Fifteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2023)

Data-Driven Approach for Runway Braking Condition Assessment with Forecasting Capability

Marek Travnik, Prof. R. John Hansman
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, Massachusetts
{travnik, rjhans}@mit.edu

***Abstract*—Traditional runway condition reporting is limited due to its reliance on runway contamination information and pilot reports of braking action. A database of 4.9 million aircraft landings by Aviation Safety Technologies, labeled with runway condition codes computed from aircraft sensor outputs provides a unique opportunity to enhance and modernize condition reporting using data-driven methods. This paper introduces a machine learning model trained on this landing database, which predicts runway condition codes using a cascading Xgboost architecture. The method incorporates a novel multiple-ROC threshold setting procedure for linked classifiers which maintains the shape of the runway condition code distribution. Notably, the model can be used in a forecasting setting as it only requires weather information from METAR reports, a description of the runway, and aircraft type as input. To test its effectiveness, the method is applied to a collection of 30 historical runway excursion incidents, consistently assigning at best ”Medium to Poor” braking action to all cases with reduced friction. The model can serve as a valuable decision aid for aircraft operators, complementing traditional runway condition reporting. Additionally, it can function as a forecasting tool to inform runway maintenance decisions.**

***Keywords*—runway condition assessment; degraded braking; runway overrun prevention; forecasting; applied machine learning; xgboost**

I. Introduction

Runway excursion incidents upon landing have been an issue of major concern in aviation safety in the 21st century. According to the Airbus Statistical Analysis of Commercial Aviation Accidents [1], runway excursions were the third leading cause of fatal aviation accidents and the primary cause of hull loss accidents between 2002 and 2021. The contributing factors to these incidents are widely acknowledged to involve the interaction of braking action between the runway and the aircraft wheels and the situational awareness and decision-making of the pilot [2].

In 2007, the Federal Aviation Administration (FAA) launched the Take-off and Landing Performance Assessment (TALPA)[1] initiative in response to Southwest Airlines Flight 1248, which overran the runway upon landing in a snowstorm at Chicago-Midway International Airport[2]. The TALPA initiative introduced Field Condition Notices to Air Missions (FICON NOTAMs) as a means to communicate up-to-date and reliable information about braking performance on the runway to pilots, helping them make the right decisions during landing.

FICON NOTAMs contain information about the contamination of the runway and the associated braking action which is determined using the Runway Condition Assessment Matrix (RCAM) [3]. Airport operators utilize the RCAM to evaluate the level of contamination in each third of the runway and assign a Runway Condition Code (RwyCC) to each third. The RwyCC scale ranges from 6, indicating a ”Dry” condition, to 0, representing extremely slippery contaminants like wet ice.

In certain cases, FICON NOTAMs also include Pilot Reported Braking Action (or simply Pilot Report) provided by a pilot who landed within the past 15 minutes. Pilot-reported braking action ranges from ”Good” braking action to ”Nil”. The RCAM establishes a mapping between RwyCCs and pilot reports. As pilot reports are issued exclusively for a non-dry runway, a RwyCC of 6 has no associated pilot reported label, but a 5 is mapped to ”Good” and 0 is mapped to ”Nil”.

RwyCCs are not a perfect surrogate for pilot-reported braking action as they, on average, overestimate it as shown in [4]. Pilot reports themselves are not an ideal reporting mechanism due to the potential unavailability of up-to-date reports and the subjective nature of braking action experienced by individual pilots. Additionally, the current methods of assessing runway conditions do not allow for forecasting as they rely on the assessment of runway contamination and/or on pilot reports.

The availability of large amounts of aircraft landing data from flight data recorders with objective braking information, and the remarkable progress of the machine learning field in recent years, provide a unique opportunity to enhance runway condition reporting using data-driven methods. This paper introduces a supervised learning method aimed at predicting runway conditions in the form of RwyCCs using widely available information from weather reports and runway specifications. Notably, the method can be applied in a forecasting

This work was sponsored by the Federal Aviation Administration (FAA) under the Joint University Program for Air Transportation

[1] www.faa.gov/about/initiatives/talpa

[2] This overrun resulted in 12 injuries and 1 death as the aircraft crashed into car traffic beyond the runway.

context by incorporating weather forecasts as input. The method was trained on processed data for 4.9 million landings labeled with computed RwyCCs provided by Aviation Safety Technologies.

The work in this paper builds upon recent efforts in data-driven runway condition assessment. Klein-Paste et al. [5] used a rule-based model called the IRIS model that maps weather and contaminant information into pilot-reported braking action (based on SNOWTAM records [6], the ICAO analog to FICON NOTAMs). In addition to using a rule-based method, Vorobyeva et al. [7] used various ML techniques to predict RwyCCs from SNOWTAM records based on available weather and contaminant data but also the previously reported RwyCC. Zhang et al. [8] then proposed a data pipeline that fuses FICON NOTAM information with METAR and runway surface data in order to be used to predict RwyCCs and pilot reports in the future.

All of the aforementioned approaches rely on RwyCCs or pilot reports from FICON NOTAMs as their ground truth labels, which are not ideal truth labels for a data-driven method as mentioned earlier. The work in this paper is most closely related to work presented by Midtfjord et al. [9] who used Xgboost to predict whether landings would be friction limited. Friction limit is the condition whereupon applying more brake pressure during landing, the rate of deceleration does not increase and the anti-skid system is engaged. Following the identification of potentially friction-limited landings, Midtfjord et al. employed regression to predict the friction coefficient (μ), which can be correlated with pilot-reported braking action. The Friction limit and μ labels are objective as they are based on aircraft sensor information, similar to labels used in this paper. In contrast to Midtfjord et al.'s two-stage approach, the method presented in this paper used a cascading model structure and did not explicitly deal with the μ parameter.

All of the methods mentioned above share a common component; they rely on runway contamination information for primary features in addition to using easily accessible weather data and runway specifications. Access to up-to-date runway contamination information is only available minutes before the landing, which restricts these methods to be used in a nowcasting setting. In contrast, the method presented here can be used in a forecasting setting, relying solely on weather forecasts and runway specifications without requiring runway contamination information from field condition reports.

A previous iteration of this work [10] showed the possibility to predict friction-limited conditions for landings, similar to [9]. This iteration of this work expands the previous model in order to allow for the prediction of RwyCCs. More detail about the methodology and a feature importance analysis for the model presented here can be found in [4].

II. Data Processing

The work in this paper centers around a database of 4.9 million landing records provided by Aviation Safety Technologies (AST). AST specializes in processing of flight data records to derive objective and standardized measures of braking action experienced during each landing, including synthetic RwyCCs computed directly from sensor data. The AST database also contains information about the time of landing, airport and runway identifiers, and aircraft type. In addition to the AST database, the Iowa State Iowa Environmental Mesonet (IEM)[3] was used to obtain weather information about each landing in the form of Meteorological Aerodrome Reports (METAR) and runway data was retrieved from the National Airspace System Resource (NASR) 28-day subscription[4].

The goal of the data processing step was to generate labels and features for each of the 4.9 million landing records, suitable for supervised learning. Figure 1 provides an overview of the data processing workflow.

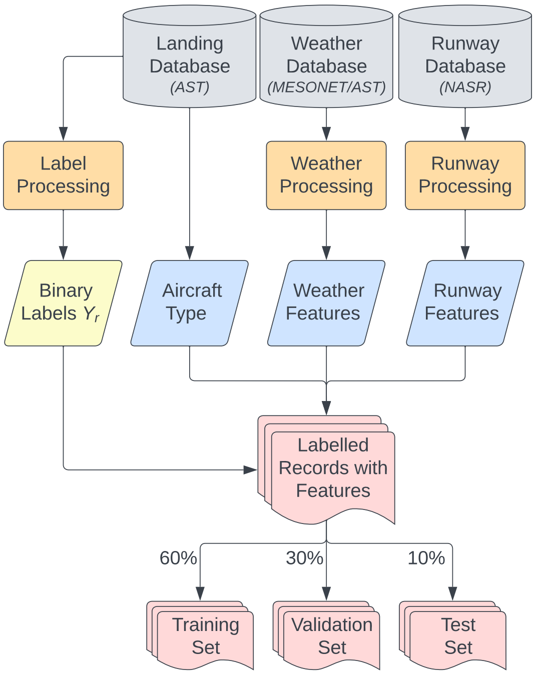


Figure 1: Data processing diagram

The truth labels used to train the supervised model were RwyCCs computed by AST, which were available for 4.9 million recorded landings between February 2017 and June 2019. AST utilizes aircraft sensor information from jet aircraft operated by various US airlines to calculate friction conditions for aircraft landings using their proprietary software. Some landings experience a friction limit, defined by AST as follows:

> *"A friction limit occurs when the pilot commands more deceleration from wheel braking but the rate of deceleration doesn't increase, e.g., the runway does not support additional deceleration due to surface contaminants, rubber build-up, feathering, etc."*

[3] www.mesonet.agron.iastate.edu

[4] https://www.faa.gov/air_traffic/flight_info/aeronav/Aero_Data/NASR_Subscription/

Landings, where a friction limit occurred, were labeled friction limited. In such cases, AST processed aircraft sensor information into a synthetic RwyCC (using proprietary software[5]) on a 0-5 scale based on how severe the lack of friction was. Non-friction limited landings were assigned a RwyCC value of 6. Among the total number of 4.9 million recorded landings, there were 8693 which were labeled as friction limited, which is less than 0.2% of all landings. This presented a severe class imbalance, requiring special approaches to be used as discussed in section III. The distribution of synthetic RwyCC values for friction-limited landings is depicted in Figure 2. The values shown are in percent of all landings to illustrate the class imbalance for each RwyCC category compared to RwyCC=6 (which makes up 99.8% of landings).

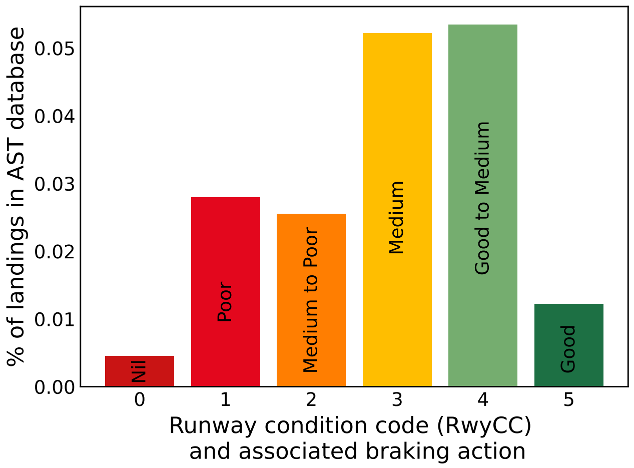


Figure 2: Distribution of Runway Condition Codes (RwyCCs) computed by AST, as a percentage of all 4.9 million landings, used to create truth labels for model training. RwyCC=6 is not shown but constitutes 99.8% (interpreted as a dry runway).

It will be shown in section III that models were not trained to explicitly predict a given RwyCC value, but rather to predict whether the RwyCC would be below or equal to a given value between 0 and 5. To enable this approach, six binary labels, denoted as Y_r, were defined for each landing:

$$Y_r = \begin{cases} 1 & \text{if } RwyCC \leq r \\ 0 & \text{otherwise} \end{cases} \quad \text{for } r \in \{0,1,2,3,4,5\} \tag{1}$$

It is worth noting that $Y_5 = 1$ is equivalent to saying that a landing is friction-limited.

A number of informative features were generated for each landing by processing data from the three databases shown in Figure 1. These features can be categorized into three groups: Weather, Runway and Aircraft Type.

The Weather features were derived from the METAR data and included:

Weather Features:

- Temperature, Relative humidity
- Pressure, Pressure altitude
- Sky coverage, Sky altitude (Cloud Ceiling)
- Visibility, Ice accretion
- Headwind & Crosswind
- Precipitation type & intensity (qualitative)
- Precipitation intensity [in/hr], Time since precip.
- Cumulative precip

The majority of the features mentioned above were directly available in the METAR reports and did not require additional processing. Categorical features, such as sky coverage, and precipitation type and intensity were one-hot encoded[6] from their corresponding weather codes. Headwind and crosswind features were generated by decomposing the wind vector using the runway's true heading.

Although a useful feature itself, precipitation intensity in [in/hr] only captured information about current precipitation at the time of landing. In order to capture information about precipitation history, a number of other features were generated by numerically integrating precipitation intensity. These features captured information about how much total precipitation descended on the runway in a given period of time before the landing. The cumulative precipitation in the N minutes before landing CP_N was calculated by integrating the precipitation intensity over the corresponding time period as shown in Equation 2:

$$CP_N = \int_{\tau_{land}-N}^{\tau_{land}} \frac{1}{60} (\text{Precip inten. at } \tau)\, d\tau \tag{2}$$

Where τ is time and τ_{land} is the time of landing. Cumulative precipitation in the 15, 30, 45, 60, 120, 180, and 1440 (1 day) minutes before landing was calculated. In Figure 3 one can see what these cumulative features would have looked like on March 12th, 2017 at Atlanta International Airport (KATL).

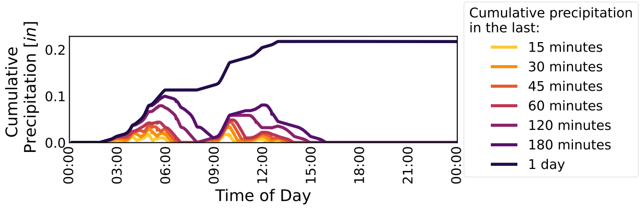


Figure 3: Cumulative precipitation features at KATL on 3/12/2017

[5] www.aviationsafetytechnologies.com/solution/methodology/

[6] One-hot encoding is a technique used to convert categorical data into numerical features that can be easily understood by machine learning algorithms. In this process, each category is represented by a set of binary features, where each feature corresponds to a specific category. These features are binary indicators, where a value of 1 represents the presence of the category and a value of 0 represents its absence. For example, if there are three sky coverage categories (clear, scattered, and overcast), they would be encoded as three binary features, with one feature active (set to 1) for each category and the rest inactive (set to 0)

The runway features used in the analysis were obtained from the 28-day NASR subscription, which provides information about runways in the United States, Canada, and the Caribbean. This meant that landing records from airports outside of these areas had to be discarded as they lacked this runway information. The database was hence limited to 4.78 million total records, including 7234 friction-limited records.

The following runway features used:

Runway Features:

- Length, Width, Elevation, Slope
- Surface type, Condition & Modification
- Weight-bearing capacity
- ILS type, Glide slope, Threshold crossing height

Categorical runway features such as surface type (e.g., asphalt, concrete), surface condition (e.g., excellent, good) and modification (e.g. grooved) were one-hot encoded to represent them as binary features.

All landing records in the database corresponded to Boeing 737 (B737) or Airbus A319-A321 series aircraft. The Boeing aircraft appeared more frequently, accounting for 88% of the records (88%). The specific aircraft types were:

Aircraft Type Features:

- B737-900, B737-800, B737-700, B737-400
- A320, A319, A321

Similar to the categorical weather and runway features, the aircraft type was one-hot encoded, resulting in seven binary features.

It is worth noting that certain additional features that could have potentially improved the model's performance were intentionally excluded. Firstly, runway contamination features, which provide information about the type and depth of contamination on the runway, were not included. Although these features would have been valuable, they are often correlated with easily observable weather features (such as precipitation history), and their exclusion was not expected to significantly hinder model performance.

Secondly, information about the aircraft's state during landing, such as the speed and weight was also omitted. This was done to allow the model to be used in a forecasting setting when the weight and speed of the aircraft are generally unknown. Previous research [4] demonstrated that including such features had only marginal performance benefits.

III. Modelling Approach

The labeled data with features were used to train a model pipeline for predicting RwyCCs. The prediction pipeline consisted of 6 classifiers arranged in a cascade, as depicted in Figure 4. Each classifier was trained to predict one of the Y_r labels, indicating whether the RwyCC for a given landing would be less than or equal to r, where $r \in \{0, 1, 2, 3, 4, 5\}$. The process started by passing the landing's features to the $RwyCC \leq 0$ model. If the model's output was "Yes", the landing was assigned a prediction of RwyCC=0. If the output was "No," the landing proceeded to the $RwyCC \leq 1$ model. A "Yes" output from this model resulted in a prediction of RwyCC=1, while a "No" output led to activation of the $RwyCC \leq 2$ model. This sequence continued for the subsequent $RwyCC \leq 2$, $RwyCC \leq 3$, $RwyCC \leq 4$, and $RwyCC \leq 5$ models. If none of the six models produced a "Yes" output, the landing was eventually assigned a RwyCC=6.

Each of the $RwyCC \leq r$ models was a classifier with an operating point chosen to balance sensitivity and false positive rate. These models were individually trained to predict Y_r labels. The training process of a single $RwyCC \leq r$ model is illustrated in Figure 5.

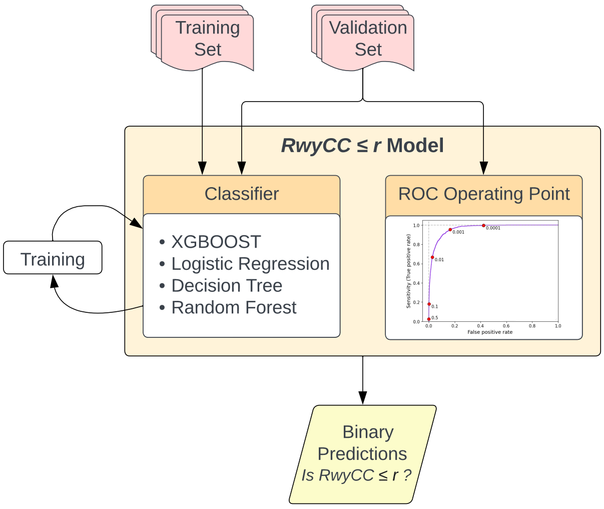


Figure 5: Single $RwyCC \leq r$ classifier training flowchart. A classifier for each $r \in \{0, 1, 2, 3, 4, 5\}$ was trained.

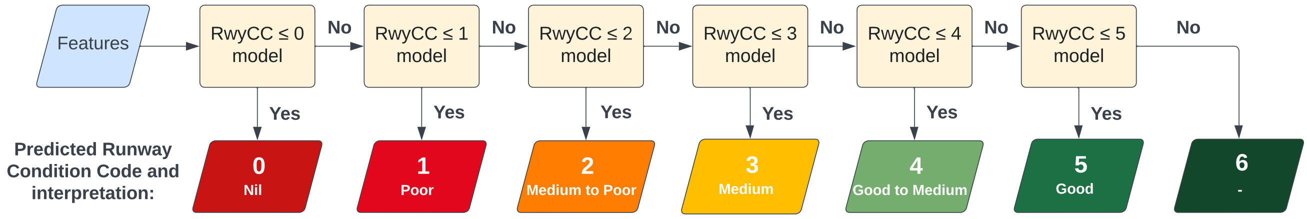


Figure 4: Cascading Runway Condition Code (RwyCC) prediction pipeline composed of 6 $RwyCC \leq r$ classifiers

For each $RwyCC \leq r$ model, training records were used to train a classifier that predicts the corresponding label Y_r based on the given features. Xgboost [11] was chosen as the classifier, but it was compared to three other classifiers: Logistic Regression, Decision Tree, and Random Forest. Each algorithm had a different set of hyperparameters[7] which were tuned using a grid search while validating on the validation records. Validation was done to maximize the Area under the Receiver Operating Curve (AUC), which is a metric invariable to class imbalance.

Using methods suited for large class imbalance was essential, as the Y_r labels were very sparse; $Y_r = 1$ values appeared in fewer than 0.2% of all records as earlier illustrated in Figure 2. Accuracy could not be used as the guiding metric for model performance as 99.8% or higher accuracy was achievable purely by guessing the majority ($Y_r = 0$) class. In a safety-critical setting such as runway condition assessment, it is imperative to detect $Y_r = 1$ cases (cases with decreased RwyCC) reliably. In other words, the correct detection of the $Y_r = 1$ class is valued more than the correct detection of the $Y_r = 0$ class. This problem is typically addressed using the Receiver Operating Curve (ROC) which is constructed for each classifier and can be used to evaluate its performance using the area under the curve (AUC). By choosing an operating point on the ROC curve it is then possible to find a suitable balance between correct detections and false alarms.

The output of a classifier is the probability of the given label is 1, given the set of input features X. Take the $RwyCC \leq 5$ model as an example. The output of the $RwyCC \leq 5$ model is $P(Y_5 = 1|X)$, with $0 \leq P(Y_5 = 1|X) \leq 1$. Since the prior probability of the label being 1 is very small ($P(Y_5 = 1)$ is small), the output $P(Y_5 = 1|X)$ tends to be small as well, and its average is close to $P(Y_5 = 1)$ (smaller than 0.2%). Typically, one would use $P(Y_5 = 1|X)$ to decide that the RwyCC for that landing should be marked as smaller or equal to 5 if $P(Y_5 = 1|X) \geq 0.5$, where 0.5 is referred to as the detection threshold t. One can see that a 0.5 threshold setting is only suitable in cases where the prediction classes are balanced. On the other hand, when the prior $P(Y_5 = 1)$ is very small, the output of the model will almost never cross 0.5 and all cases will be marked as having a RwyCC larger than 5. Since the correct detection of low RwyCC landings is of more importance than the correct dismissal of landings with high RwyCCs, the detection threshold t had to be lowered to allow for more correct detections of low RwyCC. This means that a landing was labeled as having RwyCC$\leq$ 5 if $P(Y_5 = 1|X) > t$ where $t \in [0, 0.5]$.

As the detection threshold is lowered below 0.5, the classifier makes more correct detections (higher sensitivity or true positive rate) but also experiences a higher false alarm rate (or false positive rate). The combination of t, the corresponding sensitivity, and the false positive rate is referred to as the Operating Point. Sweeping the values of t and plotting the false positive and true positive rates generates a Receiver Operating Curve (ROC) [12]. An illustration of a ROC curve with various annotated threshold t settings is shown in Figure 6.

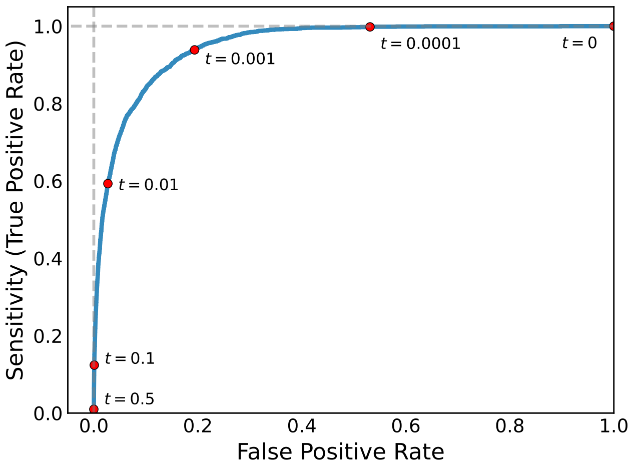


Figure 6: ROC with different threshold t settings

One can see that when t is decreased, the sensitivity and false positive rate both increase. The closer the curve is to the top left corner of the graph (100% sensitivity and 0% false alarms) the more informative the model is considered to be. Figure 7 shows an illustration of ROC curves for models of differing performance. The AUC, or the area under the curve, is the common metric that can be used to compare different models. The higher the AUC, the better the model, with the optimal value being 1.

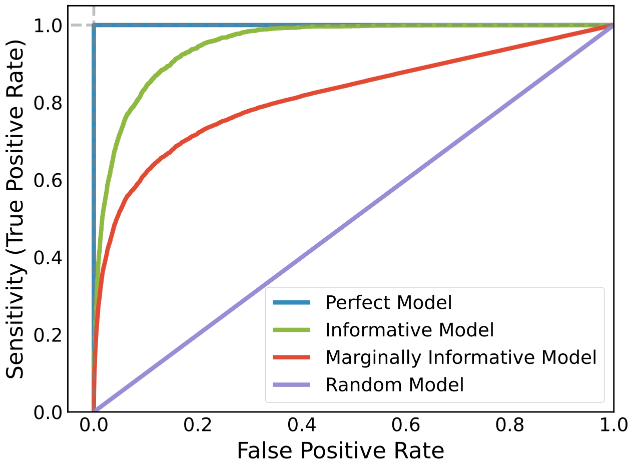


Figure 7: ROCs for models of varying performance

When the model is used in operation, a threshold t has to be chosen, which will carry an associated sensitivity and false alarm rate. There are different approaches to choosing an appropriate threshold. Swets [13] discussed how an optimal operating threshold may be found using the prior probabilities of the two classes and their relation to the slope of the ROC curve, depending on how much the operator of the model values true positives over true negatives. Kuchar [14]

[7] Hyperparameters are parameters that affect the performance of the model but are not learned through training, e.g. the depth of a tree

adopted a similar methodology to aircraft alerting systems. Alternative ways to choose an operating point are simply to set a maximum false positive rate or a minimum sensitivity.

However, these methods are only suitable for setting operating points for individual and independent classifiers. In the cascading pipeline of $RwyCC \leq r$ models, individual operating points cannot be chosen independently. This is because the nature of model errors differs from a simple binary case where only false positives and false negatives occur. In the cascading model, there are different misclassification errors with varying severity. For instance, misclassifying a sample with RwyCC=0 as RwyCC=1 is less severe than misclassifying it as RwyCC=6. Fortunately, due to the nature of the cascade, severe errors are unlikely because a sample would have to go undetected by all six classifiers.

In the cascade, earlier models are expected, but not guaranteed, to detect low braking action landings. Additionally, earlier models may detect some landings with higher RwyCCs (e.g., some RwyCC=5 landings are detected by the $RwyCC \leq 0$ model). If the false alarm rate of upstream models is very high, the downstream models may become obsolete since few samples with the corresponding RwyCC would reach them.

By individually setting the operating points of each model in the cascade, arbitrary distributions of output RwyCCs below 5 (friction limited) can be created. For instance, thresholds can be set to heavily skew the output distribution toward lower RwyCC values, resulting in a highly conservative model. Alternatively, the distribution can be skewed toward high RwyCCs, leading to a lenient model that rarely predicts low braking action. However, it is desirable for the output distribution to follow the true RwyCC distribution from AST, as shown in Figure 2.

Let R_r denote the fraction of friction-limited (RwyCC$\leq$ 5) samples where RwyCC=r. The actual R_r values read off from Figure 2 graph were: $R_0 = 0.028$, $R_1 = 0.155$, $R_2 = 0.142$, $R_3 = 0.300$, $R_4 = 0.302$ and $R_5 = 0.073$.

The threshold setting method proposed here was designed to generate predictions that maintain the shape of the RwyCC distribution for samples with RwyCC below or equal to 5 but increase their overall fraction with respect to samples with RwyCC=6. To achieve this, the parameter R_{FL} is defined, which represents the rate at which the cascade predicts friction-limited RwyCC codes (≤ 5). Due to the severe class imbalance, the fraction of samples with RwyCC=6 that are misclassified as having a lower RwyCC (false positive rate) is close to R_{FL}.

The threshold setting procedure that leverages the validation set is relatively simple. Let N_{val} be the number of validation samples. The threshold for each $RwyCC \leq r$ model is determined using Algorithm 1.

Note that X are the features for the samples that reach the $RwyCC \leq r$ model in the cascade (they do not include samples detected by upstream models). The algorithm was executed for each model in the cascade consecutively, finding the appropriate t for each model, which carried an associated false positive rate fpr and true positive rate tpr.

Since R_r is set based on the true distribution, then the only parameter that can affect the resulting t settings is the friction-limited prediction rate R_{FL}. This rate can be selected based on user preference for model conservativeness. If R_{FL} is high, the model is conservative and RwyCC predictions between 0 and 5 are more likely than when R_{FL} is low. One way to select a R_{FL} is such that the positive and negative errors of the model are balanced. Model error for a given sample was defined according to Equation 3:

$$\text{Error} = \text{Predicted RwyCC} - \text{True RwyCC} \tag{3}$$

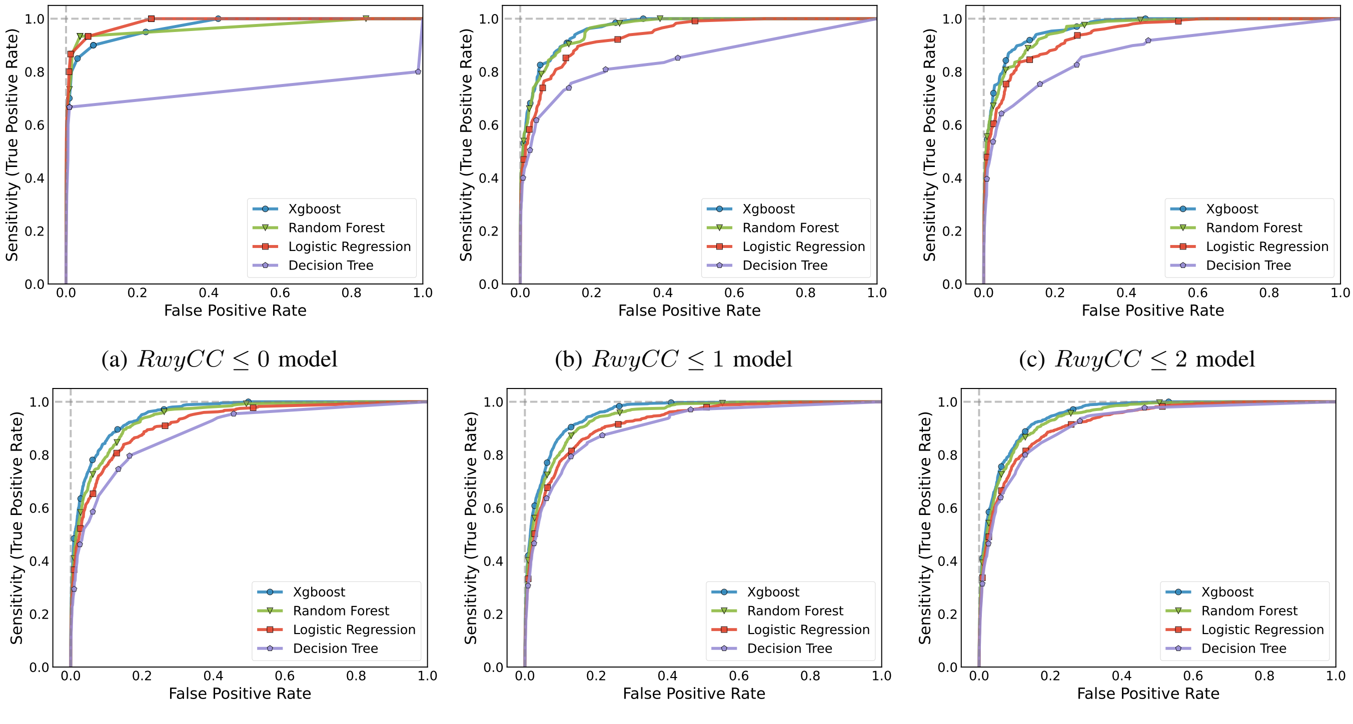
Predicting a RwyCC that is larger than the true RwyCC is referred to as a positive error. Predicting a RwyCC that is smaller than the true RwyCC is a negative error. The balance of positive and negative errors can be described by the ratio of the sum of positive errors over the sum of negative errors on all validation samples with true RwyCC$\leq$ 5.

In this case, when $R_{FL} \approx 0.075$ the error ratio was 1, meaning that positive and negative errors were balanced. This was hence the ratio that was selected. Note that there are other ways to select R_{FL}; if one is more willing to accept negative errors than positive errors, a higher R_{FL} can be chosen and vice versa. Alternatively, a cost function for different misclassification errors can be written, and the R_{FL} that minimizes cost can be chosen.

The shown method of choosing R_{FL} based on the balance of positive and negative errors is just one option among many. For more background on the threshold setting method presented here refer to [4].

Algorithm 1 Threshold Setting Algorithm

```
procedure FIND_THRESHOLD(Model ($RwyCC \leq r$), $R_r$, $R_{FL}$, $X$, $Y_r$)
    $N_{req} \leftarrow N_{val} \cdot R_{FL} \cdot R_r$            ▷ Required number of samples to detect
    $p \leftarrow$ Model.predict($X$)            ▷ Predicted probability that $RwyCC \leq r$
    $roc \leftarrow$ roc_curve($p, Y_r$)            ▷ ROC curve with thresholds $t$ in descending order
    for $fpr, tpr, t \in roc$ do
        $N_{pos}$ = Count($p \geq t$)            ▷ Number of positive predictions given $t$
        if $N_{pos} \geq N_{req}$ then return $t$            ▷ $t$ at which requirements satisfied
        end if
    end for
    return $None$            ▷ The requirements cannot be satisfied
end procedure
```



(a) $RwyCC \leq 0$ model

(b) $RwyCC \leq 1$ model

(c) $RwyCC \leq 2$ model

(d) $RwyCC \leq 3$ model

(e) $RwyCC \leq 4$ model

(f) $RwyCC \leq 5$ model

Figure 8: Classifier ROC comparison on the test set for all $RwyCC \leq r$ models.

IV. Results

The Xgboost algorithm was compared to three other classifiers: Random Forest, Logistic Regression, and Decision Tree. All $RwyCC \leq r$ models were trained for each classifier. The hyper-parameters for each algorithm were optimized by grid search, to avoid overfitting to the extent that was possible. An ROC curve comparison for each $RwyCC \leq r$ model for the test set can be seen in Figure 8. One can see from the ROC curves that Xgboost performed best across all $RwyCC \leq r$ models, except for the $RwyCC \leq 0$ model where the much simpler Logistic Regression outperformed it. This is likely due to overfitting of the Xgboost model, as there were only 122 training samples with RwyCC of 0, out of the nearly three million training samples. Nevertheless, using regularization parameters of the Xgboost algorithm it was still possible to achieve very good performance on the validation and test set, reaching an AUC of around 0.97. Since Xgboost performed best across all but one model, it was selected as the algorithm of choice for the cascade. One could instead use logistic regression for the $RwyCC \leq 0$ model and Xgboost for all the consecutive models but this unnecessarily complicates the pipeline as the two algorithms require different handling of input features and interfaces between them would have to be developed. This was rejected as it would only bring marginal improvement on the $RwyCC \leq 0$ model but complicate the pipeline significantly.

Another way to evaluate model performance is to plot the sensitivity of the cascade to samples with RwyCC≤ 5. I.e. to plot the ratio of samples with a true RwyCC≤ 5 that received a prediction RwyCC≤ 5 for each R_{FL} setting. If a sample had a true label RwyCC ≤ 5 and it was predicted any RwyCC below or equal to 5 it was considered to have been detected - it does not matter which $RwyCC$ was actually assigned as long it was not 6. The sensitivity vs. R_{FL} for each version is shown in Figure 9.

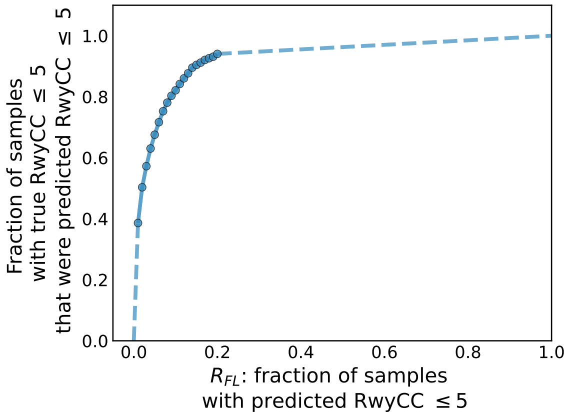


Figure 9: Sensitivity to samples with RwyCC≤ 5 vs. R_{FL} on the test set.

One may notice that this curve is reminiscent of an ROC curve. This was expected, as the extreme class imbalance between RwyCC=6 and RwyCC$\leq$ 5 means that R_{FL} was close to the false positive rate.

A. Testing on Historical Runway Excursion Events

The cascading model was further tested using historical runway excursion events. A collection of 30 runway excursion events between 2008 and 2013 was created using The Aviation Herald[8] as the main source.

Table I shows a list of the incidents used. Note that this is not an exhaustive list of runway excursions between 2008 and 2013.

For each of these excursion events, available historical METAR information was collected and converted to features used in the model. Note that in some cases, the closest METAR reports were more than an hour away from the time of the incident. Similarly, runway information was collected from airport and/or government websites. In many cases, the runway surface condition and modification at the time of the incident were unknown, in which cases the condition was assumed to be good and no surface modification was assumed.

Landings were tagged as ”Pilot Error” if the accident report from the Aviation Herald mentioned one of the following: 1. the pilot landed too fast or failed to apply enough braking, 2. the pilot landed too far along the runway. There were 13 such cases.

In addition, cases, where friction was confirmed to have been limited and/or the runway, was slippery were tagged as ”Degraded Braking”. There were 10 such cases.

The ”Model RwyCC Classification” column of Table I shows the results of applying the cascading model to this set of incidents.

One can see that more than half of the records (19/30) were classified as having RwyCC lower than or equal to 2. In fact, all ten degraded braking cases were given a RwyCC$\leq$ 2 (2 translates to ”Medium to Poor” braking action in the RCAM). This shows that the cascading model does reliably predict degraded braking cases.

Additionally, of the 11 cases that were assigned a RwyCC higher or equal to 3, seven were tagged as pilot errors. Two records that were classified as RwyCC=6 did not contain any reported ”Pilot Error” or ”Degraded Braking”. However, one of these was the landing at LPLA on 3/10/2011 which was impacted by the closure of half of the runway leading to veer-off from the runway due to ”approach aids not being aligned with the new resulting runway center line”[9]. The other case was a high-altitude (9230 ft) landing at SEQU in Ecuador on 11/30/2012[10]. One would expect the cascade to characterize this landing as at least RwyCC$\leq$ 5 due to having experienced heavy rain. However, since the model was trained on landings in the US, it may be the case that it was not able to extrapolate to extreme altitude cases such as this one in Ecuador. In fact, this is likely as the airport with the highest elevation encountered in the training set was KLAR in Laramie, WY, with an elevation of 7284 ft, almost 2000 ft lower than SEQU.

[8] www.avherald.com

[9] www.avherald.com/h?article=4391e170

[10] www.avherald.com/h?article=459d34b4

TABLE I. Collection of 30 historical runway excursion events in 2008-2013 with model runway condition code classifications.

Airport code	Date	Aircraft type	Airline	Tagged causes	Model RwyCC Classification	Interpreted braking action
EPKT	3/12/2013	737-800	Travel Service	Degraded Braking	0	Nil
YBCG	1/28/2013	737-800	Virgin Australia	Degraded Braking	1	Poor
WIOO	12/30/2012	737-400	Lionair	N/A	1	Poor
RJSY	12/8/2012	737-800	Nippon Airways	Degraded Braking	1	Poor
UWUU	11/18/2012	737-800	UTAir	N/A	1	Poor
WIOO	10/19/2012	737-400	Sriwijaya	Pilot Error	1	Poor
WIBB	7/17/2012	737-800	Garuda	N/A	1	Poor
KMDW	4/26/2011	737-700	Southwest	Pilot Error	1	Poor
WIBB	2/14/2011	737-900	Lionair	Degraded Braking	1	Poor
EGNT	11/25/2010	737-800	Thomson	Degraded Braking	1	Poor
WIOO	11/2/2010	737-400	Lionair	Pilot Error	1	Poor
VOML	5/22/2010	737-800	Air India Express	Pilot Error	1	Poor
WAUU	4/13/2010	737-300	Merpati Nusantara	N/A	1	Poor
EGPK	12/23/2009	737-800	Ryanair	Degraded Braking	1	Poor
LTBA	10/4/2009	737-300	JAT Airways	Degraded Braking	1	Poor
URRR	9/3/2013	737-800	Orenair	Pilot Error	2	Medium to Poor
WIOO	6/1/2012	737-400	Sriwijaya	Degraded Braking	2	Medium to Poor
WIBB	2/15/2011	737-900	Lionair	Degraded Braking	2	Medium to Poor
URRR	4/12/2011	737-400	Donavia	Degraded Braking & Pilot Error	2	Medium to Poor
KCMH	4/19/2013	737-800	Delta Airlines	N/A	3	Medium
CYYE	1/9/2012	737-700	Enerjet	Pilot Error	3	Medium
SYCJ	7/30/2011	737-800	Caribbean Airlines	Pilot Error	3	Medium
EHAM	10/2/2010	737-400	Corendon Air	Pilot Error	3	Medium
GUCY	7/28/2010	737-700	Mauritania Airways	Pilot Error	3	Medium
MKJP	12/23/2009	737-800	American Airlines	Pilot Error	3	Medium
GCRR	3/21/2008	737-800	Air Europa	Pilot Error	4	Good to Medium
URRR	12/1/2012	737-800	Yakutia	N/A	5	Good
SEQU	11/30/2012	737-800	Copa Airlines	N/A	6	-
URRR	4/6/2012	737-400	Globus Airlines	Pilot Error	6	-
LPLA	3/10/2011	737-800	Travel Service	N/A	6	-

V. Conclusion

This paper demonstrated a data-driven approach for runway condition assessment. The presented method was a cascading architecture of six Xgboost models trained on a database of 4.9 million landings by Aviation Safety Technologies (AST) to predict Runway Condition Code labels (RwyCCs) which had been computed by AST using aircraft sensor outputs.

The method addresses multiple limitations of traditional runway condition assessment practices. Firstly, it provides consistent and objective assessment thanks to labels based on aircraft sensor information. Assessment objectiveness is not guaranteed through pilot reports of braking action in current FICON NOTAMs as they are issued by individual pilots, each of whom may experience a landing differently. The method also does not require a physical examination of the runway by airport operators as it only relies on easily accessible up-to-date METAR data, and does not require runway contamination information. This also allows for the model to be used in a forecasting setting - where weather forecasts would be used instead of METARs. This opens the possibility for advanced runway maintenance planning.

A novel procedure was used to set the detection threshold on the ROC curve of each individual model in the cascading architecture, which maintained the shape of the true distribution of RwyCCs while detecting a significant portion of samples with reduced RwyCC and maintaining a low false positive rate. The method was successfully tested on a collection of 30 historical runway excursion events, where it predicted a "Medium to Poor" or worse braking action for all 10 cases where degraded braking was reported as one of the causes of the incident.

The model could be used at airports in a nowcasting manner together with traditional runway condition assessment to determine whether the two are consistent. It may also be used in the novel forecasting setting to examine whether it brings an operational benefit to airport operators.

Finally, note that the model can only be applied to B737 and A319-A321 series aircraft as those were the only aircraft types appearing among training samples. The method can be extended to wide-body aircraft, regional jets, and others if a sufficient number of landing samples with braking measurements becomes available.

References

[1] Airbus, "A statistical analysis of commercial aviation accidents 1958-2021," 2021.

[2] M. Jenkins and R. F. Aaron, "Reducing runway landing overruns," *Boeing Aero*, 2012.

[3] Federal Aviation Administration, "Mitigating the risk of a runway overrun upon landing," *Advisory Circular 91-79A*, 2018.

[4] M. Travnik and R. J. Hansman, "A data-driven approach for predicting and understanding braking conditions of aircraft landings," Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 2022.

[5] A. Klein-Paste, H.-J. Bugge, and A. B. Huseby, "A decision support model to assess the braking performance on snow and ice contaminated runways," *Cold Regions Science and Technology*, vol. 117, pp. 43–51, 2015.

[6] International Civil Aviation Organization, "Guidance on the issuance of SNOWTAM," 2020.

[7] O. Vorobyeva, J. Bartok, P. Šišan, P. Nechaj, M. Gera, M. Kelemen, V. Polishchuk, and L. Gaál, "Assessing the contribution of data mining methods to avoid aircraft run-off from the runway to increase the safety and reduce the negative environmental impacts," *International Journal of Environmental Research and Public Health*, vol. 17, no. 3, p. 796, 2020.

[8] W. Zhang, C. Tegen, T. Puranik, D. Anvid, R. Roy, and D. Mavris, "Fusion and analysis of data sources for assessing aircraft braking performance on non-dry runways," in *Fourteenth USA/Europe Air Traffic Management Research and Development Seminar*, 2021.

[9] A. D. Midtfjord, R. D. Bin, and A. B. Huseby, "A decision support system for safer airplane landings: Predicting runway conditions using XGBoost and explainable AI," *Cold Regions Science and Technology*, vol. 199, p. 103556, 2022.

[10] M. Travnik and R. J. Hansman, "A data driven approach for predicting friction-limited aircraft landings," in *AIAA SCITECH 2022 Forum*, 2022.

[11] 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*, ser. KDD '16. Association for Computing Machinery, 2016, p. 785–794.

[12] C. E. Metz, "Basic principles of roc analysis," *Seminars in Nuclear Medicine*, vol. 8, no. 4, pp. 283–298, 1978.

[13] J. A. Swets, "The science of choosing the right decision threshold in high-stakes diagnostics." *The American psychologist*, vol. 47 4, pp. 522–32, 1992.

[14] J. K. Kuchar, "Methodology for alerting-system performance evaluation," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 2, pp. 438–444, 1996.