Delia SCHÖSSER, M.Sc.[1]
(Corresponding author)
E-mail: delia.schoesser@tu-dresden.de
Jörn SCHÖNBERGER, Prof. Dr.[1]
(Corresponding author)
E-mail: joern.schoenberger@tu-dresden.de
[1] Technical University Dresden
"Friedrich List" Faculty of Transport and Traffic Sciences
Institute of Transport and Economics
Chair of Transport Services and Logistics
01062 Dresden, Germany

Science in Traffic and Transport
Original Scientific Paper
Submitted: 23 Mar. 2022
Accepted: 8 July 2022

ON THE PERFORMANCE OF MACHINE LEARNING BASED FLIGHT DELAY PREDICTION – INVESTIGATING THE IMPACT OF SHORT-TERM FEATURES

ABSTRACT

People and companies today are connected around the world, which has led to a growing importance of the aviation industry. As flight delays are a big challenge in aviation, machine learning algorithms can be used to forecast those. This paper investigates the prediction of the occurrence of flight arrival delays with three prominent machine learning algorithms for a data set of domestic flights in the USA. The task is regarded as a classification problem. The focus lies on the investigation of the influence of short-term features on the quality of the results. Therefore, three scenarios are created that are characterised by different input feature sets. When forgoing the inclusion of short-term information in order to shift the prediction timing to an early point in time, an accuracy of 69.5% with a recall of 68.2% is achieved. By including information on the delay that the aircraft had on its previous flight, the prediction quality increases slightly. Hence, this is a compromise between the early prediction timing of the first model and the good prediction quality of the third model, where the departure delay of the aircraft is added as an input feature. In this case, an accuracy of 89.9% with a recall of 83.4% is obtained. The desired timing of prediction therefore determines which features to use as inputs since short-term features significantly improve the prediction quality.

KEYWORDS

flight delay prediction; machine learning; aviation; feature importance; classification; SHAP.

1. INTRODUCTION

Globalisation and digitisation have led to a highly connected world where a tremendous amount of data is generated by companies, machines and individuals. One approach to manage substantial data sets is the use of automated methods like machine learning (ML). ML algorithms are able to learn from historical data and apply the knowledge gained to new records or situations. The algorithms analyse large data sets and thereby identify patterns within the data to generate a respective output [1].

One area that has a high potential for the application of ML algorithms is the aviation industry, which is growing steadily. In 2019, 811 million people took domestic flights in the USA, a 4.3% growth compared to the year before [2]. Airplane travel generates a lot of data, for example information on the flight schedule and the aircraft as well as data about the passengers or the weather conditions. This large amount of data can be used to predict an output for future situations, such as the arrival delay of a flight or the amount of fuel an airplane consumes. The knowledge gained thereby is helpful for implementing optimisations in current processes to save money, assets or manual work.

A major challenge in civil aviation are flight delays. According to the Bureau of Transportation Statistics (BTS), an airplane is considered as delayed if it arrives at least 15 minutes after the scheduled arrival at its destination airport [3]. The total cost of flight delays in the USA increased in recent years from 19.2 billion USD in 2012 to 33 billion USD in 2019 [4]. When affected by delays, dissatisfaction grows among passengers because of missed connection flights. Both airlines and airports have to deal with increased costs and challenges in planning. Even though there is an ongoing effort in improving the current air traffic management (ATM) processes to minimise delays, e.g. by demand and capacity planning [5], the costs resulting from delays are still increasing. Consequently, flight delay prediction is advantageous as efficient countermeasures can be implemented [6, 7]. Previous publications also underline the relevance of this subject [6–12]. All involved parties benefit from flight delay prediction as passengers can plan sufficient time for transfers, and airlines and airports are able to identify root causes and reduce adverse consequences. Hence, ML algorithm application can be useful because a lot of data are generated in aviation and many features and combinations of those potentially influence postponements. Thereby, models can be built which are able to predict delays of future flights in order to achieve the advantages just named.

As flight delays pose a significant problem, this paper aims to examine and evaluate the prediction of flight arrival delays with ML methods. Thereby, a classification approach is taken, which means the objective is to predict if a flight will arrive with a delay or on time. The focus of the examinations lies on the analysis of the influence of different features on the prediction quality. Especially the features "Departure Delay" and "Arrival Delay of the Previous Flight" (with the same aircraft) are targeted as those are short-term information, whose inclusion shifts the possible prediction timing forwards. In the scope of this paper, a short-term feature is a feature that is only available a short time (i.e. several hours or less) before the time the actual arrival delay is known. Furthermore, one aim is to identify features that have an influence on the prediction quality of flight arrival delays to give recommendations on potential improvement areas. Thereby, standard ML methods are applied as the goal is not to develop those algorithms further, but to investigate the influence of the features just explained. A data set that includes 5.82 million domestic flights in the USA is used for this purpose [13].

Various factors potentially influence the delay of a flight, but they usually interact with each other, and the relations are often non-linear. As looking at each influential feature individually is too complex, the application of ML is advantageous for this task. *Figure 1* depicts the share of delayed flights over the year for the given data set. As one can see, there is no linear relationship between the number of flights and the share of delayed flights. The rate is highest in June and lowest in October, but the numbers of flights in those two months are comparable.

Figure 1 depicts an example of one executed analysis, which demonstrates the complexity of the task to predict flight delays. This is supported by the correlation matrix depicted in *Figure 2*. Most features are not clearly correlated, the exceptions being

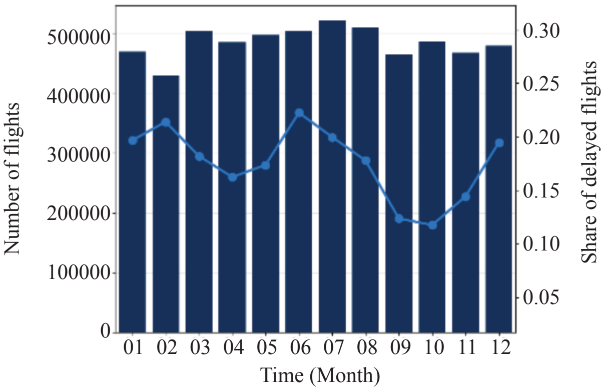


Figure 1 – Share of delayed flights

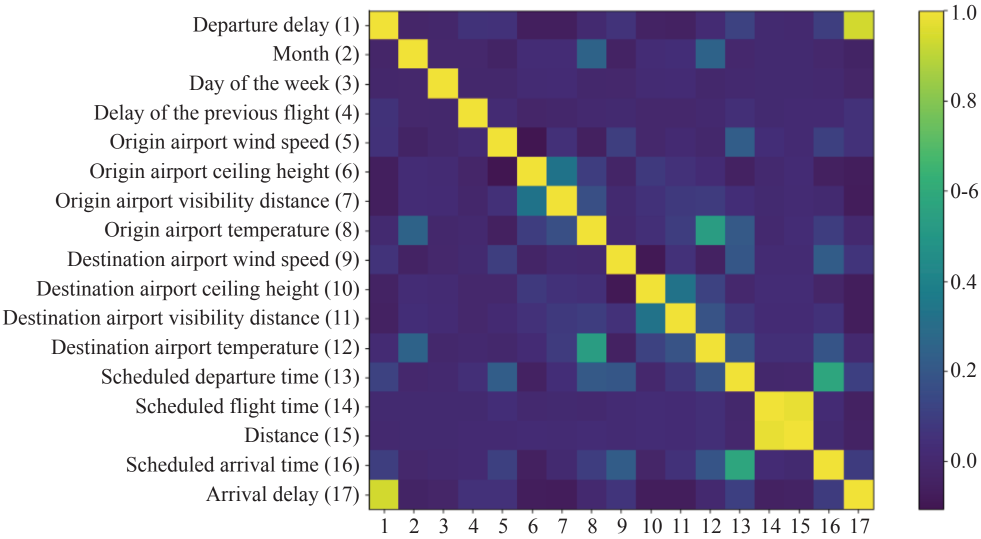


Figure 2 – Correlation matrix

"Departure Delay" and "Arrival Delay" as well as "Scheduled Flight Time" and "Distance", which is logical as flights usually take longer if the distance is larger. Hence, the dependence of the output feature "Arrival Delay" from the other features cannot be explained easily, therefore ML algorithms are used in order to predict flight arrival delays.

After analysing and pre-processing the records, the three ML algorithms, Random Forest (RF), XGBoost and Neural Network (NN), are applied to the data. The performance of the algorithms is measured and improved by modifying the considered features and adapting the hyperparameters of the algorithms. The objective of the described approach is to practically examine the ability of ML algorithms to predict flight delays for the given data set with varying explanatory variables.

The remainder of this paper is structured as follows: Section 2 summarises works on the subjects of ML applications in aviation as well as specifically on flight delay prediction. Section 3 deals with the methodology applied, which includes a description of the data and algorithms used as well as relevant performance metrics. In Section 4 the results are described, analysed and discussed in order draw a conclusion in Section 5.

2. RELATED WORKS

ML is an effective way to use large amounts of data for solving prediction tasks. One key performance indicator to evaluate the quality of the forecast is the accuracy of the model. Its information value depends on the respective task or application [1, 14].

Several authors have investigated ML applications in the aviation sector. The objective of Burnett and Si [15] was to predict if injuries or fatalities are likely to happen in the case of an aviation accident and to identify causes that increase their likelihood consequently. Horiguchi et al. [16] examined the prediction of airplane fuel consumption as the amount of fuel onboard has security and monetary impacts. Furthermore, Jan and Chen [17] addressed the detection of unusual weather, such as downdraft or turbulence during the flight.

2.1 Flight delay prediction without the application of machine learning

Two authors who dealt with the topic of flight delay analysis and prediction without applying ML methods are Yablonsky et al. [18] and Ding [7]. Yablonsky et al. [18] identified causes that have a high impact on flight delays. Overall, the National Aviation System (NAS) was responsible for most delays, which includes postponements due to air traffic control or airport operations. The second most important cause were late arriving airplanes that produced a delay for subsequent flights. Ding [7] used multiple linear regression for predicting flight delays. The author stated that departure delay and flight distance were the two most important features. Ding was able to predict flight delays with an accuracy of nearly 80% when considering those two features as explanatory variables. However, as departure delay is usually not known before take-off, prediction can only be carried out at short notice.

2.2 Machine learning based prediction of flight delays

Yazdi et al. [8] used a denoising autoencoder as well as the Levenberg-Marquart algorithm to predict flight delays as flight data often include a lot of noise. They achieved an accuracy of 96%, but "Departure Delay", a short-term feature, is one of their input features.

The performance of several ML methods (specifically Random Forest (RF), Logistics Regression, K-nearest Neighbours, Decision Tree (DT) and Naïve Bayes) was evaluated by Huo et al. [9]. They obtained flight data for 2018 for 161 airports, which only included information on flight schedule. The algorithm with the best performance was RF with an accuracy of roughly 70%. In contrast to [8], they did not include any short-term features.

Belcastro et al. [6] also exclusively utilised features which are available several days before departure. The data set used contained roughly 30 million records of domestic flights in the USA from five years with specific information about the flight schedule as well as the weather. From several ML algorithms applied, RF achieved the best results. When classifying flights up to a delay of 15 minutes as on-time, they achieved an accuracy of 74%. Additionally, they discovered that including weather observations improved the results and thereby demonstrated that weather has a significant influence on flight delays.

Another team of authors who took weather conditions into account is Gui et al. [10]. They collected information on airports, air routes and flights for 7500 records, but they did not state explicitly if short-term features were included. The authors dealt with the task as a binary classification problem with four categories. With the application of the ML algorithm RF, they achieved an accuracy of 90% for binary classification and 70% when considering four categories.

Other authors who also included weather features (besides information on flight schedule and "Departure Delay") are Kalyani et al. [11]. They took a two-layered approach, which consisted of first identifying a delay (binary classification) with the algorithm XGBoost to subsequently predict the exact length of the delay (regression) with linear regression. A data set with roughly 200,000 US domestic flights from six months in 2019 was utilised. The authors achieved an accuracy of 94%.

Manna et al. [12] dealt with the task solely as a regression problem. Their data set contained information on the flight schedule. The features used were all available before departure (thus non-short-term), therefore delay prediction could be made in advance. For solving that task, the authors used the ML algorithm Gradient Boosted DT (also known as XGBoost). Eventually, Manna et al. obtained a root mean squared error of 10.7 minutes.

Multiple approaches for solving the task of predicting flight delays have been presented in this section, with both ML and non-ML methods. Especially the application of ML algorithms allows the generation of models to accurately predict flight delays. Thus, those methods are in the focus of this paper. Summarised, it is interesting that some authors include short-term features such as "Departure Delay" and some do not. Another difference is that the authors apply various ML algorithms, but tree-based methods, like RF, are predominant.

2.3 Contribution of this work

The studies presented above employ different feature sets. Hence, this work particularly aims to examine the influence of the two short-term features "Departure Delay" and "Arrival Delay of the Previous Flight" as those are data which potentially improve prediction quality, but also shift the prediction timing forwards. The framework of the ML application has to be defined depending on the objective as some goals do not allow including short-term features. To the best of our knowledge, authors have either included those features or did not mention them, but the influence of the features has not been discussed. However, this is of importance as it strongly impacts the possible applications of flight delay prediction.

3. METHODOLOGY

This section presents the techniques used to achieve the results presented in Section 4. This covers an introduction of the data set as well as an explanation of relevant features and applied algorithms. Additionally, relevant performance metrics are described in order to evaluate the results.

3.1 Data

The main data set contains 5.82 million records for domestic passenger flights from and to 322 airports in the USA with 14 airlines [13]. *Figure 3* depicts all relevant features that are included in the data set, separated into categories for a better overview. The features belonging to *Flight, Airport* and *Schedule* describe the general framework for each flight, hence they are available already several months

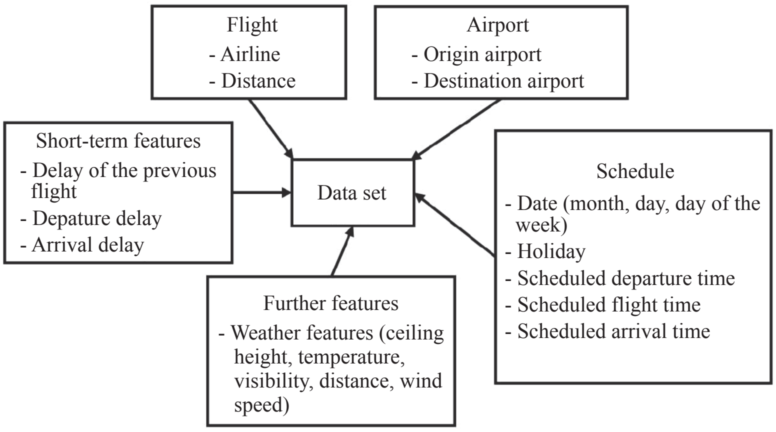


Figure 3 – Relevant features of the data set

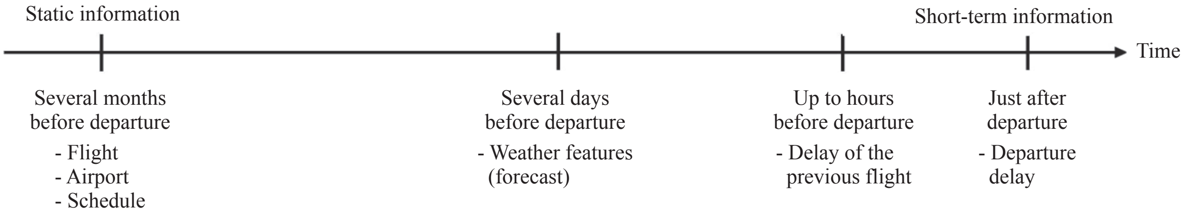

Static information
Short-term information
Time
Several months before departure
- Flight
- Airport
- Schedule
Several days before departure
- Weather features (forecast)
Up to hours before departure
- Delay of the previous flight
Just after departure
- Departure delay

Figure 4 – Temporal availability of the data

before the departure (see *Figure 4*). The dependent variable, which is to be predicted, is the "Arrival Delay", classified as *Short-term Feature* in *Figure 3*. The records have either positive or negative values for this characteristic. A negative delay, also called absorbed delay, means that the airplane arrives earlier at its destination than scheduled as any phase of the flight consumes less time than planned. Positive values mean that a delay is generated as more time than scheduled is needed for one or multiple phases of the flight [19].

Weather features are furthermore added given that those are identified as highly influential in the related research [6, 10, 11]. The information is collected by meteorological stations across the country and is provided by the National Oceanic and Atmospheric Administration (NOAA), an agency within the United States Department of Commerce [20]. Various weather information for each airport is available for the whole year. As being relevant for influencing flight delays, the four features named in *Figure 3* under *Further Features* are selected. The intention is to use forecasted weather data for real-world implementation as shown in *Figure 4*. However, for the experiments executed in this paper, the historical true weather data are used due to a lack of weather forecast data. Measurements are obtained at hourly intervals, so that weather conditions for the scheduled departure and arrival times for each flight are available. If disadvantageous weather conditions are forecasted at a scheduled departure or arrival time, this potentially leads to a delay, which justifies using the weather forecast for the scheduled times. Generally, as shown in *Figure 4*, information on the weather conditions is available several days before the departure of a flight through forecasts. Their reliability depends on the prediction timing. According to NOAA, a forecast that is made seven days in advance has an accuracy of 80% [21].

The features "Delay of the Previous Flight" and "Departure Delay" are two *Short-term Features*, whose influence on "Arrival Delay" is to be determined. This investigation is essential as delay propagation is one cause for delayed flights and can be represented partially by those two *Short-term Features* [19]. There are several authors who include the departure delay of a flight in order to predict arrival delay, which usually leads to a satisfying performance of the algorithm. However, as this feature is only available just after departure, it is questionable if departure delay should be used as input. In order to determine its influence and discuss this issue further, this feature is part of the relevant features. "Delay of the Previous Flight" (arrival delay of the same aircraft on its previous flight) is also a feature which is further investigated as this information is available earlier than departure delay and hence might be a good compromise between long-term predictions and good performance. For this paper, delay of the previous flight (in minutes) is added to the record if the aircraft arrived within four hours before the scheduled departure of the upcoming flight (see *Figure 4*). If a longer period lies between two flights, the delay of the first one will probably not affect the second one severely anymore and is thus not considered and set to 0.

The result of the data set analysis is that most flight records are complete, so entries are available for all features. Overall, a maximum of 1.8% of the values of a feature are missing. Concerning the correctness of the data, it is detected that the airport codes in October have a wrong format, they are five-digit numbers instead of International Air Transport Association (IATA) three-letter codes. Hence, this is corrected during the pre-processing phase. All other values seem to be reasonable.

In total, 80.6% of all flights arrive at their destination punctually, hence 14 minutes after the scheduled arrival time at the latest. This threshold is selected according to the Bureau of Transportation Statistics (BTS), which considers an airplane as delayed if it arrives at least 15 minutes after the scheduled arrival at its destination airport [3].

As the majority (roughly 80%) of all flights are punctual, this class is over-represented, and the data set is unbalanced. That issue requires special attention when evaluating the results, which is further discussed in Sections 4.2 and 4.3.

The original main data set contains 5.82 million flights to 322 airports in the USA. As many small locations with little traffic are included, the focus is placed on the Core 30 airports to ensure that for every airport sufficient samples are available so that the algorithms are able to generalise. Those 30 airports have great importance as well as significant activities and serve major metropolitan areas. The Federal Aviation Administration (FAA) defines the list of the Core 30 airports [22]. After deleting all records that include non-Core 30 airports, 2.23 million records are left, which are roughly 40% of the original data set. Thereby, the computation time for every step is reduced as well. The new data set is used for all further pre-processing tasks.

3.2 Algorithms

The two tree-based ML algorithms Random Forest (RF) and XGBoost as well as the algorithm Neural Network (NN), which belongs to the ML-subfield of deep learning, are applied in order to evaluate their performances with respect to the regarded data. Those algorithms are selected as they are standard methods and frequently used by other authors dealing with similar problems (see Section 2). The algorithms are not modified or developed further in the scope of this paper as the focus lies on the analysis of the influence of several input features on the prediction quality. The mathematical details of the algorithms can be found in the referenced literature.

The ML algorithm creates a model, which contains all the rules that are built when training the algorithm with data. With the help of ML, patterns are automatically detected in data sets in order to learn from those and predict an outcome of future unknown records [23]. Hence, ML is not directly about optimising processes, but about learning patterns to make forecasts and thus to derive knowledge that can be used to implement optimizations.

In general, ML can be distinguished into supervised, unsupervised and reinforcement learning [23]. The regarded problem is solved with methods of supervised learning. It means that input as well as output values of each record have to be available so that the algorithm can learn their relation. As there are many different records, the algorithm can build a model that is able to generalise the underlying patterns and learn from them in order to give predictions for unknown records [1, 24].

A very common and simple algorithm is Decision Tree (DT). It is a tree-based method that creates an output based on decision rules, which are evaluated sequentially at each node. DT is a rather simple algorithm that is usually outperformed by other methods [1, 23]. It is not applied for this paper, but it forms the basis for further algorithms that are used in the following.

A more complex tree-based algorithm is RF, which is an ensemble learning method because it combines several DTs. One of its strengths is that the variance of predictions is reduced in comparison to a single DT. This is achieved as RF consists of many simple trees that are each trained with a random subset of the whole data set. Every subset contains a share of the records and of the explanatory features. Consequently, each tree is trained with a different data set and is thus constructed uniquely [1]. For the implementations of RF in Python, the library *scikit-learn* is used [25].

Gradient Boosting DT is another tree-based ensemble learning method. The library, which contains its implementation, is called XGBoost [26]. This name is used for the algorithm in the scope of this paper. Similar to RF, it is an improved version of the DT algorithm. XGBoost applies Gradient Boosting on DTs, which takes multiple weak models to build a stronger prediction model by minimising their error. In contrast to RF, the construction of the trees happens sequentially and not in parallel. Thus, the method is exhaustive as it analyses all features at every stage. In each iteration, a new model is integrated into the existing one. The additional tree is developed by giving more weight to the samples that were not predicted correctly at the earlier stage. Hence, the forecasting errors are reduced in each iteration [12].

The last algorithm that is applied is NN. It differs from the other procedures as it is not tree-based. NNs try to reproduce the structure of the human brain. They are therefore built of one input layer, one or multiple hidden layers and an output layer. All layers consist of a predefined number of nodes, which are also called neurons [1]. In the beginning, the input data are normalised to fit the values in an interval between zero and one. In order to create a model, each link between two

Table 1 – Confusion Matrix [1]

		Predictions	
		Positive	Negative
Actual values	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

neurons of subsequent layers is initially assigned a random weight. Every input is weighted when being forwarded to a node. Then, a bias is added to the sum of all weighted inputs. Thereafter, an activation function is applied before the output is forwarded to all subsequent nodes in the next layer. This procedure continues until the output layer is reached, where the overall result is calculated. Depending on the error between this output and the actual observed value, the weights of the neurons are adapted during further training iterations, which are called epochs [14, 24]. For this paper, the library used for building NNs is called TensorFlow, Keras forms an interface to TensorFlow [27, 28].

Each of the presented algorithms is characterised by hyperparameters, which influence their performance. They have to be set by the user in advance to the learning process. This requires several runs with different hyperparameters to find an optimal configuration for the given problem. The execution of the so-called hyperparameter tuning is explained in Section 4.3. The tunable hyperparameters and their explanations are given in [25–27].

3.3 Performance metrics

For measuring the ability of a model to predict a correct output, such as the expected delay of a flight, different performance metrics can be taken into account. Those allow to compare the results of different models, for example when testing different sets of hyperparameters. Specifically, the real values of the output feature are compared to the predictions made by the model.

In case the output values are grouped into classes (classification approach), the predictions by the algorithm can only be correct or wrong. The basis for all considered performance metrics is the confusion matrix, as shown in *Table 1* [29].

In this case, the two possible classes are Positive (i.e. flights that are delayed) and Negative (i.e. punctual flights). Depending on the predicted values, all records are assigned to one field in the table. According to the Bureau of Transportation Statistics (BTS), an airplane is considered as delayed if it arrives at least 15 minutes after the scheduled arrival at its destination airport [3]. This definition is also applied for the paper.

Accuracy, given in *Equation 1*, provides the share of correct predictions, so it gives information on the overall performance of the model. However, it does not say how well a particular class is predicted [6, 29].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

To obtain information on performance regarding a specific class, the evaluation criterion recall, also called sensitivity, is generally useful. It describes the share of delayed flights that is correctly predicted. *Equation 2* gives the corresponding calculation. The counterpart for the Negative class (i.e. punctual flights) is called specificity. Within the scope of this paper, the recall is more informative as it focuses on the minority class. This is important as our aim is to identify delayed flights. The precision (also given in *Equation 2*) describes the rate of actually delayed flights among all flights that are predicted as delayed [1,14].

$$\begin{aligned} Recall &= \frac{TP}{TP + FN} \\ Precision &= \frac{TP}{TP + FP} \end{aligned} \tag{2}$$

The F_β score combines the recall and precision metrics by calculating their weighted harmonic mean. Thus, extreme values of one of the single metrics are more penalised compared to the arithmetic mean. By adding the weight β, one of the classes can be emphasised. A value of 1 means that the classes are considered equally important (known as F1 score). As the delayed flights offer more information to us, β is set to 2 [30].

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{3}$$

4. RESULTS

This section presents the results obtained by the application of three ML algorithms to predict flight arrival delays. The focus is hereby on analysing the impact of short-term features and identifying influential features. The implementations of the

ML algorithms are taken from different libraries without modification [25–27]. The evaluation of the results is crucial and helps to understand the quality of the predictions made by the models.

4.1 Experimental approach

The target feature to be predicted is "Arrival Delay" with the two categories "punctual" and "delayed". In the context of this paper, a delay of less than 15 minutes is still considered as punctual, while an airplane that arrives later at its destination is counted as delayed [3]. During the training phase, the algorithms are employed to identify patterns in the data in order to develop a prediction model, which is validated in the subsequent step, the test phase [1]. Three combinations of sets of input features are used in order to investigate the influence of the short-term features "Delay of the Previous Flight" and "Departure Delay". The features belonging to each set are given below; relevant is set (1) and the combinations of sets (1)+(2) as well as sets (1)+(3). The features are explained in more detail in Section 3.1: (1) Standard set (all features from *Figure 3*, except *Short-term Features*), (2) "Delay of the Previous Flight" and (3) "Departure Delay".

In order to train the algorithms and to validate the model afterwards, the data set is split into a train and a test subset. The test data are unseen by the model in the training phase, which is required for a realistic evaluation of the model's performance. Commonly used train-test splits are 80:20 or 75:25 ratios [9,11]. For this paper, a train-test split of 75:25 is chosen, which means the model is trained with 75% of the records and tested with the remaining 25%.

In the next step, the respective algorithm is trained and then the performance of the model is evaluated by means of the test subset. The evaluation metrics are accuracy, recall and the F_β score (see Section 3.3). When training the algorithm, several hyperparameters can be set in order to specify a framework for the algorithm and thus to improve the results.

4.2 Non-optimised models

In the beginning, the default values of the hyperparameters [25–27] are chosen to produce first results. Neural Network (NN) consists of one hidden layer with 100 neurons. The performances of the models are summarised in *Table 2*.

At first glance, the results seem to be pretty good (independently from the set of input features) as the accuracy of all models is higher than 80%, which means that more than 80% of all records in the test set are correctly predicted as punctual or delayed. However, the recall and F_β score are rather low for input feature sets (1) as well as (1)+(2), especially for NN. The reason for it is that most flights that are actually delayed are predicted as punctual, so the prediction is wrong in those cases. The high accuracy at the same time can be explained by the structure of the data set, as it is unbalanced. This means that one class is over-represented and the other one is under-represented. In the case of the given data set, 80% of all records arrived punctually (over-represented class) and only 20% of the flights were delayed (under-represented class). Hence, if a model always predicts that a flight is punctual, an accuracy of 80% and a recall of 0% would be the result. This baseline result emphasises the importance of not solely regarding accuracy as performance metric as it does not sufficiently describe the quality of prediction. The issue of handling an unbalanced data set is further stressed in the following section.

Concerning the different feature sets, the results are slightly improved with the inclusion of "Delay of the Previous Flight" (sets (1)+(2)). The improvement is larger if the feature "Departure Delay" is included instead (sets (1)+(3)), especially when regarding the recall and F_β score. This already shows

Table 2 – Results of non-optimised models (%)

Features	Random forest			XGBoost			Neural Network		
	Accuracy	Recall	F_β	Accuracy	Recall	F_β	Accuracy	Recall	F_β
(1)	82.3	14.0	16.6	82.0	12.1	14.5	81.0	3.3	4.0
(1)+(2)	82.7	17.2	20.2	82.4	16.0	18.9	81.0	3.9	4.8
(1)+(3)	92.7	69.1	72.6	92.8	70.0	73.3	91.1	58.1	62.8

the potential of the two short-term features, where the impact of "Departure Delay" is especially visible.

Further experiments with input feature sets (1)+(2)+(3) are conducted. The results are comparable to those achieved with input feature sets (1)+(3). Consequently, "Delay of the Previous Flight" does not lead to an improvement if "Departure Delay" is considered but involves additional complexity. Hence, the combination from input feature sets (1)+(2)+(3) is not considered any further.

Moreover, experiments are conducted that include the congestion at the airports as input feature. This is identified as being influential on arrival delays in [31]. Therefore, the number of flights departing and arriving at each airport every hour is summed up and put into relation to the maximum demand that arose at the specific airport in one hour over the year. Thereby, the relative congestion at the time of a flight is calculated. However, the inclusion of this information does not bring an improvement, which is why the feature is not further considered even though congestion at an airport might influence delays. The expected reason for the absent improvement is that the underlying data set only contains domestic flights in the USA, and international flights (which are rather frequent at large airports) are not included. Hence, the estimation of the congestion is too inaccurate to improve the results.

Generally, the obtained results are best for the tree-based algorithms Random Forest (RF) and XGBoost. As the results with a low recall and F_β score are not satisfactory yet, especially when regarding feature sets (1) and (1)+(2), the models are further optimised in the following.

4.3 Optimised models

For optimising the results, the hyperparameters of each algorithm are tuned by means of the two widely used methods, Grid Search (GS) and Particle Swarm Optimisation (PSO) from the Optunity library [32]. GS is an approach that tries different sets of hyperparameters in a structured manner, which means that the performance of all possible combinations of parameters that are defined beforehand is tested. It helps in identifying rough ranges for the hyperparameters, but as the approach has a fixed design, information gained during the optimisation procedure is not further considered. This is done with PSO, which is a population-based optimisation method. A predefined number of particles move around the set search space. In each iteration, every particle moves further and tries to improve the previously found solution. Thereby, PSO is able to focus on promising regions in the search space [33]. The hyperparameters that are optimised with respect to a high accuracy and a high recall are given in *Table 3*. Multiple further hyperparameters are tested in pre-experiments, but it appeared that the selected hyperparameters are the most influential ones and thus the tuning focused on those to keep the computational complexity on a reasonable level.

As the available data set is unbalanced (80% of all records belong to the punctual class and only 20% of the flights are delayed), undersampling is applied to the train subset in order to balance it. This method is often used to improve the performance with unbalanced data sets; it randomly reduces the records of the over-represented class until the amount of records of both classes are equal [6, 8, 10].

Table 3 – Tuned hyperparameters

Algorithm	Hyperparameters
Random Forest	– Maximum depth – Maximum number of leaf nodes – Number of estimators (trees)
XGBoost	– Maximum depth – Learning rate η – Minimum loss reduction required for a split γ – Subsample ratio of the training records
Neural Network	– Number of layers – Number of nodes per layer – Number of epochs – Learning rate

Additionally, the reliability of the results is examined through the application of 10-fold cross-validation. Therefore, the data set is divided into ten subsets, which is a commonly used split. The algorithm is thus trained with 90% of the data and tested with the remaining 10% of the records. This procedure is conducted ten times, and each time a different subset is used for testing. The values of the evaluation criteria (accuracy, recall and F_β score) of all runs are averaged. Thereby, the variance of the predictions is analysed [1, 24]. In the cases of the tested models, the reliability of the results is high as the values of the evaluation criteria differ only slightly when applying 10-fold cross-validation. One implication thereof is that an adjustment of the

Table 4 - Results of optimised models (%)

Features	Random Forest			XGBoost			Neural Network		
	Accuracy	Recall	F_β	Accuracy	Recall	F_β	Accuracy	Recall	F_β
(1)	68.7	67.2	56.2	69.5	68.2	57.2	66.1	65.8	54.0
(1) + (2)	70.1	67.2	56.9	70.9	68.5	58.1	66.6	66.5	54.7
(1) + (3)	90.2	81.7	79.4	89.9	83.4	80.3	86.5	85.3	78.8

train-test split by increasing the share of the training subset does not result in a significantly optimised outcome.

Table 4 shows the achieved results of all three algorithms for the different feature sets. Generally, undersampling has led to a decreased accuracy, but a strongly increased recall and F_β score, which are very important metrics. The recall gives the share of delayed flights that was classified correctly, the F_β score takes into account both classes. Overall, the conducted optimisations are successful as significant improvements compared to the non-optimised results in *Table 2* are achieved. When regarding the works presented in Section 2, the results achieved by the authors have a similar quality in comparison to those shown in *Table 4*, but due to different research questions, evaluation metrics, frameworks and data sets, a direct comparison is not possible.

The obtained results of the algorithms RF, XGBoost and NN shown in the table below are similar, but the tree-based algorithms perform slightly better than NN. The better performance of algorithms like RF or XGBoost over other algorithms was already identified by several authors [6, 9, 10].

Comparing RF and XGBoost, the given problem is better solved by the latter algorithm for feature sets (1) as well as (1)+(2) ("Delay of the Previous Flight" included). For feature sets (1)+(3) ("Departure Delay" included), the two algorithms nearly perform equally. Hence, the results produced with XGBoost are further focused for more detailed analyses.

Table 5 exemplarily shows the values of the hyperparameters for XGBoost (for input feature sets (1)+(2)) that produce the best results concerning both accuracy and recall after conducting PSO. For the sake of clarity, the other parameters are not shown here.

Table 5 – Tuned hyperparameters for XGBoost with feature sets (1)+(2)

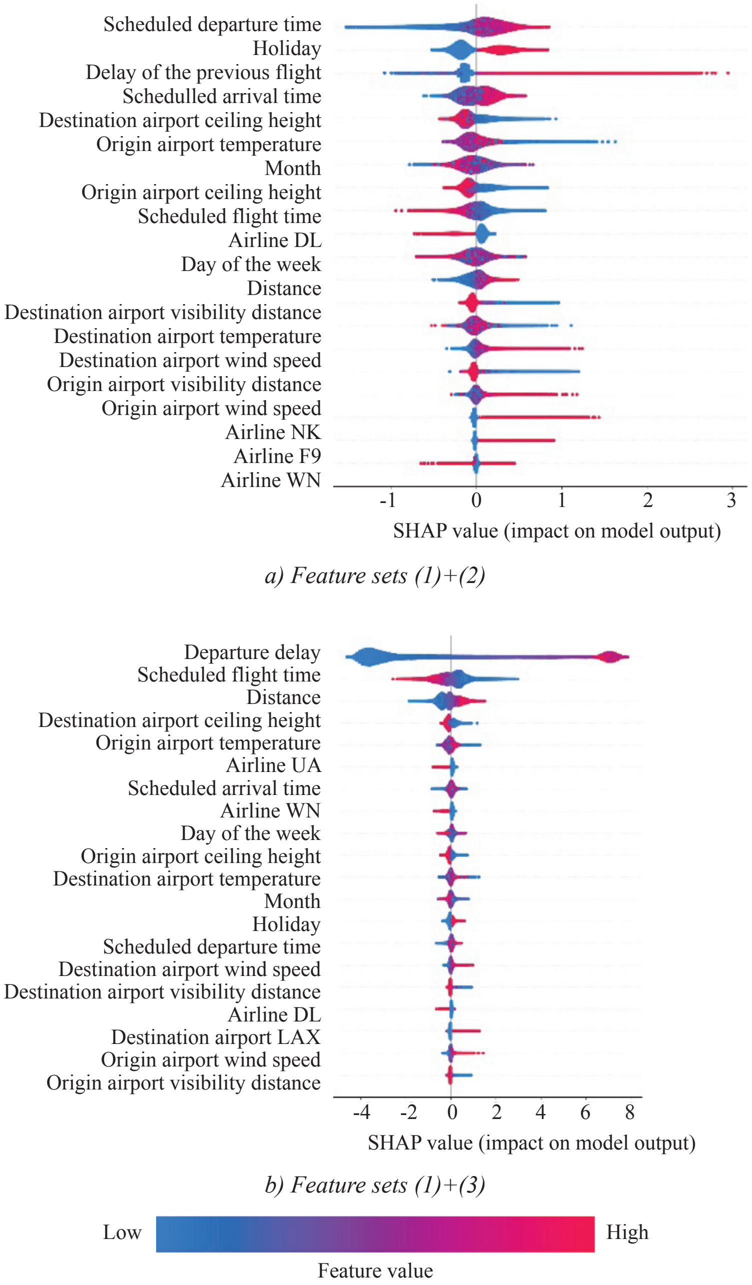
Hyperparameter	Chosen value	Default value
Max. depth	70	6
Learning rate η	0.09	0.3
Min. loss reduction required for a split γ	0.59	0
Subsample ratio of the training records	0.88	1

Regarding the different feature sets of the optimised models (see *Table 4* for the results), the performance is best with sets (1)+(3), with an accuracy of 89.9% and an F_β score of 80.3%. This was already expected because "Departure Delay" correlates with "Arrival Delay" as shown in *Figure 2*. However, as it is a short-term feature and known just after departure, its inclusion only brings a small advantage for the prediction of "Arrival Delay" as the forecast cannot be made several days or at least hours in advance. For this reason, the effect of the feature "Delay of the Previous Flight" is also investigated, as shown in *Table 4* in row (1)+(2). This feature is included as the postponement that results from a delayed previous flight is known earlier than "Departure Delay". In this case, the model does not perform as well anymore, but an improvement compared to input feature set (1) is achieved. The accuracy increased from 69.5% to 70.9% and the F_β score from 57.2% to 58.1%.

Besides measuring the performance with the performance metrics accuracy, recall, and F_β score, the influence of the individual input features is of interest in order to draw conclusions. Therefore, SHapley Additive exPlanations (SHAP) are used to help interpreting the results and to explain the outputs of the ML models [34]. *Figure 5* plots the SHAP values for the models with input feature sets (1)+(2) (*Figure 5a*) as well as (1)+(3) (*Figure 5b*) for XGBoost. The figures depict the rankings of the input features (the most important one is at the top) and the related impact on the prediction made. On the left side, the 20 most important features per model are listed in descending order. The SHAP values are shown on the x-axis. Each dot represents one cell in the data set, so the value of one feature for one record. A blue dot means that the record of the regarded feature has a low value, a red dot stands for a high value. The position of the dots on the x-axis explains if it contributes to the prediction negatively or positively [35].

The feature "Scheduled Departure Time" in *Figure 5a* has a positive influence on the "Arrival Delay" for high (late) departure times (red dots). As this feature is on the top of the list, it has a strong impact on the output value. This means that "Scheduled Departure Time" influences "Arrival Delay" in a way that a late departure on a day implies delays as those tend to propagate over the day. In *Figure 5b*, "Scheduled Flight Time" is another feature with a high influence. Here, a higher value for flight time (red) contributes to "Arrival Delay" in a negative way as aircraft can potentially reduce departure delays during a longer flight.

Comparing both figures, the ranking of the features differs, but in general, the features that belong to the most important features in *Figure 5a* are also included in *Figure 5b*. Schedule information is important, such as flight date details ("Day of the Week" and "Month"), information if the flight takes place on a holiday and the scheduled times. Moreover, features that describe the weather conditions at the origin and destination airports are



a) Feature sets (1)+(2)

b) Feature sets (1)+(3)

Figure 5 – SHAP values for XGBoost

included in both figures. One group of features that is not represented in the plots is information on airport identifiers, i.e. respective origin and destination airports.

Those two figures are selected to be added in the paper as the underlying models both have one short-term feature as input, whose influence is to be investigated. In *Figure 5a*, “Delay of the Previous Flight” is ranked third among the most influential features. A high value for this feature generally leads to a positive value for “Arrival Delay”. However, its influence on the output is lower (-1 to +3) compared to “Departure Delay” (-4 to +8) in *Figure 5b*. Summarised, those plots support the results given in *Table 4* and also explain why the models with input feature sets (1) + (2) are outperformed by the models with feature sets (1)+(3). “Delay of the Previous Flight” in *Figure 5a* only ranks third, whereas “Departure Delay” in *Figure 5b* ranks first and has a larger impact on the model output. This explains the good performances of models where “Departure Delay” is included as input feature.

5. CONCLUSION

This paper investigated the potential of predicting the occurrence of flight arrival delays with the use of ML methods. The relevance of this subject was shown by multiple publications in this field. However, the importance of different input features, especially short-term features, was only marginally or not at all examined to date.

Therefore, the performance of the three algorithms, Random Forest (RF), XGBoost and Neural Network (NN), was analysed together with three different sets of input features. Overall, the best results were obtained with XGBoost. Generally speaking, it is possible to predict flight arrival delays with ML algorithms. However, the performances of the trained models are highly sensitive to the included input features. This is why the influence of two short-term features on the performance of ML models was especially investigated, as their inclusion strongly impacts the potential applications.

The inclusion of “Departure Delay” leads to satisfying results with an accuracy of roughly 90%. However, the practicability is limited as a prediction at this point of time only brings slight advantages, in particular for short-distance flights. For long-distance flights there is still time for the Air Navigation Service Provider to reschedule the incoming flights at the destination airport. In order to shift the prediction timing backwards, the feature “Delay of the Previous Flight” (which is the arrival delay the aircraft had on the last flight) is proposed as an alternative input. An accuracy of 70.9% with a recall of 68.5% is achieved. This set of input features can be used to improve processes at airports and airlines to avoid upcoming delays. When forgoing the use of short-term features, both accuracy and recall decrease, but this approach allows to investigate the impact of non-short-term features on delays.

One limitation of this work is the used data basis. There are additional features that might influence delays but were not available for this paper. Those are, for example, details on the flight routes, airport characteristics or the expected demand. Delay of crew members might be important as well because this can also lead to a delay of the current flight. In our future work, we aim to broaden the data basis to further investigate the relevance of other feature groups on the achieved prediction quality. Moreover, the specific causes of delays are an aspect where extensive examinations are potentially beneficial for airlines and airports to save costs and improve the operating performance.

Summarised, the features to be considered for flight delay prediction depend on the goal of the prediction. If a forecast is to be made as early as possible (e.g. for implementing problem solving actions), it is not possible to include short-term features and hence the prediction quality decreases. For short-term predictions, the inclusion of “Departure Delay” results in a precise forecast. A compromise is obtained by adding “Delay of the Previous Flight”.

Delia SCHÖSSER, M.Sc.[1]
(korrespondierende Autorin)
E-Mail: delia.schoesser@tu-dresden.de
Jörn SCHÖNBERGER, Prof. Dr.[1]
(korrespondierender Autor)
E-Mail: joern.schoenberger@tu-dresden.de
[1] Technische Universität Dresden
Fakultät Verkehrswissenschaften "Friedrich List"
Institut für Wirtschaft und Verkehr
01062 Dresden, Deutschland

ÜBER DIE PERFORMANZ DER AUF MASCHINELLEM LERNEN BASIERENDEN FLUGVERSPÄTUNGSVORHERSAGE: UNTERSUCHUNG DES EINFLUSSES VON KURZZEITMERKMALEN

ZUSAMMENFASSUNG

Heutzutage sind Menschen und Unternehmen rund um die Welt miteinander verbunden, was zu einer *wachsenden Bedeutung der Luftfahrtindustrie führt. Da Flugverspätungen eine große Herausforderung in der Luftfahrt darstellen, können Algorithmen des maschinellen Lernens dazu verwendet werden, Verspätungen vorherzusagen. In diesem Paper wird die Vorhersage des Auftretens von Verspätungen bei der Ankunft von Flügen mit drei bekannten Algorithmen des maschinellen Lernens für einen Datensatz von Inlandsflügen in den USA untersucht. Die Aufgabe wird als Klassifikationsproblem betrachtet. Der Schwerpunkt liegt auf der Untersuchung des Einflusses von Kurzzeitmerkmalen auf die Qualität der Ergebnisse. Dazu werden drei Szenarien erstellt, die durch unterschiedliche Eingangsmerkmale gekennzeichnet sind. Bei Verzicht auf die Einbeziehung von Kurzzeitinformationen, um den Zeitpunkt der Vorhersage auf einen frühen Zeitpunkt zu verlegen, wird eine Genauigkeit von 69,5 % bei einem Recall von 68,2 % erreicht. Durch die Einbeziehung von Informationen über die Verspätung, die das Flugzeug auf seinem vorherigen Flug hatte, steigt die Vorhersagequalität leicht an. Es handelt sich dabei um einen Kompromiss zwischen dem frühen Vorhersagezeitpunkt des ersten Modells und der guten Vorhersagequalität des dritten Modells, bei dem die Abflugverspätung des Flugzeugs als Eingangsmerkmal hinzugefügt wird. In diesem Fall wird eine Genauigkeit von 89,9 % mit einem Recall von 83,4 % erreicht. Der gewünschte Zeitpunkt der Vorhersage bestimmt daher, welche Merkmale als Eingabedaten zu verwenden sind, da kurzfristige Merkmale die Vorhersagequalität erheblich verbessern.*

SCHLÜSSELWÖRTER

Flugverspätungsvorhersage; Maschinelles Lernen; Luftfahrt; Wichtigkeit von Merkmalen; Klassifikation; SHAP.

REFERENCES

[1] Awad M, Khanna R. *Efficient learning machines theories, concepts, and applications for engineers and system designers*. Berkeley, CA: Apress; 2015.

[2] Bureau of Transportation Statistics (BTS). *2019 traffic data for U.S. airlines and foreign airlines U.S. flights.* 2020. https://www.bts.dot.gov/newsroom/final-full-year-2019-traffic-data-us-airlines-and-foreign-airlines-us-flights [Accessed 21st Mar. 2022].

[3] Bureau of Transportation Statistics (BTS). *Airline on-time performance and causes of flight delays.* 2021. https://www.bts.gov/topics/airlines-and-airports/airline-time-performance-and-causes-flight-delays [Accessed 21st Mar. 2022].

[4] Federal Aviation Administration (FAA). *Air traffic by the numbers.* 2020. https://www.faa.gov/air_traffic/by_the_numbers/media/Air_Traffic_by_the_Numbers_2020.pdf [Accessed 21st Mar. 2022].

[5] Jacquillat A, Odoni AR. A roadmap toward airport demand and capacity management. *Transportation Research Part A: Policy and Practice.* 2018;114: 168-185. doi: 10.1016/j.tra.2017.09.027.

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

[7] Ding Y. Predicting flight delay based on multiple linear regression. In: Jia XL, Zhou SQ, Patty AA (eds.) *IOP Conference Series: Earth and Environmental Science, Volume 81, 2nd International Conference on Materials Science, Energy Technology and Environmental Engineering (MSETEE 2017), 28–30 Apr. 2017, Zhuhai, China*. IOP Publishing; 2017. 012198.

[8] Yazdi MF, Kamel SR, Chabok SJM, Kheirabadi M. Flight delay prediction based on deep learning and Levenberg-Marquart algorithm. *Journal of Big Data*. 2020;7(106): 1-28. doi: 10.1186/s40537-020-00380-z.

[9] Huo J, et al. The prediction of flight delay: Big data-driven machine learning approach. *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 14–17 Dec*. 2020. IEEE; 2020. p. 190-194.

[10] Gui G, et al. Flight delay prediction based on aviation big data and machine learning. *IEEE Transactions on Vehicular Technology*. 2020;69(1): 140-150. doi: 10.1109/tvt.2019.2954094.

[11] Kalyani NL, et al. Machine learning model - based prediction of flight delay. *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 7-9 Oct. 2020.* IEEE; 2020. p. 577-581.

[12] Manna S, et al. A statistical approach to predict flight delay using gradient boosted decision tree. *2017 International Conference on Computational Intelligence in Data Science (ICCIDS), 2-3 June 2017, Tamilnadu, India*. IEEE; 2017. p. 1-5.

[13] US Department of Transportation (US DOT). *2015 flight delays and cancellations*. 2017. https://www.kaggle.com/usdot/flight-delays [Accessed 21st Mar. 2022].

[14] Marsland S. *Machine learning - An algorithmic perspective*. New York: CRC Press; 2015.

[15] Burnett RA, Si D. Prediction of injuries and fatalities in aviation accidents through machine learning. ICCDA '17: *Proceedings of the International Conference on Compute and Data Analysis, 19-23 May 2017, Lakeland, USA*. New York: ACM Press; 2017. p. 60-68.

[16] Horiguchi Y, et al. Predicting fuel consumption and flight delays for low-cost airlines. *AAAI'17: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 4-9 Feb. 2017, San Francisco, USA.* AAAI Press; 2017. p. 4686–4693.

[17] Jan SS, Chen YT. Development of a new airport unusual-weather detection system with aircraft surveillance information. *IEEE Sensors Journal*. 2019;19(20): 9543-9551. doi: 10.1109/jsen.2019.2926391.

[18] Yablonsky G, et al. Flight delay performance at Hartsfield-Jackson Atlanta International Airport. *Journal of Airline and Airport Management*. 2014;4(1): 78-95. doi: 10.3926/jairm.22.

[19] Xu N, Sherry L, Laskey KB. Multifactor model for predicting delays at U.S. Airports. *Transportation Research Record: Journal of the Transportation Research Board.*

2008;2052(1): 1-15. doi: 10.3141/2052-08.

[20] National Oceanic and Atmospheric Administration (NOAA). *data/ global-hourly/ archive/ csv.* 2019. https://www.ncei.noaa.gov/data/global-hourly/archive/csv/ [Accessed 21st Mar. 2022].

[21] NOAA SciJinks. *How reliable are weather forecasts?* https://scijinks.gov/forecast-reliability/ [Accessed 21st Mar. 2022].

[22] Federal Aviation Administration (FAA). *Core 30.* https://aspm.faa.gov/aspmhelp/index/Core_30.html [Accessed 21st Mar. 2022].

[23] Alpaydin E. *Introduction to machine learning*. Cambridge: MIT Press; 2020.

[24] Russell SJ, Norvig P. *Artificial intelligence - A modern approach.* London: Prentice Hall; 2010.

[25] Pedregosa F, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research.* 2011; 12: 2825-2830. https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf [Accessed 21st Mar. 2022].

[26] Chen T, Guestrin C. XGBoost: A scalable tree boosting system. *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data, Mining 13-17 Aug. 2016, San Francisco, USA*. New York: ACM Press; 2016. p. 785-794.

[27] Chollet F. *Keras*. https://keras.io [Accessed 21st Mar. 2022].

[28] Abadi M, et al. *TensorFlow: Large-scale machine learning on heterogeneous systems.* https://www.tensorflow.org/ [Accessed 21st Mar. 2022].

[29] Kubat M. *An introduction to machine learning*. Cham: Springer Nature; 2021.

[30] Dembczynski K, et al. Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. *PMLR Proceedings of the 30th International Conference on Machine Learning, Atlanta, USA*. 2013. p. 1130-1138.

[31] Esmaeilzadeh E, Mokhtarimousavi S. Machine learning approach for flight departure delay prediction and analysis. *Transportation Research Record: Journal of the Transportation Research Board.* 2020;2674(8): 145-159. doi: 10.1177/0361198120930014.

[32] Claesen M, et al. Hyperparameter tuning in Python using Optunity. *International Workshop on Technical Computing for Machine Learning and Mathematical Engineering (TCMM 2014), Leuven, Belgium*. 2014. p. 1-2.

[33] Freitas D, Guerreiro Lopes L, Morgado-Dias F. Particle swarm optimisation: A historical review up to the current developments. *Entropy*. 2020;22(3): 1-36. doi: 10.3390/e22030362.

[34] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. *NIPS'17*: *Proceedings of the 31st International Conference on Neural Information Processing Systems, 4-9 Dec. 2017, Long Beach, USA*. Red Hook: Curran Associates Inc.; 2017. p. 4765-4774.

[35] Gianfagna L, Di Cecco A. *Explainable AI with Python*. Cham: Springer International Publishing; 2021.