make precise and prompt flight delay predictions which are caused by complex spatial–temporal correlations, ranging from weather, airport operations, passengers, and impact flight delay departure or arrival on each other. At the same time, it is hard to obtain these features, especially with a wide network of flights, therefore, is necessary to develop flight delay prediction models based only on flight schedule features to be available with high performance to make crucial decisions. This paper proposed FDPP-ML a novel algorithm with a supervised learning model that works on reshaping datasets and creates new significant and discriminating features (PFD) and (FTD) which contribute to improving ML models to predict flight delay. To evaluate the performance of the FDPP-ML, we focus on utilizing only basic flight schedule features for flight delay prediction using a wide flight network of US flights with many experiments of predicting flight delay in three forecast horizons 2, 6, and 12 h, using 10 machine and deep learning models with error measure tools MAE, MSE, and RMSE. The FDPP-ML improves the accuracy of all 10 models and error reduction than traditional training models, the average error reduction percent by FDPP-ML in testing 10 models in forecast horizon 2 h were 32%, 38%, in MAE, MSE respectively, in forecast horizon 6 h were 21%, 23%, in MAE, MSE respectively, in forecast horizon 12 h were 15%, 14%, in MAE, MSE respectively. The FDPP-ML is more effective in a short forecast horizon because the flight delay prediction is considered (PFD) feature for the next flight that is on the same path, during this inherited looping of course the model will be weaker the more of the forecast horizon. In forecast horizon 2 h the outperforming prediction model was Stacking in FDPP-ML in MAE 16.69 min compared to traditional training in MAE 26.29 min the FDPP-ML improvement and reduction of errors with a 37% percent. Based on stacking prediction can improve individual airlines and airports, the average error reduction percent by FDPP-ML in the core 30 airports were 35%, 42%, in MAE, MSE, respectively. And the average error reduction percent by FDPP-ML in 10 airlines involved were 36%, 47%, in MAE, MSE, respectively. The proposed approach is yielding encouraging results and shows the ability of prediction to utilize previous flight delay PFD features when integrated into FDPP-ML. Finally recommended using the FDPP-ML model proves successful to predict flight delays with only flight schedules to our perception and motivation for using this proposed approach that leads to promoting efficiency for stakeholders and passenger satisfaction by improving airport management efficiency.

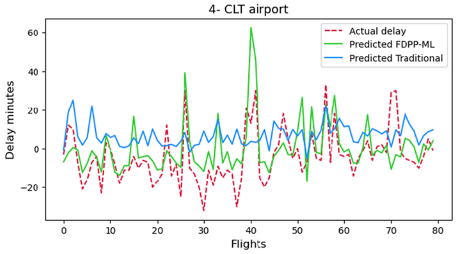
Appendix

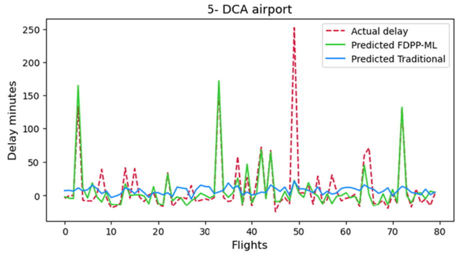
Appendix A

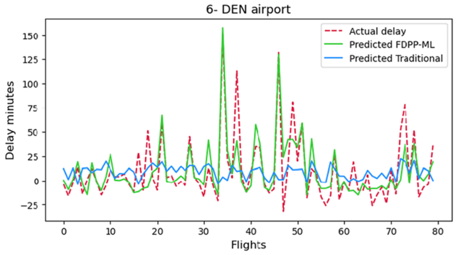

1- ATL airport

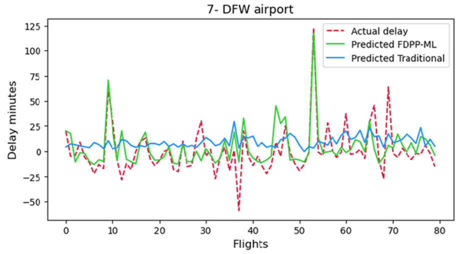

2- BOS airport

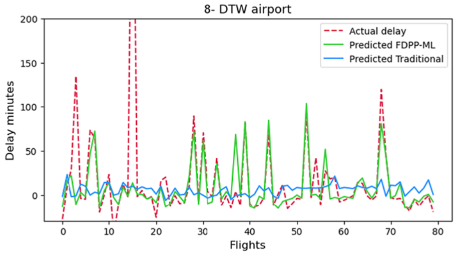

3- BWI airport

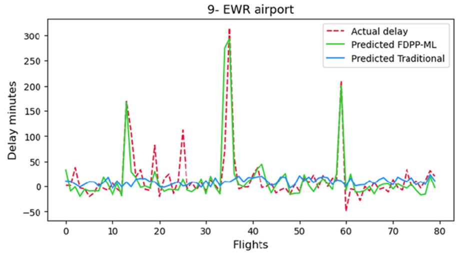

4- CLT airport

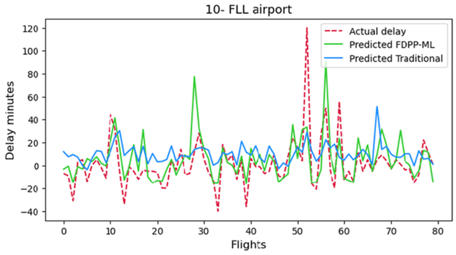

5- DCA airport

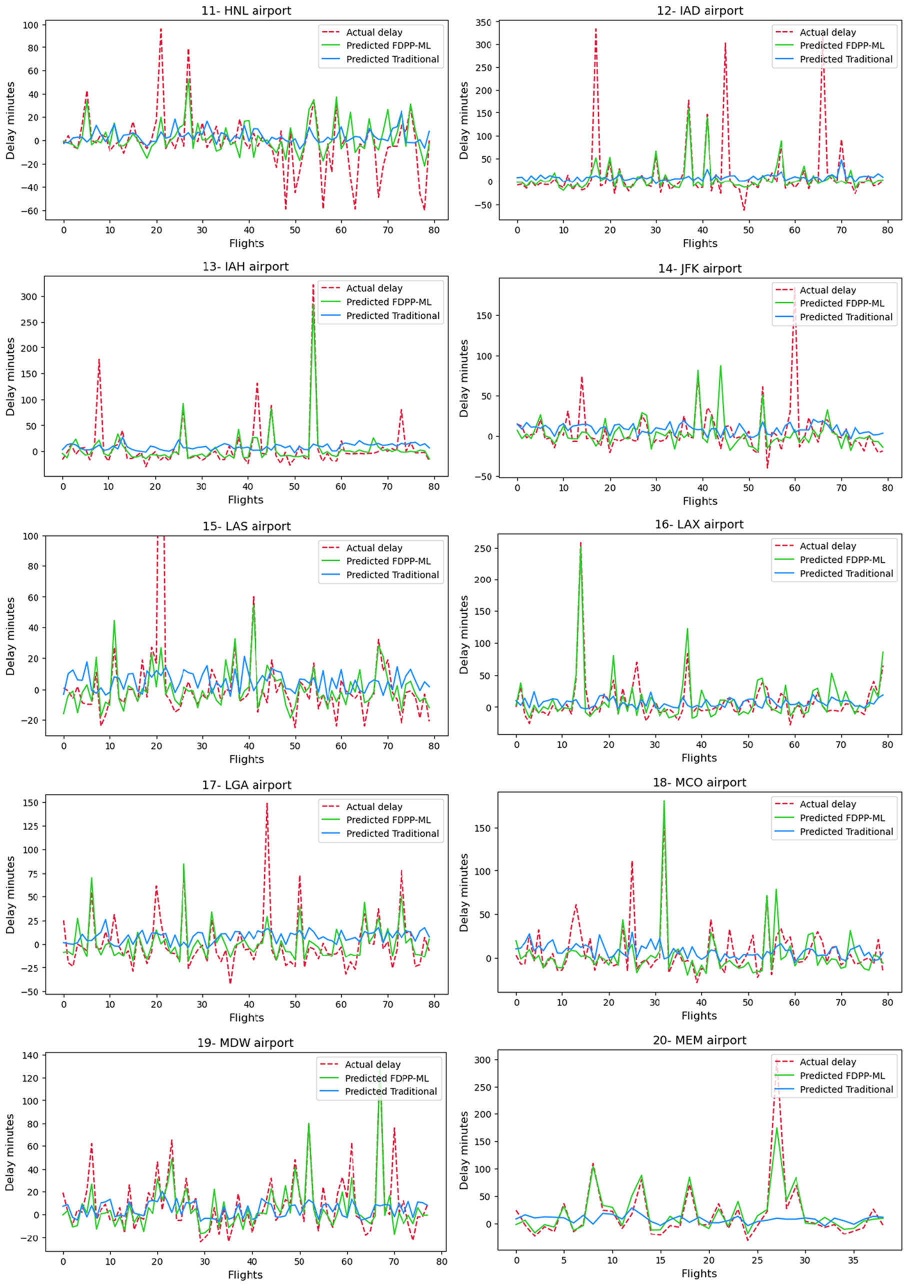

6- DEN airport

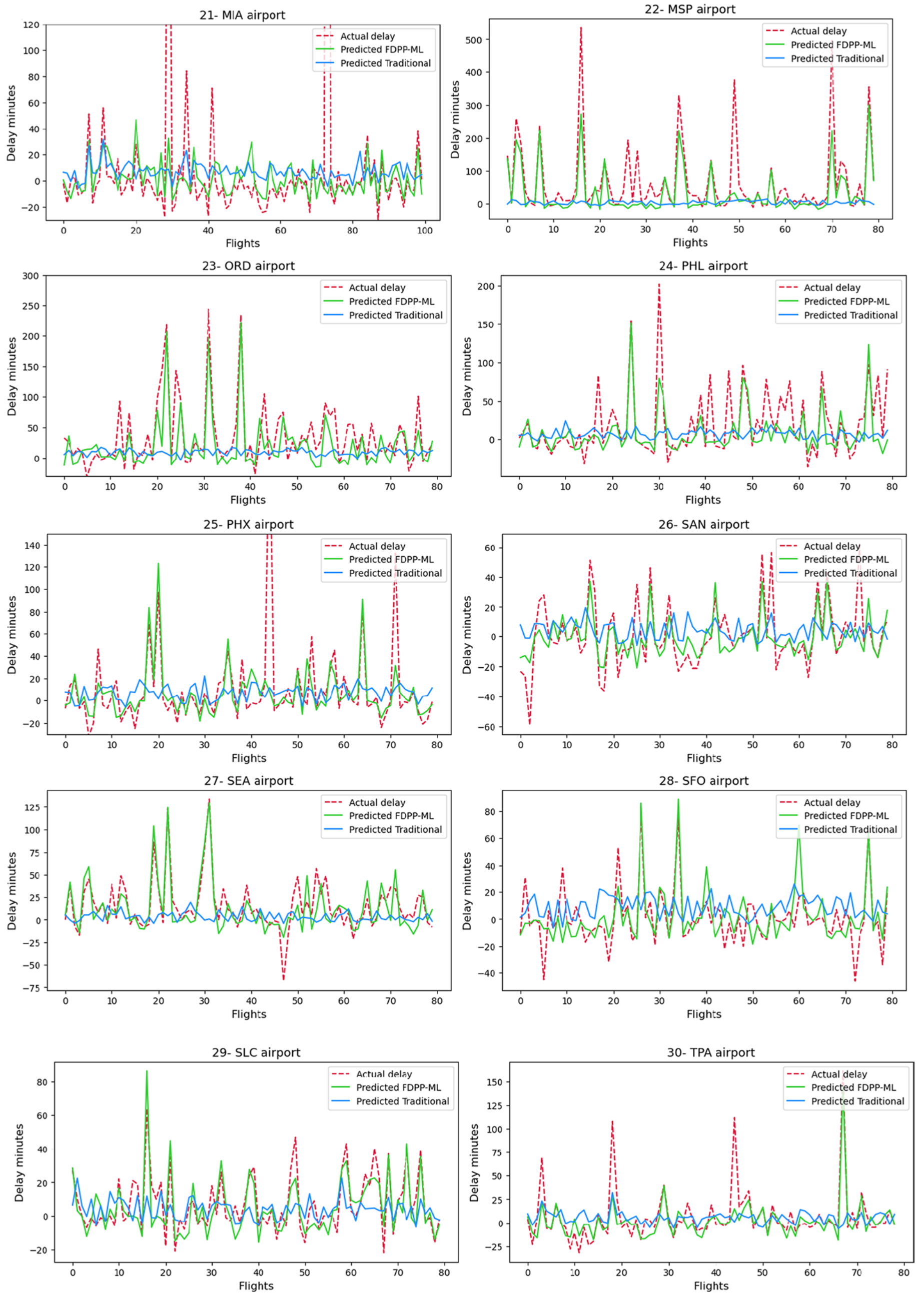

7- DFW airport


8- DTW airport


9- EWR airport


10- FLL airport


11- HNL airport
12- IAD airport
13- IAH airport
14- JFK airport
15- LAS airport
16- LAX airport
17- LGA airport
18- MCO airport
19- MDW airport
20- MEM airport
Actual delay
Predicted FDPP-ML
Predicted Traditional
Delay minutes
Flights



21- MIA airport

22- MSP airport

23- ORD airport

24- PHL airport

25- PHX airport

26- SAN airport

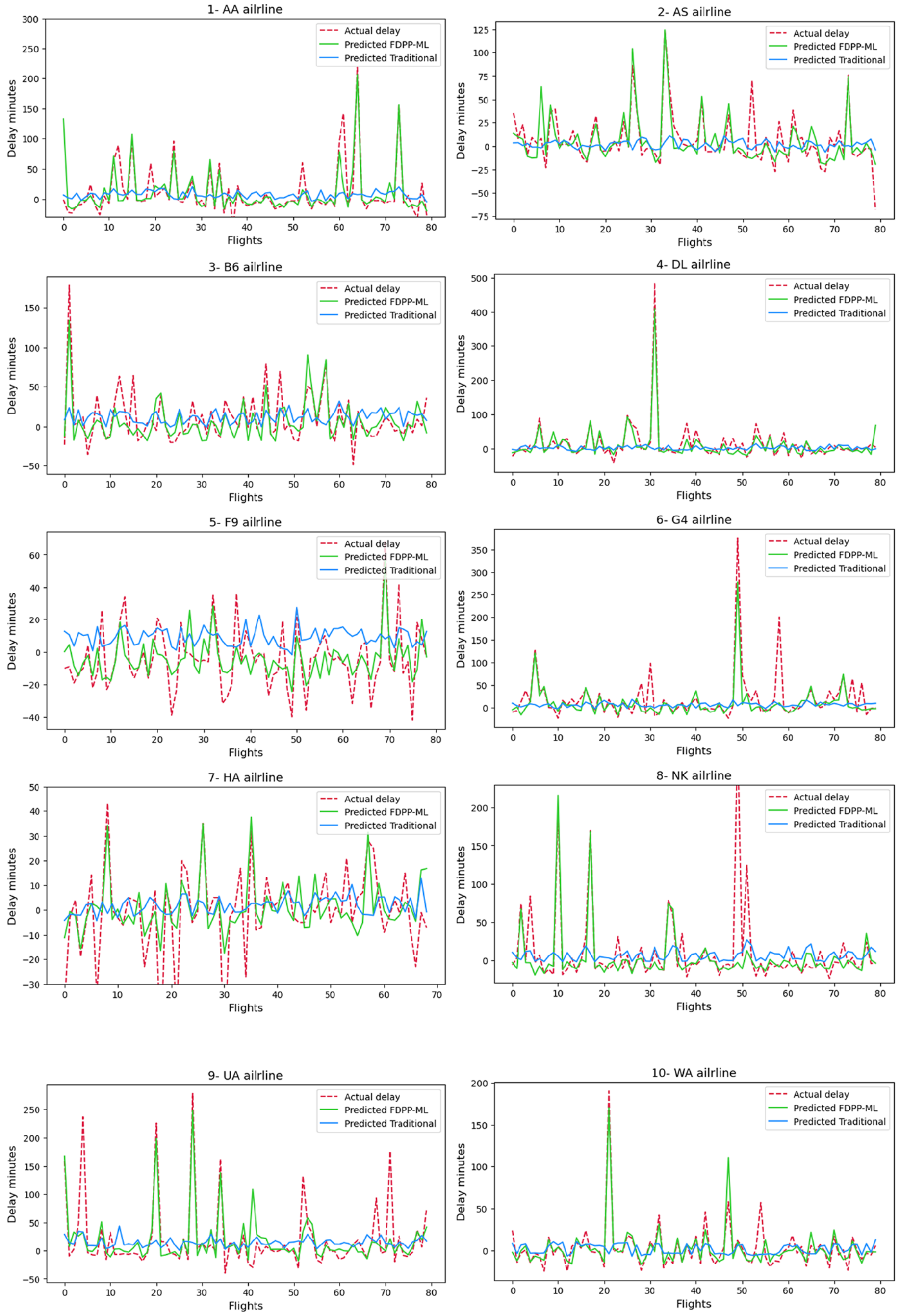
27- SEA airport

28- SFO airport

29- SLC airport

30- TPA airport

Appendix B



Acknowledgements

All authors thank the editors and reviewers for their attention to the paper.

Author contributions

MM had the primary role in developing the hypothesis, creating an algorithm approach with coding and conducting the tests on the available datasets and analyzing, evaluating, and writing the article. ME and HH supported to development of the concept and theory, providing the search trends and the highlights of the manuscript, reading the paper's draughts with reviewing, and approving the final manuscript.

Funding
Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Availability of data and materials
The datasets used in this paper's experiments are available online at [58].

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The author declares that he has no competing interests.

Received: 27 August 2023 Accepted: 21 November 2023
Published online: 21 December 2023

References

1. Khan WA, Ma HL, Chung SH, Wen X. Hierarchical integrated machine learning model for predicting flight departure delays and duration in series. Transp Res Part C Emerg Technol. 2021;129:103225.
2. Etani N. Development of a predictive model for on-time arrival flight of airliner by discovering correlation between flight and weather data. J Big Data. 2019;6(1):85. https://doi.org/10.1186/s40537-019-0251-y.
3. FAA. 2021. https://www.faa.gov/dataresearch/aviationdatastatistics.pdf. Accessed 15 Jan 2022.
4. Zhu X, Li L. Flight time prediction for fuel loading decisions with a deep learning approach. Transp Res Part C Emerg Technol. 2021;128:103179.
5. Guo Z, Yu B, Hao M, Wang W, Jiang Y, Zong F. A novel hybrid method for flight departure delay prediction using random forest regression and maximal information coefficient. Aerosp Sci Technol. 2021;116:106822.
6. Mamdouh M, Ezzat M, Hefny H. Optimized planning of resources demand curve in ground handling based on machine learning prediction. Int J Intell Syst Appl. 2021;13(1):1–16.
7. Evler J, Asadi E, Preis H, Fricke H. Airline ground operations: Schedule recovery optimization approach with constrained resources. Transp Res Part C Emerg Technol. 2021;128:103129.
8. Sharma M, Kumar CJ, Deka A. Land cover classification: a comparative analysis of clustering techniques using Sentinel-2 data. Int J Sustain Agric Manag Inform. 2021;7(4):321.
9. Wang C, Hu M, Yang L, Zhao Z. Prediction of air traffic delays: an agent-based model introducing refined parameter estimation methods. PLoS ONE. 2021;16(4):e0249754. https://doi.org/10.1371/journal.pone.0249754.
10. Qu J, Wu S, Zhang J. Flight delay propagation prediction based on deep learning. Mathematics. 2023;11(3):494.
11. Abdel-Aty M, Lee C, Bai Y, Li X, Michalak M. Detecting periodic patterns of arrival delay. J Air Transp Manag. 2007;13(6):355–61.
12. Mamdouh M, Ezzat M, Hefny HA. Airport resource allocation using machine learning techniques. Intel Artif. 2020;23(65):19–32.
13. Lin Y, Li L, Ren P, Wang Y, Szeto WY. From aircraft tracking data to network delay model: a data-driven approach considering en-route congestion. Transp Res Part C Emerg Technol. 2021;131:103329.
14. Dahl GE, Sainath TN, Hinton GE. Improving deep neural networks for LVCSR using rectified linear units and dropout. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE; 2013. p. 8609–13.
15. Report. https://www.bts.gov/explore-topics-and-geography/topics/time-data. Accessed 20 Feb 2022
16. Li Q, Jing R. Generation and prediction of flight delays in air transport. IET Intell Transp Syst. 2021;15(6):740–53. https://doi.org/10.1049/itr2.12057.
17. Yu B, Guo Z, Asian S, Wang H, Chen G. Flight delay prediction for commercial air transport: a deep learning approach. Transp Res Part E Logist Transp Rev. 2019;125:203–21.
18. Gui G, Liu F, Sun J, Yang J, Zhou Z, Zhao D. Flight delay prediction based on aviation big data and machine learning. IEEE Trans Veh Technol. 2020;69(1):140–50.
19. Guleria Y, Cai Q, Alam S, Li L. A multi-agent approach for reactionary delay prediction of flights. IEEE Access. 2019;7:181565–79.
20. Cheevachaipimol W, Teinwan B, Chutima P. Flight delay prediction using a hybrid deep learning method. Eng J. 2021;25(8):99–112.
21. Sahadevan D, Ponnusamy P, Nelli M, Gopi V. Predictability improvement of scheduled flights departure time variation using supervised machine learning. Int J Aviat Aeronaut Aerosp. 2021;8(2):9.
22. Alla H, Moumoun L, Balouki Y. A multilayer perceptron neural network with selective-data training for flight arrival delay prediction. Sci Program. 2021;2021:1–12.
23. Bisandu DB, Moulitsas I, Filippone S. Social ski driver conditional autoregressive-based deep learning classifier for flight delay prediction. Neural Comput Appl. 2022;34(11):8777–802. https://doi.org/10.1007/s00521-022-06898-y.
24. Airmiles. 2022. https://www.airmilescalculator.com/distance/hnl-to-bos/. Accessed 25 May 2022.
25. Yi J, Zhang H, Liu H, Zhong G, Li G. Flight delay classification prediction based on stacking algorithm. J Adv Transp. 2021;2021:1–10.

26. Shao W, Prabowo A, Zhao S, Tan S, Koniusz P, Chan J, 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. New York, NY, USA: ACM; 2019. p. 432–5. https://doi.org/10.1145/3347146.3359079
27. Bao J, Yang Z, Zeng W. Graph to sequence learning with attention mechanism for network-wide multi-step-ahead flight delay prediction. Transp Res Part C Emerg Technol. 2021;130:103323.
28. Chakrabarty N, Kundu T, Dandapat S, Sarkar A, Kole DK. Flight arrival delay prediction using gradient boosting classifier. In: Abraham A, Dutta P, Mandal JK, Bhattacharya A, Dutta S, editors. Advances in intelligent systems and computing. Singapore: Springer; 2019. p. 651–9. https://doi.org/10.1007/978-981-13-1498-8_57.
29. 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). IEEE; 2019. p. 102–7. https://ieeexplore.ieee.org/document/8876970/. Accessed 02 Feb 2023.
30. Wang X, Wang Z, Wan L, Tian Y. Prediction of flight delays at Beijing capital international airport based on ensemble methods. Appl Sci. 2022;12(20):10621.
31. Rahul R, Kameshwari S, Pradip Kumar R. Flight delay prediction using random forest classifier. In: Kumar A, Senatore S, Gunjan VK, editors. ICDSMLA 2020. Singapore: Springer; 2022. p. 67–72. https://doi.org/10.1007/978-981-16-3690-5_7.
32. Zhou H, Li W, Jiang Z, Cai F, Xue Y. Flight departure time prediction based on deep learning. Aerospace. 2022;9(7):394.
33. Kalyani NL, Jeshmitha G, Sai U. BS, Samanvitha M, Mahesh J, Kiranmayee BV. Machine learning model - based prediction of flight delay. In: 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). IEEE; 2020. p. 577–81. https://ieeexplore.ieee.org/document/9243339/. Accessed 15 Feb 2023.
34. Yazdi MF, Kamel SR, Chabok SJM, Kheirabadi M. Flight delay prediction based on deep learning and Levenberg-Marquart algorithm. J Big Data. 2020;7(1):106. https://doi.org/10.1186/s40537-020-00380-z.
35. Liu P, Qiu X, Huang X. Recurrent neural network for text classification with multi-task learning. arXiv Prepr. 2016. https://doi.org/10.48550/arXiv.1605.05101.
36. Sagnika S, Mishra BSP, Meher SK. An attention-based CNN-LSTM model for subjectivity detection in opinion-mining. Neural Comput Appl. 2021;33(24):17425–38. https://doi.org/10.1007/s00521-021-06328-5.
37. Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv Prepr arXiv14123555. 2014
38. Freidman JH. Greedy function approximation : a gradient boosting machine. Vol. 29, Institue of Mathematical Statistics. 2008. p. 1189–232. https://statweb.stanford.edu/~jhf/ftp/trebst.pdf. Accessed 10 May 2023.
39. Sahoo R, Pasayat AK, Bhowmick B, Fernandes K, Tiwari MK. A hybrid ensemble learning-based prediction model to minimise delay in air cargo transport using bagging and stacking. Int J Prod Res. 2022;60(2):644–60. https://doi.org/10.1037/xge0000076.
40. Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat. 2001;29(5):1189–232. https://doi.org/10.1214/aos/1013203451.full.
41. Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A. Catboost: Unbiased boosting with categorical features. Vols. 2018-Decem, Advances in Neural Information Processing Systems. 2018. p. 6638–48.
42. Hancock JT, Khoshgoftaar TM. CatBoost for big data: an interdisciplinary review. J Big Data. 2020. https://doi.org/10.1186/s40537-020-00369-8.
43. Kumar CJ, Das PR, Hazarika A. Autism spectrum disorder diagnosis and machine learning: a review. Int J Med Eng Inform. 2022;14(6):512.
44. Kumar CJ, Das PR. The diagnosis of ASD using multiple machine learning techniques. Int J Dev Disabil. 2022;68(6):973–83. https://doi.org/10.1080/20473869.2021.1933730.
45. Elbeltagi A, Pande CB, Kumar M, Tolche AD, Singh SK, Kumar A, et al. Prediction of meteorological drought and standardized precipitation index based on the random forest (RF), random tree (RT), and Gaussian process regression (GPR) models. Environ Sci Pollut Res. 2023;30(15):43183–202. https://doi.org/10.1007/s11356-023-25221-3.
46. Al-Mukhtar M. Modeling the monthly pan evaporation rates using artificial intelligence methods: a case study in Iraq. Environ Earth Sci. 2021;80(1):39. https://doi.org/10.1007/s12665-020-09337-0.
47. Zhu J, Su Y, Liu Z, Liu B, Sun Y, Gao W, et al. Real-time biomechanical modelling of the liver using LightGBM model. Int J Med Robot Comput Assist Surg. 2022. https://doi.org/10.1002/rcs.2433.
48. Porwik P, Doroz R, Wrobel K. An ensemble learning approach to lip-based biometric verification, with a dynamic selection of classifiers. Expert Syst Appl. 2019;115:673–83. https://doi.org/10.1016/j.eswa.2018.08.037.
49. Zhu Y, Zhou L, Xie C, Wang GJ, Nguyen TV. Forecasting SMEs' credit risk in supply chain finance with an enhanced hybrid ensemble machine learning approach. Int J Prod Econ. 2019;211:22–33.
50. Sharma M, Kumar CJ, Talukdar J, Singh TP, Dhiman G, Sharma A. Identification of rice leaf diseases and deficiency disorders using a novel DeepBatch technique. Open Life Sci. 2023;18(1):20220689. https://doi.org/10.1515/biol-2022-0689/html.
51. Sharma M, Kumar CJ, Deka A. Early diagnosis of rice plant disease using machine learning techniques. Arch Phytopathol Plant Prot. 2022;55(3):259–83. https://doi.org/10.1080/03235408.2021.2015866.
52. Sharma M, Nath K, Sharma RK, Kumar CJ, Chaudhary A. Ensemble averaging of transfer learning models for identification of nutritional deficiency in rice plant. Electronics. 2022;11(1):148.
53. Sharma M, Kumar CJ. Improving rice disease diagnosis using ensemble transfer learning techniques. Int J Artif Intell Tools. 2022;31(08):2250040. https://doi.org/10.1142/S0218213022500403.
54. Bhadra S, Kumar CJ. An insight into diagnosis of depression using machine learning techniques: a systematic review. Curr Med Res Opin. 2022;38(5):749–71. https://doi.org/10.1080/03007995.2022.2038487.
55. Bhadra S, Kumar CJ. Enhancing the efficacy of depression detection system using optimal feature selection from EHR. Comput Methods Biomech Biomed Engin. 2023. https://doi.org/10.1080/10255842.2023.2181660.
56. Ribeiro MHDM, dos Santos Coelho L. Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series. Appl Soft Comput. 2020;86:105837. https://doi.org/10.1016/j.asoc.2019.105837.

57. Bai B, Li G, Wang S, Wu Z, Yan W. Time series classification based on multi-feature dictionary representation and ensemble learning. Expert Syst Appl. 2021;169:114162. https://doi.org/10.1016/j.eswa.2020.114162.
58. Kaggle. 2020. https://www.kaggle.com/datasets/ioanagheorghiu/historical-flight-and-weather-data/code. Accessed 4 Apr 2022.
59. NGUYEN TD. Catching that flight: Visualizing social network with Networkx and Basemap. 2018. https://towardsdatascience.com/catching-that-flight-visualizing-social-network-with-networkx-and-basemap-ce4a0d2eaea6.
60. Keras. 2021. https://keras.io/. Accessed 5 Jan 5 2022.
61. Core_30. 2023. https://aspm.faa.gov/aspmhelp/index/Core_30.html. Accessed 1 Jul 2023.
62. Mrňa D. Internet of things as an optimization tool for smart airport concept. Eur Transp Eur. 2021;82(82):1–15.
63. Madana AL, Shukla VK, Sharma R, Nanda I. IoT Enabled Smart Boarding Pass for Passenger Tracking Through Bluetooth Low Energy. In: 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE). IEEE; 2021. p. 101–6.
64. Zeng L, Zhao M, Liu Y. Airport ground workforce planning with hierarchical skills: a new formulation and branch-and-price approach. Ann Oper Res. 2019;275(1):245–58.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.