Building a Machine Learning Model integrating In-depth Analysis and Sampling Approaches to predict Flight Delay

Huynh Hue Truc[1,2], Le Nguyen Thanh Ty[1,2], Ho Song Tin[1,2], Nguyen Le Phuong Anh[1,2], Do Thanh Danh[1,2], Duy Thanh Tran[1,2]

[1]University of Economics and Law, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
First Author Email: truchh22411c@st.uel.edu.vn
Corresponding Author Email: thanhtd@uel.edu.vn

Abstract: In the rapidly evolving landscape of air transportation globally, the demand for flight tickets is likely to surge over the next decade, reinforcing the pivotal role of air travel in modern transportation. Consequently, there is an urgent need for advanced flight delay prediction methods to improve the reliability and accuracy of flight planning and operations. Acknowledging how imbalanced data can pose a significant obstacle to the efficacy of machine learning models' flight delay prediction, numerous studies have implemented data sampling methods in predicting flight delays. Nevertheless, there is a discernible gap in the current body of research, as comprehensive comparisons among various pairs of machine learning and sampling methods have yet to be undertaken. This work contributes to the realm of worldwide delay prediction by performing an extensive comparative analysis of five machine-learning methodologies and three sampling techniques. We aim to find effective strategies for identifying delayed flights, understand feature importance and thereby ultimately minimize the costs linked to flight delays. Overall, we seek to contribute not only to the field of machine learning in aviation but also to the broader goal of fostering a more seamless and reliable global air transportation network. Our results show three different models with over 80% of precision for both classes, proving the ability to accurately predict both positive and negative cases.

Keywords: *Flight Delay Prediction, Machine Learning, Imbalanced Data, Data Sampling, Comparative Analysis, Air Transportation.*

1 INTRODUCTION

1.1 Context and motivation

Rapid growth in the aviation industry is a result of the increasing demand for air transportation. In the aviation sector, passenger and cargo demands are annually increasing at an average rate of 7% and 4.43%, respectively (IATA, 2019). Over the next 20 years from 2019 to 2039, the Federal Aviation Administration (FAA) anticipates a nearly 35% increase in flight operations, with an average annual growth rate of around 1.5% for all commercial airlines (Federal Aviation Administration, 2019). These figures indicate our increasing dependence on air transportation. However, amid this growth, the issue of flight delays has become a significant focal point. Before the COVID-19 crisis, the continuous growth of air traffic led to challenging scheduling situations and an increase in flight delays: In 2018, Europe saw more than 11 million flights with an average delay of 14.7 minutes, marking a 3.8% and 17% increase from 2017, respectively (Network manager annual report, 2018; Network operations report 2018, 2018). For the full year 2018, reporting carriers posted an on-time arrival rate of 79.4 percent, down from 80.2 percent in 2017 (Fig. 1). After the crisis, it is expected that air traffic volume will return to pre-crisis levels within 5 years (Eurocontrol five-year forecast 2020-2024, 2020). An increase in flight delays adversely impacts the quality of service and revenue for both airlines and airports (Marco Alderighi and Alberto A Gaggero, 2018) (Fig. 2, Fig. 3).

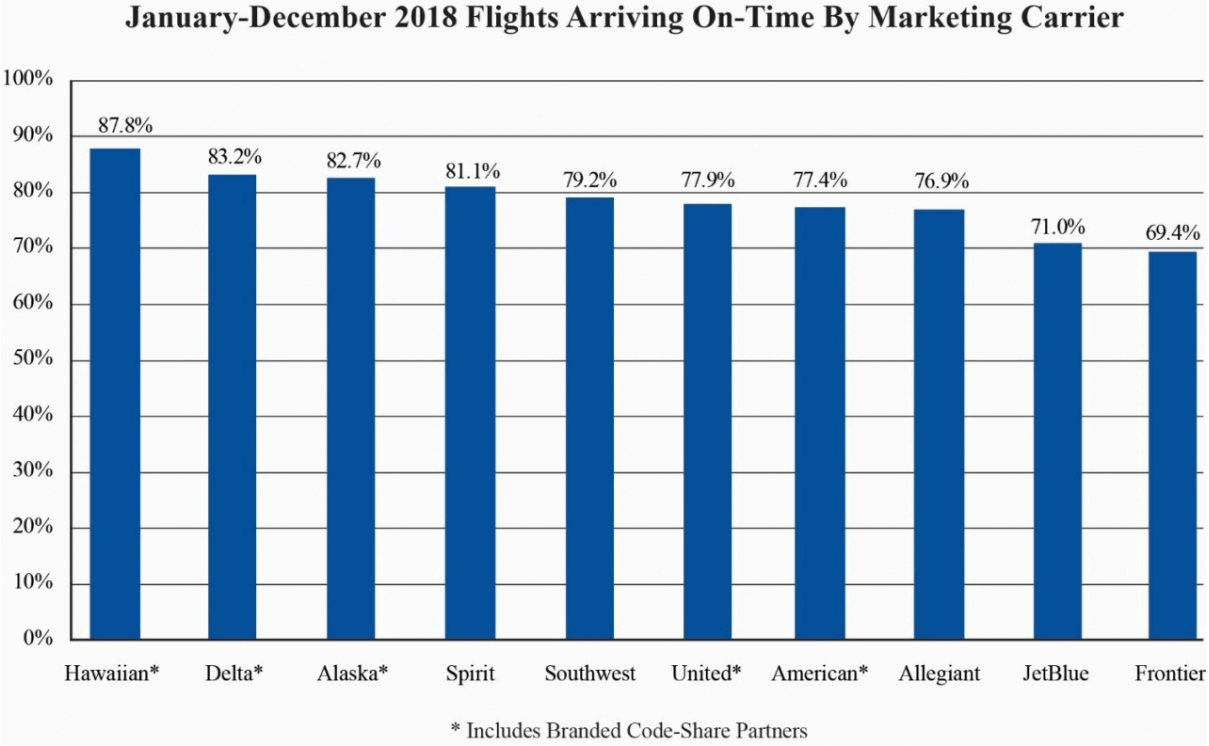

Fig. 1 - Air Travel Consumer Report: Full Year 2018. Source: Bureau of Transportation Statistics

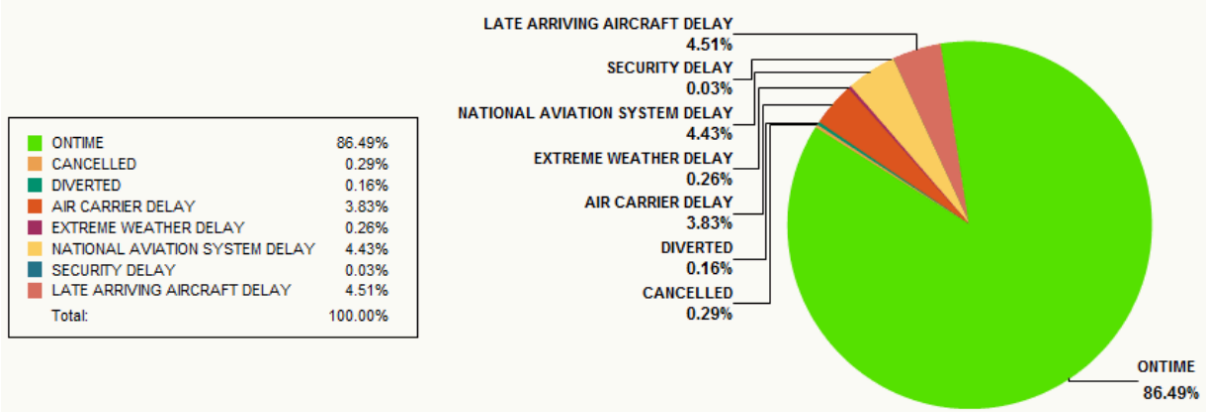

Fig. 2 - Overall causes of delay by reporting carrier November 2016. Source: Bureau of Transportation Statistics

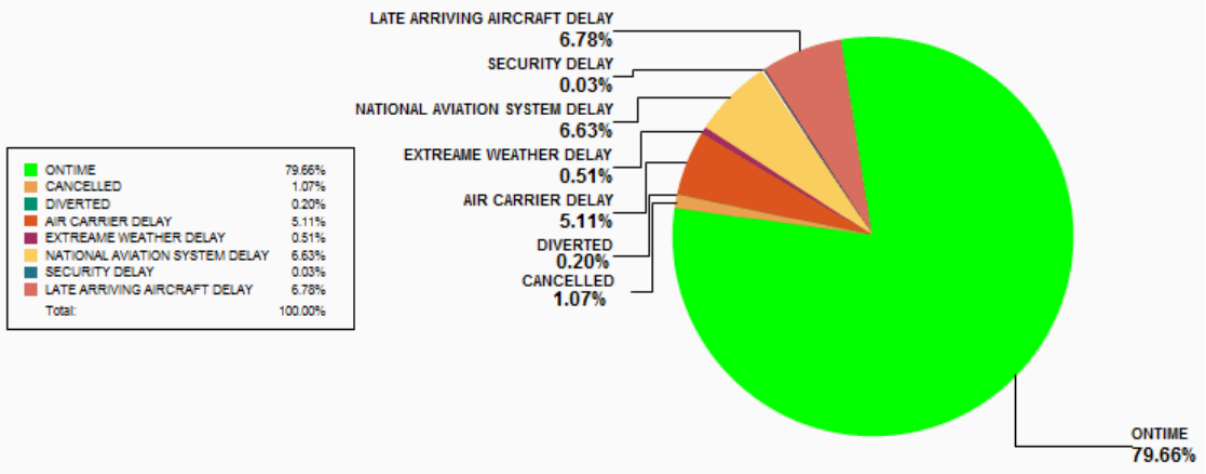

Fig. 3 - Overall causes of delay by reporting carrier November 2018. Source: Bureau of Transportation Statistics

Examining the U.S. economy, flight delays cost airlines $30 billion annually (IATA, 2019). Flight delays will not only increase economic losses, but can also weaken the performance of air transportation operational systems, adversely affecting passengers, airlines, and airport planning (K. K. H. Ng et al., 2019; J. Huo et al., 2020). Research (C. Diaz, 2020) reveals the fact that passengers who experience delays, especially service failures, promptly respond with negative emotional reactions and form a bad judgment for the airport. Therefore, having the ability to anticipate which flights may be delayed is valuable for airports and airlines. This foresight enables proactive decision-making to mitigate the impact of delays. Our study aims to predict if a flight will be on-time or delayed by using various machine-learning approaches to prevent the problems mentioned earlier.

1.2 Existing model limitations and research questions

In recent years, several studies have developed machine learning algorithms for predicting flight delays, emphasizing the importance of maintaining on-time performance (Alice Sternberg et al., 2017). Although the application of machine learning techniques to anticipate flight delays is new, it holds significant potential. Successfully estimating delays enables companies to prevent problems before they develop. As a result, concrete advantages, such as lower costs and higher customer satisfaction, will emerge. These improvements target the most vulnerable aspects of the aviation business.

Flight delay prediction has gained significant research attention in the last few years. One study introduced a machine learning model with a large number of classification models to predict flight delays (MC ATLIOĞLU et al., 2020). However, the result shows that some machine learning algorithms perform very poorly, while others perform well. They suggest

that the main reason is due to an imbalance in the dataset in terms of delay and on-time class distributions. Many studies observed similar challenges (Y Tang, 2021; Haixiang et al., 2017; Yu et al., 2015 and Xinglin Zhang et al., 2015b). In the future, they plan to repeat the experiments on the old dataset by using some techniques such as oversampling, SMOTE, etc. The use of sampling techniques in developing hybrid methods has been limited (Haixiang et al., 2017; Yu et al., 2016a). In another study, they developed a scalable parallel version of Random Forest (RF) to predict arrival delays caused by weather conditions (Belcastro et al., 2016). However, studies focusing solely on uncontrollable variables, such as weather conditions, may not contribute significantly to reducing flight delays or aiding decision-makers. Expanding the scope of variables to areas that can be controlled by decision-makers will provide a more robust solution to manage airline operations better.

To assist involved individuals and organizations in overcoming existing limitations, our study aims to address the key research questions:

1. How can the performance of an airline be enhanced through understanding and controlling the dataset variables?

2. Which techniques can effectively smooth decision boundaries and improve the prediction accuracy of machine learning methods?

3. Is there a risk of making incorrect decisions by sampling both flight delay training and testing sets?

4. As we know, each predictive machine learning model has certain advantages and disadvantages. How to combine weak models so that they support each other to form a strong model that solves the problem of improving the accuracy of flight delay prediction?

1.3 Research Contributions and Innovations

In this study, we collect official information on domestic flights performance nationally among all existing commercial airports in the United States. Addressing the challenges that limit the applicability of existing models for predicting flight delays is one of the objectives of this study. The study identified and tackled those challenges as a contribution and innovation to flight delay prediction:

First, innovations in aviation technology, including sensors, the Internet of Things, and Industry 4.0, make real-time tracking of flights and airplane performance possible. As a result, one can make more accurate predictions if a flight might be delayed due to

maintenance issues or en route, which can then be used to avoid last-minute disruption losses and passenger dissatisfaction (Thiagarajan et al., 2017). Most prior classification models used variables from the FAA dataset (Choi et al., 2016; Gopalakrishnan & Balakrishnan, 2017; Huang et al., 2017; Koetse & Rietveld, 2009; Sternberg et al., 2017). The complete list of the variables within an airline's decision control, examined in previous studies, is as follows: Date of Flight, Quarter of Year, Year, Day of Month, Day of Week, Unique Carrier, Origin Airport ID, Destination Airport ID, CRS Dep Time, CRS Arrival Time, Arrival Delay, CRS Elapsed Time, Departure Delay, Taxi Out Time, Arrival Time, Flight Distance. (Choi et al., 2016; Gopalakrishnan & Balakrishnan, 2017; Sternberg et al., 2017). The list of most commonly used variables in different models in the literature does not include uncontrollable variables such as weather since our focus is on controllable variables (Choi et al., 2016; Gopalakrishnan & Balakrishnan, 2017; Huang et al., 2017; Koetse & Rietveld, 2009). The literature review reveals some opportunities for extending the available work in the domain of airline delays. While previous studies have included origin and destination airports, their focus has been limited to one or a few specific airports (Choi et al., 2016; Gopalakrishnan & Balakrishnan, 2017; Sternberg et al., 2017). Therefore, there is an opportunity to expand the dataset to include data from 2016 to 2017 for all domestic flights across the United States at all commercial airports. The expanded dataset in terms of timeframe, and airports contributes to developing a model that accurately reflects actual practices rather than relying on a partial view of airline on-time and flight delay data. Our study does not provide a decision-making framework for optimizing operational management in the air travel industry. Instead, we focus on providing an analysis of the variables by feature importance that can help offer insights in developing a better framework for effective decision-making.

Second, the research and methodologies to address an imbalanced dataset can be classified into three main categories: Sampling techniques, Cost-sensitive learning, and the application of ensemble machine learning algorithms (Haixiang et al., 2017). Some papers also use other techniques to limit the imbalance dataset problem, such as loss functions to improve the classification rate of minority classes when sampling is difficult or not used (Haixiang et al., 2017). Various ensemble methods, like iterative ensemble classifiers and parallel-based ensemble classifiers, etc., are employed (Choi et al., 2016; XiaoLi Zhang et al., 2015). There are also a few hybrid techniques combining ensemble methods and cost-sensitive learning that are explored (Haixiang et al., 2017). Additionally, a proposed methodology combines sampling techniques and cost-sensitive methods to develop a new hybrid framework that can

be used with any machine learning algorithm (Alok Dand et al., 2020). In many cases, simple random over-sampling (Rebollo and Balakrishnan, 2014) and random under-sampling techniques (Belcastro et al., 2016) are recommended to balance the flight delay dataset to make it suitable for classification. In random oversampling, the risk of overfitting increases because of the use of duplicate examples from the minority class, while random undersampling raises the chances of losing potentially useful data by eliminating examples from the majority class (Batista et al., 2004). The current study applies various hybrid techniques incorporating both oversampling and under-sampling to balance the dataset and create smoother decision boundaries to address research question no.02 as outlined in subsection 1.3. One notable contribution of this study is the meticulous comparison of sampling techniques, including SMOTETomek, SMOTEENN, and SMOTE-RUS, in combination with six machine-learning algorithms for predicting flight delays. By leveraging these techniques, we aim to mitigate the challenges posed by imbalanced data, thereby enhancing the predictive performance of our models.

Third, among the above sampling techniques, the one with high prediction accuracy is often chosen for flight delay prediction by researchers. In existing works, sampling techniques are commonly applied to both training and testing sets of the dataset. In contrast, our study applies balancing techniques specifically to the training set and evaluates the performance by comparing it with the original testing set, addressing research question no.03 highlighted in subsection 1.3.

Fourth, numerous scholars have studied flight delay issues using different machine-learning methods. Esmaeilzadeh and Mokhtarimousavi employed a support vector machine to mine the nonlinear relationship between flight delay and various features. Given the black-box nature of machine learning, they conducted sensitivity analysis on corresponding variables and independent variables and comprehensively considered weather factors, airport scene operation, demand, and other factors. Another research presented a classification model based on Hartsfield-Jackson International Airport that utilized Decision Tree, Random Forest, and Multilayer Perceptron, with the Multilayer Perceptron providing the highest accuracy (Henriques and Feiteira, 2018). Additionally, a proposed methodology attempted two supervised learning algorithms, Decision Tree and KNN, and two ensemble learning algorithms, Random Forest, and Adaboost (Choi et al., 2016). The results showed that the ensemble algorithm classifier outperformed the single algorithm classifier. The Lithuania Airport flight delays datasets served as the research object in another study, where seven

machine learning algorithms were selected, such as probabilistic neural network, multilayer perceptron neural network, Gradient-Boosted Tree, Decision Tree, and the Gradient-Boosted Tree obtained optimal results (Stefanovič et al., 2020). The above research studies are inspirational, most of them obtain one optimal model through model comparison, while the other models were eliminated, leading to a potential waste of computing power. Additionally, flight datasets are enormous and versatile, and algorithm stability is significant for real-world applications. However, most studies did not pay attention to the stability of algorithms, especially some novel algorithms. Our approach incorporates ensemble learning, enabling the aggregation of predictions from multiple models to improve overall performance and robustness. In addition, we calculated SHAP values to ascertain the significance of variables in the model's impact on the outcomes. These values offer valuable insights into the contribution of each feature, thereby illuminating the factors that influence the model's predictions. By leveraging this information, we can identify and conclude the most crucial variables. Subsequently, we can utilize these important features as reference points for shaping airport strategies aimed at mitigating commercial revenue losses resulting from flight delays.

To summarize, the primary novelty of our work lies in its comprehensive comparative analysis, which sheds light on the relative strengths and weaknesses of different combinations of machine learning methodologies and sampling techniques in the context of flight delay prediction. Our findings reveal promising results, with the Random Forest, Random Forest with SMOTETomek, and Stacked Generation with SMOTEENN having the highest rating indexes, but the Random Forest model demonstrates superior performance in predicting flight delays, closely followed by the 2 models above.

Ultimately, the overarching goal of this research is to contribute to the advancement of machine learning in aviation, with the broader objective of fostering a more seamless and reliable global air transportation network.

The structure of the paper consists of seven chapters. It can be shown through the flowchart below (Fig. 4).

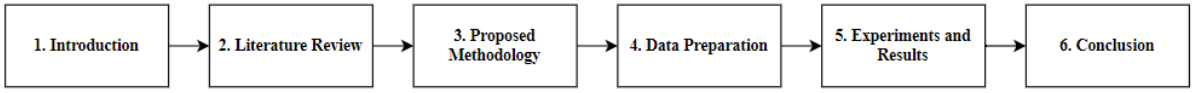

Fig. 4 - Paper Layout. Source: Own research

2 LITERATURE REVIEW

Flight delays have become a focal point for researchers due to the importance of the growing aviation industry. In existing studies, researchers mainly used optimization methods, network analysis, probabilistic models, statistical regression, and machine learning to study flight delays. Among various approaches, machine learning has gained much popularity in the last few years due to its ability to extract useful information from high-dimensional data (Khan et al., 2019b, 2019c; LeCun et al., 2015; Tkac and Verner, 2016). However, machine learning and flight delay topics started to be studied not long ago. One of the reasons for this is the development of machine learning methods so that big data operations can be done easily with machine learning. Since the topic has critical importance in air traffic control, airline decision-making processes and ground operations have been studied from various perspectives. The very first study was done by (Choi et al., 2016). Flight delays that are caused by weather are forecasted with the domestic flight data, and the weather data is used from 2005 to 2015. A subsequent study in 2019 utilized the SMOTE method for balanced data in binary classification, resulting in increased accuracy from 80.89% to 85.73% (Chakrabarty N, 2019). The author advocated for predicting flight delays by considering air traffic flow to enhance individual delay predictions. However, the domain of implementation is limited to five airports in the US. A hierarchical integrated machine learning model was presented to predict flight departure delays and duration at three hierarchical levels (W.A. Khan et al., 2021). The study compared various machine learning algorithms and sampling techniques, demonstrating the effectiveness of their proposed model in achieving better accuracy for classifying delay status and predicting delay duration. Recently, (X. Dong et al., 2023) introduced another prediction model, concentrating on forecasting the occurrence of Ground Delay Programs (GDPs) in flight schedules. This study integrated local weather data and flight information with the ATMAP algorithm to assess the impact of weather on GDPs. They employ various machine learning models, including Support Vector Machine, Random Forest, and XGBoost to estimate the probability of GDPs. Notably, the decision tree model outperformed, showing an 8.8% improvement in its correlation coefficient.

Another study by R.A. Sugara and D. Purwitasari (2022) focused on predicting the status of flight delays to mitigate airport commercial revenue losses (R.A. Sugara and D. Purwitasari, 2022). This research integrated weather characteristics with airport operational flight data and employed various classification algorithms, including Decision Tree, Random Forest,

Gradient Boosted Tree, and XGBoost. The study emphasized the importance of employing multiple sampling techniques and conducting feature importance analysis, particularly through SHAP, to comprehend the impact of weather on flight category status and guide airport strategies for minimizing revenue losses. A comprehensive approach was adopted by M.C. Atlıoğlu et al. (2020), who tested 11 machine learning models on operational data obtained from an airline company (M.C. Atlıoğlu et al., 2020). Their study provided insights into the importance of specific features, such as gate_resource_id and stand_resource_id, in flight delay prediction. The challenge posed by imbalanced datasets was acknowledged, emphasizing the need for balancing techniques.

A data mining approach, utilizing a Gradient Boosting Classifier and hyperparameter tuning results in improved accuracy for predicting flight arrival delays (N. Chakrabarty, 2019). The study also implemented oversampling techniques, such as Randomized SMOTE, to balance the dataset, contributing to enhanced prediction performance. R. Hendrickx, M. Zoutendijk, and M. Mitici (2021) addressed the challenges of imbalanced datasets in flight delay and cancellation prediction by optimizing imbalance ratios and employing sampling techniques like RUS and SMOTE. The findings highlighted the necessity of sampling techniques for achieving optimal prediction outcomes.

A novel method incorporating sampling and cost-sensitive techniques was introduced to address imbalanced datasets (Alok Dand et al., 2020). Their research emphasized the importance of variable consideration by decision-makers, exploring the impact of weather, aircraft, airport, and scheduling variables on predicting delays. In our research direction, similar to theirs, many variables in various databases are considered, but not including weather variables, which are uncontrollable variables.

In terms of classification accuracy, Random Forest has performed the best, followed by decision tree and neural network algorithms in various studies (Choi et al., 2016; Gopalakrishnan & Balakrishnan, 2017; Sternberg et al., 2017). Notably, Kalyani et al. proposed a flight arrival delay prediction classification model based on XGBoost and a flight arrival delay prediction regression model based on linear regression. Linear regression, a widely used algorithm in machine learning, is valued for its simplicity and ease of application, while XGBoost, an ensemble learning algorithm, leverages Decision Tree principles to optimize results through constant hyperparameter adjustments. In a more recent study by (I Hatıpoğlu, Ö Tosun, N Tosun, 2022), the focus shifted to predicting flight delays using advanced machine learning techniques such as GBDT, XGBoost, LightGBM, and Catboost.

The study highlighted the potential of these up-to-date machine learning techniques for accurate delay predictions and suggested potential future applications. Through various papers, we have selected three highly rated algorithms Random Forest, Decision Tree, and XGBoost to train the model. By combining SMOTE, feature selection using the Boruta algorithm, and a stacking classification algorithm, the research achieved high accuracy, precision, recall, and F1 Score (G. Li et al., 2021). The study underscored the importance of feature selection in machine learning applications and highlighted the stability and reliability of the stacking algorithm in flight delay prediction. The stacking algorithm is a new technique that not many papers have used, but realizing the potential of this technique, we decided to study it more closely in this research article. In another study, (P. Stefanovič et al., 2020) conducted a study on predicting flight delays at small airports in Lithuania with fresh data using machine learning models. The research focuses on integrating data from small airports and employs techniques such as SMOTE and grid search to find the best parameters that give the highest accuracy for each algorithm. They tested seven different algorithms, and the algorithm with the highest accuracy was gradient-boosted trees with arrivals of 88.59% and departures of 96.02%.

The algorithms have utilized different methods to estimate their performance. The accuracy of the classification model can be computed as a simple ratio of total number of correctly predicted points over total number of points labeled in the specific category (Choi et al., 2016; Gopalakrishnan & Balakrishnan, 2017; Sternberg et al., 2017). Another way to examine the accuracy of the classification model is through an assessment of the Confusion Matrix (Choi et al., 2016; Gopalakrishnan & Balakrishnan, 2017; Huang et al., 2017; Movva & Menon, 2016; Sternberg et al., 2017). Prior studies rely on the simple ratio method to support the accuracy of the classification model (Choi et al., 2016; Gopalakrishnan & Balakrishnan, 2017; Sternberg et al., 2017).

The contributions of researchers to predicting flight delays are remarkable. Most current research focuses on using multiple sampling techniques combined with many different machine-learning models. In this study, with a carefully prepared dataset, the principle of the Stacking classification algorithm is introduced, the SMOTE algorithm is chosen to handle imbalanced data sets, and the GridSearchCV algorithm is used to find the relevant information. The best number gives the highest accuracy for each algorithm. There are five supervised machine learning algorithms in the first-level Stack learner, including Random

Forest, Logistic Regression, Decision Tree, KNN, and SVM. The second-level learner is Logistic Regression.

3 PROPOSED METHODOLOGY

In this section, we present the proposed methodologies along with a schematic flowchart in Fig. 5, introducing the method flow of the study. After collecting the dataset from the Bureau of Transportation Statistics (BTS), we proceeded to prepare the complete dataset. Subsequently, we trained the model for each case: without sampling, after applying SMOTETomek, SMOTEENN, and SMOTE-RUS. The machine learning models employed include Decision Tree, XGBoost, Random Forest, KNN, SVM and finally used Stacking Ensemble Learning. We evaluated the performance of these models using parameters such as Accuracy, Precision, Recall, F1 score, and AUC-ROC to determine the best combination.

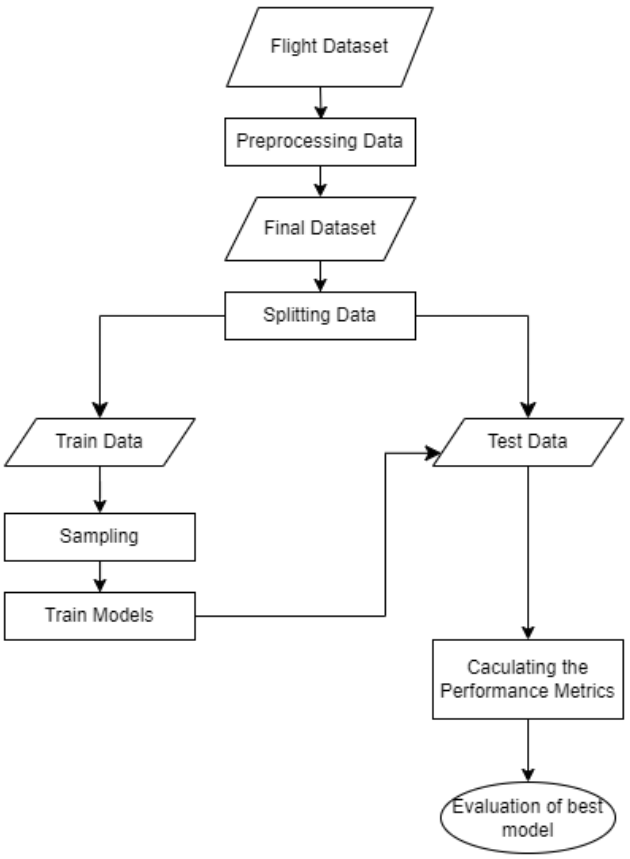

Fig. 5 - Flowchart of the proposed approaches. Source: own research

3.1 Variations of SMOTE sampling in handling imbalanced data

3.1.1 Imbalanced Dataset and SMOTE method

Datasets with unequal distribution of samples across label classes are termed imbalanced datasets. As in our dataset, the ratio between the minority class and the majority class is approximately 1:2, making the dataset mildly imbalanced. Training models on such datasets often results in bias towards the majority class, thereby complicating predictions for the minority class. By using various resampling techniques, categorized into oversampling, undersampling, and hybrid methods, we alleviate the effects of imbalance.

SMOTE is an oversampling technique that generates synthetic samples specifically for the minority class. It operates by creating a synthetic example, denoted as x_{new}, for each minority instance x_i. This process involves identifying the K-nearest neighbors of x_i , where K represents the number of minority instances with the smallest Euclidean distance to x_i. From these neighbors, one is randomly selected, denoted as y_i. Subsequently, a random number δ within the range [0, 1] is applied in Equation (1). This process results in x_{new} being positioned along the segment connecting x_i and y_i, thus enhancing the dataset with synthetic instances.

$$x_{new} = x_i + (y_i - x_i) * \delta \quad (1)$$

3.1.2 SMOTETomek

SMOTETomek is a hybrid approach designed to address class imbalances in datasets. While SMOTE is utilized to oversample the minority class, it can inadvertently introduce noise and further imbalance into the dataset. To mitigate these challenges, the Tomek links method is employed to clean the oversampled synthetic samples. Tomek links, a neighbor-based undersampling technique, identify pairs of opposite class observations that are each other's nearest neighbors. In the SMOTETomek process, Tomek links containing observations from both classes are removed. This results in a balanced dataset with well-defined class clusters.

3.1.3 SMOTEENN

SMOTEENN offers a dual-pronged approach to tackling imbalanced datasets by leveraging the strengths of SMOTE and ENN. Through the integration of SMOTE's oversampling of the minority class and ENN's simultaneous elimination of samples from both classes, SMOTEENN provides a comprehensive solution for refining datasets. This combined strategy aims to mitigate class imbalances while enhancing the quality of the data, resulting in a balanced dataset with improved classification performance and generalization ability.

3.1.4 SMOTE-RUS

SMOTE-RUS stands out as an effective strategy in handling imbalanced datasets by amalgamating SMOTE with Random Undersampling to mitigate their respective drawbacks. By initially applying SMOTE to oversample the minority class until a balanced ratio is attained, followed by random undersampling of the majority class, SMOTE-RUS ensures a desirable balance ratio as defined by Equation (2). This combination optimizes the dataset's class distribution, fostering improved classifier performance and robustness against class imbalance challenges.

$$\text{precUnder} = (\text{Num Of Positive Instances}/\text{Num Of Negative Instances} * 100) \quad (2)$$

3.2 Machine Learning Classification

3.2.1 Decision Tree

Decision tree algorithms are extensively employed in classification tasks owing to their simplicity and interpretability. They provide a clear framework, represented as a tree-like structure, facilitating users to understand the decision-making process effortlessly. The primary objective of decision trees is to create a model that predicts the value of a target variable based on multiple input variables. Typically, constructing decision trees involves two phases: (i) tree growth, where the training set is recursively split using local optimal criteria until most records within a partition share the same class label, and (ii) tree pruning, aimed at reducing the tree's size for improved comprehension. Various methods for constructing decision trees differ based on the selected operators. Notably, CART and C4.5 emerged as the most commonly used algorithms for decision tree construction.

3.2.2 XGBoost

XGBoost, an abbreviation for eXtreme Gradient Boosting, is a highly advanced ensemble learning technique rooted in decision trees and gradient boosting methodology. Unlike traditional decision trees, XGBoost optimizes a regularized objective function, comprising a loss function and a regularization term, to minimize prediction errors while preventing overfitting. Its scalability and efficiency, achieved through parallel and distributed computing, enable the handling of massive datasets with millions of observations and features, making it invaluable across various industries. Moreover, XGBoost incorporates advanced regularization techniques like shrinkage and column subsampling to enhance generalization performance. Despite its complexity, XGBoost offers interpretable insights into feature importance, facilitating the understanding of predictive patterns. However, mastering

XGBoost requires a comprehensive grasp of its principles and parameter-tuning nuances, presenting a challenge for newcomers. Nonetheless, its predictive prowess and interpretability render XGBoost indispensable for sophisticated prediction tasks in diverse domains.

3.2.3 Random Forest

Random Forest is a potent ensemble learning method that harnesses the collective power of multiple decision trees. Its performance directly correlates with the number of trees in the forest, with more trees generally leading to higher accuracy. Unlike traditional decision tree algorithms, Random Forest combines Bagging with random feature selection, reducing variance and enhancing generalization. Introduced by Leo Breiman in 2001, it has become widely popular for its simplicity, flexibility, and robust performance across diverse classification tasks. By aggregating predictions from various trees, Random Forest mitigates overfitting and noise, making it suitable for handling large datasets efficiently. Its straightforward implementation and reduced sensitivity to hyperparameters further contribute to its appeal among data scientists and machine learning practitioners.

3.2.4 KNN

The k-nearest neighbor (KNN) technique determines a sample's class based on its nearest neighbors, with the number of neighbors defined by parameter k. It's categorized into structure-based and structure-less approaches, depending on how data is organized. KNN is effective for large and noisy training data but faces challenges in scaling for high-dimensional datasets. Strategies like the K-Nearest Neighbor Mean Classifier (k-NNMC) have been proposed to mitigate space requirements, offering improved classification accuracy compared to other methods. Despite its simplicity and robustness, KNN suffers from computational complexity, memory limitations, and runtime performance issues with large datasets and irrelevant attributes.

3.2.5 SVM

Support Vector Machines (SVM) are highly regarded in supervised learning, especially for analyzing high-dimensional datasets. SVM aims to find the optimal hyperplane that effectively separates different classes based on training data, ensuring a wide margin between classes for robust classification. It can utilize various kernel functions to handle nonlinear relationships and map data into higher-dimensional spaces. SVM's focus on support vectors, the closest data points to the decision boundary, enhances efficiency, particularly for large datasets. Despite its effectiveness, SVM's performance depends on proper parameter selection

and kernel choice. Nonetheless, with appropriate tuning, SVM remains a versatile and powerful tool for classification tasks across various domains.

3.3 Stacking Ensemble Learning

In recent years, the relevance of ensemble models has grown significantly due to their remarkable performance in tasks such as classification and regression. These methods involve combining different learning models to collectively improve upon the results obtained by each individual model. Among the fundamental ensemble methods, bagging, boosting, and stacking stand out as the most widely used and recognized techniques.

In this paper, we adopt a stacking approach, considering it the most suitable method for the classification problem under investigation. Stacking involves constructing models using various learning algorithms, followed by training a combiner algorithm to make final predictions using the base algorithms' predictions. Figure 6 illustrates a general schematic of this approach.

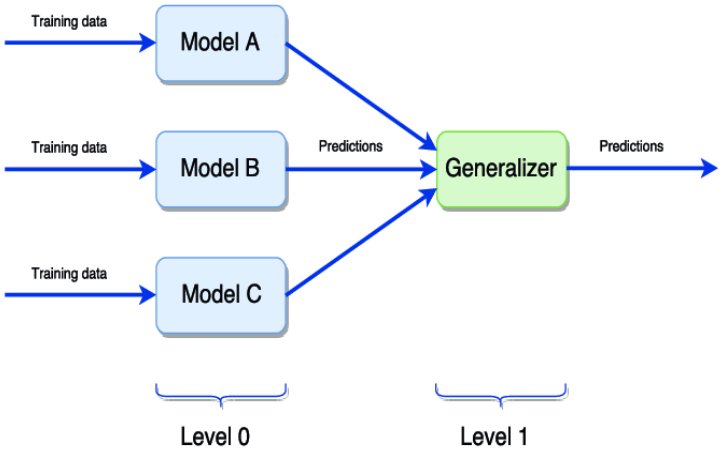

Fig. 6 - An example scheme of stacking ensemble learning. Source: own research

Additionally, the stacking approach offers the advantage of allowing for the incorporation of domain knowledge and expert insights into the modeling process. By combining diverse learning algorithms, stacking can effectively leverage the collective intelligence of different models while accommodating prior knowledge about the problem domain. This adaptability allows practitioners to incorporate specific features and constraints from their domain, enhancing the ensemble's relevance and making the predictive model more applicable. Moreover, stacking's ability to combine multiple models trained on different subsets of the

data can help mitigate overfitting and enhance the model's ability to generalize to unseen data, making it a robust and reliable technique for classification tasks across various domains.

3.4 Model Validation

In this study, we employ the confusion matrix, as depicted in Figure 3, to compute metrics such as Accuracy, Precision, Recall, F1 Score, and ROC-AUC. These are to evaluate the predictive performance.

Confusion Matrix		Predicted	
		Positive	Negative
True	Positive	*TP*	*FN*
	Negative	*FP*	*TN*

Fig. 7 - Confusion matrix. Source: own research

True Positive (TP) signifies instances where both the ground truth and the prediction are positive, representing the count of correctly predicted positive samples. False Positive (FP) denotes cases where the ground truth is negative, but the prediction is positive, indicating the number of negative samples erroneously classified as positive. True Negative (TN) reflects instances where both the ground truth and the prediction are negative, representing the count of correctly predicted negative samples. False Negative (FN) indicates situations where the ground truth is positive, but the prediction is negative, representing the number of positive samples erroneously classified as negative.

3.4.1 Accuracy

Accuracy is calculated as the ratio of correctly predicted samples to the total number of samples. Its formula is expressed as Equation (3):

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \qquad (3)$$

Accuracy is widely utilized as an evaluation metric in classification tasks. However, when dealing with imbalanced datasets, such as the flight delay data where the number of on-time flights significantly outweighs the delayed flights, solely optimizing for accuracy can lead to biased results. In such scenarios, models may tend to classify minority samples as the majority class to achieve higher accuracy, consequently neglecting effective prediction of

delayed samples. Hence, it's essential to complement accuracy with metrics like Precision, Recall, F1 Score and ROC - AUC to provide a more comprehensive assessment of the predictive performance in classification problems.

3.4.2 Precision

Precision is a pivotal gauge of accuracy, and reflects the ratio of correctly classified minority instances to the total instances predicted as belonging to the minority class. This vital metric, as elucidated by Equation (3), offers insight into the effectiveness of classification, particularly in discerning the accuracy of minority class predictions.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

3.4.3 Recall

This metric signifies the percentage of minority instances correctly classified as part of the minority class. In the literature, it's referred to by various names such as sensitivity, true positive rate (TPrate), or positive accuracy. Equation (4) outlines its definition.

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

3.4.4 F1 score

F1 score, also referred to as F-Measure, represents the harmonic mean of precision and recall. As it increases with both precision and recall, a higher F1 score suggests better performance in the minority class. Equation (5) defines this metric.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \quad (6)$$

3.4.5 AUC - ROC

The AUC-ROC (Area Under the Receiver Operating Characteristic Curve) is a performance metric commonly used to evaluate the quality of a binary classification model.

The ROC curve is widely used for assessing the performance of classifiers, providing a visual representation of their trade-offs between success rate and false alarm rate. Originally developed for signal detection applications, it has since been adopted across various domains.

The ROC curve is a two-dimensional graph where True Positive Rate (TP rate) is plotted on the y-axis and False Positive Rate (FP rate) on the x-axis. In the case of discrete classifiers, each point in the ROC space corresponds to a pair of (FP rate, TP rate). However, probabilistic classifiers yield continuous numerical values. Consequently, the use of a threshold enables the generation of a series of points in the ROC space, resulting in a curve instead of a single point.

From the ROC curve, another metric called the AUC is defined in Equation (7) to compare the performance of two classifiers. If the area under the curve associated with classifier C1 is greater than that associated with classifier C2, then the performance of C1 is considered superior to that of C2.

$$AUC = \frac{TPrate + TNrate}{2} = \frac{1 + TPrate - FPrate}{2} \qquad (7)$$

3.5 Feature Importance

Feature Importance is a pivotal concept in machine learning, providing valuable insights into the relevance of different features in predictive modeling. By utilizing various methods such as tree-based models, linear models, permutation importance, and SHAP values, practitioners can assess the contribution of each feature to the model's predictions. This understanding enables better model interpretation and facilitates informed decision-making regarding feature selection and model optimization.

In this study, we use the SHAP value, which indicates which features have a significant positive or negative impact on the delayed flight classification, what the magnitude of the impact is, and how much a specific feature value drives the classification of a flight as delayed. For a specific flight, a high positive (largely negative) SHAP value for a feature indicates its substantial contribution to classifying a flight as delayed/not delayed. A SHAP value of a feature close to zero suggests that the feature has minimal influence on the classification and does not contribute to decide in classifying a flight as being delayed or not. In this paper, the SHAP values are expressed in log odds, where the log odd of a variable A is defined as (8):

$$log(\frac{P(A)}{1-P(A)}), \text{ with } P(A) < 1 \qquad (8)$$

Feature importance serves as a crucial tool for diagnosing model performance and identifying potential areas for improvement. By identifying the most influential features, data scientists can focus their efforts on refining these aspects, thereby enhancing the overall efficacy and interpretability of machine learning models. Overall, a thorough understanding of feature importance is essential for building robust and interpretable machine-learning models across diverse applications and domains.

4 DATA PREPARATION

This paper uses official datasets of flights from a public information source - the Bureau of Transportation Statistics (BTS) of the United States Department of Transportation, which provides information on flight performance recorded from 2016 to 2017.

Focusing on flight delay prediction, we exclude canceled and diverted flights from the RCOP dataset. Next, we eliminate duplicate rows and those containing missing values. Additionally, rows exhibiting questionable attributes, such as negative flight time lengths, are removed. Flights with a delay time length (variable DEP_DELAY) exceeding 180 minutes are removed from the dataset. They account for approximately 0.88 percent of the final dataset but can cause tremendous issues to the algorithms' performance.

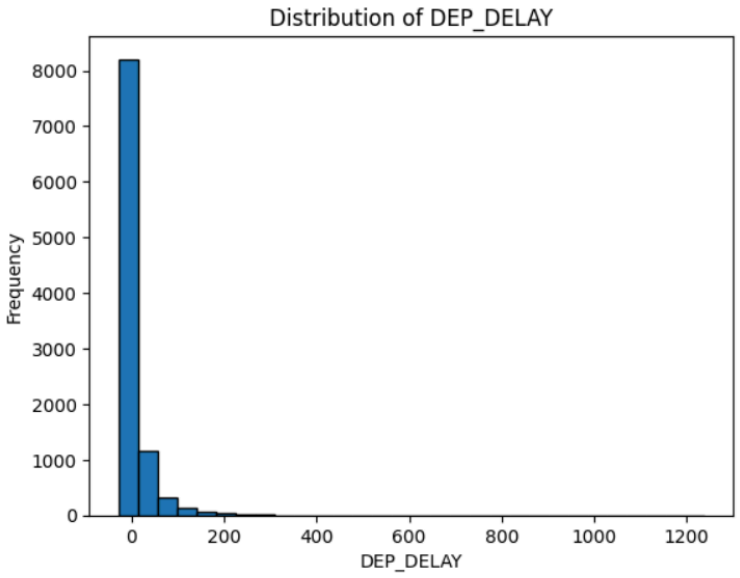

Fig. 8 - Distribution of DEP_DELAY in the final dataset. Source: own research

Additional adjustments to essential data types are made to enhance the subsequent algorithms' efficiency. Most categorical variables in the dataset are applied with the label encode method. This assigns a unique numerical label to each category, converting categorical data into the numerical input required by machine learning algorithms. A distinct column, “Status” is

added, signifying the flight status, with “1” indicating a “delayed” flight and “0” representing an “on-time” flight status. This classification aligns with the FAA's definition, where flights experiencing delays exceeding 15 minutes are categorized as “delayed”, and others as “on-time”.

A representative distribution of classes is crucial for accurate model training and evaluation. Because of computing limitations, a stratified sampling technique is employed to address the class imbalance issue within the dataset. This involves grouping the data based on the “Status” column and sampling from each group proportionally, ensuring a sufficient representation of each class in the resulting dataset. Specifically, samples are drawn from each class with a size of 10,000 observations.

Following the stratified sampling step, standard scaling was applied to the dataset. By standardizing the features in this manner, the influence of the scale of the features was mitigated, ensuring that all features contributed equally to the subsequent analyses.

Finally, the dataset was split into training and testing sets, with a stratified approach used to ensure that the distribution of classes remained consistent across both sets. This train-test split is essential for assessing the model's performance on unseen data and gauging its generalization ability.

Tab. 1 - Variables used. Source: own research

Variable	***Description***	***Data Type***
Independent Variable		
QUARTER	The quarter at which the flight happened.	Categorical
MONTH	The month in which the flight happened.	Categorical
DAY_OF_MONTH	The day of the month on which the flight happened.	Categorical
DAY_OF_WEEK	The day of the week on which the flight happened.	Categorical
OP_UNIQUE_CARRIER	The flight’s carrier.	Categorical
TAIL_NUM	N-number of the aircraft involved.	Categorical
OP_CARRIER_FL_NUM	Flight number.	Categorical
CRS_ELAPSED_TIME	The flight’s scheduled total time length.	Continuous
CRS_DEP_TIME	The flight’s scheduled departure time.	Continuous
CRS_ARR_TIME	The flight’s scheduled arrival time.	Continuous

DEP_DELAY	Time gap between scheduled and actual departure.	Continuous
DISTANCE	Distance between the two airports.	Continuous
ORIGIN_AIRPORT_ID	The airport from which the flight's aircraft departed.	Categorical
DEST_AIRPORT_ID	The airport to which the flight's aircraft arrived.	Categorical
Dependent Variable		
STATUS	Delayed/On-time	Categorical

5 EXPERIMENTS AND RESULTS

5.1 Model training and evaluation

By splitting the standardized data, we separate train and test sets from the final dataset at the proportion of 1:4. Three hybrid sampling methods are used in the expectation of preventing performance failure due to imbalanced data.

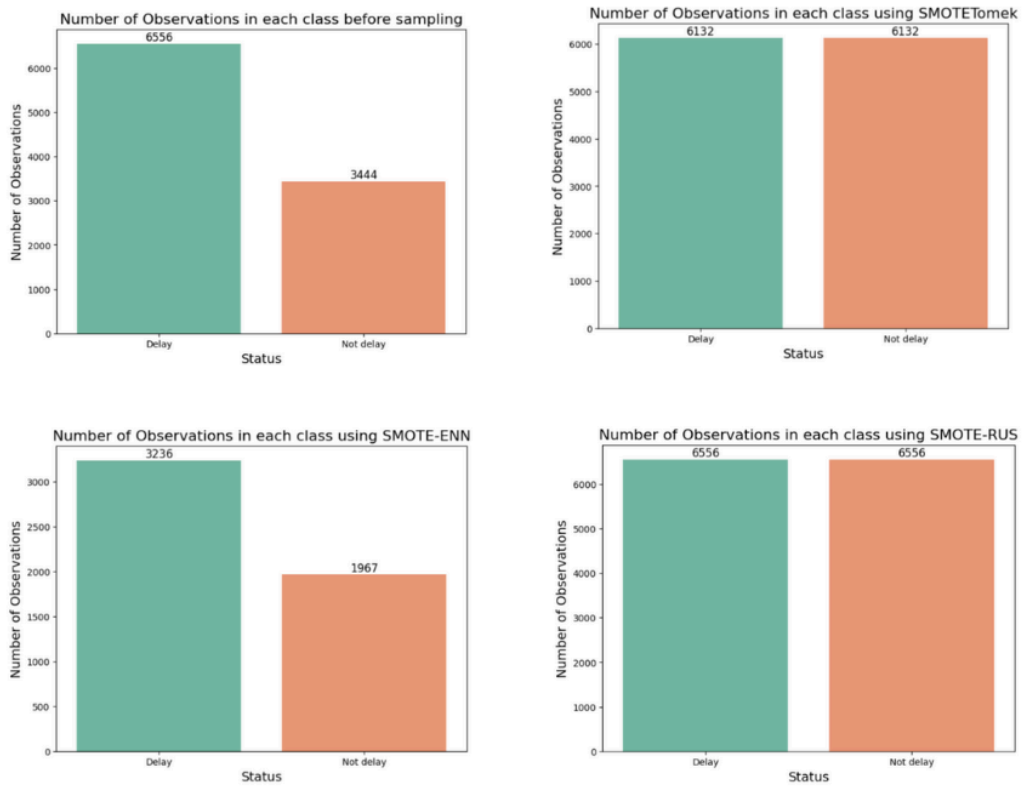

Fig. 9 - Number of observations in each class, before and after sampling. Source: own research

The dataset initially has one class comprising 65.56% of observations and the other class comprising 34.44%. Employing hybrid sampling techniques like SMOTETomek and SMOTE-RUS balances the class proportions, aiming for a fairer distribution while

maintaining overall balance. This process increases the number of observations. However, when using SMOTE-ENN, the number of observations decreases, and the resulting class proportions generally resemble the original distribution.

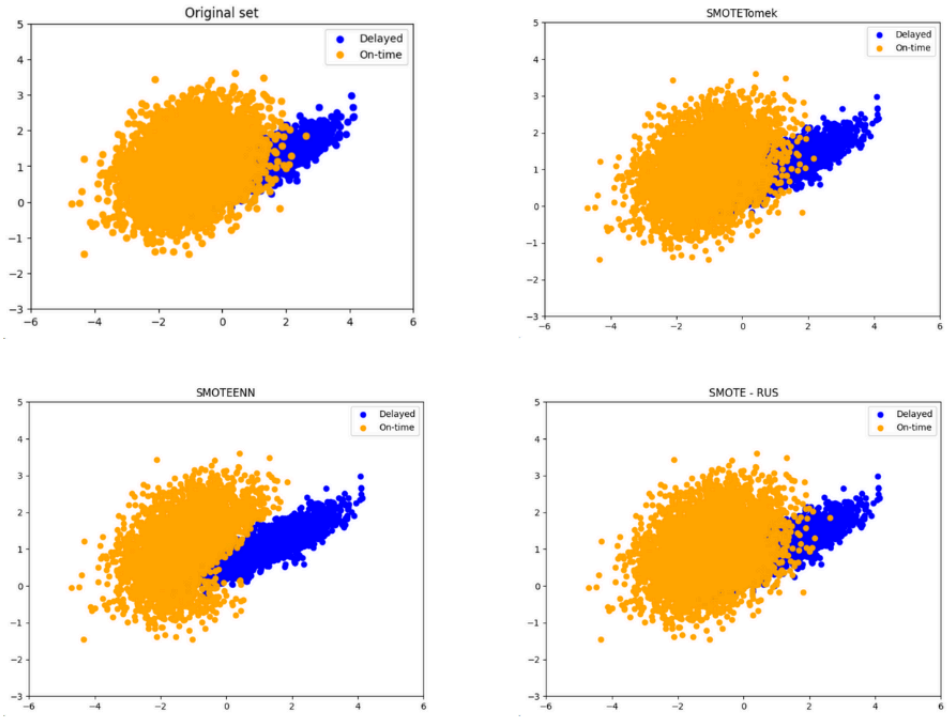

Fig. 10 - Data distribution, before and after sampling. Source: own research

A diverse set of machine learning models: Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Trees (DT), Random Forest (RF), and XGBoost are employed to perform classification prediction. Furthermore, we also include an ensemble version, consisting of a total of five mentioned models, aiming to reduce overfitting and enhance generalization. In this manuscript, we refer to the proposed methodology as "Stacked Generalization," abbreviated as SG. Especially, within this ensembled method, a Logistic Regression model acts as a meta-model, connecting predictions from base models trained on both sampled with unsampled data.

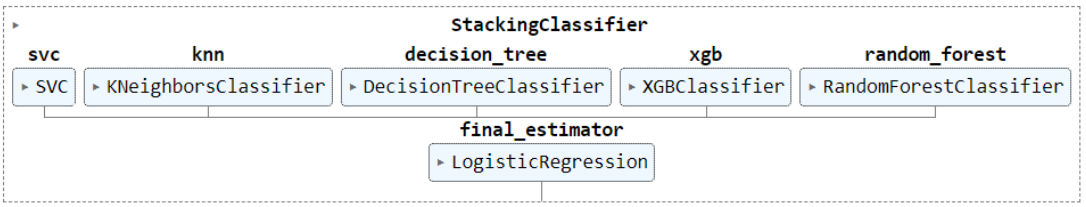

Fig. 11 - Meta-model of Ensemble method. Source: own research

Our primary aim is to systematically assess and compare the influence of diverse sampling techniques on the efficacy of various machine learning algorithms. To achieve this objective, we have structured our analysis into four distinct cases. In each case, classification predictions are executed using a consistent set of the mentioned 6 models. The differentiating factor lies in the sampling method employed for each case. The initial case involves models without any sampling. The second case incorporates SMOTE TOMEK, while the third and fourth cases integrate SMOTE ENN and SMOTE RUS, respectively. This systematic approach allows for a comprehensive examination of the impact of specific sampling methodologies on the performance of our machine-learning models, serving our comparison purpose.

The evaluation process involves comprehensive metrics such as precision, F1-score, recall, accuracy, and ROC-AUC scores. Thanks to their assistance, we have assessed the effectiveness of each model under different sampling strategies. The modeling result on data testing of each model in 4 cases can be seen in the following figure.

Classifier + Sampling technique	Acc	F1-Score 0	F1-Score 1	Precision 0	Precision 1	Recall 0	Recall 1	Roc-Auc
DT	0.74	0.80	0.61	0.81	0.60	0.79	0.62	0.7088
DT with ENN	0.67	0.71	0.61	0.84	0.51	0.62	0.77	0.6934
DT with RUS	0.74	0.80	0.64	0.82	0.61	0.78	0.67	0.7249
DT with Tomek	0.75	0.80	0.64	0.82	0.61	0.78	0.67	0.7268
KNN	0.73	0.81	0.48	0.74	0.66	0.90	0.38	0.6413
KNN with ENN	0.61	0.67	0.51	0.75	0.44	0.60	0.61	0.6066
KNN with RUS	0.70	0.79	0.49	0.74	0.57	0.84	0.43	0.6338
KNN with Tomek	0.71	0.79	0.50	0.75	0.58	0.84	0.44	0.6387
RF	0.83	0.88	0.74	0.83	0.85	0.95	0.62	0.7760
RF with ENN	0.75	0.80	0.67	0.86	0.60	0.74	0.76	0.7517
RF with RUS	0.83	0.88	0.72	0.84	0.80	0.92	0.66	0.7890
RF with Tomek	0.83	0.87	0.72	0.84	0.79	0.91	0.66	0.7844
SG	0.83	0.88	0.71	0.83	0.85	0.94	0.61	0.7775
SG with ENN	0.83	0.87	0.72	0.84	0.79	0.91	0.66	0.7844
SG with RUS	0.83	0.88	0.72	0.83	0.82	0.93	0.64	0.7823
SG with Tomek	0.83	0.88	0.72	0.84	0.82	0.83	0.64	0.7842
SVM	0.83	0.88	0.68	0.81	0.90	0.97	0.55	0.7592
SVM with ENN	0.65	0.69	0.61	0.86	0.49	0.57	0.81	0.8094
SVM with RUS	0.82	0.87	0.71	0.83	0.79	0.91	0.64	0.8288
SVM with Tomek	0.82	0.87	0.70	0.83	0.77	0.91	0.64	0.8291
XGBoost	0.82	0.87	0.69	0.82	0.82	0.93	0.60	0.7654
XGBoost with ENN	0.73	0.78	0.66	0.86	0.58	0.72	0.77	0.7429
XGBoost with RUS	0.82	0.87	0.70	0.83	0.79	0.92	0.62	0.7705
XGBoost with Tomek	0.82	0.87	0.71	0.84	0.78	0.91	0.65	0.7773

Fig. 12 - Data testing result. Source: own research

5.2 Feature Importance

The performance of various machine learning models was evaluated in terms of accuracy (Acc), ROC_AUC score (Roc-Auc), F1-score, precision, and recall for predicting delays in a given dataset. Notably, RandomForest models exhibited high accuracy across different sampling techniques, while KNN models generally showed lower performance, especially

when coupled with SMOTE-ENN. Additionally, SVM models with SMOTETomek demonstrated the highest ROC_AUC score, indicating strong predictive capability.

Examining F1-scores, Stacked Generalization, and SVM models without sampling stood out for accurately predicting instances of on-time flights (class 0), while RandomForest performed best for identifying delayed flights (class 1). Conversely, KNN without sampling struggled to accurately classify both classes.

Precision analysis revealed that RandomForest models consistently achieved high precision, particularly when employing SMOTE-ENN, while KNN with SMOTE-ENN exhibited the lowest precision for both classes. Notably, precision for class 1 was generally lower across all models compared to class 0.

Regarding recall, SVM without sampling demonstrated the highest recall for class 0, highlighting its effectiveness in capturing instances without delays. XGBoost and Decision Tree models with SMOTE-ENN showcased the highest recall for class 1, indicating their proficiency in identifying delayed instances. Conversely, KNN without sampling displayed the lowest recall for both classes, suggesting its limitations in correctly identifying instances of delays. These findings collectively underscore the importance of model selection and sampling techniques in accurately predicting delays in the given dataset.

We calculated SHAP values based on our best-performing model - Random Forest. These values provide insights into the contribution of each feature towards the model's predictions, shedding light on the factors influencing the outcomes. Fig. 13 is a summary plot that displays aggregated SHAP values for all features used for classification. In Fig. 14, dots representing flights are aggregated. Their colors indicate feature values, where blue signifies small values and red signifies large values.

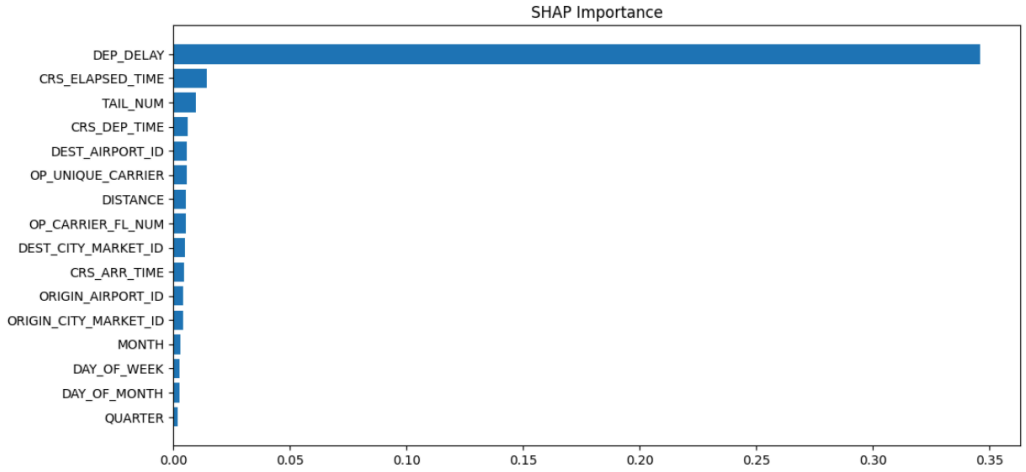

Fig. 13 - SHAP importance on all features. Source: Own research

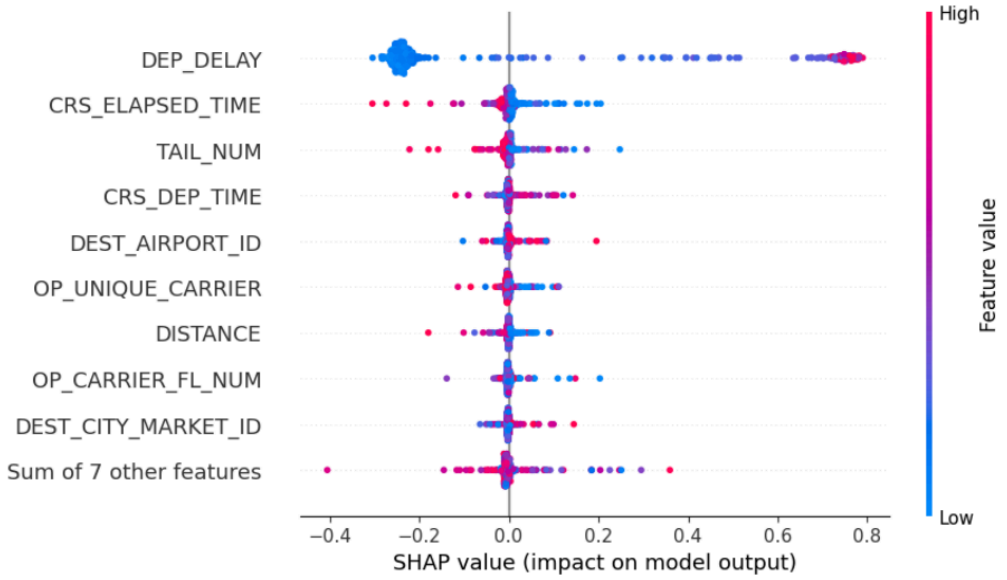

Fig. 14 - SHAP value of all features impact on model output. Source: Own research

For instance, in Fig. 14, for the feature DEP_DELAY, there is a significant number of flights where this feature exhibits a range of SHAP values between -1 and 0 (represented by an accumulation of blue dots). The blue dots (flights) with negative SHAP values, which have low values in DEP_DELAY (blue color), are classified as not delayed (negative SHAP). In particular, for the blue dots (flights) where the SHAP value is close to zero, the DEP_DELAY is low (blue color), but it does not significantly impact the classification of these flights (SHAP value close to zero).

In Fig. 14, the features are sorted by the sum of the SHAP values magnitudes over all samples. The feature at the top of the graph has the highest impact on the flight classification, whereas the feature at the bottom of the graph has the lowest impact. For example, the feature DEP_DELAY is at the top of the graph since it has the highest impact on the flight delay

classification. The features CRS_ELAPSED_TIME and TAIL_NUM follow as the second and third most impactful features. The feature QUARTER has the lowest feature importance for flight delay classification. Therefore, we utilize summary plots to assess the importance of these features, as explained previously, to conclude that the top 3 most important features are DEP_DELAY, CRS_ELAPSED_TIME and TAIL_NUM.

6. DISCUSSIONS

Based on the evaluation metrics of classification report and ROC-AUC, the Random Forest model, the SMOTETomek Random Forest model, and the Stack Generation SMOTEENN model exhibit the highest performance. Particularly, the latter two models have identical performance metrics. In terms of precision, both models achieve over 80% for both classes, indicating their ability to accurately predict positive and negative cases. However, their recall for class 1 is comparatively lower, implying they struggle to capture a significant portion of actual class 1 cases. The f1-score also indicates better performance for class 0, suggesting the models are more adept at predicting class 0. Despite this, both models achieve an overall accuracy of 83%, signifying a high level of correct predictions.

In ROC-AUC evaluation, the Random Forest model and the Stack Generation SMOTE-ENN model achieve scores of 0.7760 and 0.7844 respectively, indicating effective performance in predicting flight delays. Although there is a slight difference in scores, both models demonstrate considerable efficacy.

However, the Random Forest model emerges as the most effective due to its comparable performance without the need for sampling methods like the Stack Generation SMOTEENN model. This not only saves time but also computational resources. Therefore, the Random Forest model is deemed superior for flight delay prediction in this study. Comparing models based on the Accuracy index, Random Forest, Stacked Generation, Support Vector Machine, and XGBoost models consistently exhibit high accuracy, surpassing 80%. Conversely, the KNN model performs the poorest with accuracy ranging from 61-73%. This underscores the compatibility of data in flight delay research with these models.

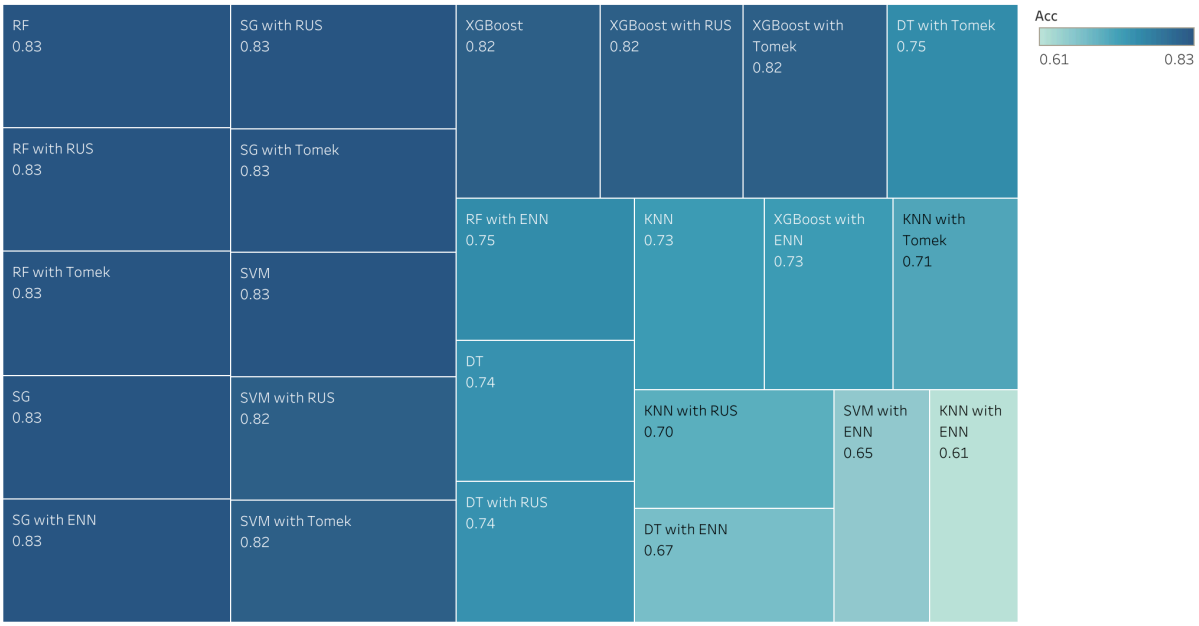

Fig. 15 - The Accuracy index of the models in the research. Source: own research

Regarding the ROC-AUC index, the Support Vector Machine model, especially when combined with sampling methods like SMOTETomek, SMOTEENN, and SMOTE, achieves the highest index, exceeding 80%. Following closely are the Random Forest and Stacked Generation models, which demonstrate around 78% efficiency with certain sampling methods. These findings highlight the practical utility of these models in the context of flight delay prediction.

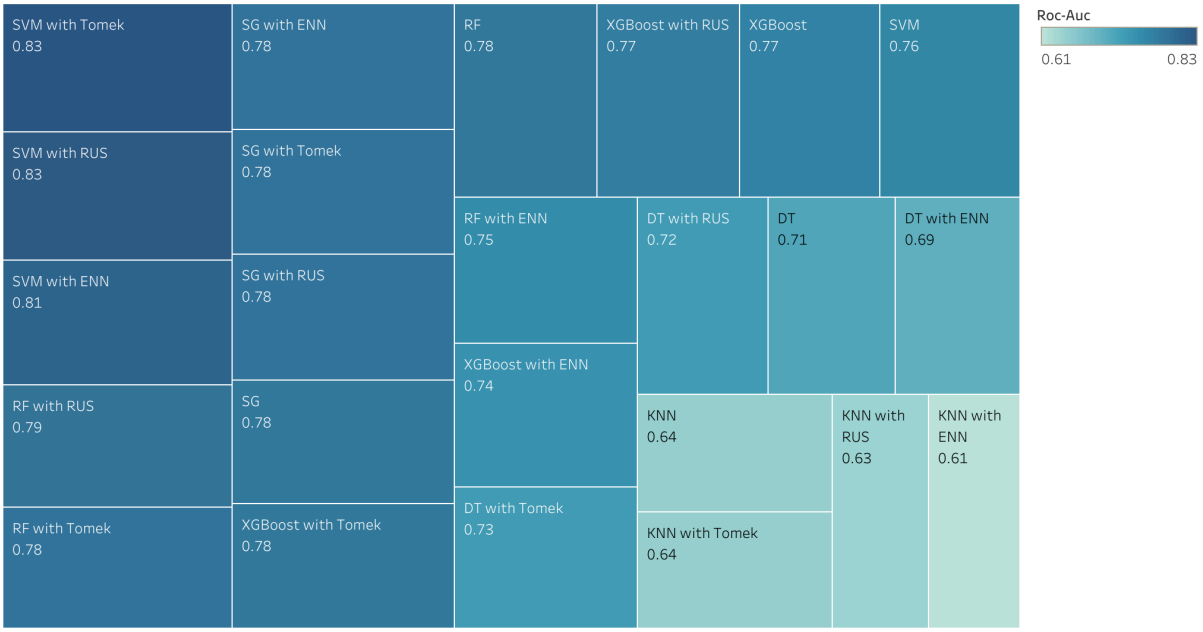

Fig. 16 - The ROC-AUC index of the models in the research. Source: own research

The research results indicate the most influential variable as DEP_DELAY. This suggests that airports need to take measures to ensure flights depart on time by timely monitoring weather conditions, plans adjustment, and addressing departure delays promptly. Implementing these measures will improve operational efficiency and passenger experience, contributing to a more resilient aviation system.

Data preprocessing techniques were pivotal in achieving the expected effectiveness in predicting experiments. Procedures such as data transformation, label encoding, and stratified sampling significantly enhanced the overall performance of the model.

The combination of machine learning models with sampling methods is effective in the study, as demonstrated in the Random Forest model combined with SMOTETomek, yielding promising results. Additionally, the Stacked Generation model, composed of various individual models, also demonstrated high accuracy and efficiency in flight prediction. This is evident through the identical performance between Stacked Generation SMOTEENN and Random Forest SMOTETomek.

Acknowledgement

This research is funded by the University of Economics and Law, Vietnam National University Ho Chi Minh City, Vietnam.

References

Alderighi, M., & Gaggero, A. A. (2018). Flight cancellations and airline alliances: Empirical evidence from Europe. Transportation Research Part E: Logistics and Transportation Review, 116, 90-101.

ATLIOĞLU, M. C., Bolat, M., Şahin, M., Tunali, V., & KILINÇ, D. (2020). Supervised learning approaches to flight delay prediction. Sakarya University Journal of Science, 24(6), 1223-1231.

Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD explorations newsletter, 6(1), 20-29.

Bhatia, N. (2010). Survey of nearest neighbor techniques. arXiv preprint arXiv:1007.0085.

Brodley, C. E., & Utgoff, P. E. (1992). Multivariate versus univariate decision trees. Amherst, MA: University of Massachusetts, Department of Computer and Information Science.

Chakrabarty, N. (2019). A data mining approach to flight arrival delay prediction for American airlines. In 2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON) (pp. 102-107). IEEE.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

Choi, S., Kim, Y. J., Briceno, S., & Mavris, D. (2016). Prediction of weather-induced airline delays based on machine learning algorithms. In 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC) (pp. 1-6). IEEE.

Dand, A. (2020). Airline delay prediction using machine learning algorithms. Doctoral dissertation, Wichita State University.

Daskalaki, S., Kopanas, I., & Avouris, N. (2006). Evaluation of classifiers for an uneven class distribution problem. Applied artificial intelligence, 20(5), 381-417.

Divina, F., Gilson, A., Goméz-Vela, F., García Torres, M., & Torres, J. F. (2018). Stacking ensemble learning for short-term electricity consumption forecasting. Energies, 11(4), 949.

Dong, X., Zhu, X., Hu, M., & Bao, J. (2023). A Methodology for Predicting Ground Delay Program Incidence through Machine Learning. Sustainability, 15(8), 6883.

Dudani, S. A. (1976). The distance-weighted k-nearest-neighbor rule. IEEE Transactions on Systems, Man, and Cybernetics, (4), 325-327.

Eurocontrol. (2018). Network Manager Annual Report. Accessed in February 2020.

Eurocontrol. (2018). Network Operations Report. Accessed in February 2020.

Eurocontrol. (2020). Five-Year Forecast 2020-2024. Accessed on 24-02-2021.

Fu, K., Qu, J., Chai, Y., & Dong, Y. (2014). Classification of seizure based on the time-frequency image of EEG signals using HHT and SVM. Biomedical Signal Processing and Control, 13, 15-22.

Gad, W., & Hashem, M. (2023). Smote-rus: Combined oversampling and undersampling techniques to classify the imbalanced autism spectrum disorder dataset. International Journal of Intelligent Computing and Information Sciences, 23(3), 83-94.

Hatıpoğlu, I., Tosun, Ö., & Tosun, N. (2022). Flight delay prediction based with machine learning. LogForum.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on knowledge and data engineering, 21(9), 1263-1284.

Hendrickx, R., Zoutendijk, M., Mitici, M., & Schäfer, J. (2021). Considering Airport Planners' Preferences and Imbalanced Datasets when Predicting Flight Delays and Cancellations. In 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC) (pp. 1-10). IEEE.

Henriques, R., & Feiteira, I. (2018). Predictive modelling: flight delays and associated factors, hartsfield–Jackson Atlanta International airport. Procedia computer science, 138, 638-645.

Huang, J., & Ling, C. X. (2005). Using AUC and accuracy in evaluating learning algorithms. IEEE Transactions on knowledge and Data Engineering, 17(3), 299-310.

Huo, J., Keung, K. L., Lee, C. K. M., Ng, K. K., & Li, K. C. (2020). The prediction of flight delay: Big data-driven machine learning approach. In 2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (pp. 190-194). IEEE.

Kesavaraj, G., & Sukumaran, S. (2013). A study on classification techniques in data mining. In 2013 fourth international conference on computing, communications and networking technologies (ICCCNT) (pp. 1-7). IEEE.

Khan, W. A., Ma, H. L., Chung, S. H., & Wen, X. (2021). Hierarchical integrated machine learning model for predicting flight departure delays and duration in series. Transportation Research Part C: Emerging Technologies, 129, 103225.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30.

Ng, K. K. H., Lee, C. K. M., & Chan, F. T. (2019). An alternative path modelling method for air traffic flow problem in near terminal control area. In 2019 2nd International Conference on Intelligent Autonomous Systems (ICoIAS) (pp. 171-174). IEEE.

Stefanovič, P., Štrimaitis, R., & Kurasova, O. (2020). Prediction of flight time deviation for Lithuanian airports using supervised machine learning model. Computational intelligence and neuroscience, 2020.

Sternberg, A., Soares, J., Carvalho, D., & Ogasawara, E. (2017). A review on flight delay prediction. arXiv preprint arXiv:1703.06118.

Suga, R. A., & Purwitasari, D. (2022). Flight Delay Prediction for Mitigation of Airport Commercial Revenue Losses Using Machine Learning on Imbalanced Dataset. In 2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM) (pp. 1-8). IEEE.

Twa, M. D., Parthasarathy, S., Roberts, C., Mahmoud, A. M., Raasch, T. W., & Bullimore, M. A. (2005). Automated decision tree classification of corneal shape. Optometry and vision science: official publication of the American Academy of Optometry, 82(12), 1038.

Viswanath, P., & Sarma, T. H. (2011). An improvement to k-nearest neighbor classifier. In 2011 IEEE Recent Advances in Intelligent Computational Systems (pp. 227-231). IEEE.

Weiss, G. M., & Provost, F. (2001). The effect of class distribution on classifier learning: an empirical study. Rutgers University.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., ... & Steinberg, D. (2008). Top 10 algorithms in data mining. Knowledge and information systems, 14, 1-37.

Yi, J., Zhang, H., Liu, H., Zhong, G., & Li, G. (2021). Flight delay classification prediction based on stacking algorithm. Journal of Advanced Transportation, 2021, 1-10.

Contact information

Huynh Hue Truc
University of Economics and Laws, Faculty of Information Systems
669, Highway 1, Quarter 3, Linh Xuan Ward, Thu Duc City, Ho Chi Minh City
E-mail: truchh22411c@st.uel.edu.vn
Phone number: +(84) 0974089548
ORCID: https://orcid.org/0009-0007-7030-800X

Le Nguyen Thanh Ty
University of Economics and Laws, Faculty of Information Systems
669, Highway 1, Quarter 3, Linh Xuan Ward, Thu Duc City, Ho Chi Minh City
E-mail: tylnt22411c@st.uel.edu.vn
Phone number: +(84) 0963162057
ORCID: https://orcid.org/0009-0009-9078-3224

Nguyen Le Phuong Anh
University of Economics and Laws, Faculty of Information Systems
669, Highway 1, Quarter 3, Linh Xuan Ward, Thu Duc City, Ho Chi Minh City
E-mail: anhnlp22411c@st.uel.edu.vn
Phone number: +(84) 0794619605
ORCID: https://orcid.org/0009-0002-6821-1499

Ho Song Tin
University of Economics and Laws, Faculty of Information Systems
669, Highway 1, Quarter 3, Linh Xuan Ward, Thu Duc City, Ho Chi Minh City
E-mail: tinhs22411c@st.uel.edu.vn
Phone number: +(84) 0377710618
ORCID: https://orcid.org/0009-0006-8689-5950

Do Thanh Danh
University of Economics and Laws, Faculty of Information Systems
669, Highway 1, Quarter 3, Linh Xuan Ward, Thu Duc City, Ho Chi Minh City
E-mail: danhdt22411@st.eul.edu.vn
Phone number: +(84) 0368657431
ORCID: https://orcid.org/0009-0009-2382-7477

Tran Duy Thanh (Phd.)
University of Economics and Laws, Faculty of Information Systems
669, Highway 1, Quarter 3, Linh Xuan Ward, Thu Duc City, Ho Chi Minh City
E-mail: thanhtd@uel.edu.vn
Phone number: +(84) 0987773061
ORCID: https://orcid.org/0000-0003-0680-9452