




Research Article

Missed Approach, a Safety-Critical Go-Around Procedure in Aviation: Prediction Based on Machine Learning-Ensemble Imbalance Learning

Afaq Khattak ¹, Pak-Wai Chan,² Feng Chen ¹, Haorong Peng ³,
and Caroline Mongina Matara^{4,5}

¹The Key Laboratory of Infrastructure Durability and Operation Safety in Airfield of CAAC, College of Transportation Engineering, Tongji University, 4800 Cao'an Road, Jiading, Shanghai 201804, China

²Hong Kong Observatory, 134A Nathan Road, Kowloon, Hong Kong, China

³Shanghai Research Center for Smart Mobility and Road Safety, Shanghai 200092, China

⁴Department of Civil and Resource Engineering, Technical University of Kenya, P.O. Box 52428-00200, Haile Sellasie Avenue, Nairobi, Kenya

⁵Department of Civil and Construction Engineering, University of Nairobi, P.O. Box 30197-00100, Harry Thuku Road, Nairobi, Kenya

Correspondence should be addressed to Feng Chen; fengchen@tongji.edu.cn

Received 18 November 2022; Revised 11 May 2023; Accepted 30 June 2023; Published 19 July 2023

Academic Editor: Yaolin Lin

Copyright © 2023 Afaq Khattak et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The final approach phase of an aircraft accounts for nearly half of all aviation incidents worldwide due to low-level wind shear, heavy downpours, runway excursions, and unsteady approaches. Adopting the missed approach (MAP) procedures may prevent a risky landing, which is usually executed in those situations, but it is safety-critical and a rare occurrence. This study employed machine learning-ensemble imbalance learning to predict MAPs under low-level wind shear conditions based on environmental and situational parameters. The models were developed using the 2017–2021 Hong Kong International Airport (HKIA) Pilot Reports (PIREPs). Initially, imbalance data were applied to machine learning models such as the random forest (RF), light gradient boosting machine (LGBM), and extreme gradient boosting (XGBoost), but these were unable to accurately predict the occurrence of MAPs. Then, these models were used as base estimators for ensemble imbalance learning methods, including the self-paced ensemble (SPE) framework, the balance cascade model, and the easy ensemble model. The SPE framework utilizing XGBoost as the base estimator performed better than other frameworks in terms of recall, F1-score, balanced accuracy, and geometric mean. Afterwards, SHAP was utilized to interpret the SPE framework with XGBoost as the base estimator. Results showed that low-level wind shear magnitude, runway orientation, and vertical location of low-level wind shear contributed most to MAPs. Runways 07C and 07R had the most MAPs. Most MAPs were initiated when low-level wind shear was within 500 feet of the ground. Strong tailwind triggered MAPs more than headwind. For aviation safety researchers and airport authorities, the framework proposed in this study is a valuable tool.

1. Introduction

At the final approach, phase of the aircraft, bad weather conditions, runway excursions, and unstabilized approaches are the primary causes of nearly half of all aviation accidents worldwide. An unsafe landing can be avoided by initiating a missed approach (MAP) protocol. Although the protocol is

in place to prevent unsafe landings, the complex and tough maneuvering operations and constrained time availability can increase potential dangers, especially in extreme weather conditions. Airport throughput and on-time performance of the flights are adversely affected, as is the workload of air traffic controllers and noise levels [1–4]. Most MAPs are made at low altitudes and slow speeds, and therefore, several

actions must be taken instantly, such as altering the altitude, thrust, and flight path of the aircraft to ensure no conflicts with nearby air traffic are encountered as a result. Since safe and effective MAP execution depends on both the pilot and the air traffic controller, their roles are crucial.

While wind shear is a local meteorological phenomenon, it may be associated with larger-scale phenomena like thunderstorms and cold fronts on the mesoscale or spatial and temporal scales. The International Civil Aviation Organization (ICAO) specifies low-level wind shear as a sustained change of 15 knots or more in headwind or tailwind within 1600 feet above ground level, and this is a crucial weather phenomenon from an aviation perspective. It varies the lift of the aircraft, which can cause it to deviate from its intended approach path, having to put both incoming and departing aircrafts in hazard [5, 6]. This can ultimately lead to the execution of MAPs (Figure 1).

Low-level wind shear during the final approach phase can have two potentially catastrophic effects on the aircraft, and pilots feel enormous pressure: (1) low-level wind shear can destabilize the glide path, and (2) it can induce the approach speed to veer away from the pre-defined threshold [7]. The effects of decreasing and increasing shear from the headwind on an aircraft are depicted in Figure 2 under the assumption of no pilot intervention and a standard instrument landing system (ILS) approach with a glide slope of 3 degrees. The first case, depicted in Figure 2(a), is the decreasing headwind experienced by an incoming aircraft. The aircraft's airspeed (its speed relative to the air flow around it) decreases as it gets closer to the ground, which reduces lift and typically results in a steeper descent angle due to the momentary force imbalance. If this occurs, the plane may crash land short of the runway. The pilot in this case may increase the level of throttle to go-around (MAP) and try for a second attempt. With the same glide slope, but now with a rising headwind (Figure 2(b)) (3 degrees), the aircraft's airspeed increases in relation to the surrounding air flow, creating more lift that tends to result in a flatter angle of descent or even a climb. In this case, the pilot can choose to abort the landing and proceed with a MAP.

In either case, MAP may be activated, and pilots and controllers must be able to collaborate in order to make MAP decisions based on anticipated weather conditions, which is essential for the safe approach of aircraft.

Numerous works of research have examined the criteria and variables that contribute to MAPs occurrences. Numerous methods have been tried by researchers in the past for predicting and modeling MAPs based on a wide variety of input factors. Using statistical and machine learning models, Zaal et al. [8] assessed the influence of environmental factors on MAPs. They noticed that visibility, wind speed, and localizer deviation significantly impact MAP decision-making. Chou et al. [9] used machine learning models to analyze the causes of MAPs. They observed that the categorical boosting model performed better than other models and that factors such as visibility, wind speed, and pressure were among the most important causes of MAPs. Donavalli et al. [10] utilized a statistical method to evaluate

the weather factors influencing MAPs. The results indicated that thunderstorms and winds exceeding 29 miles per hour significantly increase the likelihood of MAP. However, visibility did not indicate a significant impact. Numerous occurrences of MAPs were attributed to adverse weather conditions, particularly convective storms on the approach path, according to Proud [11] who employed and compared various MAPs detection methods.

In addition, several researchers have emphasized the modeling of MAPs resulting from an unstable approach and a change in runway configuration. Using the sparse variation Gaussian process (SVGP) model, Singh et al. [12] developed a framework to demonstrate the aircraft's 4D trajectories during the final approach phase. The experimental analysis revealed that SVGP delivers an interpretable probabilistic bound for the parameters of aircrafts that can assess deviation and detect anomalies in real time. To predict the occurrences of MAPs, the authors of [13] used a logistic regression model based on principal component analysis. They found that factors such as flight spacing, approach stability, departure air traffic, and ceiling have a significant impact on MAPs. A number of researchers have also investigated MAPs, focusing on the performance and behavior of pilots and air traffic controllers. Causse et al. [14] discovered that the unpleasant psychological effects are tied to the MAPs. The uncertainty of a decision's outcome temporarily compromises pilot decision-making and cognitive performance. During the execution of MAPs, the authors of [15] analyzed the anomalies in pilot flying performance, including flight path deviations and visual scanning behaviors. According to Jou et al. [2], situational unawareness by air traffic controllers was the major cause of MAP occurrences. Kennedy et al. [16] observed that the age and experience of air traffic controllers have significant effects on MAP decision-making.

When compared to statistical models, machine learning models' predictions are less translucent because of their black-box nature, despite the fact that machine models are more flexible. Equally important for a more accurate evaluation of the model's effectiveness is a comprehensive explanation of how the model would actually work. Traditional methods for interpreting the outcomes of machine learning models involved feature ranking analysis, such as importance scores based on permutation. Even though the feature ranking interpretation can rank the significance of the different factors, it cannot exhibit interactions among factors or how much each factor influences the model's prediction. Recent studies have utilized posthoc SHapley Additive Explanations (SHAP) interpretation tool, which is based on the notion of game theory, to assess the effect of different factors on the outcome [17]. Using SHAP in conjunction with machine learning models, it is possible to assess the importance and relative contribution of various factors on the prediction. They have been used in a variety of fields, such as the safety assessment of infrastructure projects [18, 19]; clinical, medicine, and healthcare modeling [20–27]; transportation and traffic safety [28–38]; finance and economics risk analysis [39–42]; and offshore safety analysis [43].

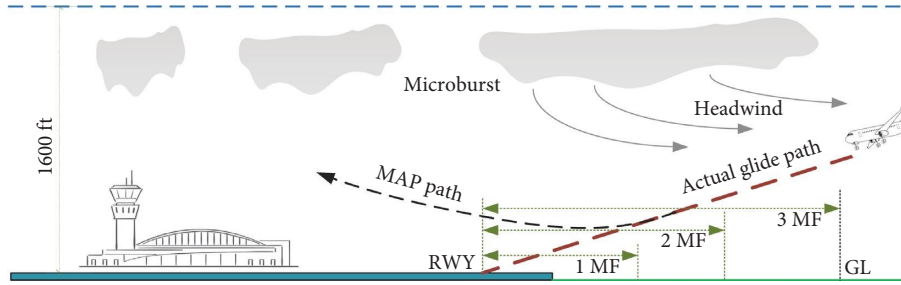


FIGURE 1: Location of low-level wind events that occur near airport runways.

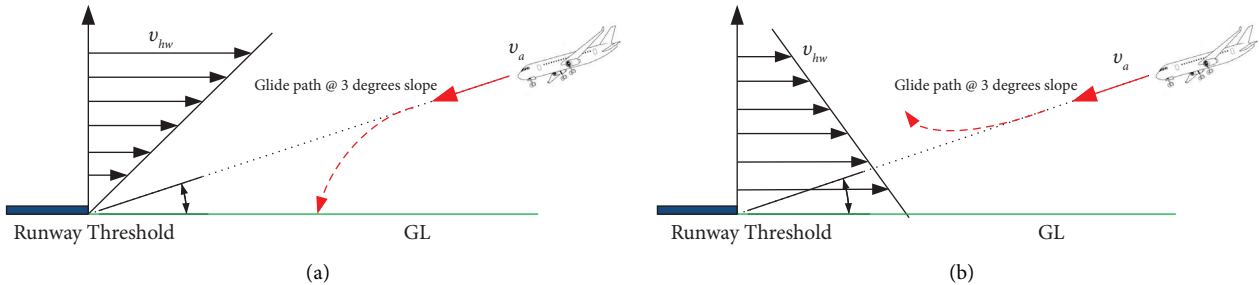


FIGURE 2: Low-level wind shear adverse and potentially hazardous effects on approaching aircraft. (a) Final approach with decreasing shear from headwind. (b) Final approach with increasing shear from headwind.

Although MAPs are exceptionally rare, being able to foresee when they might occur in response to extreme weather is crucial. Prior research has provided useful insights into the numerous factors that contribute to MAPs; however, to the best of our knowledge, no studies have accounted for the impact of low-level wind shear and used cutting-edge ensemble imbalance learning strategies in conjunction with SHAP interpretation tool, and this area remains largely unexplored in the existing literature. This study aims to quantify the factors that contribute to Hong Kong International Airport (HKIA)-based MAPs triggered by extreme weather and situational factors by employing three state-of-the-art machine learning-ensemble imbalance learning techniques, namely self-paced ensemble (SPE) framework [44], balanced cascade model [45], and easy ensemble model [45] with three state-of-the-art machine learning classifiers as base estimators, namely random forest [46], light gradient boosting machine (LGBM) [47], and extreme gradient boosting (XGboost) [48].

The intent of this paper has three aspects. The outcomes of our study would help pilots, air traffic controllers, and aviation policymakers truly comprehend the factors that raise the likelihood of a MAP occurring. Second, MAPs and the conditions that lend support to them may be regarded as unusual; therefore, analyzing the causes of MAP occurrences can help identify specific measures that can be taken to lessen the frequency with which MAPs are executed. The number of MAPs can be reduced by implementing mitigation strategies such as adjusting protocols, flight training, and technical equipment designs.

To that end, we first examine the HKIA-based pilot reports (PIREPs) to identify the causes of the MAPs. When talking about pilot reports in civil aviation, the acronym

PIREP is commonly used. Pilots who experience MAP and other forms of rough weather are directed to contact air traffic controllers via PIREPs. Low-level wind shear conditions (the magnitude, altitude, and encounter location from the runway threshold, as well as its causes), type of aircraft (narrow or wide-body), precipitation aspect (clear sky or rainfall), flight (HKIA inbound international or domestic), arrival (approaching) runway (07L, 07R, 25L, and 25R), and temporal factors including season of the year and time of day can all contribute to MAPs. Upon establishing which factors are related to MAPs, we build machine learning-ensemble imbalance learning models to test their efficacy. The SHAP interpretation method is then used to determine how much each factor contributes to MAP occurrence.

The remainder of this paper is structured as follows: Section 2 details the study’s methodology, including its data source and analysis phases, machine learning-ensemble imbalance learning models, and posthoc SHAP explanation approach. In Section 3, we present the performance assessment of machine learning-ensemble imbalance learning models as well as posthoc explanation results via SHAP. In Section 4, we summarize our findings and discuss their relevance.

2. Method and Data

In this study, machine learning-ensemble imbalance learning techniques, including self-paced ensemble (SPE), balance cascade, and easy ensemble, with state-of-the-art machine learning models as base estimators, i.e., random forest (RF), light gradient boosting machine (LGBM), and extreme gradient boosting machine (XGBoost), were used

to predict the MAPs at HKIA under low-level wind shear conditions caused by sea breeze and gust front. The data used for the research come from the HKIA-based pilot flight reports (PIREPs) during the time period 2017–2021. The PIREP data were first preprocessed for the missing values and removed irrelevant information, such as reported hazardous weather conditions during takeoff. Then, for the development and assessment of the machine learning-ensemble imbalance learning techniques, the data were split into training (70%) and testing (30%) datasets. In addition to model development, hyperparameters of LGBM, XGBoost, and RF (base estimators) were tuned employing Bayesian optimization in combination with 10-fold cross-validation. Machine learning-ensemble imbalance learning techniques with fine-tuned base estimators were used for performance evaluation and model comparison. To further interpret the effect of each factor on the MAP occurrences, the optimal model was used to calculate Shapley additive values using the posthoc SHAP analysis tool, which provides factor importance and contribution analysis as well as factor interaction analysis. The overall operational conceptual framework proposed in this study is shown in Figure 3.

2.1. Study Location. The HKIA (IATA code: HKG, ICAO code: VHHH) is the main airport of Hongkong, which is located on the artificial island of Lantau off the subtropical coast of the Chinese mainland (Figure 4). Hong Kong's regular convective weather consists of tropical cyclones and the southwest monsoon. In addition to causing flight delays, the convective weather also tends to bring thunderstorms and downpours to the area. This airport is one of the airports most susceptible to low-level wind shear. Innumerable observation-based and simulation studies demonstrated that HKIA's complex land-sea contrast and intricate orography are favorable conditions for the emergence of low-level wind shear [49]. Approximately every 400–500 flights, a significant low-level wind shear event takes place [50]. From 1998, when HKIA first opened, through 2015, 97% of reports indicated level-level wind shear between 15 and 25 knots [51].

2.2. Data Processing from Pilot Reports (PIREPs). The aviation sector generally abbreviates pilot reports as PIREPs. Pilots alert air traffic controllers when they encounter potentially dangerous weather conditions. Turbulence, icing, and the status of the flight path are typical aspects covered in PIREPs. Since HKIA is particularly prone to low-level wind shear, specific information about its occurrence is also provided in HKIA-based PIREPs. This includes the type of aircraft and the flight, the vertical low-level wind shear encounter locations (such as 200 ft, 500 ft, and 1000 ft), the horizontal encounter locations of low-level wind shear from the threshold of runway (such as 1 MF, 2 MF, and 3 MF), the magnitude of low-level wind shear, and the date and time of the occurrence of low-level wind shear. The pilot may also report MAP in the HKIA-based PIREPs if it is executed due to low-level wind shear caused by sea breeze, or gust front, as

shown in Table 1. It should be noted that the shear from a headwind and a tailwind is represented by plus and minus signs.

From 2017 to 2021, PIREPs depicted a total of 1731 occurrences of low-level wind shear on both outbound and inbound flights. However, 1388 (80.418%) were indicated by HKIA inbound flights, while 343 (19.81%) were confirmed by HKIA outbound flights out of a total of 1731 occurrences. In this research, we focused on the factors that contribute to MAP during low-level wind shear events, so we only kept data from arriving flights and eliminated the data that was reported by outbound flights. As this study considers only low-level wind shear-induced MAPs, the data reported by approaching flights were kept in the dataset while that from outbound flights were excluded. In addition, the dataset was preprocessed to clean up the extraneous data. Once redundant and erroneous data were excluded, a dataset containing 765 low-level wind shear occurrences in which MAPs have been observed 184 times was achieved. Furthermore, a binary classification problem was setup by labeling all MAPs (the minority class) as “1” and all APs (approaches being the majority class) as “0.” Table 2 lists all the variables and provides descriptions of each.

2.3. Machine Learning-Ensemble Imbalanced Learning Techniques. In this study, three ensemble imbalanced learning techniques were proposed for the prediction of MAP occurrence: the self-paced ensemble (SPE) framework, the balance cascade model, and the easy ensemble. These ensemble-imbalanced learning techniques combine resampling and ensemble learning models. We incorporated LGBM, XGBoost, and RF as the base estimators to evaluate the performance of ensemble imbalanced learning techniques in MAP prediction. The details of each ensemble imbalanced learning technique are provided below while the details of base estimators are provided in Appendix A-1.

2.3.1. Easy Ensemble Approach. Easy ensemble is a straightforward approach and is known for its high performance. It is essentially an integration of an under-sampling technique and a base estimator. First, bootstrap samples several subsets of a majority class independently and then builds and learns a base estimator for each subset. The final robust ensemble is built by adding all of the generated classifiers. The working procedure of EasyEnsemble is described as follows in Algorithm 1) [45]:

2.3.2. Balance Cascade Approach. In this approach, multiple balanced subsets of data are generated, and a weak classifier is learned for each subset. This approach reduces the majority of class training sets at each step by removing all correctly classified instances. In two ways, it differs from EasyEnsemble. First, the weights are modified in accordance with the false positive rates that a classifier must achieve. Second, instances that have been correctly classified are eliminated. This sequential dependence focuses primarily on

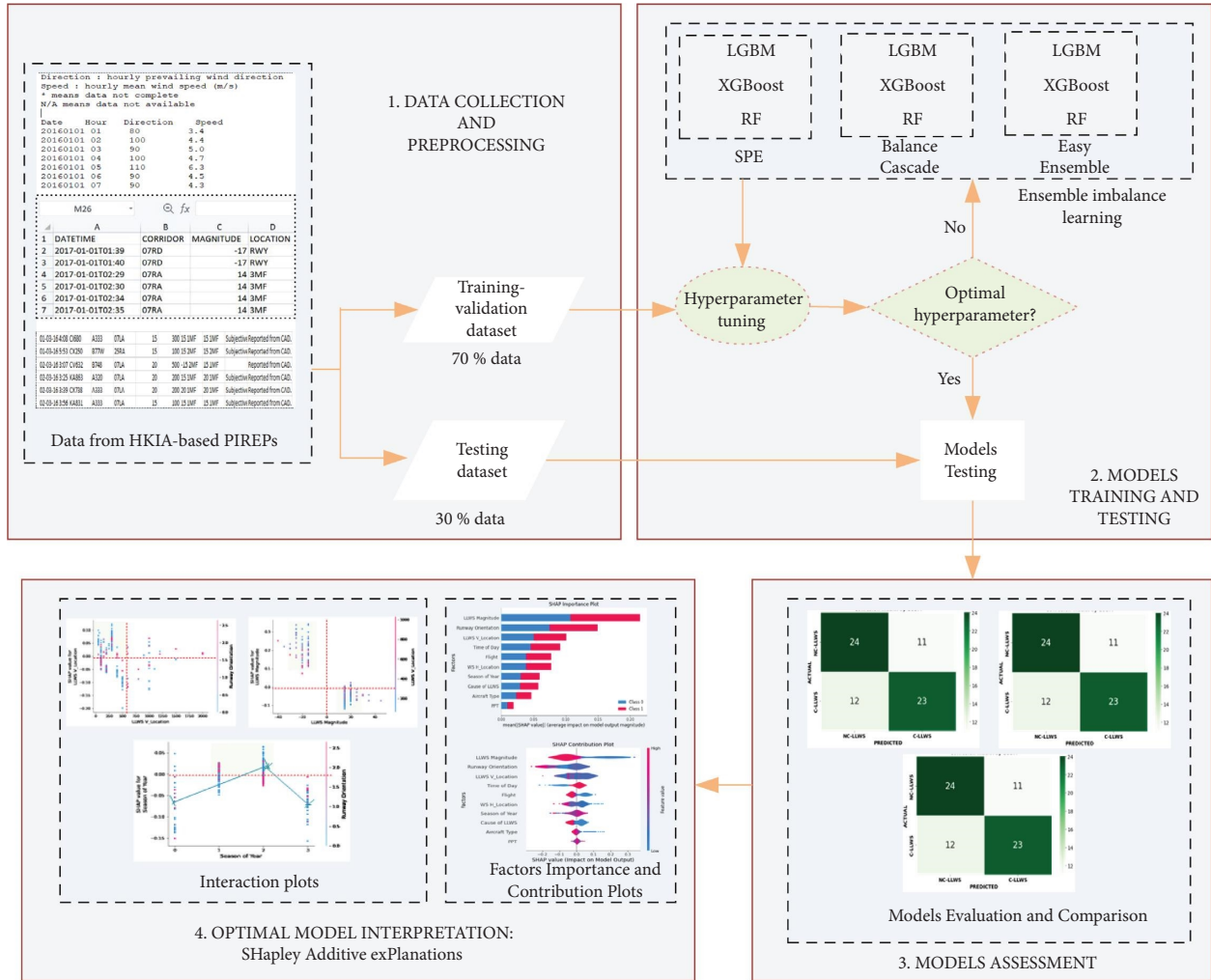


FIGURE 3: Framework for low-level wind shear-induced MAP prediction and interpretation.



FIGURE 4: Hong Kong international airport and Lantau Island.

minimizing redundant information in the majority class. Following Algorithm 2) [45] is a description of the balance cascade’s operational procedure.

2.3.3. Self-Paced Ensemble Approach. In this study, we also proposed a newly developed SPE framework [44], which is an ensemble-based imbalance learning framework. The basic concept regarding hardness harmonize and self-paced factor is highlighted as follows, which is then followed by the SPE algorithm.

(1) *Hardness Harmonize.* Each instance from the majority class is sorted into one of the “ β ” bins according to its hardness value. Every k^{th} bin indicates a distinct degree of hardness. The dataset is then rendered more equitable by under-sampling instances from the majority class while preserving the same total hardness contribution in each bin. Harmonize is the term used to describe this approach in the “gradient-based optimization” literature. The initial iteration employs a similar strategy to harmonize the hardness. Nevertheless, hardness harmonize is not always used in all iterations. The main cause of this is that as the ensemble

TABLE 1: Sample data obtained from the PIREPs of HKIA.

Date	Time	Landing runway	Flight	Type of aircraft	Magnitude of LLWS	H-location of LLWS	V-location of LLWS	PPT	LLWS causes	MAP
2017-06-15	2:52	25RA	CX616	A333	-30 knots	2 MF	600 ft	No	Gust front	Yes
2018-04-21	1:15	07LA	KA451	A333	20 knots	3 MF	1000 ft	Yes	Sea breeze	Yes
—	—	—	—	—	—	—	—	—	—	—
2019-04-12	8:19	07LA	HX337	A333	20 knots	2 MF	300 ft	No	Gust front	Yes
—	—	—	—	—	—	—	—	—	—	—
2021-09-04	8:14	25LA	CX445	A359	15 knots	RWY	200 ft	No	Sea breeze	Yes

Notes. LLWS: low-level wind shear; H-location: horizontal encounter location; V-location: vertical encounter location; PPT: precipitation.

classifier learns to fit the training set, the number of trivial instances rises. Consequently, there remains a significant amount of trivial instances after merely harmonizing the hardness contribution. These less instructive instances significantly slow down later iterations of the learning process. Instead, “self-paced factors” have been developed to enable under-sampling to be performed at a user-defined pace.

(2) *Self-Paced Factor*. Specifically, the sample probability of bins with a huge population is gradually reduced after harmonizing the hardness contribution of each bin. A self-regulating factor (Ω) sets the rate of decay. In the presence of a large Ω , the simple hardness contribution harmonize takes a back seat to a heightened focus on the harder samples. Outliers and noise have less of an effect on the model’s ability to generalize in the early stages of training because the framework prioritizes informative borderline samples. To avoid over-fitting, the framework keeps a respectable fraction of trivial (high confidence) instances as the “skeleton,” even in later iterations when Ω is large. In Algorithm 3, we examine how the SPE framework works in detail. It is important to note that in each iteration, the hardness value is updated to choose the most useful data instances for the current ensemble. The development of the self-paced factor has been modulated by means of the tangent function. As such, in the first iteration, the self-paced factor is zero, and in the last iteration, it is infinite (see Algorithm 3 [44]).

2.4. Performance Metrics. Overall classification accuracy is a common model performance metric that is calculated as the ratio of the total number of correct predictions to the total number of predictions. This metric tends to favor the majority class; therefore, it could be misleading if the data were not evenly distributed. This precludes the use of classification accuracy as a performance metric. Several other performance metrics, besides accuracy, can be used to deal with this issue.

For the binary classification task, assume that n_1 denotes the sample size of the majority class and n_2 represents the minority class. The total number of records of both majority

and minority classes in the training set is n . The binary classifier determines the likelihood that each instance is positive or negative. Thus, it produces four distinct results: true positive (Δ_p), true negative (Δ_n), false positive (∇_p), and false negative (∇_n), as illustrated in the confusion matrix (Figure 5). Some important measures, including balanced accuracy, recall, precision, F1-score, and the geometric mean (G-mean), can be determined from the results of the confusion matrix, which are shown by the following equations:

$$\text{Balanced accuracy} = \frac{(\Delta_p + \Delta_n/\Delta_p + \nabla_n + \Delta_n + \nabla_p)}{2}, \quad (1)$$

$$\text{Precision} = \frac{\Delta_p}{\Delta_p + \nabla_p}, \quad (2)$$

$$\text{Recall} = \frac{\Delta_p}{\Delta_p + \nabla_n}, \quad (3)$$

$$\text{F1-score} = \frac{\Delta_p}{\Delta_p + (1/2)(\nabla_p + \nabla_n)}, \quad (4)$$

$$\text{G-mean} = \sqrt{\left(\frac{\Delta_p}{\Delta_p + \nabla_n}\right)\left(\frac{\Delta_n}{\nabla_p + \Delta_n}\right)}. \quad (5)$$

2.5. Posthoc Interpretation of Ensemble Imbalance Learning Model. In order to interpret the ensemble imbalance learning model, the SHAP interpretation tool is one of the posthoc explanation tools that were developed by Lundberg and Lee [17]. The fundamental concept underlying the interpretation by the SHAP tool is to calculate the marginal contribution of each factor to the model’s outcome and to interpret the results in both a global and a local context. A prediction value is computed for each instance during the training of the model, and the SHAP value corresponds to the value given to each factor in the instance. Equation (6) is used to calculate each factor’s contribution, which is represented by the Shapley value.

TABLE 2: Factors description with their label coding.

Factors	Descriptions	Type of data	Label coding
MAP	Missed approach/go-around	Discrete	1: if MAP is executed, 0: if approach occurs
	Type of the flight	Discrete	1: if the inbound flight is an international flight, 0: if the inbound flight is Hong Kong-based carrier
Vehicle-specific	Type of inbound aircraft	Discrete	1: if the inbound aircraft is a wind-body aircraft, 0: if the inbound aircraft is a narrow-body aircraft
Runway-specific	Assigned arrival runway for inbound flight	Discrete	1: if arrival runway is 07R, 2: if the arrival runway is 25C, 3: if arrival runway is 25L, 0: if arrival runway is 07C
Environment-specific	LLWS magnitude	Continuous	—
	LLWS H-location	Discrete	1: if horizontal encounter location of low-level wind shear is 1 nautical mile from runway threshold, 2: if horizontal encounter location of low-level wind shear is 2 nautical miles from runway threshold, 3: if horizontal encounter location of low-level wind shear is 3 nautical miles from runway threshold, 0: if low-level wind shear occurs at the runway threshold
	LLWS V-location	Continuous	—
	Cause of LLWS	Discrete	1: if there is sea breeze-induced wind shear, 0: if there is gust front-induced wind shear
	Precipitation	Discrete	1: if there is precipitation, 0: otherwise
	Time of day	Discrete	1: if the MAP occurs during day time (08:00 to 18:59), 0: if the MAP occurs during night time (19:00 to 07:59)
Temporal-specific	Seasons	Discrete	1: if MAP event occurs in the spring season, 2: if MAP event occurs in the summer season, 3: if MAP event occurs in the autumn season, 0: if MAP event occurs in the winter season

- (1) **Input:** Minority class in the training data (A), majority class in the training data (B), number of subsets T to sample from majority class instances set N , and the number of iterations k_j required to learn the “base estimator” H_j .
- (2) $j \leftarrow 0$
- (3) **repeat**
- (5) $j \leftarrow j + 1$
- (6) Randomly sample a subset B_j from B , $|B_j| = |A|$
- (7) Learn H_j by using A and B_j . H_j is the base estimator with k_j weak classifiers $h_{j,l}$ and the corresponding weights distribution $\omega_{j,l}$. The ensemble’s threshold is Δ_j , i.e., $H_j(x) = \text{sgn}(\sum_{l=1}^{k_j} \omega_{j,l} h_{j,l}(x) - \Delta_j)$,
- (8) Until $j = T$
- (9) **Output:** The robust ensemble model:
- (10) $H(x) = \text{sgn}(\sum_{j=1}^T \sum_{l=1}^{k_j} \omega_{j,l} h_{j,l}(x) - \sum_{j=1}^T \Delta_j)$,

ALGORITHM 1: EasyEnsemble Approach.

- (1) **Input:** Minority class in the training data (A), majority class in the training data (B), number of subsets T to sample from the majority class instances set, and the number of iterations k_j required to learn the base estimator H_j .
- (2) $j \leftarrow 0$
- (3) The false positive rate $\nabla_{\text{pr}} \leftarrow \sqrt{(|A|/|B|)}$ that model H_j has to achieve. The ∇_{pr} is the basically misclassification rate of a majority class instances to the minority class
- (4) **repeat**
- (5) $j \leftarrow j + 1$
- (6) Sample a subset B_j randomly from B such that $|B_j| = |A|$
- (7) Learn the model H_j by using minority class set A and subset B_j . H_j is the base estimator with weak classifiers $h_{j,l}$ and corresponding weights distribution $\omega_{j,l}$. The ensembles’ threshold is Δ_j , i.e., $H_j(x) = \text{sgn}(\sum_{l=1}^{k_j} \omega_{j,l} h_{j,l}(x) - \Delta_j)$,
- (8) Adjust Δ_j such that false positive rate of H_j is ∇_{pr}
- (9) Discard all the instances from B that are classified correctly by H_j
- (10) $j = T$
- (11) **Output:** Single robust ensemble model:
- (12) $H(x) = \text{sgn}(\sum_{j=1}^T \sum_{l=1}^{k_j} \omega_{j,l} h_{j,l}(x) - \sum_{j=1}^T \Delta_j)$,

ALGORITHM 2: Balance Cascade Approach.

- (1) **Input:** Training dataset $(x_k, y_k)_i^n$, hardness function (∂) , total number of bins (β) , base estimator (ζ) , the number of base estimators (\forall) , minority class in the training data (A), majority class in the training data (B)
- (2) **Initialize:** With the subsets of majority class (B') and minority class (A), train the base estimator (ζ) by utilizing random under-sampling approach such that $|B'| = |A|$
- (3) **for** $k = 1$ to n **do**
- (4) Ensemble of the base estimators $\mathbb{F}_k(x) = 1/k \sum_{k=0}^{k-1} \zeta_k(x)$
- (5) The majority class dataset is separated into β bins with regards to $\partial(x, y, \mathbb{F}_k)$: $(b_1, b_2, \dots, b_\beta)$
- (6) The mean hardness contribution can be obtained in the any i^{th} bin as $\partial_i = \sum_{s \in b_i} \partial(x_s, y_s, \mathbb{F}_i) / |b_i|$, $i = 1, 2, \dots, \beta$
- (7) The self-paced factor has been updated as $\Omega = \tan(k\Pi/2\zeta)$,
- (8) For the i^{th} bin, non-normalized sampling weight can be obtained as $\theta_i = 1/\partial_i + \Omega$
- (9) From the i^{th} bin, perform the under-sampling with $\theta_i / \sum_m \theta_m |A|$ instances
- (10) Using newly under-sample data subset, train ζ_i
- (11) **End**
- (12) **Return** Final robust ensemble $\mathbb{F}(x) = 1/\forall \sum_{m=1}^{\forall} \zeta_m(x)$,

ALGORITHM 3: Self-Paced Ensemble.

$$\varphi_i = \sum_{Y \subseteq \Pi \setminus \{i\}} \frac{Y!(n-|Y|-1)!}{n!} [f(Y \cup \{i\}) - f(Y)], \quad (6)$$

where φ_i is the contribution of i^{th} input factor from the dataset. Π is the set of all the input factors from the dataset. Y is the subset of given predicted factors from the dataset.

$f(Y_i)$ and $f(Y)$ are the outcomes with and without i^{th} factor from the dataset, respectively.

Using an additive factors imputation strategy, the SHAP tool generates an interpretable machine learning model. The outcome of the model is represented as a linear sum of all the input factors, as shown by the following equation:

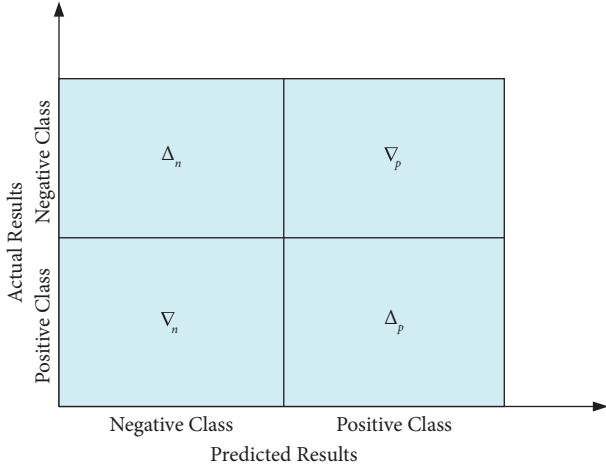


FIGURE 5: Confusion matrix.

$$g(z') = \varphi_0 + \sum_{i=1}^{\Lambda} \varphi_i z', \quad (7)$$

where $z' \in \{0, 1\}^{\Lambda}$. If a factor is supplied $z' = 1$, else $z' = 0$. Λ is the amount of the input factors that are supplied. φ_0 is the base value.

In this research, the optimal ensemble imbalanced learning technique was interpreted using the posthoc SHAP analysis tool, and the most important factors likely to cause MAPs were evaluated. Additionally, the SHAP tool in this study performed an analysis of the factor interaction.

3. Results and Discussion

The ensemble imbalance learning techniques with various base estimators were used in conjunction with HKIA-based PIREPs to address the imbalance data problem and predict the occurrence of MAP under low-level wind shear conditions. Table 3 provides the aggregate statistics for all of the factors in the HKIA-based PIREPs. We used Pearson correlation analysis to look for correlations between the PIREPs' various factors. Based on Figure 6, a weak to moderate correlation is indicated by a Pearson's correlation coefficient with an absolute value between 0.019 and 0.62. Although we found a negative Pearson correlation coefficient of -0.62 between causes of low-level wind shear and precipitation, we decided not to rule them out of further modeling due to the moderate nature of the relationship between the two factors. Both of these aspects are environment-specific and could have a major effect if incorporated into the model. It is pertinent to note that the analysis was conducted within the Python programming environment. The Python codes can be found in Appendix B.

To commence the modeling process, the PIREPs data were divided into two subsets: 70% of the data were used to train the ensemble imbalance learning models, while the remaining 30% were used for ensemble imbalance learning models performance evaluation. Before developing the models, hyperparameters tuning of machine learning models (base estimators of ensemble imbalance learning) were done,

which was a critical step and can influence generalization capability, prevents over-fitting, and reduces model complexity. Bayesian optimization strategy was employed to obtain the optimal hyperparameters for the base estimators by maximization of "G-mean." A 10-fold cross-validation approach was used in conjunction with Bayesian optimization that randomly splits the training set into 10 subsets (each time, nine subsets were being used for training and one for testing). Table 4 shows the hyperparameters list of different base estimators of ensemble imbalance learning models with the search space and optimal values.

3.1. Performance Assessment of Base Estimators of Ensemble Imbalance Learning Models. In this study, the positive and negative classes were referred to, respectively, as MAP and AP. Initially, each base estimator (LGBM, RF, and XGBoost) was evaluated separately, without the use of an ensemble imbalance learning model. Using the testing dataset, the confusion matrix (Figure 7) has been built, and the required performance metrics of balanced accuracy, precision, recall, F1-score, and G-mean have been derived. It is crucial to recognize that, unlike the AP class, the MAP class is a minority class and that we are more concerned with the correct classification of this class.

Only 28 out of 61 instances that were actually MAPs were correctly classified by the LGBM model based on the testing data set, as shown in Figure 8(a). Meanwhile, 33 instances were incorrectly classified, resulting in a recall value of 46.24%. Despite the fact that 181 out of 194 instances of the majority class AP were correctly classified, we do not concentrate on this class because it is not of interest to us. As a result, the LGBM model was able to classify the majority of instances with a higher degree of precision. However, it was unable to effectively classify minority instances, which is the parameter that we are interested in. The overall balanced accuracy was found to be quite low, equaling 41.40%, while the F1-score was 55.43% and the G-mean was 65.33%. Following the evaluation of one more cutting-edge XGBoost model, a confusion matrix was obtained. Figure 8(b) demonstrates that 23 out of the 58 actual MAP instances were correctly predicted as MAPs, while 35 were incorrectly predicted as APs, resulting in a recall value of 40.21%. The results obtained from the XGBoost model were marginally lower than those obtained from the LGBM model, with an overall imbalanced classification accuracy of equal to 34.32% and an F1-score of 46.65%. The G-mean value was 60.41%. Afterwards, the RF model was also evaluated, and a confusion matrix was obtained. According to the standalone RF model, Figure 8(c) demonstrated that only 27 instances of actual MAPs were correctly predicted as MAPs, while 31 were incorrectly predicted as AP. This case also had a higher percentage of incorrect classifications, yielding a recall value of 46.75%, precision of 66.42%, F1-score of 55.46%, balanced accuracy of 41.30%, and G-mean of 66.04% as shown in Table 5.

In all these three state-of-the-art machine learning models as base estimators of ensemble imbalance learning models, the percentage of correct classification

TABLE 3: Descriptive statistics of HKIA-based PIREPs factors.

Factors	Descriptions	Statistics			
		Mean	Standard deviation	Min	Max
Vehicle-specific	Flight	0.554	0.497	0	1
	Type of the aircraft	0.741	0.434	0	1
Runway-specific	Assigned runway for arrival	0.897	1.002	0	3
Environment-specific	LLWS magnitude (-/+)	17.17/-19.23	3.86/4.85	-15/15	-40/45
	LLWS H-location	1.473	0.896	0	3
	LLWS V-location (ft)	335.52	304.723	15	2000
	Cause of LLWS	0.457	0.492	0	1
	Precipitation	0.530	0.497	0	1
Temporal-specific	Time of day	0.623	0.482	0	1
	Seasons	1.551	0.865	0	3



FIGURE 6: Pearson correlation matrix.

TABLE 4: Hyperparameters tuning of the base estimators.

Base estimators	Hyperparameters	Range	Optimal values
LGBM	$n_estimators$	(100-1500)	900
	num_leaves	(30-100)	38
	Learning rate	(0.001-0.2)	0.07
	reg_lambda	(1.1-1.5)	1.24
	reg_alpha	(1.1-1.5)	1.18
RF	$n_estimators$	(50-1000)	1041
	max_depth	(2-15)	7
XGBoost	$n_estimators$	(100-1500)	1105
	num_leaves	(30-100)	46
	Learning rate	(0.001-0.2)	0.05
	reg_lambda	(1.1-1.5)	1.41
	reg_alpha	(1.1-1.5)	1.27

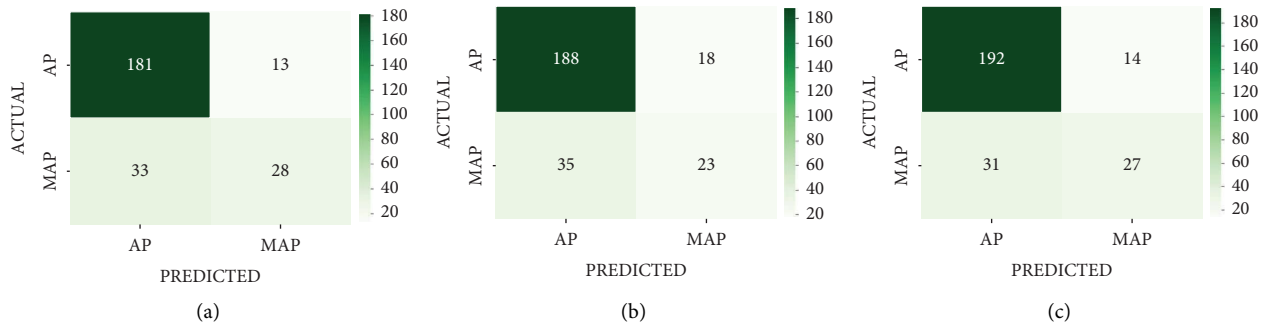


FIGURE 7: Confusion matrix: (a) LGBM; (b) XGBoost; (c) RF.

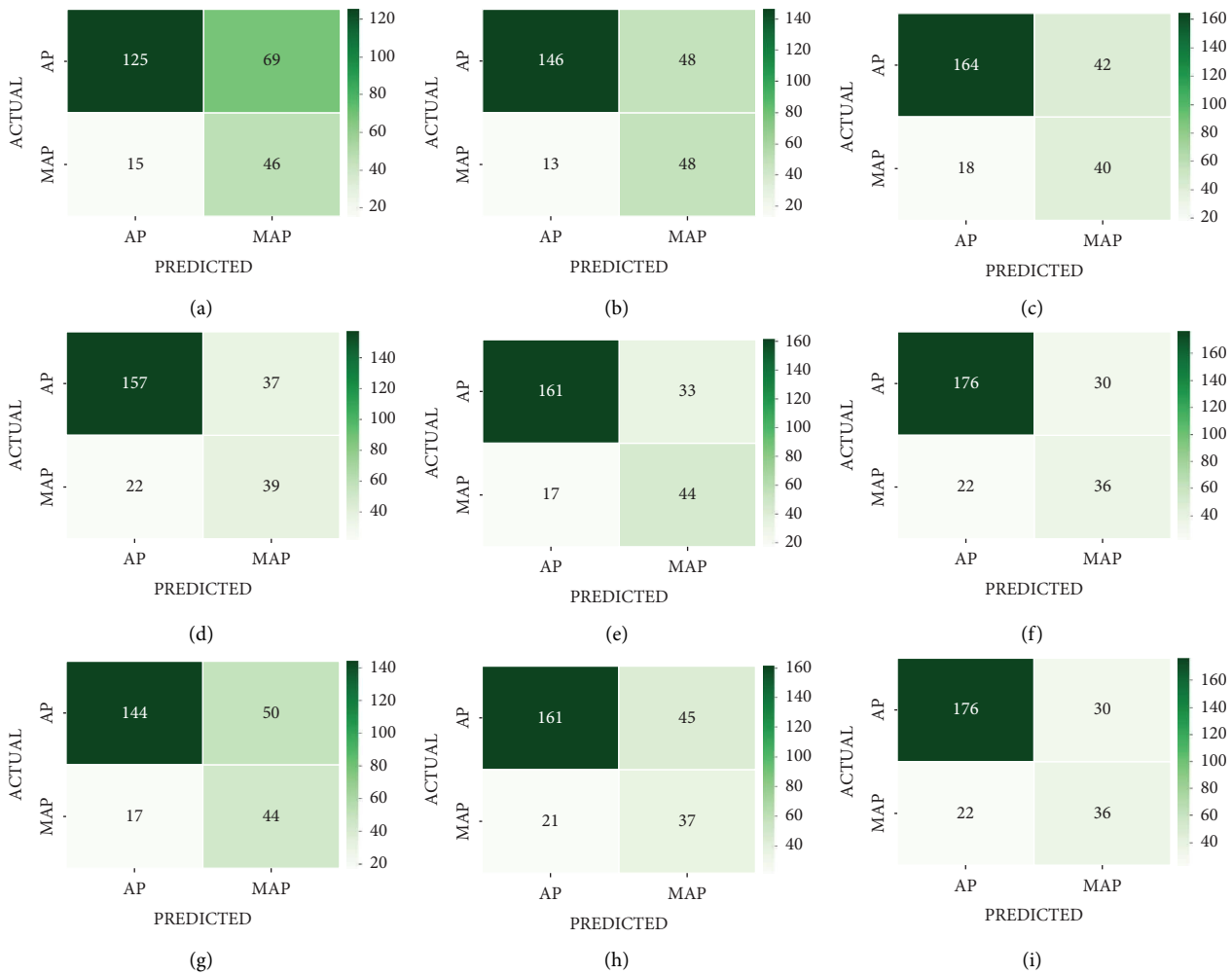


FIGURE 8: (a) Confusion matrix by SPE with LGBM as base estimator; (b) confusion matrix of SPE with XGBoost as base estimator; (c) confusion matrix of SPE with RF as base estimator; (d) confusion matrix by balance cascade with LGBM as base estimator; (e) confusion matrix of balance cascade with XGBoost as base estimator; (f) confusion matrix of balance cascade with RF as base estimator; (g) confusion matrix by easy ensemble with LGBM as base estimator; (h) confusion matrix of easy ensemble with XGBoost as base estimator; (i) confusion matrix of easy ensemble with RF as base estimator.

of MAP was quite low and cannot be used for the classification of MAPs as a standalone model. The better among these three machine learning models was RF; however, still as a standalone classifier, it fails to show a promising result.

3.2. Performance Assessment of Ensemble Imbalanced Learning Models with Different Base Estimators. Although we have employed state-of-the-art machine learning models for the prediction and classification of MAPs, however, we observed that with fined tuned LGBM, XGBoost, and RF, the

TABLE 5: Performance assessment of base estimators.

Metrics	Base estimators		
	LGBM	XGBoost	RF
Precision (%)	68.72	56.23	66.42
Recall (%)	46.24	40.21	46.75
F1-score (%)	55.43	46.65	55.46
Balanced accuracy (%)	41.40	34.32	41.30
G-mean (%)	65.19	60.41	66.04

minority class MAP was poorly classified. In order to improve the accuracy of the correct classification of minority class MAP, the SPE, balanced cascade, and easy ensemble models were used with LGBM, XGBoost, and RF as their base estimators. It is pertinent to mention that we have used those models as base estimators with their optimal hyperparameters, which we previously obtained via Bayesian optimization. First, LGBM, XGBoost, and RF were used as base estimators for the SPE framework, and then, LGM, XGBoost, and RF were used as base estimators for the balance cascade and finally for the easy ensemble model.

Figure 8 depicts the SPE, balance cascade, and easy ensemble models' confusion matrices; Table 6 pulls the performance indicators from these matrices. In the case of the SPE framework using XGBoost as the base estimator, it was found that 48 out of 61 instances could be correctly classified, yielding a recall value of 79.69%. The balanced accuracy, precision, recall, and F1-score, as well as the G-mean, were, respectively, 59.68%, 50.11%, 79.69%, 61.25%, and 78.14%. These results were superior to those of other models. The next model in the sequence was the balance cascade model, which used XGBoost as the base estimator. It had a G-mean value of 77.31% and a balance accuracy of 59.22%. The SPE model with LGBM as the base estimators performed the worst out of all the models, with a G-mean value of 70.02% and a balance accuracy of 49.33%.

As shown in Figures 9(a) and 9(b)), among three ensemble imbalance learning models with three machine learning base estimators, the best result was shown by the SPE framework with XGBoost as a base estimator, resulting in a G-mean of 78.14% and balanced accuracy of 59.68%. It was then followed by the balance cascade model with the XGboost model as a base estimator with a G-mean of 77.31% and balanced accuracy of 59.22%. The G-mean value (74.71%) and balanced accuracy (54.22%) of balance cascade with RF as base estimators and G-mean value (74.15%) and balanced accuracy (53.71%) are ranked third and fourth, respectively. The results of the top two models based on G-mean values and balanced accuracy are quite close to each other and can be considered as optimal models for the classification and prediction of MAPs. Furthermore, the best model has been in conjunction with SHAP analysis for interpretation to obtain significant factors as well as assessment of the interaction among risk factors.

3.3. Sensitivity Analysis. The formation of a precise MAP prediction model is crucial because more accurate models may more effectively describe the relationship between MAP

and various environmental and situational factors. The ability to decipher the outcomes of ensemble-imbalance learning models is just as vital. In order to interpret the results of the best models (SPE with XGBoost) and determine the effect of the particular risk factors and their interactions, the SHAP implementation is discussed.

3.3.1. Factor Importance and Contribution. SPE with XGBoost as a base estimator was the optimal model for the prediction of MAPs based on its performance measures, and therefore, we used it to analyze the significance and value of the contributions of specific factors. It is indeed important to note that the two concepts of "factor importance" and "factor contribution" are not synonymous. To what extent a given factor contributes to a model's accuracy is indicated by its importance. The results can be rationally explained by the factors identified through the factor contributions (MAP or AP). As shown in Figure 10(a), the SHAP global importance scores are applied to the factors in the SPE with the XGboost base estimator. Nonetheless, the result does not reveal how much each factor contributed to the overall probability of a MAP occurrence. It demonstrates that low-level wind shear magnitude, with a mean SHAP value of +0.210, is the most important factor causing the MAPs, followed by runway orientation, with a mean SHAP value of +0.160, and altitude of low-level wind shear, with a value of +0.110. Similarly, the SPE with XGboost was investigated further via a SHAP contribution evaluation utilizing SHAP beeswarm plots (Figure 10(b)). Using the SHAP contribution plots, we arrived at a quantitative value by combining the Shapely values and expressing them in terms of the SPE model's factor contributions. On the vertical axis, the input factors are listed in order of growing influence, from most influential to least. Using a horizontal axis for the SHAP value and a color scale from blue (low significance) to red (high significance), this graph displays the contribution of different factors.

The SHAP beeswarm plot of the SPE with XGboost as base estimator illustrated that most of the tailwinds resulted in the initiation of MAPs. The aircraft in this may not be able to touch down at the designated touchdown location. The result is also consistent with the finding of previous research conducted by [52, 53] in which it was also observed that the strong tailwind encountered due downdraft of the thunderstorm and MAPs occurred. The orientation of the runway was the second influential factor. Runways 07C and 07R were more prone to MAP initiation. Some previous studies [54–56] employing numerical simulation and wind tunnel testing also observed that the southerly or southeasterly gusts of wind at HKIA are more likely to trigger low-level wind shear, which can have a significant impact on runways 07C and 07R. Therefore, approaching aircraft could experience a severe low-level wind shear and initiate MAP. Given that MAPs have become a safety concern, runways 07C and 07R should not be utilized for landings during low-level wind shear events.

The third most crucial factor was the low-level wind shear V Location. According to Figure 11(b), low-level wind

TABLE 6: Performance assessment of ensemble imbalance learning model with different base estimators.

Base estimators	Performance measures				
	Recall (%)	Precision (%)	F1-score (%)	Balanced accuracy (%)	Geometric mean (%)
<i>SPE framework</i>					
LGBM	75.23	40.13	52.15	49.33	70.02
XGBoost	79.69	50.11	61.25	59.68	78.14
RF	69.11	49.71	57.51	54.22	74.13
<i>Balance cascade</i>					
LGBM	64.06	51.02	57.25	51.02	72.13
XGBoost	72.38	57.18	64.13	59.22	77.31
RF	69.11	49.71	57.53	54.22	74.71
<i>Easy ensemble</i>					
LGBM	72.71	47.14	57.22	53.71	74.15
XGBoost	64.71	45.91	53.41	49.52	71.66
RF	62.11	55.71	58.51	52.18	72.86

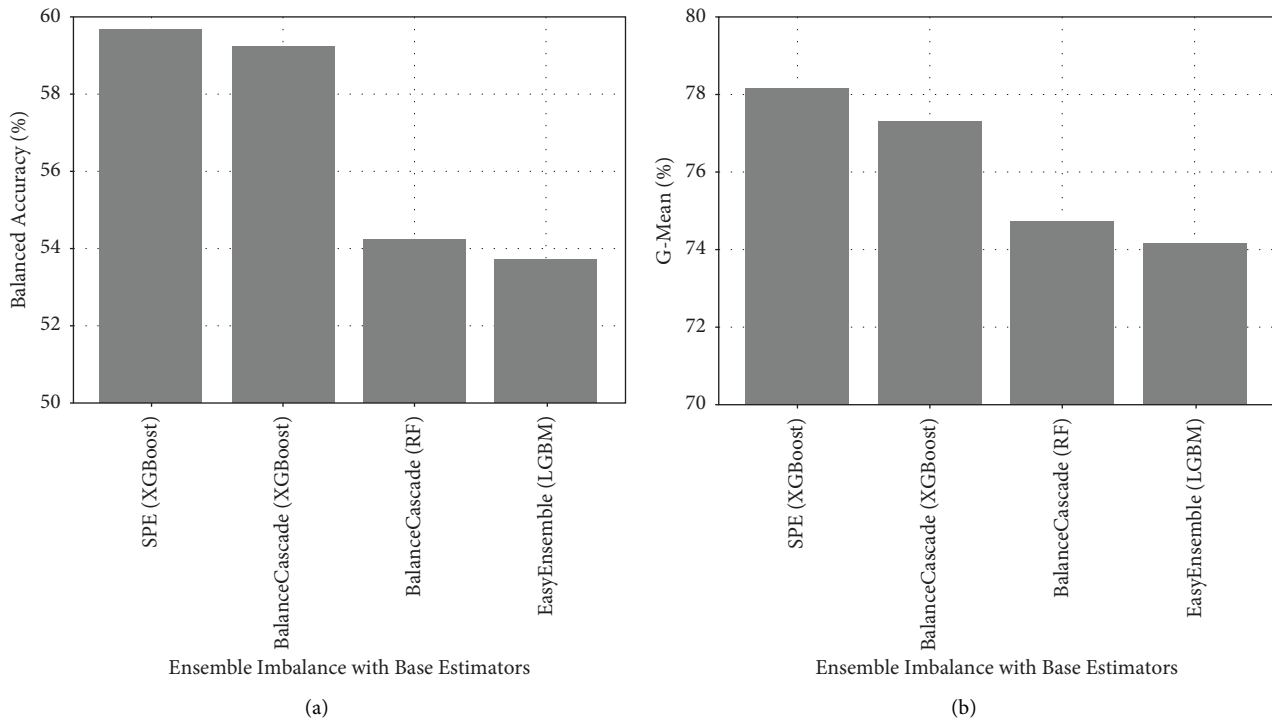


FIGURE 9: Optimal models: (a) top four models with higher balanced accuracy (%) and (b) top four models with higher G-mean value (%).

shear events that took place at lower altitudes were what led to the high number of MAPs. The outcomes are also in line with wind tunnel tests of [57], which showed that about 55% of the severe low-level wind shear occurs below 600 feet and could be regarded as a critical zone for aircrafts on the final approach. The combination of bad weather, complicated terrain, and nearby buildings would cause more turbulence along the glide path. Because of this, the cockpit crew is constantly in motion during the landing phase, and the captain and copilot must make a number of split-second decisions to complete the landing checklist. The best course of action in the event of an unprecedented low-level wind shear that occurs very close to the runway is to abort the landing attempt and perform a go-around.

It is also pertinent to mention that a previous study [58] demonstrated that 300 ft above the ground might be the acceptable value for MAP unless no environmental factors are involved. However, in case of low-level wind shear condition in the runway proximity, initiating MAP at 300 ft above the ground might be very dangerous.

3.3.2. Factor Interaction. The factor importance and contribution (beeswarm) plot revealed no evident connection between the shift in factor value and the change in SHAP value. Figure 11 complemented the contribution plot by depicting the individual interpretation outcomes for the factors and showing how the SHAP values varied with the

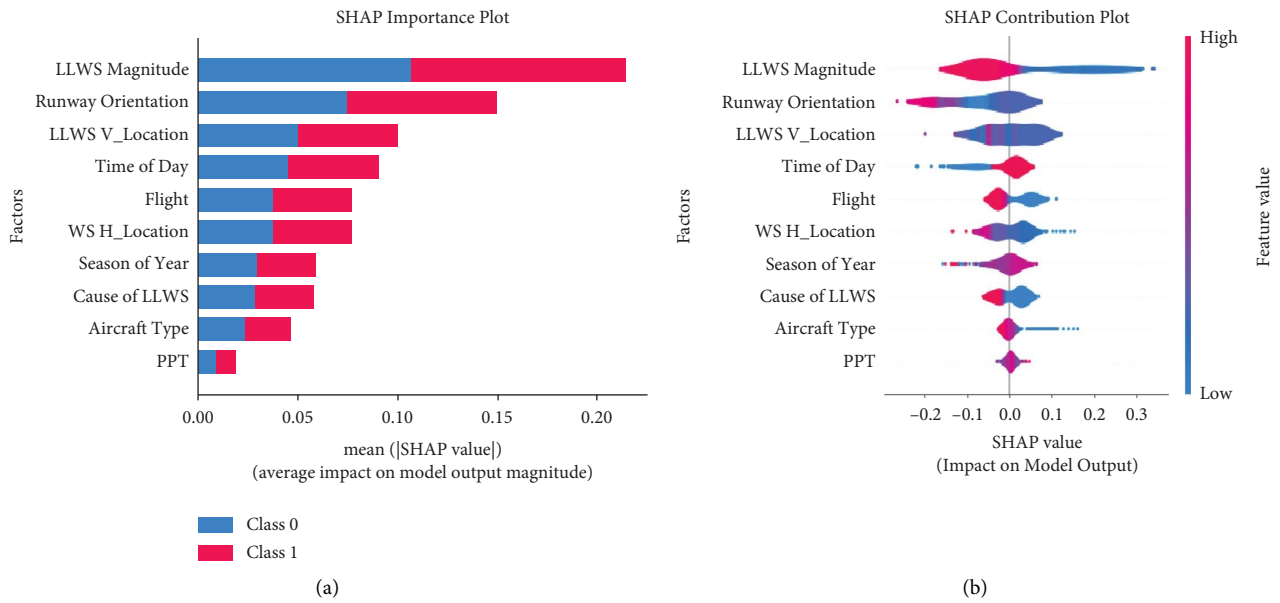


FIGURE 10: (a) SHAP importance plot and (b) SHAP contribution plot.

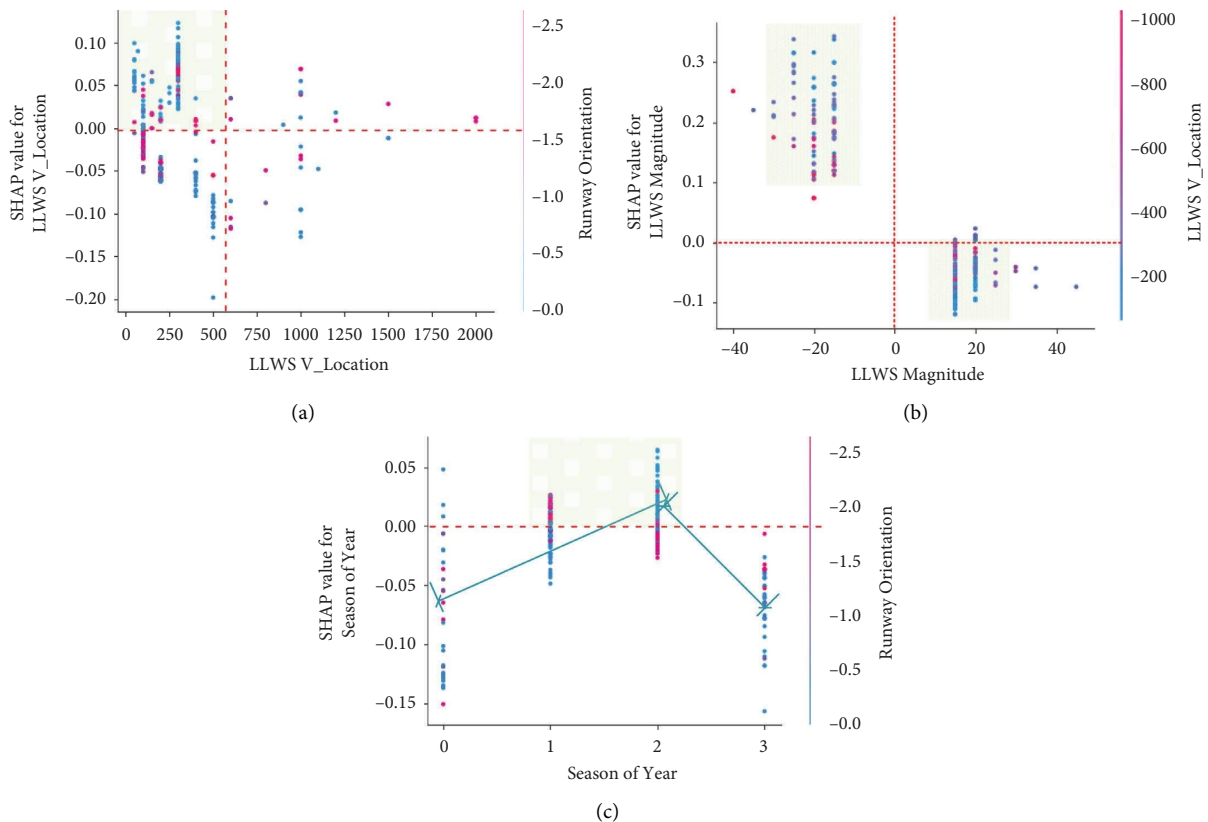


FIGURE 11: (a) SHAP LLWS V_location dependence plot, (b) SHAP LLWS magnitude dependence plot, and (c) SHAP season of year dependence plot.

eigenvalues. By analyzing the interaction plots provided by SHAP, we were able to assess how much each input factor had an effect on the final score obtained from SPE with XGboost as a base estimator.

The impact of low-level wind shear V_Location and runway orientation on model predictions is shown in Figure 11(a). From V_Location of 0 to 500 ft, the points with high densities are those that are above the SHAP 0.00

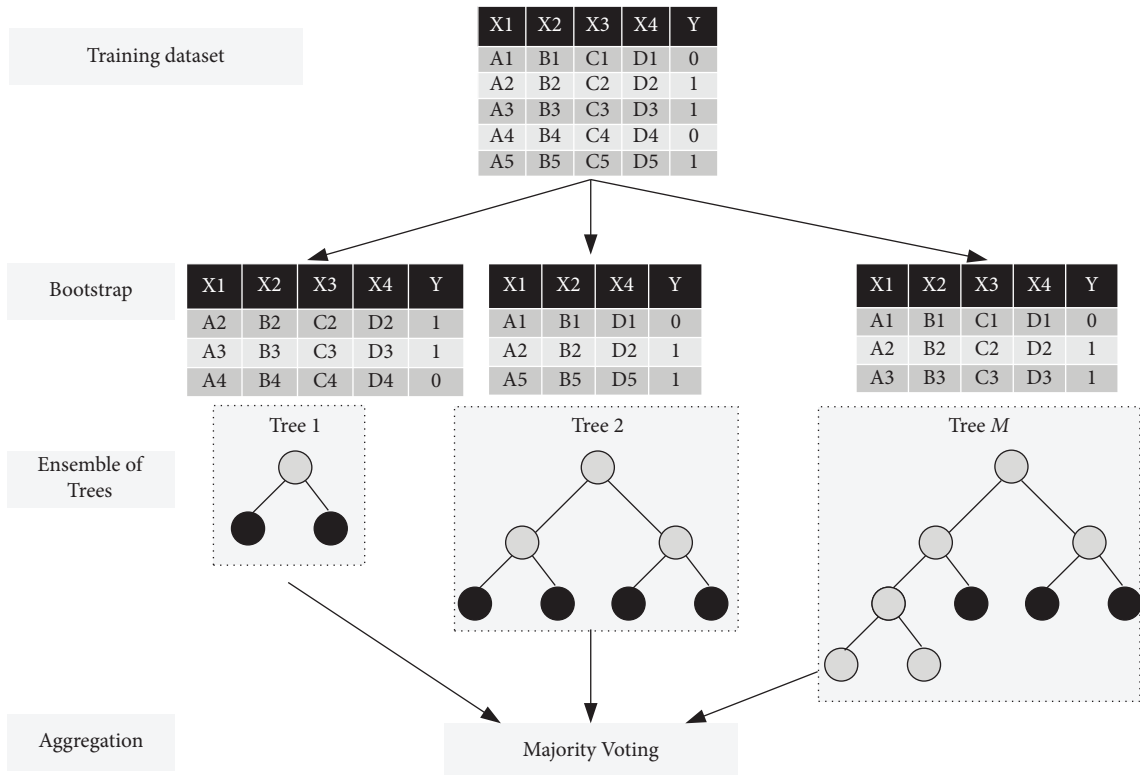


FIGURE 12: Bootstrapping and aggregation in random forest.

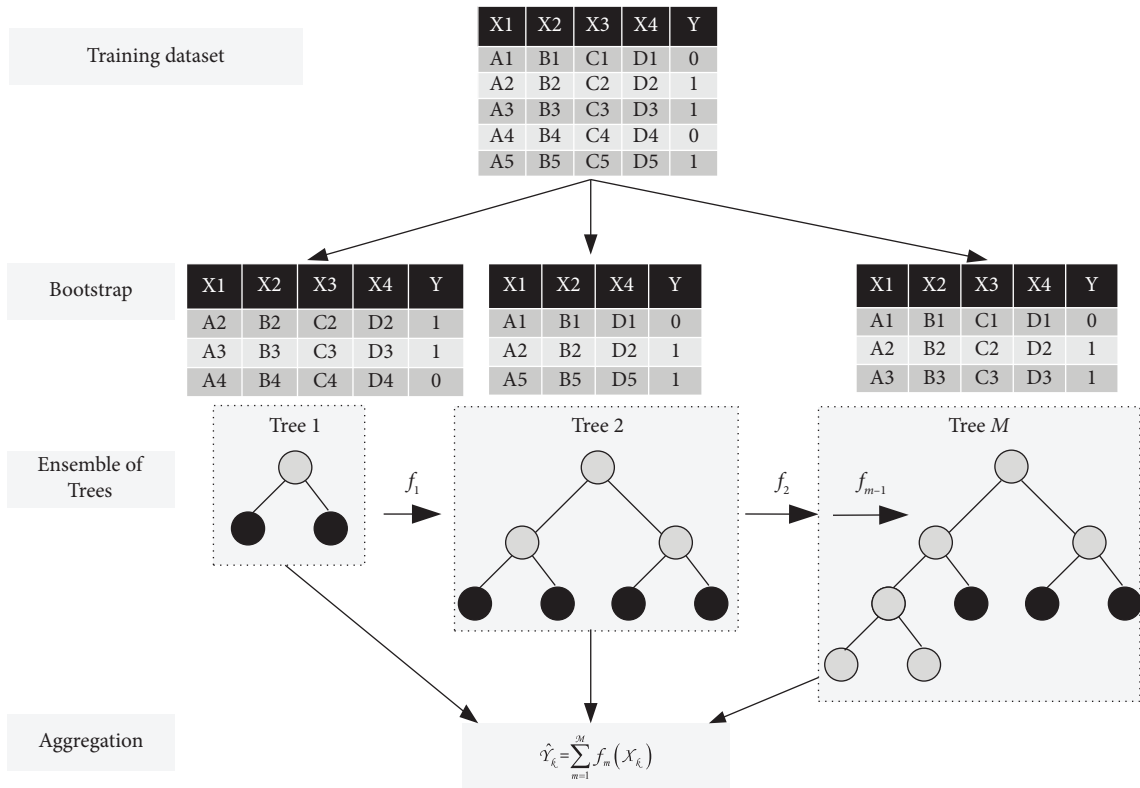


FIGURE 13: Tree learning mechanism of XGBoost.

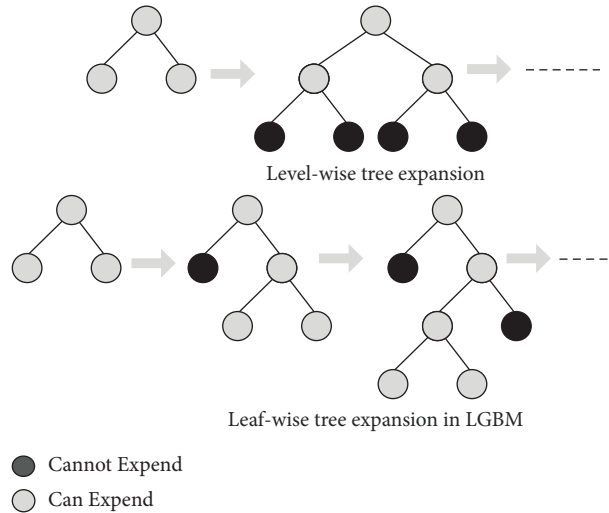


FIGURE 14: Tree expansion in LGBM.

Input: Training data: $D = \{(X_k, Y_k)\}_1^M$, loss function: $L(Y, \varphi(X))$, iterations: J , ratio of big gradient data sampling: α , ratio of slight gradient data sampling: β

- (1) Using FEB approach, combine the factors that are mutually exclusive the factors, i.e., they never concurrently accept nonzero values X_k
- (2) Set $\varphi_0(X) = \operatorname{argmin}_c \sum_k L(Y_k, c)$,
- (3) **for** $j = 1$ to J **do**
- (4) Compute the absolute value of gradient $\lambda_k = |\partial L(Y_k, \varphi(X_k)) / \partial \varphi(X_k)|_{\varphi(X) = \varphi_{j-1}(X)}$,
- (5) The data set is resampled by using GOSS approach
 - (i) top $M = \alpha \times \operatorname{len}(D)$, ; rand $M = \beta \times \operatorname{len}(D)$,
 - (ii) sorted = GetSortedIndices(abs(r)),
 - (iii) $A = \operatorname{sorted}[1: \operatorname{topM}]$,
 - (iv) $B = \operatorname{RandomPick}(\operatorname{Sorted}[\operatorname{topM} : \operatorname{len}(D)], \operatorname{randM})$,
 - (v) $D' = A + B$;
- (6) The Information Gains (IG) are computed: $V_e(d) = 1/n [((\sum_{x_k \in A_1} r_k + 1 - \alpha/\beta \sum_{x_k \in B_1} r_k)^2 / n_1^e(d)) + ((\sum_{x_k \in A_r} r_k + 1 - \alpha/\beta \sum_{x_k \in B_r} r_k)^2 / n_r^e(d))]$,
- (7) New decision tree is developed $\varphi_j(X)'$ on set D'
- (8) Update $\varphi_j(X) = \varphi_{j-1}(X) + \varphi_j(X)$,
- (9) **End for**
- (10) **Return** $\tilde{\varphi}(X) = \varphi_j(X)$,

ALGORITHM 4: Light Gradient Boosting Machine (LGBM).

reference line. The majority of the points have blue labels that indicate Runway 07C and 07R. It demonstrated that the majority of the MAPs were seen on these runways at a lower altitude. As a result, caution should be used when landing on these runways at 500 feet above ground because low-level wind shear could occur. The impact of low-level wind shear magnitude in relation to low-level wind shear V_Location, however, might also be of interest.

The relationship between low-level wind shear magnitude and low-level wind shear V_Location is thus shown in Figure 11(b). All headwinds are shown on the horizontal axis to the right of reference point 0, and all tailwinds are shown to the left of reference point 0. The graph shows that MAPs were typically started at low altitude and with a tailwind most of the time. Pilots attempt

to abort the landing and proceed for a second attempt to reduce the risk of landing short of the runway because the recovery margin time in the event of low-level wind shear occurring at low altitude is limited. As a result, at HKIA, pilots are required to be more vigilant during strong tailwind situations.

Additionally, as shown in Figure 11(c), we evaluated the effects of the Season of the Year factor and runway orientation. There were more blue dots than red, which demonstrated that runways 07C and 07R are extremely susceptible to the occurrence of MAPs year-round. However, the majority of the MAPs were seen in the summer. The pilots' final approach might have been affected by tropical cyclones and southern monsoon winds. These results are in line with earlier research findings as well [59–62].

4. Conclusions and Future Work

This study presents the application of three machine learning-ensemble imbalance learning techniques: SPE framework, balance cascade, and easy ensemble to the prediction of the occurrence of MAP events. For these three ensemble imbalance learning techniques, three state-of-the-art machine learning models, LGBM, XGBoost, and RF were used as base estimators. To the best of our knowledge, this is the first work to detect, model, and interpret MAPs occurrences from HKIA-based PIREPs data using ensemble imbalance learning techniques in conjunction with SHAP. The MAPs were predicted under low-level wind shear conditions considering both environmental and situational factors using 2017 to 2021 low-level wind shear data from HKIA-based PIREPS. Initially, LGBM, XGBoost, and RF were evaluated separately to assess their performance in case of imbalance in low-level wind shear data. Afterwards, these models were employed as base estimators for SPE, balance cascade, and easy ensemble and performance measures were obtained. Regrettably, machine learning algorithms often receive criticism for being ambiguous and challenging to comprehend. However, the increased adaptability and quite often improved reliability of engineering domain modeling over more conventional predictive statistical methods do have an impact on their universal popularity. In this research, the posthoc SHAP interpretation tool was used to decipher the best-predictive ensemble imbalance learning model, and the impact of various factors on the likelihood of a MAP event occurring was demonstrated. From the study, the following conclusions can be drawn:

- (i) On the testing dataset, all three machine learning models, LGBM, XGBoost, and RF, even with well-tuned hyperparameters showed a poor performance in predicting MAPs under low-level wind shear condition with G-mean value of 65.19%, 60.541%, and 66.04%, respectively.
- (ii) The performance of each individual machine learning varied marginally. The G-mean of LGBM was relatively 7.33% higher than XGBoost. Similarly, G-mean of RF was 1.28% higher than LGBM and 8.5% higher than XGboost.
- (iii) The SPE framework in conjunction with XGboost model as a base estimators performed best among all with the G-mean value of 78.14% and balance accuracy of 61.25%. It was then followed by balance cascade model with XGboost as a base estimator with G-mean value of 77.31% and balance accuracy of 59.22%.
- (iv) SHAP demonstrated efficacy in interpreting the optimal model's outcome (SPE with XGBoost as base estimator). The low-level wind shear magnitude was the most influential factor, followed by runway orientation and low-level wind shear V_Location.
- (v) Most of the MAPs were observed during strong tailwind situation. Runway 07C and Runway 07R were observed to high highly vulnerable to MAPs.

- (vi) Similarly, most of the MAPs were initiated within a ceiling of 500ft. In case of severe low-level wind shear, MAPs at altitudes as low as 300 ft might be very dangerous to recover. However, in case of calm weather conditions, 300 ft might be suitable for initiating a go-around protocol.

This proposed method could be used to examine MAPs on a large scale at other airports around the world. It serves as a benchmark for aviation authorities and intellectuals intrigued by air safety. Given that it is crucial for aviation and meteorological applications to comprehend the intricate interactions between several risk aspects that influence the occurrence of MAPs, researchers focusing on civil aviation safety should take advantage of this opportunity. Besides that, this article only addressed the issue of predicting MAPs under low-level wind shear conditions, taking context-specific and environmental factors into account. Additional research could be carried out by combining a number of different machine learning or deep learning approaches with a wide variety of other potential risk factors.

Appendix

A. Machine Learning Models as Base Estimators

A1. Random Forest (RF). RF is an example of a bagging-ensemble classifier, which trains a collection of classifiers in parallel using bootstrapping and aggregation to generate a single final prediction (see Figure 12). According to bootstrapping, multiple decision trees can be learned independently in parallel by using different subsets of the training dataset and various subsets of the available factors. Because each decision tree in a random forest is guaranteed to be unique using bootstrapping, the random forest's overall variance is reduced. The RF classifier has better generalization because it combines the ratings of many trees into a single verdict.

A2. Extreme Gradient Boosting (XGBoost). Extreme gradient boosting (XGBoost) is a distributed gradient-boosted decision tree (GBDT) approach that is optimized for reliability, computational efficiency, and model performance. It consists primarily of an iterative decision tree algorithm with multiple decision trees. Each tree acquires knowledge from the residuals of all preceding trees. As opposed to adopting the majority of random forest's voting output results, as shown in Figure 13 and equation (A.1), XGBoost's predicted output is the sum of all the results.

$$\hat{Y}_k = \sum_{m=1}^M f_m(X_k), f_m \in F, \quad (\text{A.1})$$

where F denotes the trees space, f_m illustrates single m^{th} tree, $f_m(X_k)$ is the output of m^{th} tree, and \hat{Y}_k is the prediction of k^{th} sample x_k . The optimization objective function of the XGBoost model is given by the following equation:

$$\text{Obj}(\varphi) = \Gamma(\varphi) + \Theta(\varphi), \quad (\text{A.2})$$

where $\Gamma(\varphi) = \sum_{k=1}^K l(Y_k, \hat{Y}_k)$ depicts the loss function, Y_k is the target factor, and $\Theta(\varphi) = \sum_{m=1}^M \Theta(f_m)$ penalizes the models' complexity. Then, the classifier is trained in an additive fashion. Let \hat{Y}_k^t be the predicted output of k^{th} sample at the t^{th} iteration, then \hat{Y}_k^t can be illustrated as the following equation:

$$\hat{Y}_k^{(t)} = \hat{Y}_k^{(t-1)} + f_t(x_k). \quad (\text{A.3})$$

In this case, it minimizes the objective function, illustrated by the following equation:

$$\begin{aligned} \text{Obj}^t &= \sum_{k=1}^K (Y_k, \hat{Y}_k) + \Theta(f_t) \\ &= \sum_{k=1}^K (Y_k, \hat{Y}_k^{(t-1)} + f_t(x_k)) + \Theta(f_t). \end{aligned} \quad (\text{A.4})$$

To quickly optimize the objective function, second-order approximation is employed, as shown by the following equation:

$$\text{Obj}^t = \sum_{k=1}^K \left(Y_k, \hat{Y}_k^{(t-1)} + \psi_k f_t(x_k) + \zeta_k f_t^2(x_k) \right) + \Theta(f_t), \quad (\text{A.5})$$

where ψ_k and ζ_k are the first- and second-order gradient statistics on the loss function, respectively. Regularized boosting and column subsampling are two techniques that can enhance the performance of XGBoost classifier and prevent over-fitting [48] paper provides more details).

A3. Light Gradient Boosting Machine (LGBM). Since the gradient boosting decision tree (GBDT) struggles to handle cases with high-dimensional input factors and large amounts of data, the LGBM was developed to address these issues. Unlike other boosting strategies, which divide the tree at the level, this one does so at the leaf level because it makes use of decision tree algorithms. Leaf-wise splitting minimizes loss more than level-wise splitting when developing on the same leaf, leading to significantly higher classification precision than most well-known boosting algorithms. LGBM and gradient boosting both show level-wise and leaf-wise tree expansion in Figure 14.

The working procedure of the LGBM framework is described as follows:

B. Source Code

```
#####IMPORT LIBRARIES #####
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import time
```

```
from sklearn.ensemble import RandomForestClassifier
from lightgbm import LGBMClassifier
from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import recall_score
from sklearn.metrics import balanced_accuracy_score
import collections
from sklearn import metrics
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.metrics import roc_curve, auc
from sklearn.pipeline import make_pipeline
from imblearn.metrics import classification_report_imbalanced
from sklearn.metrics import precision_score, recall_score, f1_score, roc_auc_score, accuracy_score, classification_report
from collections import Counter
from sklearn.model_selection import KFold, StratifiedKFold
from imbalanced_ensemble.ensemble import SelfPacedEnsembleClassifier
from imbalanced_ensemble.ensemble import EasyEnsembleClassifier
from imbalanced_ensemble.ensemble import BalanceCascadeClassifier
from imbalanced_ensemble.utils import evaluate_print
from sklearn.metrics import matthews_corrcoef
import shap
from imblearn.metrics import classification_report_imbalanced.
####IMPORT DATA####
data=pd.read_csv(r"D:\pythondata\MAP_dataset.csv")
x=data.drop(columns=['CLASS'])
y=data['CLASS']
x_train,x_test,y_train,
y_test=train_test_split(x,y,test_size=0.30,random_state=0)
plt.figure(figsize=(12, 8))
aa=sns.heatmap(x.corr(), annot=True)
plt.xticks(rotation=20)
plt.yticks(rotation=20)
aa.set_xlabel("Factors",fontsize=14,weight='bold')
aa.set_ylabel("Factors",fontsize=14,weight='bold')
```

```

#####HYPERPARAMETER TUNING LGBM#####
def objective(space):
    model1 = LGBMClassifier (learning_rate = space
['learning_rate'],
        n_estimators=int(space['n_estimators']),
        num_leaves=int(space['num_leaves']), max_
depth=int(space['max_depth']),
        reg_lambda=space['reg_lambda'],
        reg_alpha=space['reg_alpha'])
    accuracy=cross_val_score(model1, x_train, y_train,
cv=10, scoring="accuracy").mean()
    return {'loss': -accuracy,"status":STATUS_OK}
from hyperopt import hp, fmin, tpe, Trials,
STATUS_OK
space={'learning_rate': hp.quniform('learning_rate',
0.001, 0.2, 0.001),
        'n_estimators': hp.quniform('n_estimators',
200,1500,100),
        'num_leaves':
hp.quniform('num_leaves',30,100,1),
        'max_depth':hp.quniform('max_depth',2,15,1),
        'reg_lambda':hp.uniform('reg_lambda',1.1,1.5),
        'reg_alpha':hp.quniform('reg_alpha',1.1,1.5,0.1)}
trials=Trials()
best1=fmin(fn=objective,
        space=space,
        algo=tpe.suggest,
        max_evals=25,
        trials=trials)
best1
lg=LGBMClassifier(max_depth=int(best1['max_
depth']), n_estimators=int(best1['n_estimators']),
        reg_lambda=best1['reg_lambda'],
        reg_alpha=best1['reg_alpha'],
        learning_rate=best1['learning_rate'],
        num_leaves=int(best1['num_leaves']))
lg=lg.fit(x_train, y_train)
pred_lg=lg.predict(x_test)
clg=confusion_matrix(y_test, pred_lg)
clg
%matplotlib inline
class_names=[False, True] # name of classes
plt.subplots(figsize=(8,5))
#ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.rcParams.update({'font.size': 16})

```

```

plt.ylabel('Actual')
plt.xlabel('Predicted')
tick_marks=[0.5, 1.5]
plt.xticks(tick_marks, class_names, fontsize =
16,weight = 'bold')
plt.yticks(tick_marks, class_names, fontsize=16,
weight='bold')
matrix_lg_test=pd.DataFrame(data=clg, columns=
['AP', 'MAP'], index=['AP', 'MAP'])
a1=sns.heatmap(matrix_lg_test, annot=True, fmt='d',
linewidths=1, cmap='Greens',annot_kws={"fontsize":
28})
a1.set_xlabel("PREDICTED" fontsize = 16,weight =
'bold')
a1.set_ylabel("ACTUAL",fontsize=16,weight='bold')
print(classification_report_imbalanced(y_test,
pred_lg))
best1
#####HYPERPARAMETER TUNING XGBOOST
#####
def objective(space):
    model1 = XGBClassifier(learning_rate = space
['learning_rate'],
        n_estimators=int(space['n_estimators']),
        num_leaves=int(space['num_leaves']), max_
depth=int(space['max_depth']),
        reg_lambda=space['reg_lambda'],
        reg_alpha=space['reg_alpha'])
    accuracy=cross_val_score(model1, x_train, y_train,
cv=10, scoring="accuracy").mean()
    return {'loss': -accuracy,"status":STATUS_OK}
from hyperopt import hp, fmin, tpe, Trials,
STATUS_OK
space={'learning_rate': hp.quniform('learning_rate',
0.001, 0.2, 0.001),
        'n_estimators': hp.quniform('n_estimators',
200,1500,100),
        'num_leaves':
hp.quniform('num_leaves',30,100,1),
        'max_depth':hp.quniform('max_depth',2,15,1),
        'reg_lambda':hp.uniform('reg_lambda',1.1,1.5),
        'reg_alpha':hp.quniform('reg_alpha',1.1,1.5,0.1)}
trials=Trials()
best2=fmin(fn=objective,
        space=space,
        algo=tpe.suggest,
        max_evals=25,

```

```

        trials=trials)
best2
xg=XGBClassifier(max_depth=int(best2['max_
depth']), n_estimators=int(best2['n_estimators']),
                reg_lambda=best2['reg_lambda'],
                reg_alpha=best2['reg_alpha'],
                learning_rate=best2['learning_rate'],
                num_leaves=int(best2['num_leaves']))
xg=xg.fit(x_train, y_train)
pred_xg=xg.predict(x_test)
cxg=confusion_matrix(y_test, pred_xg)
cxg
%matplotlib inline
class_names=[False, True] # name of classes
plt.subplots(figsize=(8,5))
#ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.rcParams.update({'font.size': 16})
plt.ylabel('Actual')
plt.xlabel('Predicted')
tick_marks=[0.5, 1.5]
plt.xticks(tick_marks, class_names, fontsize = 16,
weight='bold')
plt.yticks(tick_marks, class_names, fontsize = 16,
weight='bold')
matrix_lg_test=pd.DataFrame(data=cxg, columns=
['AP', 'MAP'], index=['AP', 'MAP'])
a1=sns.heatmap(matrix_lg_test, annot=True, fmt='d',
linewidths=1, cmap='Greens',annot_kws={"fontsize":
28})
a1.set_xlabel("PREDICTED", fontsize = 16,weight =
'bold')
a1.set_ylabel("ACTUAL",fontsize=16,weight='bold')
print(classification_report_imbalanced(y_test,
pred_xg))
best2
#####HYPERPARAMETER TUNING RF#####
def objective(space):
    model1=RandomForestClassifier(max_depth=int(-
space['max_depth']), n_estimators=int(space
['n_estimators']))
    accuracy=cross_val_score(model1, x_train, y_train,
cv=10, scoring="f1").mean()
    return {'loss': -accuracy,"status":STATUS_OK}
from hyperopt import hp, fmin, tpe, Trials,
STATUS_OK
space={'n_estimators': hp.quniform('n_estimators',
200,1500,100), 'max_depth':
hp.quniform('max_depth',2,15,1)}

```

```

trials=Trials()
best3=fmin(fn=objective,
          space=space,
          algo=tpe.suggest,
          max_evals=25,
          trials=trials)
best3
rf=RandomForestClassifier(max_depth=int(best3
['max_depth']), n_estimators=int(best3['n_estima-
tors'])).fit(x_train, y_train)
rf=rf.fit(x_train, y_train)
pred_rf=rf.predict(x_test)
crf=confusion_matrix(y_test, pred_rf)
crf
%matplotlib inline
class_names=[False, True] # name of classes
plt.subplots(figsize=(8,5))
#ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.rcParams.update({'font.size': 16})
plt.ylabel('Actual')
plt.xlabel('Predicted')
tick_marks=[0.5, 1.5]
plt.xticks(tick_marks, class_names, fontsize = 16,
weight='bold')
plt.yticks(tick_marks, class_names, fontsize = 16,
weight='bold')
matrix_lg_test = pd.DataFrame(data = crf, columns=
['AP', 'MAP'], index=['AP', 'MAP'])
a1=sns.heatmap(matrix_lg_test, annot=True, fmt='d',
linewidths=1, cmap='Greens',annot_kws={"fontsize":
28})
a1.set_xlabel("PREDICTED",fontsize=16,weight='
bold')
a1.set_ylabel("ACTUAL",fontsize=16,weight='bold')
print(classification_report_imbalanced(y_test,
pred_rf))
best3
#####MODELS DEVELOPMENT#####
slg = SelfPacedEnsembleClassifier (base_estimator =
LGBMClassifier(max_depth=int(best1['max_depth']),
                n_estimators=int(best1['n_estimators']),
                reg_lambda=best1['reg_lambda'],
                reg_alpha=best1['reg_alpha'],
                learning_rate=best1['learning_rate'],
                num_leaves = int(best1['num_leaves'])))
slg=slg.fit(x_train, y_train)
pred_slg=slg.predict(x_test)

```

```

cslg=confusion_matrix(y_test, pred_slg)
cslg
%matplotlib inline
class_names=[False, True] # name of classes
plt.subplots(figsize=(8,5))
#ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.rcParams.update({'font.size': 16})
plt.ylabel('Actual')
plt.xlabel('Predicted')
tick_marks=[0.5, 1.5]
plt.xticks(tick_marks, class_names, fontsize = 16,
weight='bold')
plt.yticks(tick_marks, class_names, fontsize = 16,
weight='bold')
matrix_lg_test=pd.DataFrame(data=cslg, columns=
['AP', 'MAP'], index=['AP', 'MAP'])
a1=sns.heatmap(matrix_lg_test, annot=True, fmt='d',
linewidths=1, cmap='Greens',annot_kws={"fontsize":
28})
a1.set_xlabel("PREDICTED",fontsize=16,weight='
bold')
a1.set_ylabel("ACTUAL",fontsize=16,weight='bold')
print(classification_report_imbalanced(y_test,
pred_slg))
blg=BalanceCascadeClassifier(base_estimator=LGBM-
Classifier (max_depth=int(best1['max_depth']),
n_estimators=int(best1['n_estimators']),
reg_lambda=best1['reg_lambda'],
reg_alpha=best1['reg_alpha'],
learning_rate=best1['learning_rate'],
num_leaves=int(best1['num_leaves'])))
blg=blg.fit(x_train, y_train)
pred_blg=blg.predict(x_test)
cblg=confusion_matrix(y_test, pred_blg)
cblg
%matplotlib inline
class_names=[False, True] # name of classes
plt.subplots(figsize=(8,5))
#ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.rcParams.update({'font.size': 16})
plt.ylabel('Actual')
plt.xlabel('Predicted')
tick_marks=[0.5, 1.5]
plt.xticks(tick_marks, class_names, fontsize = 16,
weight='bold')

```

```

plt.yticks(tick_marks, class_names, fontsize = 16,
weight='bold')
matrix_lg_test=pd.DataFrame(data=cblg, columns=
['AP', 'MAP'], index=['AP', 'MAP'])
a1=sns.heatmap(matrix_lg_test, annot=True, fmt='d',
linewidths=1, cmap='Greens',annot_kws={"fontsize":
28})
a1.set_xlabel("PREDICTED", fontsize = 16, weight =
'bold')
a1.set_ylabel("ACTUAL",fontsize=16,weight='bold')
print(classification_report_imbalanced(y_test,
pred_blg))
elg=EasyEnsembleClassifier(base_estimator=LGBMC-
lassifier (max_depth=int(best1['max_depth']),
n_estimators=int(best1['n_estimators']),
reg_lambda=best1['reg_lambda'],
reg_alpha=best1['reg_alpha'],
learning_rate=best1['learning_rate'],
num_leaves=int(best1['num_leaves'])))
elg=elg.fit(x_train, y_train)
pred_elg=elg.predict(x_test)
celg=confusion_matrix(y_test, pred_elg)
celg
%matplotlib inline
class_names=[False, True] # name of classes
plt.subplots(figsize=(8,5))
#ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.rcParams.update({'font.size': 16})
plt.ylabel('Actual')
plt.xlabel('Predicted')
tick_marks=[0.5, 1.5]
plt.xticks(tick_marks, class_names, fontsize = 16,
weight='bold')
plt.yticks(tick_marks, class_names, fontsize=16,
weight='bold')
matrix_lg_test=pd.DataFrame(data=celg, columns=
['AP', 'MAP'], index=['AP', 'MAP'])
a1=sns.heatmap(matrix_lg_test, annot=True, fmt='d',
linewidths=1, cmap='Greens',annot_kws={"fontsize":
28})
a1.set_xlabel("PREDICTED", fontsize=16,
weight='bold')
a1.set_ylabel("ACTUAL",fontsize =16,weight='bold')
print(classification_report_imbalanced(y_test,
pred_elg))
fig=plt.figure(figsize=(10, 7)) ax=fig.add_axes([0,0,1,1])
runway=['SPE(XGBoost)', 'BalanceCascade(XGBoost)',
'BalanceCascade(RF)', 'EasyEnsemble(LGBM)']

```

```

llws=[59.68, 59.22, 54.22, 53.71]
ax.bar(runway, llws, ls='-.', lw=0.25, color='Grey')
plt.xticks(rotation=20, fontsize=14)
plt.yticks(fontsize=14)
plt.ylim(50,60)
ax.grid(True)
plt.ylabel('Balanced Accuracy (%)',fontsize=14,
weight='bold')
plt.xlabel('Ensemble Imbalance with Base
Estimators',fontsize=14,weight='bold')
plt.show()
#####SHAP ANALYSIS #####
#####
shap.initjs()
explainer=shap.KernelExplainer(sxg.predict_proba,
sampletest)
shap_values=explainer.shap_values(x_test)
plt.ylabel('Factors',fontsize=14,weight='bold')
plt.xlabel('Mean SHAP Value',fontsize=14,
weight='bold')
plt.title('SHAP Importance Plot',fontsize=14,
weight='bold')
shap.summary_plot(shap_values, x_test)
plt.figure(figsize=(12, 8))
summary=shap.summary_plot(shap_values[1], x_test,
plot_type='violin', show=False)
plt.title('SHAP Contribution Plot',fontsize=14,
weight='bold')
plt.ylabel('Factors',fontsize=14,weight='bold')
plt.xlabel('SHAP value (Impact on Model
Output)',fontsize=14,weight='bold')
plt.gcf().axes[-1].set_aspect('auto')
plt.tight_layout()
# As mentioned, smaller "box_aspect" value to make
colorbar thicker
plt.gcf().axes[-1].set_box_aspect(50)
shap.dependence_plot('LLWS V_Location', shap_val-
ues[1], x_test)
shap.dependence_plot('LLWS Magnitude', shap_val-
ues[1], x_test)
shap.dependence_plot('Season of Year', shap_values
[1], x_test)

```

Data Availability

The data supporting the current study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Research Fund for International Young Scientists (RFIS) of National Natural Science Foundation of China (Grant No. 52250410351), National Foreign Expert Project (Grant No. QN2022133001L), National Natural Science Foundation of China (Grant No. U1733113), Shanghai Municipal Science and Technology Major Project (Grant No. 2021SHZDZX0100) and the Fundamental Research Funds for the Central Universities.

References

- [1] T. Blajev and W. Curtis, "Go-around decision-making and execution project," *Final Report to Flight Safety Foundation*, 2017, https://flightsafety.org/wp-content/uploads/2017/03/Go-around-study_final.pdf.
- [2] R. C. Jou, C. W. Kuo, and M. L. Tang, "A study of job stress and turnover tendency among air traffic controllers: the mediating effects of job satisfaction," *Transportation Research Part E: Logistics and Transportation Review*, vol. 57, pp. 95–104, 2013.
- [3] X. Prats, V. Puig, J. Quevedo, and F. Nejari, "Multi-objective optimisation for aircraft departure trajectories minimising noise annoyance," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 6, pp. 975–989, 2010.
- [4] J. Shortle and L. Sherry, "A model for investigating the interaction between go-arounds and runway throughput," in *Proceedings of the 2013 Aviation Technology, Integration, and Operations Conference*, p. 4235, Los Angeles, CA, USA, August 2013.
- [5] W. Frost, *Flight in Low-Level Wind Shear (No. NASA-CR-3678)*, National Aeronautics And Space Administration, Washington, DC, USA, 1983.
- [6] D. A. Stratton and R. F. Stengel, "Probabilistic reasoning for intelligent wind shear avoidance," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 1, pp. 247–254, 1992.
- [7] L. Bretschneider, R. Hankers, S. Schönhals, J. M. Heimann, and A. Lampert, "Wind shear of low-level jets and their influence on manned and unmanned fixed-wing aircraft during landing approach," *Atmosphere*, vol. 13, no. 1, p. 35, 2021.
- [8] P. Zaal, A. Campbell, J. A. Schroeder, and S. Shah, "Validation of proposed go-around criteria under various environmental conditions," in *Proceedings of the AIAA Aviation 2019 Forum*, p. 2993, Dallas, TX, USA, June 2019.
- [9] C. S. Chou, A. Tien, and H. Bateman, "A machine learning application for predicting and alerting missed approaches for airport management," in *Proceedings of the 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, pp. 1–9, IEEE, San Antonio, TX, USA, October 2021.
- [10] B. Donavalli, S. P. Mattingly, and A. Massidda, *Impact of Weather Factors on Go-Around Frequency (No. 17-03934)*, Transportation Research Board, Washington DC, USA, 2017.
- [11] S. R. Proud, "Analysis of aircraft flights near convective weather over Europe," *Weather*, vol. 70, no. 10, pp. 292–296, 2015.

- [12] N. P. Singh, S. K. Goh, and S. Alam, "Real-time unstable approach detection using sparse variational Gaussian process," in *Proceedings of the 2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*, pp. 1–10, IEEE, Singapore, February 2020.
- [13] L. Dai, Y. Liu, and M. Hansen, "Modeling go-around occurrence using principal component logistic regression," *Transportation Research Part C: Emerging Technologies*, vol. 129, Article ID 103262, 2021.
- [14] M. Causse, F. Dehais, P. Péran, U. Sabatini, and J. Pastor, "The effects of emotion on pilot decision-making: a neuro-ergonomic approach to aviation safety," *Transportation Research Part C: Emerging Technologies*, vol. 33, pp. 272–281, 2013.
- [15] F. Dehais, J. Behrend, V. Peysakhovich, M. Causse, and C. D. Wickens, "Pilot flying and pilot monitoring's aircraft state awareness during go-around execution in aviation: a behavioral and eye tracking study," *The International Journal of Aerospace Psychology*, vol. 27, no. 1-2, pp. 15–28, 2017.
- [16] Q. Kennedy, J. L. Taylor, G. Reade, and J. A. Yesavage, "Age and expertise effects in aviation decision making and flight control in a flight simulator," *Aviation Space and Environmental Medicine*, vol. 81, no. 5, pp. 489–497, 2010.
- [17] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [18] H. Liu and G. Tian, "Building engineering safety risk assessment and early warning mechanism construction based on distributed machine learning algorithm," *Safety Science*, vol. 120, pp. 764–771, 2019.
- [19] G. Ma, Z. Wu, J. Jia, and S. Shang, "Safety risk factors comprehensive analysis for construction project: combined cascading effect and machine learning approach," *Safety Science*, vol. 143, Article ID 105410, 2021.
- [20] A. L. Beam and I. S. Kohane, "Big data and machine learning in health care," *JAMA*, vol. 319, no. 13, pp. 1317–1318, 2018.
- [21] J. Futoma, M. Simons, T. Panch, F. Doshi-Velez, and L. A. Celi, "The myth of generalisability in clinical research and machine learning in health care," *The Lancet Digital Health*, vol. 2, no. 9, pp. e489–e492, 2020.
- [22] S. N. Goodman, S. Goel, and M. R. Cullen, "Machine learning, health disparities, and causal reasoning," *Annals of Internal Medicine*, vol. 169, no. 12, pp. 883–884, 2018.
- [23] T. N. K. Hung, N. Q. K. Le, N. H. Le et al., "An AI-based prediction model for drug-drug interactions in osteoporosis and paget's diseases from SMILES," *Molecular informatics*, vol. 41, no. 6, Article ID 2100264, 2022.
- [24] L. H. T. Lam, D. T. Do, D. T. N. Diep et al., "Molecular subtype classification of low-grade gliomas using magnetic resonance imaging-based radiomics and machine learning," *NMR in Biomedicine*, vol. 35, no. 11, p. 4792, 2022.
- [25] H. Malik, N. Fatema, and J. A. Alzubi, *AI and Machine Learning Paradigms for Health Monitoring System: Intelligent Data Analytics*, Springer, Berlin, Germany, 2021.
- [26] M. B. McDermott, S. Wang, N. Marinsek, R. Ranganath, L. Foschini, and M. Ghassemi, "Reproducibility in machine learning for health research: still a ways to go," *Science Translational Medicine*, vol. 13, no. 586, p. 1655, 2021.
- [27] T. H. Vo, N. T. K. Nguyen, Q. H. Kha, and N. Q. K. Le, "On the road to explainable AI in drug-drug interactions prediction: a systematic review," *Computational and Structural Biotechnology Journal*, vol. 20, pp. 2112–2123, 2022.
- [28] R. De la Torre, C. G. Corlu, J. Faulin, B. S. Onggo, and A. A. Juan, "Simulation, optimization, and machine learning in sustainable transportation systems: models and applications," *Sustainability*, vol. 13, no. 3, p. 1551, 2021.
- [29] S. Dong, A. Khattak, I. Ullah, J. Zhou, and A. Hussain, "Predicting and analyzing road traffic injury severity using boosting-based ensemble learning models with SHAPley Additive exPlanations," *International Journal of Environmental Research and Public Health*, vol. 19, no. 5, p. 2925, 2022.
- [30] Z. Fan, C. Liu, D. Cai, and S. Yue, "Research on black spot identification of safety in urban traffic accidents based on machine learning method," *Safety Science*, vol. 118, pp. 607–616, 2019.
- [31] L. Govinda, M. Sai Kiran Raju, and K. Ravi Shankar, "Pedestrian-vehicle interaction severity level assessment at uncontrolled intersections using machine learning algorithms," *Safety Science*, vol. 153, Article ID 105806, 2022.
- [32] A. Khattak, H. Almujiabah, A. Elamary, and C. M. Matara, "Interpretable dynamic ensemble selection approach for the prediction of road traffic injury severity: a case study of Pakistan's national highway N-5," *Sustainability*, vol. 14, no. 19, Article ID 12340, 2022a.
- [33] A. Khattak, P. W. Chan, F. Chen, and H. Peng, "Time-series prediction of intense wind shear using machine learning algorithms: a case study of Hong Kong international airport," *Atmosphere*, vol. 14, no. 2, p. 268, 2023.
- [34] C. Li and P. Xu, "Application on traffic flow prediction of machine learning in intelligent transportation," *Neural Computing and Applications*, vol. 33, no. 2, pp. 613–624, 2021.
- [35] M. Nama, A. Nath, N. Bechra et al., "Machine learning-based traffic scheduling techniques for intelligent transportation system: opportunities and challenges," *International Journal of Communication Systems*, vol. 34, no. 9, p. 4814, 2021.
- [36] R. Singh, R. Sharma, S. Vaseem Akram et al., "Highway 4.0: digitalization of highways for vulnerable road safety development with intelligent IoT sensors and machine learning," *Safety Science*, vol. 143, Article ID 105407, 2021.
- [37] D. Truong, "Using causal machine learning for predicting the risk of flight delays in air transportation," *Journal of Air Transport Management*, vol. 91, Article ID 101993, 2021.
- [38] S. Zhang, A. Khattak, C. M. Matara, A. Hussain, and A. Farooq, "Hybrid feature selection-based machine learning Classification system for the prediction of injury severity in single and multiple-vehicle accidents," *PLoS One*, vol. 17, no. 2, Article ID 262941, 2022.
- [39] P. Gogas and T. Papadimitriou, "Machine learning in economics and finance," *Computational Economics*, vol. 57, no. 1, pp. 1–4, 2021.
- [40] J. W. Goodell, S. Kumar, W. M. Lim, and D. Pattnaik, "Artificial intelligence and machine learning in finance: identifying foundations, themes, and research clusters from bibliometric analysis," *Journal of Behavioral and Experimental Finance*, vol. 32, Article ID 100577, 2021.
- [41] S. Kumar, D. Sharma, S. Rao, W. M. Lim, and S. K. Mangla, "Past, present, and future of sustainable finance: insights from big data analytics through machine learning of scholarly research," *Annals of Operations Research*, pp. 1–44, 2022.
- [42] T. Warin and A. Stojkov, "Machine learning in finance: a metadata-based systematic review of the literature," *Journal of Risk and Financial Management*, vol. 14, no. 7, p. 302, 2021.

- [43] A. Rawson, M. Brito, Z. Sabeur, and L. Tran-Thanh, "A machine learning approach for monitoring ship safety in extreme weather events," *Safety Science*, vol. 141, Article ID 105336, 2021.
- [44] Z. Liu, W. Cao, Z. Gao et al., "Self-paced ensemble for highly imbalanced massive data classification," in *Proceedings of the 2020 IEEE 36th international conference on data engineering (ICDE)*, pp. 841–852, IEEE, Dallas, TX, USA, April 2020.
- [45] X. Y. Liu, J. Wu, and Z. H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics- Part B: Cybernetics: A Publication of the IEEE Systems, Man, and Cybernetics Society*, vol. 39, no. 2, pp. 539–550, 2009.
- [46] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [47] G. Ke, Q. Meng, T. Finley et al., "Lightgbm: a highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [48] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, New York, NY, USA, August 2016.
- [49] K. C. Szeto and P. W. Chan, "High resolution numerical modelling of windshear episodes at the Hong Kong International Airport," in *Proceedings of the 12th Conference on Aviation, Range, and Aerospace Meteorology*, Atlanta, GA, USA, January 2006.
- [50] K. K. Hon and P. W. Chan, "Historical analysis (2001–2019) of low-level wind shear at the Hong Kong International Airport," *Meteorological Applications*, vol. 29, no. 2, p. 2063, 2022.
- [51] P. W. Chan, "Severe wind shear at Hong Kong International airport: climatology and case studies," *Meteorological Applications*, vol. 24, no. 3, pp. 397–403, 2017.
- [52] S. M. Tse, P. W. Chan, and W. K. Wong, "A case study of missed approach of aircraft due to tailwind associated with thunderstorms," *Meteorological Applications*, vol. 21, no. 1, pp. 50–61, 2014.
- [53] A. Khattak, P. W. Chan, F. Chen, and H. Peng, "Prediction of aircraft go-around during wind shear using the dynamic ensemble selection framework and pilot reports," *Atmosphere*, vol. 13, no. 12, p. 2104, 2022.
- [54] F. Chen, H. Peng, P. W. Chan, and X. Zeng, "Wind tunnel testing of the effect of terrain on the wind characteristics of airport glide paths," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 203, Article ID 104253, 2020.
- [55] L. Lei, P. W. Chan, Z. Li-Jie, and M. Hui, "Numerical simulation of terrain-induced vortex/wave shedding at the Hong Kong International Airport," *Meteorologische Zeitschrift*, vol. 22, no. 3, pp. 317–327, 2013.
- [56] L. Li and P. W. Chan, "LIDAR observation and numerical simulation of vortex/wave shedding at the Eastern Runway Corridor of the Hong Kong International Airport," *Meteorological Applications*, vol. 23, no. 3, pp. 379–388, 2016.
- [57] F. Chen, H. Peng, P. W. Chan, and X. Zeng, "Low-level wind effects on the glide paths of the North Runway of HKIA: a wind tunnel study," *Building and Environment*, vol. 164, Article ID 106337, 2019.
- [58] A. Campbell, P. Zaal, J. A. Schroeder, and S. Shah, "Development of possible go-around criteria for transport aircraft," in *Proceedings of the Proceedings of the 2018 Aviation Technology, Integration, and Operations Conference*, p. 3198, Atlantic, NJ, USA, June 2018.
- [59] P. W. Chan and K. K. Hon, "Observation and numerical simulation of terrain-induced windshear at the Hong Kong International Airport in a planetary boundary layer without temperature inversions," *Advances in Meteorology*, vol. 2016, Article ID 1454513, 9 pages, 2016.
- [60] P. W. Chan and K. K. Hon, "observations and numerical simulations of sea breezes at hong kong international airport," *Weather*, vol. 78, no. 2, pp. 55–63, 2022.
- [61] P. W. Chan, "An event of tail strike of an aircraft due to terrain-induced wind shear at the Hong Kong International Airport," *Meteorological Applications*, vol. 19, no. 3, pp. 325–333, 2012.
- [62] K. K. Hon, P. W. Chan, Y. Y. Chiu, and W. Tang, "Application of short-range LIDAR in early alerting for low-level wind-shear and turbulence at Hong Kong International Airport," *Advances in Meteorology*, vol. 2014, Article ID 162748, 8 pages, 2014.