SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ DERGİSİ

Sakarya University Journal of Science
SAUJS

e-ISSN 2147-835X | Period Bimonthly | Founded: 1997 | Publisher Sakarya University |
http://www.saujs.sakarya.edu.tr/en/

Title: Supervised Learning Approaches to Flight Delay Prediction

Authors: Mehmet Cemal ATLIOĞLU, Mustafa BOLAT, Murat ŞAHİN, Volkan TUNALI, Deniz KILINÇ
Recieved: 2020-03-27 16:00:07

Accepted: 2020-09-11 20:02:56

Article Type: Research Article

Volume: 24
Issue: 6
Month: December
Year: 2020
Pages: 1223-1231

How to cite
Mehmet Cemal ATLIOĞLU, Mustafa BOLAT, Murat ŞAHİN, Volkan TUNALI, Deniz KILINÇ; (2020), Supervised Learning Approaches to Flight Delay Prediction. Sakarya University Journal of Science, 24(6), 1223-1231, DOI: https://doi.org/10.16984/saufenbilder.710107
Access link
http://www.saujs.sakarya.edu.tr/en/pub/issue/57766/710107

Sakarya University Journal of Science 24(6), 1223-1231, 2020

Supervised Learning Approaches to Flight Delay Prediction

Mehmet Cemal ATLIOĞLU[1], Mustafa BOLAT[1] , Murat ŞAHİN[2], Volkan TUNALI[*3], Deniz KILINÇ[4]

Abstract

Delays in flights and other airline operations have significant consequences in quality of service, operational costs, and customer satisfaction. Therefore, it is important to predict the occurrence of delays and take necessary actions accordingly. In this study, we addressed the flight delay prediction problem from a supervised machine learning perspective. Using a real-world airline operations dataset provided by a leading airline company, we identified optimum dataset features for optimum prediction accuracy. In addition, we trained and tested 11 machine learning models on the datasets that we created from the original dataset via feature selection and transformation. CART and KNN showed consistently good performance in almost all cases achieving 0.816 and 0.807 F-Scores respectively. Similarly, GBM, XGB, and LGBM showed very good performance in most of the cases, achieving F-Scores around 0.810.

Keywords: air transportation, flight delay prediction, machine learning, data science

[1] Tav Technology, İstanbul, E-Mail: MehmetCemal.Atlioglu@tav.aero
E-Mail: Mustafa.Bolat@tav.aero ORCID: https://orcid.org/0000-0003-1289-2715
ORCID: https://orcid.org/0000-0001-8169-0629
[2] Manisa Celal Bayar University, Faculty of Technology, E-Mail: muratpq@gmail.com
ORCID: https://orcid.org/0000-0002-2866-8796
[*] Corresponding Author: volkan.tunali@gmail.com
[3] Maltepe University, Faculty of Engineering and Natural Sciences
ORCID: https://orcid.org/0000-0002-2735-7996
[4]İzmir Bakırçay University, Faculty of Engineering and Architecture,
E-Mail: drdenizkilinc@gmail.com
ORCID: https://orcid.org/0000-0002-2336-8831

1. INTRODUCTION

Airline operations are too complex to manage with complete accuracy. There are a lot of resources and constraints to be synchronized to have an adequate accuracy of timing. Trying to increase the synchronization accuracy of airline operations is a major field of study and practice in the business of Airline Management. One of the alternative approaches to gaining accuracy is to predict some anomalies and seemingly unexpected delays in operations in advance and to react and adapt accordingly [1]. Therefore, predicting the delays in well-defined flight milestone points can provide improvements in the overall predictability of the airline operations and may result in considerable level of cost savings and increase in passenger satisfaction [2].

In this study, we addressed the flight delay prediction problem from a supervised machine learning perspective. Using a real-world airline operations dataset provided by a leading airline company, we identified optimum dataset features that improve prediction accuracy. Moreover, we trained and tested a lot of machine learning models such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (CART), and Gaussian Naïve Bayes (GNB) on the datasets that we created from the original dataset via feature selection and transformation.

This paper is organized as follows: Section 2 presents some timely work on flight delay prediction in the literature. Section 3 explains the details of the datasets and machine learning models that we experimented with. In Section 4, we present our empirical findings and discuss them. Finally, Section 5 concludes the study and provides some future directions for further research.

2. RELATED WORK

Due to increasing demand to airline transportation, flight delay prediction is an important problem that researchers have been actively investigating. In this section, we present some notable and recent work on the topic. Most of these studies rely on data mining and machine learning techniques because large quantities of operational data regarding aviation operations are now possible to collect, store, and process. Besides, applications based on machine learning techniques have already achieved promising performances in very diverse domains like computer vision, natural language processing, medical diagnosis, fraud detection, and so on. Therefore, it is perfectly normal to see many studies that try to utilize machine learning for flight delay prediction.

In a study by Ding, flight delay prediction was considered as a regression problem, and a solution based on multiple linear regression was proposed and compared with Naïve Bayes and C4.5 [3].

A Gradient Boosting classification model was created using hyper-parameter optimization by Chakrabarty in [4]. In addition, to overcome the imbalance problem of the training data, over-sampling via Randomized SMOTE technique was also employed in the study.

In [5] by Yu et al., key factors causing the delays were analyzed, and a prediction model that used a deep belief network along with a support vector regression was proposed for optimal prediction accuracy.

Khaksar and Sheikholeslami investigated the factors affecting the flight delay prediction using several datasets and machine learning techniques like Bayesian modeling, decision tree, random forest, and so on [6].

In a very recent study by Gui et al., flight delay prediction was also studied from big data and machine learning perspective [7]. In the study, several machine learning models were examined and it was experimentally shown that random forest-based model achieved very high prediction accuracy without overfitting.

When we examine the previous studies mentioned above, we observe that the datasets used and the methodologies applied were highly similar to those in our study. All of them used the flight datasets of from the commercial airline

companies of their respective countries, and these datasets had very similar features. In some studies, however, the datasets were enriched with new features like weather conditions, fleet age, and so on. In our study, we did not apply such kind of feature enrichment, instead, we only used the data as obtained from the airline operations. All studies consistently reported that their datasets were imbalanced in terms of class distribution, and some of them applied several dataset balancing techniques. In our study, we did not make use of any kind of balancing. While some of the studies approached the problem as a binary classification problem (i.e., delay or no-delay), some used several delay classes like delays of 0-15 (minutes), 15-30, 30-60, and so on, making them a multi-class classification problem. In our study, however, we addressed the flight delay problem as a binary classification problem. The previous studies usually trained and tested a relatively small number of classification algorithms and they reported classification accuracy around 70-90%. Different from them, we trained and tested a large number of classification models on several datasets with hand-picked features, we obtained accuracy around 82%.

3. MATERIALS AND METHOD

In this section, we give the details of the dataset used and the method applied in this study.

3.1. Dataset

The dataset used in this study was provided by a leading airline company in Turkey. It contains 8,086 lines of data records that belonged to operations of the Dammam King Fahd International Airport, Saudi Arabia, during the three-year period from January 1st, 2017 to December 9th, 2019. Detailed information about the dataset structure is given in Table 1.

Table 1
Structure of the original dataset

Column name	Description	Number of distinct values
line_no	Unique row ID	NA
origin	IATA code of the airport from which the incoming aircraft departed	12
airline_icao	Airline carrier ICAO code of the incoming aircraft	1
etad_stad	Difference between Estimated Time and Scheduled Time in minutes	NA
flight_category	International vs. Domestic	2
ac_subtype	Aircraft subtype	5
terminal_resource	International vs. Domestic	2
pax_count	Number of passengers	NA
gate_resource_id	Gate ID used by incoming passengers	19
stand_resource_id	Area ID where the aircraft parked	32
etad	Estimated time of arrival/departure	NA
stad	Scheduled time of arrival/departure	NA
tobt	Target Off Block Time	NA
aobt	Actual On Block Time	NA
atad	Actual time of arrival/departure	NA

3.2. Data Preprocessing

In this study, we considered the flight delay prediction problem as a binary classification problem such that if *etad_stad* (difference between Estimated Time and Scheduled Time in minutes) was greater than 15 minutes, we considered it a delay, and no-delay otherwise. Therefore, we generated a class variable with two values as delay and no-delay. This 15-minute period was decided after consultations with domain experts from the airline company. After the class variable generation, the dataset had an imbalanced class distribution of 1,174 delay and 6,912 no-delay records.

As seen in Table 1, the data columns *origin*, *flight_category*, *ac_subtype*, *terminal_resource*, *gate_resource_id*, and *stand_resource_id* are categorical variables with a number of distinct values. Because there was no ordinal relationship between the values of these variables, we treated them as nominal variables during experiment datasets creation. Applying one-hot-encoding

technique to each categorical variable, we created a large number of binary features.

The variables other than the categorical ones are simply ignored in the experiment datasets creation phase since they did not present any useful information for delay prediction. We manually combined the categorical variables in different ways to create six different experiment datasets. Besides, we augmented the experiment datasets with the *year* of *etad* value as a numerical variable by simply adding this new variable to each dataset. Details of the experiment datasets are given in Table 2.

Table 2
Details of the experiment datasets

	Experiment Dataset ID					
Variable used	**1**	**2**	**3**	**4**	**5**	**6**
Origin	✓	✓	✓	✓	✓	✓
ac_subtype	✓	✓		✓	✓	✓
gate_resource_id	✓	✓	✓		✓	
stand_resource_id	✓	✓	✓	✓		
Year	✓		✓	✓	✓	✓
flight_category	✓	✓	✓	✓	✓	✓
Number of features	71	70	66	52	39	20

3.3. Supervised Learning Models

Machine Learning is a subfield of Artificial Intelligence dealing with algorithms that learn through experience that is supplied in the form of past data [8]. Machine Learning algorithms are divided into two well-established categories depending on their approach and the type of the problem they are expected to solve: supervised learning and unsupervised learning [9]. In supervised learning, the data contains a class (or target) variable, and the task is to predict the class value upon learning the correlations between the class labels and the other variables from the data via a process called training. Regression and classification are the main types of supervised learning algorithms. In unsupervised learning, on the other hand, the algorithm learns the patterns and structures like clusters in the data without being supplied with any class labels.

In this study, our main objective was to predict flight delays using past labeled data. Therefore, we employed a supervised learning approach to the problem. Rather than predicting the flight delay time in minutes and thus considering the flight delay prediction problem as a regression problem, we approached it as a classification problem where we tried to predict whether a delay would occur or not with respect to some flight information given. Classification is the task of predicting the class of a data instance whose class is previously unknown using a model trained with data instances with previously known class labels [9].

There are a lot of classification algorithms in the literature. In this study, we trained 11 different classification models for each experiment dataset using 11 different classification algorithms, namely, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (CART), Gaussian Naïve Bayes (GNB), Logistic Regression (LR), Multilayer Perceptron (MLP), Random Forest (RF), Gradient Boosting (GBM), XGBoost (XGB), CatBoost (CB), and LightGBM (LGBM) [9-14]. First five of these algorithms are called base classifiers because only one classifier instance is trained for each one. Multilayer Perceptron, also a base classifier, is also called Artificial Neural Network, where there is a series of interconnected neuronal layers. The rest of the algorithms are called ensemble classifiers because more than one instance of base classifiers are trained and their collective decision is reported as the final prediction [9].

3.4. Experimental Setup

Throughout this study, we used Python 3.7 for all data preprocessing, model training, and model evaluation tasks because Python is a very powerful programming language with extensive data manipulation capabilities with pandas package (version 1.0.3) [15] and machine learning capabilities with scikit-learn package (version 0.22.2) [16]. In our experiments, we ran all classifiers with their default parameters in their scikit-learn implementations. The most important of these default parameters are presented in Table 3. In addition, we applied 10-fold cross validation to get more reliable model

performance prediction since our datasets were imbalanced and number of training and testing data instances were quite small.

Table 3
Default parameter values of classifiers

Algorithm	Default parameter values
KNN	n_neighbors:5, metric:'minkowski'
SVM	regularization:1.0, kernel:'rbf', degree:3, gamma:'scale'
CART	criterion:'gini', splitter:'best', max_depth:None,
GNB	priors:None, var_smoothing:1e-09
LR	regularization:1.0, class_weight:None, fit_intercept:True, intercept_scaling:1, l1_ratio:None, max_iter:100, solver:'liblinear'
MLP	hidden_layer_sizes:(100,) activation:'relu', solver:'adam', alpha:0.0001
RF	n_estimators:100, criterion:'gini', max_depth:None, min_samples_split:2, min_samples_leaf:1
GBM	loss:'deviance', learning_rate:0.1, n_estimators:100, subsample:1.0, criterion:'friedman_mse', min_samples_split:2, min_samples_leaf:1
XGB	max_depth:3, learning_rate:0.1, n_estimators:100
CB	iterations:None, learning_rate:None, depth:None, min_child_samples:None, max_leaves:None, num_leaves:None, max_depth:None, n_estimators:None
LGBM	boosting_type:'gbdt', num_leaves:31, max_depth:-1, learning_rate:0.1, n_estimators:100

3.5. Model Evaluation Metrics

In order to measure the classification performance of the selected algorithms, we used Precision, Recall, F-Score, and ROC Area metrics. Since our dataset was imbalanced in terms of class distribution, these were very reliable metrics to predict the model performance in a real-world scenario. For each of these metrics, the higher the metric value, the higher the performance of a classifier is.

When we test a binary classifier, we obtain four different counts as True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). Using these counts, it is possible to compute the above metrics as given in Equations 1, 2, 3, and 4, respectively.

$$Precision = \frac{TP}{TP+F} \quad (1)$$

$$Recall = \frac{TP}{TP+} \quad (2)$$

$$F\text{-}Score = \frac{2\times Precision\times Recall}{Precision+Rec} \quad (3)$$

$$ROC\ Area = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) \quad (4)$$

We used the related metric functions of the scikit-learn package with the *average* parameter set to 'weighted' in order to take into consideration the imbalanced nature of the class labels. With this option, respective metrics are calculated for each class independently, and then their average weighted by the number of true instances for each class is calculated.

For example, for a binary classification with TP = 1744, FP = 278, FN = 0, and TN = 0, precision (P) is calculated as follows. First, precision for each class label is calculated independently as P_{pos} = 1744 / 2022 = 0.862, and P_{neg} = 278 / 2022 = 0.137. Then, the weighted average of these values is calculated using the number of true instances for each class label as P = (0.862 × 1744 + 0.137 × 0) / 2022, and it is found 0.743.

As a result of this parameter decision for imbalanced datasets, for instance, calculated F-Score values may not fall into between related Precision and Recall values. For the very same reason, calculated ROC Area values may seem rather low when Precision and Recall values are considered. For the previous example, ROC Area is calculated as 0.5 × (1744 / 1744 + 0 / 278) = 0.5, which is much lower than the calculated Precision and Recall values.

4. FINDINGS AND DISCUSSION

We trained 11 machine learning models and tested them on six experiment datasets using 10-fold cross validation. We present model scores for each dataset in Tables 4 to 9, respectively. In

the tables, highest values for each score are emphasized with boldface font.

Table 4
Experiment results for Experiment Dataset 1

Algorithm	Precision	Recall	F-Score	ROC Area
KNN	0.807	0.861	0.807	0.514
SVM	0.744	0.860	0.797	0.498
CART	0.814	0.861	**0.816**	**0.532**
GNB	**0.882**	0.143	0.044	0.503
LR	0.744	**0.863**	0.799	0.500
MLP	0.744	**0.863**	0.799	0.500
RF	0.744	**0.863**	0.799	0.500
GBM	0.814	0.862	0.812	0.524
XGB	0.817	**0.863**	0.812	0.523
CB	0.744	**0.863**	0.799	0.500
LGBM	0.817	**0.863**	0.811	0.521

Table 5
Experiment results for Experiment Dataset 2

Algorithm	Precision	Recall	F-Score	ROC Area
KNN	0.815	**0.863**	0.805	0.511
SVM	0.744	0.860	0.797	0.498
CART	0.810	0.862	0.805	0.510
GNB	**0.882**	0.143	0.044	0.503
LR	0.836	**0.863**	0.801	0.503
MLP	0.744	**0.863**	0.799	0.500
RF	0.744	**0.863**	0.799	0.500
GBM	0.744	0.862	0.799	0.500
XGB	0.810	0.862	0.805	0.510
CB	0.744	**0.863**	0.799	0.500
LGBM	0.820	**0.863**	**0.807**	**0.514**

Table 6
Experiment results for Experiment Dataset 3

Algorithm	Precision	Recall	F-Score	ROC Area
KNN	0.807	0.861	0.807	0.514
SVM	0.744	0.860	0.797	0.498
CART	0.814	0.861	**0.816**	**0.532**
GNB	**0.831**	0.145	0.049	0.503
LR	0.744	0.863	0.799	0.500
MLP	0.019	0.137	0.033	0.500
RF	0.744	0.863	0.799	0.500
GBM	0.810	0.861	0.811	0.523
XGB	0.820	0.863	0.812	0.523
CB	0.744	0.863	0.799	0.500
LGBM	0.830	**0.864**	0.809	0.516

Table 7
Experiment results for Experiment Dataset 4

Algorithm	Precision	Recall	F-Score	ROC Area
KNN	0.799	0.860	**0.805**	**0.511**
SVM	0.744	**0.863**	0.799	0.500
CART	0.744	**0.863**	0.799	0.500
GNB	**0.882**	0.143	0.044	0.503
LR	0.744	**0.863**	0.799	0.500
MLP	0.744	**0.863**	0.799	0.500
RF	0.744	**0.863**	0.799	0.500
GBM	0.744	0.861	0.798	0.499
XGB	0.836	**0.863**	0.801	0.503
CB	0.744	**0.863**	0.799	0.500
LGBM	0.836	**0.863**	0.801	0.503

Table 8
Experiment results for Experiment Dataset 5

Algorithm	Precision	Recall	F-Score	ROC Area
KNN	0.799	0.862	0.800	0.503
SVM	0.744	0.861	0.798	0.499
CART	0.779	0.860	0.800	0.503
GNB	**0.882**	0.142	0.043	0.503
LR	0.744	**0.863**	0.799	0.500
MLP	0.744	**0.863**	0.799	0.500
RF	0.744	**0.863**	0.799	0.500
GBM	0.767	0.861	0.799	0.500
XGB	0.804	0.862	**0.801**	**0.504**
CB	0.744	**0.863**	0.799	0.500
LGBM	0.804	0.862	**0.801**	**0.504**

Table 9
Experiment results for Experiment Dataset 6

Algorithm	Precision	Recall	F-Score	ROC Area
KNN	0.744	**0.863**	**0.799**	0.500
SVM	0.744	**0.863**	**0.799**	0.500
CART	0.744	**0.863**	**0.799**	0.500
GNB	**0.882**	0.142	0.043	**0.503**
LR	0.744	**0.863**	**0.799**	0.500
MLP	0.744	**0.863**	**0.799**	0.500
RF	0.744	**0.863**	**0.799**	0.500
GBM	0.744	**0.863**	**0.799**	0.500
XGB	0.744	**0.863**	**0.799**	0.500
CB	0.744	**0.863**	**0.799**	0.500
LGBM	0.744	**0.863**	**0.799**	0.500

4.1. Evaluation of Datasets

High F-Score and ROC Area together are good indicators of a good prediction model. When we examine the results in the tables, we see consistently higher F-Score and ROC Area scores in Tables 4 and 6 for Experiments Datasets 1 and 3, respectively. Experiment Dataset 1 is the dataset where all variables are used as the features (71 features in total). Experiment Dataset 3, on the other hand, is the one with features without *ac_subtype* variable (66 features in total). Almost all prediction algorithms performed similarly on these two datasets. Based on this finding, we can infer that *ac_subtype* (aircraft subtype) does not contribute

much to the predictive ability of the machine learning algorithms. As a result, it can be excluded from real-world model creation.

We see that model scores for Experiment Datasets 4, 5, and 6 are consistently very low when compared to the scores of the other datasets. The common factor for these three datasets is the absence of either *gate_resource_id* or *stand_resource_id* variable, or both. Therefore, we can infer that these two variables play highly important role for flight delay prediction, although their absence resulted in less number of features, which was good for model training and testing.

The model score results in Table 9 show such an interesting pattern that all algorithms other than GNB performed exactly the same in terms of all metrics. This also suggests that the absence of both *gate_resource_id* and *stand_resource_id* variables created machine learning models with no predictive power at all. ROC Area score with 0.5 value also supports this finding. The same pattern is also observed in Tables 7 and 8 to some degree, where either *gate_resource_id* or *stand_resource_id* variable was absent in the respective experiment datasets.

4.2. Evaluation of Algorithms

We have an important observation that GNB algorithm performed very poorly for all experiment datasets. Although it presented higher precision than all other competing algorithms, its recall values were very low, which also resulted in very low F-Score values.

According to the results in Tables 4, 5, and 6, the decision tree (CART) algorithm performed well, reaching almost the highest F-Score and ROC Area scores.

From our previous experience, we would expect that RF algorithm, which is an ensemble learning algorithm, would perform well. However, its performance was almost always very low. On the other hand, we see that other ensemble learning algorithms GBM, XGB, and LGBM showed very good performance in most of the cases. CB, which is also an ensemble classifier, however, performed very poorly in all cases.

Besides CART algorithm, KNN also showed consistently good performance in almost all cases. Although it is the simplest learning algorithm and no learning model is actually created at all due to its lazy nature, it performed surprisingly well in flight delay prediction.

Finally, the other algorithms, which are very well-known for their good performances in classification tasks, failed drastically in flight delay prediction in all cases. That is, SVM, LR, and MLP algorithms did not exhibit any good performance as opposed to their good reputations.

5. CONCLUSION AND FUTURE DIRECTIONS

Delays in flights and other airline operations have significant consequences in quality of service, operational costs, and customer satisfaction. Therefore, it is important to predict the occurrence of delays and take necessary actions accordingly. In this study, we considered the flight delay prediction problem as a machine learning problem. More specifically, we offered a solution to the problem by transforming it to classification problem.

Using the operational data from a three-year period made available by an airline company, we generated six different experiment datasets by applying several preprocessing techniques like feature selection and data transformation. Once we obtained these datasets, we trained and tested 11 different machine learning models on each one. Whereas some machine learning models showed promising performance in flight delay prediction, some failed very badly. For example, CART (decision tree) and KNN algorithms showed consistently good performance in almost all cases achieving 0.816 and 0.807 F-Scores respectively. Similarly, GBM, XGB, and LGBM algorithms showed very good performance in most of the cases, achieving F-Scores around 0.810. GNB, RF, CB, SVM, LR, and MLP

algorithms, on the other hand, did not show acceptable performance.

Performances of the machine learning models were not independent of the datasets on which they were trained. The datasets that the machine learning algorithms showed good performance were the ones that contained the features *gate_resource_id* and *stand_resource_id.* These two variables were the ones with the most predictive power in flight delay prediction. Therefore, these variables should definitely be included in the production model creation. Nevertheless, *ac_subtype* variable did not contribute much to the predictive ability of the machine learning algorithms, and it can be excluded in production.

When we examine the previous studies, we see that they all used the flight datasets of from the commercial airline companies of their respective countries. Similarly, in this study, we introduced a new, genuine, and recent flight dataset from a Turkish airline company. Moreover, the previous studies usually trained and tested a relatively small number of classification algorithms on their datasets. We, on the other hand, trained and tested a quite large number of classification models on several datasets with hand-picked and engineered features.

The dataset used in this study was imbalanced in terms of delay and no-delay class distributions. This could be the main reason why some machine learning algorithms did very poorly while some did well. As a future work, we plan to repeat the experiments on a dataset that we made balanced using several techniques like over-sampling, SMOTE, and so on. Furthermore, we also consider that the machine learning algorithms that are based on decision trees may perform better when the categorical data are not transformed into binary features.

Funding

This work was partially supported by the Research and Development Center of TAV Airports Holding accredited on Turkey-Ministry of Science.

The Declaration of Conflict of Interest/ Common Interest

No conflict of interest or common interest has been declared by the authors.

Authors' Contribution

M.C.A: Investigation, experimental design.

M.B: Experimental design, investigation, data analysis.

M.Ş: Experimental design, investigation, data analysis.

V.T: Review, writing, editing, investigation, supervision.

D.K: Review, writing, investigation.

The Declaration of Ethics Committee Approval

The authors declare that this document does not require an ethics committee approval or any special permission.

The Declaration of Research and Publication Ethics

The authors of the paper declare that they comply with the scientific, ethical and quotation rules of SAUJS in all processes of the paper and that they do not make any falsification on the data collected. In addition, they declare that Sakarya University Journal of Science and its editorial board have no responsibility for any ethical violations that may be encountered, and that this study has not been evaluated in any academic publication environment other than Sakarya University Journal of Science.

REFERENCES

[1] N. Pyrgiotis, K. M. Malone, and A. Odoni, "Modelling delay propagation within an airport network," Transportation Research Part C: Emerging Technologies, vol. 27, pp. 60-75, 2013.

[2] J. J. Rebollo and H. Balakrishnan, "Characterization and prediction of air traffic delays," Transportation Research Part C: Emerging Technologies, vol. 44, pp. 231-241, 2014.

[3] Y. Ding, "Predicting flight delay based on multiple linear regression," in 2nd International Conference on Materials Science, Energy Technology and Environmental Engineering (MSETEE 2017), Zhuhai, China, pp. 1-8, 2017.

[4] N. Chakrabarty, "A Data Mining Approach to Flight Arrival Delay Prediction for American Airlines," CoRR, vol. abs/1903.06740, 2019.

[5] B. Yu, Z. Guo, S. Asian, H. Wang, and G. Chen, "Flight delay prediction for commercial air transport: A deep learning approach," Transportation Research Part E: Logistics and Transportation Review, vol. 125, pp. 203-221, 2019.

[6] H. Khaksar and A. Sheikholeslami, "Airline delay prediction by machine learning algorithms," Scientia Iranica, vol. 26, pp. 2689-2702, 2017.

[7] G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou, and D. Zhao, "Flight Delay Prediction Based on Aviation Big Data and Machine Learning," IEEE Transactions on Vehicular Technology, vol. 69, pp. 140-150, 2020.

[8] E. Alpaydın, Introduction to Machine Learning, 3rd ed. London, England: The MIT Press, 2014.

[9] J. Han and M. Kamber, Data Mining: Concepts and Techniques. USA: Morgan Kaufmann Publishers, 2006.

[10] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in Advances in kernel methods, S. Bernhard, J. C. B. Christopher, and J. S. Alexander, Eds., ed: MIT Press, pp. 185-208, 1999.

[11] L. Breiman, "Random Forests," Machine Learning, vol. 45, pp. 5-32, 2001.

[12] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," presented at the Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, 2016.

[13] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: a highly efficient gradient boosting decision tree," presented at the Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA, 2017.

[14] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: gradient boosting with categorical features support," CoRR, vol. abs/1810.11363, 2018.

[15] W. McKinney, "pandas: a foundational Python library for data analysis and statistics," Python for High Performance and Scientific Computing, vol. 14, 2011.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.