LOW-LEVEL WIND SHEAR ALERT PREDICTION SYSTEM USING MACHINE LEARNING

EJHA LARASATI SIADARI
ATMS 600

TABLE OF CONTENTS

01 INTRODUCTION

02 DATA

03 METHODS

04 RESULTS

05 CONCLUSIONS

LLWS ALERT PREDICTION

Enhance aviation safety by improving the early detection and prediction of low-level wind shear conditions:

- Wind shear detection in the next one hour
- Within 15 minutes windows
- Utiliizing 10 minutes data points

https://www.miami-airport.com/about_us.asp

01

INTRODUCTION

The location of Soekarno-Hatta airport is in Tangerang, Indonesia consisting of 2 runway zones.

- LLWAS Wind shear warning data is generated by analyzing the divergence of wind speed and direction data using an algorithm developed by Wilson (1991).
- LLWAS considers an area as a wind shear zone when its divergence value exceeds the threshold.

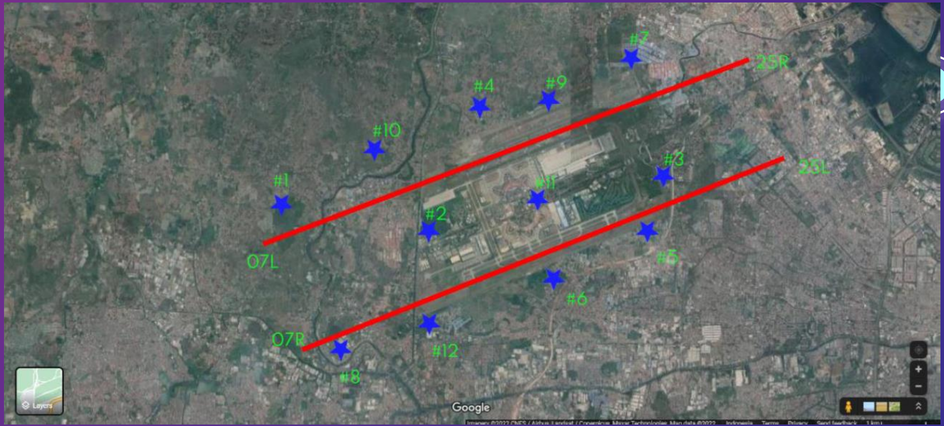


Fig. 1. LLWAS anemometer placement at Soekarno-Hatta airport (Ryan, et.all, 2022).

“Since there are 2 runways in Soekarno-Hatta airport, LLWAS divides the runway area into 4 parts for each end of the existing runway.
.”

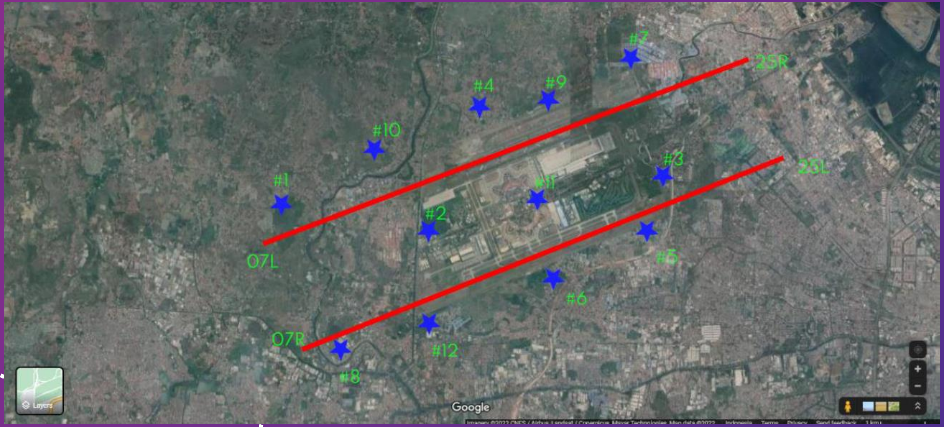


Fig. 1. LLWAS anemometer placement at Soekarno-Hatta airport (Ryan, et.all, 2022).

02

DATA

Raw Data

LLWAS Warning

Every 10 seconds from 8 parts of the runway over a period of 40 days, from April 1st to May 10th, 2018

Anemometer sensors

Contained measurements of wind speed and direction at a height of 10 meters above ground level from 12 sensors collected over a period of 40 days, from April 1st to May 10th, 2018

PREPOCESSING

Cleaning

cleaning the data—30% data was discarded. discarded all of the sequences that have missing values

U and V conversion

converting the wind direction to x-y components of wind

Interpolation

if it is only missing 10 seconds will be linearly interpolated

10 minutes data points

1 hour

15 mins interval

Current time

10 mins interval of predictors (u,v, components of wind from 12 sensors) as our input (x)

TARGETS

there is a gap of one hour, and within 15 mins interval, we look if there is warning or not, 0 or 1 (target, y)

352,601

Total dataset, 0.014% of this dataset consists of wind shear warning labeled data, with the remainder being labeled as no wind shear warning

Undersampling

- All the wind shear warning labeled data was included, while an equal number of randomly selected no wind shear warning labeled data were also included.
- Before doing under sampling, the negative values are 345601 and our positive values corresponds to rwy_25 ra with the most positive values are 416.
- After under-sampling, the total dataset used for each scheme is 332195, 80% training, 20% test.

03

METHOD

Temporal Convolutional Network (TCN)

TCN is a one-dimensional Convolutional-based neural network model for temporal data (Lea et al., 2016)

TCN has gained popularity due to its ability to capture long-term dependencies in sequential data while maintaining parallelism and computational efficiency.

At its core, TCN utilizes one-dimensional dilated convolutions to process sequential data. This dilation factor can be increased progressively in deeper layers of the network, enabling TCN to capture both local and global patterns in the data.

Model Evaluation

RECALL

also known as true positive rate or sensitivity, measures the proportion of actual positive cases (occurrence of wind shear) that are correctly identified by the model.
TP / (TP + FN)

PRECISION

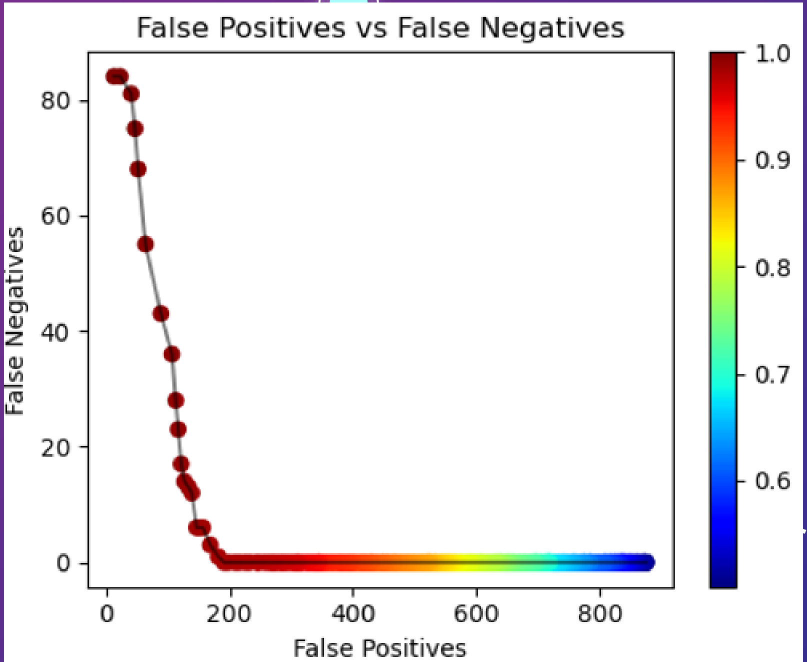
quantifies the accuracy of positive predictions made by the model.
TP / (TP + FP)

ACCURACY

measures the overall correctness of the model's predictions (both true positives and true negatives) over the total number of predictions.
(TP + TN) / (TP + FP + TN + FN)

Trade off between FP and FN

- False negatives occur when the wind shear prediction system fails to identify its presence.
- Undetected wind shear increases the risk of accidents and compromises flight safety.



ACCURACY

in the context of wind shear prediction, the occurrence of wind shear events is often rare compared to non-wind shear conditions. This creates an imbalanced dataset where the majority of instances belong to the non-wind shear class. In such cases, a prediction model that always predicts the majority class (non-wind shear) will achieve high accuracy but fail to detect the critical wind shear events.

Best thresholds

best threshold is the threshold value that results in zero false negatives. The code calculates the best threshold by finding the first threshold value where the false negative is zero.

04

RESULTS

1. version1- using u and v components of wind, by dropping the missing values

baseline_acc, baseline_fp, baseline_fn

```
In [23]: # baseline

         # TN | FP
         # ---|---
         # FN | TP
         baseline = np.zeros(len(target))
         confusion_matrix(target, baseline)

Out[23]: array([[20418,     0],
                [   20,     0]], dtype=int64)
```

- (99.90214306683629, 0, 20)
- This means that the baseline accuracy of the model is 99.90%. The baseline number of false positives is 0, since we are predicting all negatives. The baseline number of false negatives i

best_threshold, val_acc

- (0.548, 99.88746452686172)

```
In [22]: # best threshold with n false negatives
         n = 0
         best_threshold = thresholds[np.where(fn == n)[0]][-1]
         print(best_threshold)

         # TN | FP
         # ---|---
         # FN | TP
         confusion_matrix(target, probs > best_threshold)

         0.548

Out[22]: array([[20395,    23],
                [    0,    20]], dtype=int64)
```

Version1	
Metrices	Values
Recall	1.0
Precision	0.465
Accuracy	0.9989

2. version2- using u and v components of wind, linearly interpolated missing data

baseline_acc, baseline_fp, baseline_fn

```
In [24]: # baseline

         # TN | FP
         # ---|---
         # FN | TP
         baseline = np.zeros(len(target))
         confusion_matrix(target, baseline)

Out[24]: array([[66355,     0],
                [   84,     0]], dtype=int64)
```

- The output is ((99.873568, 0, 84)
- This means that the baseline accuracy of the model is 99.87%. The baseline number of false positives is 0, since we are predicting all negatives. The baseline number of false negatives is 84.

best_threshold, val_acc

```
In [22]: # best threshold with n false negatives
         n = 0
         best_threshold = thresholds[np.where(fn == n)[0]][-1]
         print(best_threshold)

         # TN | FP
         # ---|---
         # FN | TP
         confusion_matrix(target, probs > best_threshold)

         0.9410000000000001

Out[22]: array([[65581,   774],
                [    0,    84]], dtype=int64)
```

- (0.9410000000000001, 98.83502159875977)

Version2	
Metrices	Values
Recall	1.0
Precision	0.097
Accuracy	0.9984

3. version3- using variance and mean of u and v components of wind, linearly interpolated missing data

baseline_acc, baseline_fp, baseline_fn

```
In [24]: # baseline

         # TN | FP
         # ---|---
         # FN | TP
         baseline = np.zeros(len(target))
         confusion_matrix(target, baseline)

Out[24]: array([[66355,     0],
                [   84,     0]], dtype=int64)
```

- The output is (99.87356823552432, 0, 84)
- This means that the baseline accuracy of the model is 99.87%. The baseline number of false positives is 0, since we are predicting all negatives. The baseline number of false negatives is 84.

best_threshold, val_acc

```
In [23]: # best threshold with n false negatives
         n = 0
         best_threshold = thresholds[np.where(fn == n)[0]][-1]
         print(best_threshold)

         # TN | FP
         # ---|---
         # FN | TP
         confusion_matrix(target, probs > best_threshold)

         0.981

Out[23]: array([[66164,   191],
                [    0,    84]], dtype=int64)
```

- (0.981, 99.71251824982315)

Version3	
Metrices	Values
Recall	1.0
Precision	0.305
Accuracy	0.9971

best_threshold, val_acc

- (0.987, 99.77272385195442)

In [24]:
```
# best threshold with n false negatives
n = 12
best_threshold = thresholds[np.where(fn == n)[0]][-1]
print(best_threshold)

# TN | FP
# ---|---
# FN | TP
confusion_matrix(target, probs > best_threshold)
```

```
0.987
```

Out[24]:
```
array([[66216,   139],
       [   12,    72]], dtype=int64)
```

Version3	
Metrices	Values
Recall	0.857
Precision	0.341
Accuracy	0.9972

4. version4- using variance and mean of u and v components of wind, linearly interpolated missing data, and 200 seconds interval data

baseline_acc, baseline_fp, baseline_fn

```
In [24]: # baseline

         # TN | FP
         # ---|---
         # FN | TP
         baseline = np.zeros(len(target))
         confusion_matrix(target, baseline)

Out[24]: array([[66355,     0],
                [   84,     0]], dtype=int64)
```

- The output is (99.87356823552432, 0, 84)
- This means that the baseline accuracy of the model is 99.87%. The baseline number of false positives is 0, since we are predicting all negatives. The baseline number of false negatives is 84.

best_threshold, val_acc

```
In [24]: # best threshold with n false negatives
         n = 0
         best_threshold = thresholds[np.where(fn == n)[0]][-1]
         print(best_threshold)

         # TN | FP
         # ---|---
         # FN | TP
         confusion_matrix(target, probs > best_threshold)

         0.898

Out[24]: array([[65854,   501],
                [    0,    84]], dtype=int64)
```

- (0.898, 99.24592483330574)

Version4	
Metrices	Values
Recall	1.0
Precision	0.143
Accuracy	0.9924

conclusions

- Data cleaning, using variance as additional targets, and applying longer sequence than shorter sequence helps to improve the performance of the model

Metrices	Version1	Version2	Version3	Version4
Recall	1.0	1.0	1.0	1.0
Precision	0.465	0.097	0.305	0.143
Accuracy	0.9989	0.9984	0.9972	0.9924

References

1. 1Boille, A., Mahfouf, J., 2013. Wind shear over the Nice Cote d’Azur airport: Case studies. Nat. Hazards Earth Syst. Sci. 13, 2223–2238.
2. Chan, P.W., Hon, K.K., 2016. Performance of super high resolution numerical weather prediction model in forecasting terrain-disrupted airflow at the Hong Kong International Airport: case studies. Meteorology. Appl. 23, 101–114.
3. Liu, N., Kwong, K., Chan, P., 2012. Chaotic oscillatory-based neural network for wind shear and turbulence forecast with lidar data. IEEE Trans. Syst. Man Cybern. Part C, 42, 1412–1423.
4. 4Meng, L., Xu, J., Xiong, X., Ma, Y., Zhao, Y. 2018. A novel ramp method based on improved smoothing algorithm and second recognition for windshear detection using lidar. Curr. Opt. Photonics, 2, 7–14.
5. 5Nechaj, P., Gaál, L., Bartok, J., Vorobyeva, O., Gera, M., Kelemen, M., Polishchuk, V., 2019. Monitoring of low-level wind shear by ground-based 3D lidar for increased flight safety, protection of human lives and health. Int. J. Environ. Res. Public Health 16, 4584.
6. 6Sadique, F., Sengupta, S., 2021. Modeling and Analyzing Attacker Behavior in IoT Botnet using Temporal Convolution Network (TCN) 2021, 1–22.
7. 7Shun, C.M., Chan, P.W., 2008. Applications of an infrared Doppler lidar in detection of wind shear. J. Atmos. Ocean. Technol. 25, 637–655.
8. Tai, Y., Yang, J., Liu, X., 2017. Image super-resolution via deep recursive residual network. Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017 2017-Janua, 2790–2798.