Prediction of Weather-induced Airline Delays Based on Machine Learning Algorithms

Sun Choi, Young Jin Kim, Simon Briceno and Dimitri Mavris
Aerospace Systems Design Laboratory
Georgia Institute of Technology
Atlanta, Georgia 30332–0250

***Abstract*—The primary goal of the model proposed in this paper is to predict airline delays caused by inclement weather conditions using data mining and supervised machine learning algorithms. US domestic flight data and the weather data from 2005 to 2015 were extracted and used to train the model. To overcome the effects of imbalanced training data, sampling techniques are applied. Decision trees, random forest, the AdaBoost and the k-Nearest-Neighbors were implemented to build models which can predict delays of individual flights. Then, each of the algorithms' prediction accuracy and the receiver operating characteristic (ROC) curve were compared. In the prediction step, flight schedule and weather forecast were gathered and fed into the model. Using those data, the trained model performed a binary classification to predicted whether a scheduled flight will be delayed or on-time.**

I. Introduction

According to the 'Bureau of Transportation Statistics (BTS)', approximately twenty percent of the entire scheduled commercial flights are delayed. Airline delays cost airlines multi-billion dollars per year and cause a great inconvenience to passengers. BTS has categorized airline delays into five main causes, which are air carrier, extreme weather, National Aviation System, late-arriving aircraft and security [1].

Weather is not only one of the main reasons of delays but it is also closely related to other categories, indeed. For example, National Aviation System category can include delays due to the re-routing of flights by inclement weather. Besides, weather is also a factor affecting late-arriving aircraft although airlines don't report the causes as weather. By considering those facts, weather's percentage share accounts for about 40% of total delay minutes [1]. Thus, study on the influence of inclement weather on airline delays is essential for efficient flight operations. Furthermore, a decision support tool built on the study can inform passengers and airlines about weather-induced delays in advance and help them reduce possible monetary losses. For this purpose, the classification model to predict weather-induced delays of individual flights is proposed in this study.

On-time performance of flights has been an important research subject as demands for air travel increase. Thus, several attempts were there to discover patterns in air traffic. Rebello and Balakrishnan have created variables indicative of the state of the NAS. And they predicted network-related delays of the future by utilizing the system-level dependencies among airports [2]. Zonglei et al. have trained classification models to find how serious daily delays are at the hub-airport of China in the future [3]. In contrast, Klein et al. have focused more on weather and presented a delay prediction model established on metric called Weather Impacted Traffic Index (WIFI) which measures the severity and the impact of weather [4]. However little research exists that focuses on weather-induced delay prediction of individual flight by utilizing machine learning.

In this research, we had focused on arrival delays of individual flights using supervised machine learning algorithms. There are several reasons to explain why machine learning was tried. First of all, the volume of historical flight and weather data are too large to analyze analytically. Moreover, relationships between causal factors and delay or even correlations among factors are extremely complicated and highly nonlinear to test all hypothesis. Machine learning is able to develop models vigorously with huge amount of dataset and it has the ability to discover and display the hidden patterns in the data. In summary, machine learning is a clever method that can address problems in analytical analysis with big data.

To predict delays of individual flights, supervised machine learning algorithms were implemented with features including flight schedules and weather conditions at the origin and the destination. In order to increase the predictive capability, models were trained for individual origin-destination (OD) pair not for the entire National Airspace System (NAS) by capturing each airport's weather characteristics originated from geographic location. In the prediction step, flight schedule information combined with weather forecasts was fed into the model to get the predictions on scheduled flights not yet flown.

The paper is organized as follows: Section II provides brief explanation on methodologies used in this study and Section III describes how the model is built. In Section IV, classifiers' performance are investigated in terms of accuracy and ROC. Lastly, conclusions and future work are presented in Section V and they concludes the paper.

II. Methodology

A. Sampling Techniques: SMOTE

The dataset is described as imbalanced when the classification categories are not approximately equally represented [5]. According to this criteria, the training data is imbalanced because the number of on-time flights is three to four times more than that of delayed flights. It means that one can have prediction accuracy over 75% even though every flight is

978-1-5090-2523-7/16/$31.00 ©2016 IEEE

classified as on-time. In real world, identifying the minority class is required as it is costly to misclassify examples from the minority class [6]. To address issues emerged from imbalanced data, examples of minority class should be generated synthetically to adjust the distribution between majority and minority class.

Synthetic Minority Over-sampling TEchnique (SMOTE) is an over-sampling approach that creates synthetic minority class examples. The minority class is over-sampled by introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors [5]. SMOTE is a powerful solution to imbalanced dataset addressing drawbacks of the over-sampling and under-sampling. It has been shown that SMOTE improves the performance of classifiers in various application domains [7], [8]. The combination of SMOTE and random under-sampling was used in the training step as it resulted in more precise estimation than other sampling methods [5].

B. Supervised Learning Classifiers

In this study, following four different algorithms were applied and their performance were evaluated.

1) Decision Trees (DT): Decision trees recursively partitions the input dataset at a node for a randomly chosen attribute. At an each leaf node, a local model storing the distribution over class labels is defined. In the end, the model for the input variable x can be written in the following form,

$$f(\mathbf{x}) = \mathbb{E}[y|\mathbf{x}] = \sum_{m=1}^{M} w_m \phi(\mathbf{x}; \mathbf{v}_m)$$

where w_m is the distribution over class labels in the m^{th} region and $\mathbf{v}_m$ encodes the choice of variable to split and the threshold value on the path from the root to the m^{th} leaf [9].

2) Random Forest (RF): Random forest is an ensemble of many individual decision trees [10]. It builds a large collection of de-correlated trees which are noisy but unbiased, and averages them to reduce the variance. Random forest obtains a class vote from each tree, and then classifies a sample using majority vote [11]. Let $\hat{C}_b(x)$ be the class prediction of the b^{th} tree, then the class obtained from random forest, $\hat{C}_{rf}(x)$, is

$$\hat{C}_{rf}(x) = \text{majority vote}\{\hat{C}_b(x)\}_1^B$$

3) AdaBoost: AdaBoost is a method of converting weak classifier into highly accurate prediction rule. It learns week learner on weighted example set sequentially and combines weak hypotheses linearly. Produced sequence of classifiers is dependent on the previous one and focuses on the previous one's errors. Samples that are incorrectly predicted in the previous classifiers are chosen more often or weighted more heavily when estimating a new classifier [12].

4) k-Nearest-Neighbors Classifier (kNN): To assign a label to the point x, the k-Nearest-Neighbors classifier draws a sphere centered on x enclosing exactly k training points. Afterwards, it examines the label of k training points closest to x. Then the label having the largest vote is assigned to the test point x [12]. Despite its simplicity, the k-Nearest-Neighbors classifier has been successful where the decision boundary is very irregular.

C. Model Evaluation

1) 10-fold Cross Validation: Cross validation is a method that can estimate model's accuracy on unseen data. A dataset is divided into 10 approximately equal-sized subsets. The k^{th} ($k = 1, ..., 10$) subset is chosen to be a validation set and the rest subsets are utilized as a training set. At last, the cross-validation error (CV) can be calculated from the following equation:

$$CV = \frac{1}{10} \sum_{i=1}^{10} L(y_i, \hat{f}^{-\kappa(i)}(x_i))$$

where $\hat{f}^{-\kappa(i)}(x_i)$ is the fitted function, $\kappa : \{1, ..., N\} \rightarrow \{1, ..., K\}$ is an indexing function that i^{th} observation is allocated to the k^{th} partition and y_i is the target value [11].

2) Receiver Operating Characteristic(ROC) Curve: The Receiver Operating Characteristic (ROC) curve is the plot for a binary classifier's performance. It illustrate True Positive Rate (TPR) vs. False Positive Rate (FPR) for a set of threshold, τ. Let $\delta(x) = I(f(x) > \tau)$ be the decision rule, where $f(x)$ is a measure of confidence that $y = 1$. Then, TPR, also known as the sensitivity, and FPR are computed as follows [9].

$$TPR = \frac{TP}{(TP + FN)} \approx p(\hat{y} = 1|y = 1)$$

$$FPR = 1 - \text{sensitivity} = \frac{FP}{(TN + FP)} \approx p(\hat{y} = 1|y = 0)$$

The ideal point on the ROC curve is (0,1), that is all positive examples are classified correctly and no negative examples are misclassified as positive [5]. Plus, the Area Under the ROC Curve (AUC) is another strict measure of the prediction performance. The larger AUC represents the better prediction.

III. Prediction Model

An overview of the model developed to predict delays of individual flights is shown in Fig. 1. The model consists of two main parts, the training process and the prediction process. The training process starts with data collection. Historical flight data and weather data are collected and they are joined together using the scheduled departure time and airport as the join keys. In the preprocessing step, estimating missing data and normalization are performed. Then the training set is finally ready and it is used to train the predictive model with sampling techniques. Data for the prediction process is collected and preprocessed in the same way as the training set. After that it is fed into the model trained with the training data. In the end, the model assigns each data point a label.

The methodologies used are discussed in the previous section and the remaining explanation about the model is presented in the following section.

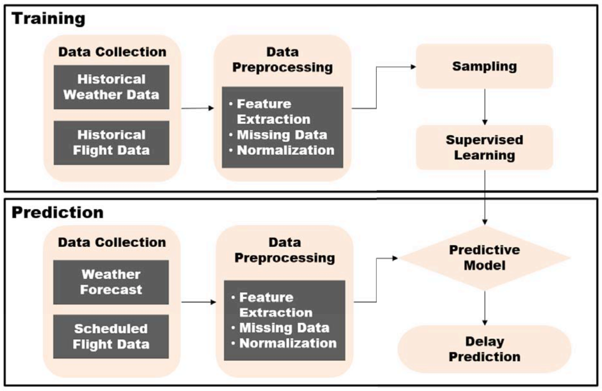


Fig. 1: Summary of the model developed

A. Data Collection

US domestic airline traffic data and weather data from 2005 to 2015 are obtained from the Bureau of Transportation Statistics (BTS)' Airline On-time Performance dataset and National Oceanic and Atmospheric Administration (NOAA)'s Integrated Surface Database, respectively. The BTS' dataset contains on-time arrival performance data for non-stop domestic flights served by major air carriers. It also provides additional information such as origin/destination airports, flight numbers, flight schedules and delay times [13]. NOAA's database contains weather information including wind, cloud height, visibility, temperature, pressure, precipitation, etc. reported approximately hourly at worldwide stations [14].

2016's Flight schedule and weather forecast are collected via API and merged to create a test set for the prediction process. Current flight data not yet released from the BTS' is available in FlightStats APIs [15]. Also, weather forecast is obtained from World Weather Online API. The forecast for the origin and the destination airport at scheduled departure and arrival time is accessible through their weather APIs [16].

B. Data Preprocessing

Air traffic data for 45 major airports and corresponding weather data are extracted. Following the rules of BTS, flights that arrive at the gate within 15 minutes of the scheduled time are considered as on-time. Canceled and diverted flights in the training set are deemed as delayed. To deal with missing values in weather data, linear interpolation is used with two adjacent known values.

The following data fields were extracted from BTS' dataset for every scheduled flight because those are factors having impacts on flight delays [17].

- Quarter of Year
- Month
- Day of Month
- Day of Week
- Departure and Arrival Schedule in Local Time
- Arrival Delay Indicator: 0 if actual arrival time minus scheduled arrival time is less than 15 minutes, 1 if actual arrival time minus scheduled arrival time is greater than or equal to 15 minutes

It is known that delays are occurred in association with convective weather at the terminal area. Also, low ceiling/visibility conditions, high surface winds and precipitations make an aircraft landing difficult [18], [17]. Extracted weather fields reflecting these facts are as follows.

- Wind Direction Angle [deg]
- Wind Speed Rate $[m/s]$
- Visibility $[m]$
- Precipitation $[mm]$
- Snow Depth $[cm]$
- Snow Accumulation $[cm]$
- Weather Intensity Code
  1:Light, 2:Moderate, 3:Heavy, 4:Vicinity
- Weather Descriptor Code
  1:Shallow, 2:Partial, 3:Patches, 4:Low Drifting, 5:Blowing, 6:Showers, 7:Thunderstorms, 8:Freezing
- Precipitation Code
  1:Drizzle, 2:Rain, 3:Snow, 4:Snow Grains, 5:Ice Crystals, 6:Ice Pellets, 7:Hail, 8:Small Hail and/or Snow Pellets, 9:Unknown Precipitation
- Obscuration Code
  1:Mist, 2:Fog, 3:Smoke, 4:Volcanic Ash, 5:Widespread Dust, 6:Sand, 7:Haze, 8:Spray
- Other Weather Code
  1:Well-Developed Dust/Sand Whirls, 2:Squalls, 3:Funnel Cloud, Tornado, Waterspout, 4:Sandstorm, 5:Duststorm
- Combination Indicator Code
  1:Not part of combined weather elements, 2:Beginning elements of combined weather elements, 3:Combined with previous weather element to form a single weather report

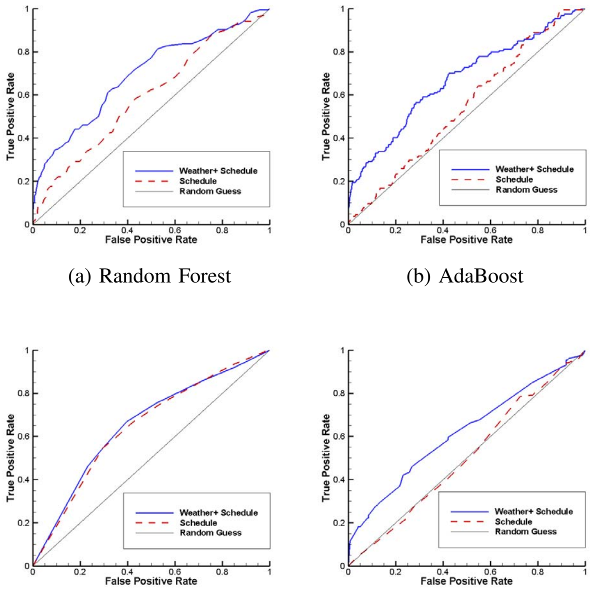
By preprocessing, all of categorical variables are converted to numerical variables since machine learning algorithms exhibit better performance with numerical variables. Furthermore, normalization scales the range of features and prevents the model from being dominated by a few features. In the end, the merged training dataset after preprocessing has approximately two millions of flights in line with weather and its size reaches three gigabytes.

C. Classification Model

Classification model for an origin-destination(OD) pair was trained with flight data and weather data to predict arrival delays of individual scheduled flights. Parameters for each classifier were tuned by trial-and-error. As a results, the number of trees in random forest was 100, the maximum number of estimators and learning rate for AdaBoost were 100 and 1, the number of neighbors for k-Nearest-Neighbors was 6 and the maximum depth of decision trees was 7.

IV. Results and Analysis

In this section, the classification model typically trained on flights departing from Denver International Airport(DEN) and



(a) Random Forest (b) AdaBoost

(c) kNN (d) Decision Trees

Fig. 2: Receiver Operating Characteristic with and without Weather Data

arriving in Charlotte Douglas International Airport(CLT) is evaluated.

A. Impacts of Weather Data

First of all, the impacts of weather data on the prediction performance of four models are investigated. In the series of graphs in Fig. 2, a red dotted line is the model trained only with schedule data while a blue solid line is the model trained with weather data as well as schedule data. A diagonal line represents random guessing, so the closer ROC curve to the diagonal, the less accurate.

In case of Fig.2(a), Fig.2(b) and Fig.2(d), weather data distinctly improved predictive ability of each model since blue solid lines are closer to the ideal point on the ROC curve than red dotted lines. Without weather data, prediction performance of AdaBoost and decision trees were about the same level as random guess'. In other words, AdaBoost and decision trees could not tell whether a scheduled flight would be delayed or not without weather data. However, kNN's gap between two curves with and without weather data is quite small compared to other classifiers. The reason of this is the curse of dimensionality. The number of features in the flight data reaches 56 and there are countless combinations of variables. In such high-dimensional data, measuring distance between sample points becomes meaningless [19] and including weather data amount to 80 features is not helpful for kNN to enhance predictive capability.

B. Impacts of Sampling Techniques

As it is already explained in Section II, SMOTE and random under-sampling were utilized to balance the number of 'delayed' samples against that of 'on-time' samples. To analyze impacts of sampling techniques, the model is trained with and without sampling techniques and the prediction performance of those two cases are compared. Table I shows accuracy and elapsed time of four classifiers trained with and without sampling techniques.

TABLE I: Training Accuracy and Elapsed Time

(a) with Sampling Techniques

Classifier	Accuracy (%)	$\text{Time}_{\text{elapsed}}$(sec)
Random Forest	81.37	8
AdaBoost	78.05	12
kNN	61.69	2
Decision Trees	77.02	0

(b) without Sampling Techniques

Classifier	Accuracy (%)	$\text{Time}_{\text{elapsed}}$(sec)
Random Forest	83.40	9
AdaBoost	83.21	12
kNN	82.42	2
Decision Trees	82.84	0

Sampling techniques' influence on the prediction performance could be figured out in comparison between Table I(a) and I(b). The accuracy of classifier trained without sampling techniques was higher than that trained with sampling techniques for all of four classifiers. However it does not imply that applying sampling techniques is a bad choice. Classifiers are biased toward 'on-time' class when they are trained on imbalanced data and it is easier for classifiers to predict 'on-time' class. Further, it is more likely to classify delayed flight as on-time.

Let us examine random forest's predictive performance further. Its classification results on validation set composed of 8833 flights are in Table II. In the table, the main diagonal is the number of flights that are predicted correctly and the antidiagonal components are incorrectly predicted flights. Even though sampling techniques lowered the sum of the main diagonal, i.e. prediction accuracy, it increased the model's true positive rate (TPR). In order words, the model's minority class recognition was improved and it means that the sensitivity of the model was increased with sampling techniques.

In Table II, false positive rate (FPR) is the proportion of on-time flights that are incorrectly classified and the percentage of delayed flights that are classified as on-time is false negative rate (FNR). Comparison between FPR and FNR of random forest revealed that the model is more likely to misclassify delayed flights as on-time than vice versa. A considerable number of delayed flights were not able to be captured by the model despite decreased FNR with sampling techniques. This is because the model fitted in this study didn't take into account other possible causes of delays than weather. If delays are occurred due to the fact that is not weather, it is hard to be predicted by the model. For example, in case of a flight arrived late because of congestion in air traffic, the model might not recognize it since it is nothing to do with weather.

TABLE II: Confusion Matrix of Random Forest

(a) with Sampling Techniques

	Predicted On-time	Predicted Delay
Actual On-time	6838	418
Actual Delay	1231	346

(b) without Sampling Techniques

	Predicted On-time	Predicted Delay
Actual On-time	7178	78
Actual Delay	1388	189

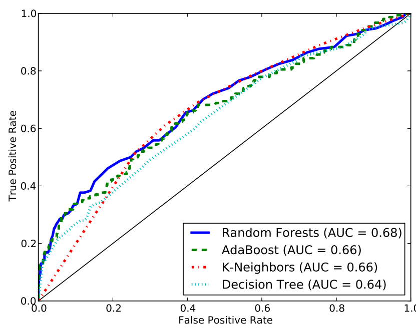


Fig. 3: Receiver Operating Characteristic of Classifiers

C. Comparison between Methods

10-folds cross-validation was performed to estimate test accuracy of the trained model. Table I(a) exhibits training accuracy and elapsed time of four classifiers obtained by 10-folds cross-validation. Random forest did best in classification showing the highest accuracy within a reasonable amount of time. However, evaluating performance of classifiers only based on predictive accuracy is not sufficient when the data is imbalanced [5]. Like we already figured out in Section II, a classifier can have high accuracy without paying special attention to the minority class. Hence, it is more appropriate to use a receiver operating characteristic (ROC) curve and the Area Under the ROC curve (AUC) for the case of imbalanced datasets instead.

The ROC curves and the AUC values of four classifiers are shown in Fig. 3. They support the truth that random forest has the best ability to distinguish delays from on-time flights among four classifiers. Because it has the largest AUC in Fig. 3 and is also the farthest one from the diagonal line. The prediction accuracy, however, does not always coincide with the ROC curves' trend. In Table I(a), the k-Nearest-Neighbors classifier is the poorest in terms of accuracy while decision trees has the lowest AUC in Fig. 3. In confusion matrix, the k-Nearest-Neighbors had relatively high TPR and high FPR at the same time. On the other hands, decision trees classifier's FPR and TPR were low. The preferable classifier could be determined by the costs of false positive and false negative. If false negative is costly, a classifier with high TPR is a better choice.

Besides, it is noteworthy that decision trees' ROC is below the diagonal line at the top-right corner of Fig 3. The diagonal line connecting (0,0) and (1,1) means random guess. And a threshold of $+\infty$ matches up with (0,0) and $-\infty$ produces (1,1). What these mean is that decision trees' prediction performance was worse than random guess for low threshold values. Lower threshold value would make classifiers yield more 'delayed' class. Meanwhile, they were expected to have better accuracy with the higher threshold. kNN's ROC, however, is quite close to the diagonal with high threshold value and this is because of the curse of dimensionality as explained before. All other three ROC curves had trends of going closer to the diagonal line as the threshold decreases. Decision trees was particularly sensitive to the threshold and its ROC even exhibited s-shaped curve.

TABLE III: Test Accuracy
and Width of Decrease from Training Accuracy

Classifier	Accuracy (%)	Decrease (%)
Random Forest	80.36	1.01
AdaBoost	71.43	6.62
KNN	35.71	25.98
Decision Trees	64.29	12.73

D. Assessment of Performance on Test Data

How well classifiers perform on unseen data is another issue. For the purpose, the classifier was trained exactly as before. Then it is get tested with a test set consists of 56 flights departing from DEN and arriving at CLT during a week, May 20 to 26, 2016. Test performance of four classifiers has shown in Table III. The last column of the table is width of decrease from training accuracy in Table I(a). All classifiers performed better on the training data and predicting unseen data was harder for them. Four classifiers' performance degraded with test set because they overfitted modeling every minor variations in the input.

The ultimate goal of the model is to predict delays of scheduled flights not yet flown with weather forecast. Hence, random forest classifier's behavior with 5 days/1 day forecast horizon was assessed. The test results are in Table IV. Table IV(a) and Table IV(b) are prediction results obtained with five days forecast horizon and one day forecast horizon, respectively. And Table IV(c) is the results from actual weather. In case of the results with forecast, there could possibly be the error arose from forecast uncertainty combined with model's pure error. Classification results attained from actual weather was required to differentiate uncertainty in forecast from the model's pure error. In Table IV, the predictions with forecast were much worse than the predictions with the actual weather. The model's predictive performance is drastically lowered due to uncertainty in forecast. The results with the actual weather exhibited higher accuracy as uncertainty in forecast had dropped out. Nonetheless, actual weather did not provide perfect prediction. Prediction error can arise from two main sources. First one is the limitation of the current model. The other one is that delayed flights could not be captured by the model since delays are caused by non-weather-related factors.

TABLE IV: Confusion Matrix of Random Forest with Forecast Horizons

(a) 5 days Forecast Horizon, Accuracy = 26.79%

	Predicted On-time	Predicted Delay
Actual On-time	7	39
Actual Delay	2	8

(b) 1 Day Forecast Horizon, Accuracy = 30.36%

	Predicted On-time	Predicted Delay
Actual On-time	7	39
Actual Delay	0	10

(c) 0 Day Forecast Horizon, Accuracy = 80.36%

	Predicted On-time	Predicted Delay
Actual On-time	44	2
Actual Delay	9	1

V. Conclusion

This study proposed a prediction model enabled to classify airline delays caused by inclement weather condition. In particular, the model was built on historical weather and traffic data of individual OD pair by utilizing machine learning algorithms. Supervised machine learning algorithms implemented in this study includes random forest, AdaBoost, k-Nearest-Neighbors and Decision Trees. Because the data was imbalanced, the combination of SMOTE and random under-sampling were applied. The model's prediction performance on the validation set and the test set was analyzed.

There are still possible approaches that can improve the model in the future. If the costs of false positive and false negative are taken into account, preferred performance of classifiers could be clearly determined. Then it could be a solid foundation for a decision support tool for predicting aircraft arrival. Also a thorough consideration of uncertainty in forecast would enhance the model's predictive performance.

References

[1] U. D. of Transportation, "February 2016 on-time performance up from previous year," jan 2016.

[2] H. B. Juan Jose Rebollo, "Characterization and prediction of air traffic delays," *Transportation Research Part C: Emerging Technologies*, vol. 44, pp. 231–241, 2014.

[3] Z. G. Lu Zonglei, Wang Jiandong, "A new method to alarm scale of flight delay based on machine learning," in *Proceedings of International Symposium on Knowledge Acquisition and Modeling*, 2008.

[4] R. S. L. Alexander Klein, Chad Craun, "Airport delay prediction using weather-impacted traffic index (witi) model," in *Preceedings of the Digital Avionics Systems Conference(DASC) 29th*, 2010.

[5] L. O. H. W. P. K. Nitesh V. Chawla, Kevin W. Bowyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[6] R. M. J. S. R. A. V Garca, J.S. Snchez, "The class imbalance problem in pattern classification and learning," in *II Congreso Espanol de Informtica, CEDI 2007*, 2007, pp. 283–291.

[7] N. V. Chawla, "C4.5 and imbalanced data sets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure," in *Proceedings of the ICML, Workshop on Learning from Imbalanced Datasets II*, Washington DC, 2003.

[8] S. S. P. Reshma C. Bhagat, "Enhanced smote algorithm for classification of imbalanced big-data using random forest," in *Proceedings of the Advance Computing Conference (IACC)*, 2015.

[9] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[10] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[11] J. F. Trevor Hastie, Robert Tibshirani, *The Elements of Statistical Learning*. Springer, 2008.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[13] B. of Transportation Statistics, "Airline on-time data," 2016.

[14] R. V. Adam Smith, Neal Lott, "The integrated surface database: Recent developments and partnerships," *Bulletin of the American Meteorological Society*, vol. 92, pp. 704–708, 2011.

[15] (2016, may) Flightstats. [Online]. Available: https://developer.flightstats.com/

[16] (2016, may) World weather online. [Online]. Available: http://developer.worldweatheronline.com/api/

[17] Y. B. X. L. M. M. Mohamed Abdel-aty, Chris Lee, "Detecting periodic patterns of arrival delay," *Journal of Air Transport Management*, vol. 13, pp. 355–361, 2007.

[18] J. S.S.Allan, S.G.Gaddy, "Delay causality and reduction at the new york city airports using terminal weather information system," Lincoln Laboratory, Massachusetts Institute of Technology, Tech. Rep., 2001.

[19] M. Pechenizkiy, "The impact of feature extraction on the performance of a classifier: knn, nave bayes and c4.5," in *18th CSCSI Conference on Artificial Intelligence AI05*. Springer Verlag, 2005, pp. 268–279.