Majalah Ilmiah Pengkajian Industri
(Journal of Industrial Technology Assessment)
Vol. 16, No.2 August 2022 – (87-94)
Journal homepage: https://ejurnal.bppt.go.id/index.php/MIPI/index

Data Exploratory Analysis and Feature Selection of Low-Speed Wind Tunnel Data for Predicting Force and Moment of Aircraft

Fitra Hidiyanto[1*], Shabrina Leksono[2], Rizqon Fajar[1], Sigit Tri Atmaja[1]

[1]*Research Center for Transportation Technology, National Research and Innovation Agency, Indonesia*
[2]*National Laboratory of Aerodynamics, Aeroelastics and Aeroacoustics, National Research and Innovation Agency, Indonesia*
*Correspondence E-mail: fitr009@brin.go.id

ABSTRACTS

This paper discusses exploratory data analysis (EDA) and feature selection of aircraft test results in Indonesia's low-speed wind tunnels (ILST). First, we briefly explain input and output parameters and data processing to make readable and higher accurate data. Then, we used feature selection using embedded and random forest methods to find parameters that most affect the force coefficient of aircraft. The research activities carried out in this study are to review literature from either scientific journals, the internet, or books and interview with an engineer who tests aircraft models at ILST. Then create a program for processing data from test results, such as data extraction, data cleaning, exploratory data analysis, and feature selection with python. After applying the feature selection method, we found that all the methods show similar results and have succeeded in separating the powerful features from the weak ones with a significant score difference. We decide to use the Random Forest method. The three most strongest features in the coefficient of an aircraft model in the ILST test (**C_L**, **C_D**, **C_{M25}**, **C_{YAW}**, **C_{ROLL}** and **C_Y**) are the following: for **C_L** are ALFA (0.984), T0 (0.008), P0 (0.004), on for **C_D** is are ALFA (0.965), T0 (0.009), RE (0.007), in **C_{M25}** are ALFA (0.416), P0 (0.285), T0 (0.168), in **C_{YAW}** are BETA (0.44), T0 (0.141), ALFA (0.141), in **C_{ROLL}** is BETA (0.79), ALFA (0.091), P0 (0.036), and in **C_Y** are BETA (0.842), ALFA (0.114) and T0 (0.014). The results of this paper can be used to help build a model for the coefficient of aircraft design using machine learning based on the data from the ILST test more effectively and efficiently.

ARTICLE INFO

Article History:
Received 08 Apr 2022
Revised 11 Jul 2022
Accepted 21 Jul 2022

Keyword:
Machine Learning,
Feature Selection,
Exploratory Data Analysis,
Aircraft Modeling.

INTRODUCTION

Testing aircraft models in wind tunnels is essential in the aviation industry. This is related to determining aircraft design parameters that must be considered, which are the magnitudes of the forces and moments of the test model, such as **C_L** (Lift Coefficient), **C_D** (Drag Coefficient), **C_Y** (Side Force Coefficient), **C_{M25}** (Pitch Moment Coefficient), **C_{ROLL}**(Rolling Moment

***Corresponding Author | Fitra Hidiyanto ✉ fitr009@brin.go.id**

Coefficient) and **C_{YAW}** (Yawing Moment Coefficient) [1].

In the wind tunnel, measurements were made for several forces and moments, as illustrated in Figure 1. There are three forces measured: lift, drag and side force. The lift force (C_L) is the force generated by the pressure difference of the aircraft wing so that the aircraft could fly. It acts on the z-axis of the plane. The drag force (C_D) is the force that acts on the x-axis of the plane due to wind resistance to the fuselage in the opposite direction from the aircraft. And lastly, the side force (C_Y) acts on the plane's y-axis.

Then there are three types of measured moments: yaw moment (C_{YAW}), roll moment (C_{ROLL}), and pitch moment(C_M). These three moments act on different axes. The yaw moment acts on the z-axis, the roll moment on the x-axis, and the pitch moment on the y-axis. These forces and moments are converted into dimensionless numbers or coefficients in their application. For example, lift force or lift becomes the coefficient of lift or **C_L**, drag force or drag becomes the coefficient of drag or **C_D**, etc.

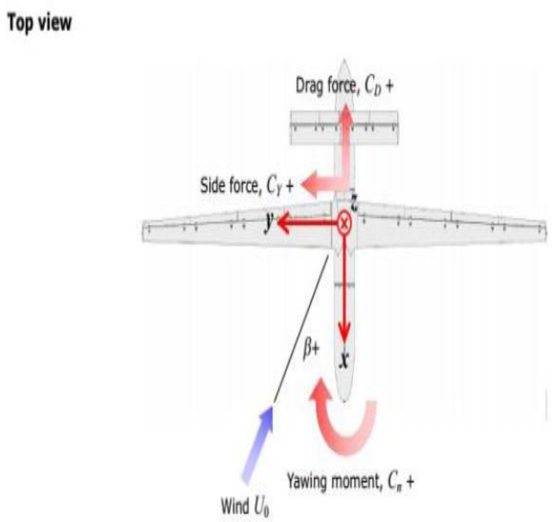

Figure 1. Axis and direction of aircraft control deflection [2].

This paper explores how much input values (features) influence each aircraft's force/moment coefficient. In wind tunnel testing, these features are essential and influence aircraft design parameters because the value of moment/force affects the test model as it represents the actual aircraft design. So, to increase the effectiveness and efficiency of the design process, a selection feature is needed to find out which features or inputs have a dominant influence on the output force/moment sought. The selection feature is a process for removing less relevant features and taking the most relevant features from data to improve predictive accuracy, reduce costs and computing time and extract valid information [3].

METHODS

The method used in writing this paper can be seen in the flow diagram in **Figure 2** below:

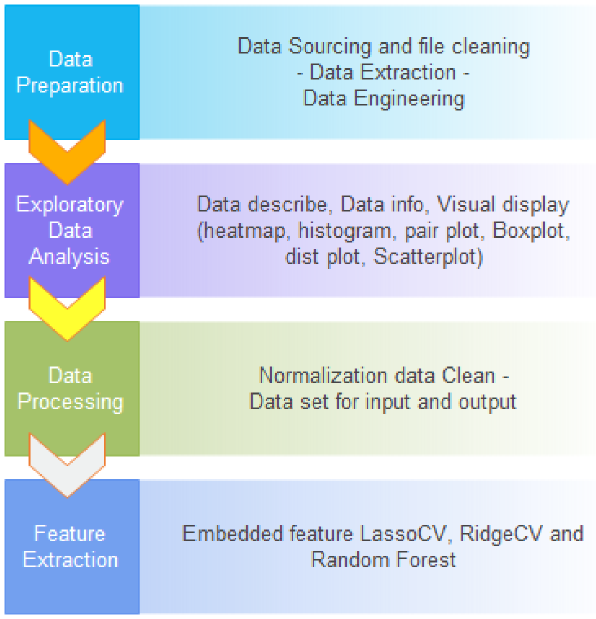

Figure 2. Method flow chart used in this paper.

Data Sourcing and file cleaning

This paper uses data from aerodynamic laboratory testing results on an aircraft model whose data are presented in the form of standard parameters. As for the input parameters in the file, among others, P0 is static air pressure inside the tunnel, Q0 is the dynamic air pressure in the tunnel, T0 is the static temperature in the sensor, Rh0 is Air density in the test section, V0 is air velocity in the test section, Re is a Reynold number that is influenced by the coefficient of dynamic viscosity (μ), Alpha is the angle of the attack direction (wind) which can be changed using a dynamic load (Up +, Down -). In contrast, Beta is the magnitude of the yaw angle (rolling angle) which can be changed using the turntable (left-right) in the test section [4].

The file to be processed in machine learning is a clean file, which means the test result file in conditions without deflection flap, aileron, vertical tailplane, or horizontal tailplane. This value is always mentioned in the 19th row of the test results file. In this paper, the file that we retrieve is only a file that contains the value 0 in all categories except in the trip section.

Data Extraction & Engineering

The file should have a .txt extension to make the program's reading easier. The program reads files with the format encoding='utf-8' with the readlines()

command. From the results of reading the entire file, the data was accommodated in a variable list, which was written into a .txt file after all the data was collected. Then, with the write() command, the data was imported and saved into a .csv file with an extension to finalize data extraction. With this file, The following process is to read the files into python-based data processing script using the read_csv() command from the panda's library. See the Github link for the complete script in the additional information section.

Data Cleaning

In this section, to speed up the process and avoid error, we clear the unnecessary data such as data containing blank characters, redundant data (double or more than 1), and outliers (data placed in an extreme position away from the general value). The redundant data was eliminated using [data.duplicated()] command. While for the outliers contained in the data, we use the interquartile range (IQR) method. Interquartile Range (IQR) is the difference between the third and first quartiles. The quartile is the partition value that divides all data into four equal parts. The first quartile is symbolized by Q1, known as the lower quartile, Q2 denotes the second quartile as the middle quartile, and Q3, the upper quartile, symbolizes the third quartile.

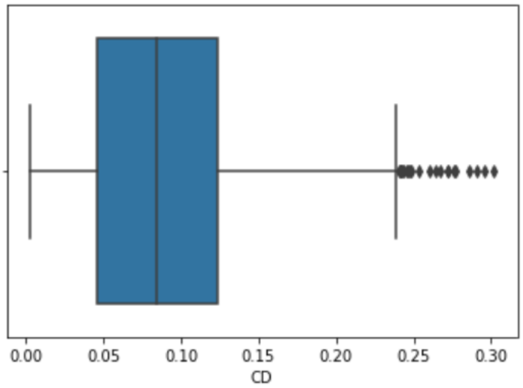

Figure 3. Example of a data boxplot that has an Outlier for **C_D** Output.

Therefore, IQR is the upper quartile minus the lower quartile or the difference between the upper and lower quartile, with the formula [5]:

$$IQR = \text{Upper quartile} - \text{Lower quartile} = Q3 - Q1 \quad (1)$$

Where Q1 is the first quartile and Q3 is the third quartile of the data series. This IQR method can be seen in the results for the **C_D** output boxplot graph, where it appears that the differences in figure 3 and figure 4 show the loss of outliers after IQR in figure 4.

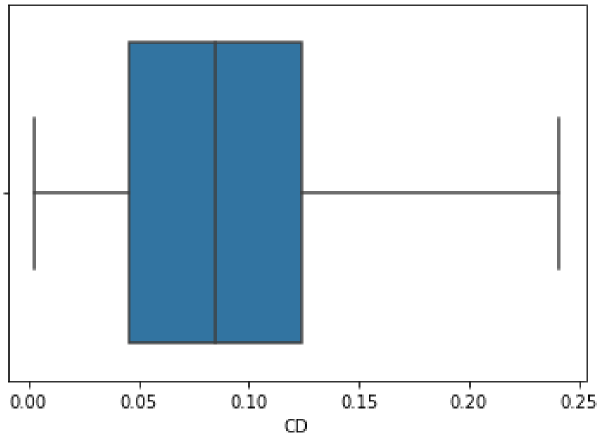

Figure 4. Boxplot graph **C_D** output after IQR.

Data Normalization

Variables measured at different scales do not make the same contribution to the model fitting and function of the model being studied, this could end up creating bias. So, to overcome this potential problem, normalization is used. In this paper, we used the Minmax scaler method. The mathematical formula of normalization is [6]:

$$X_{Scaled} = \frac{X - \min(x)}{\max(x) - \min(x)} \quad (2)$$

The difference between data before and after normalization lies only in the scale of the data value that changes to between 0 - 1 for normalized data. All data is positive, with a minimum value of 0 and a maximum of 1, and the data value is between them.

Exploratory data analysis

Exploratory data analysis is one of the critical processes of preprocessing machine learning to find out data patterns and trends, view data anomalies, test hypotheses, and check the assumptions from the statistical data and graph display [7].

Some ways to find out information, data distribution, and data visualization in python are to run the following commands:

Data.shape, the result of this command is the data dimension in the form of the number of rows and data columns.

Data.describe(), will display the amount of data, minimum value, maximum value, standard deviation, average, value 1⁄4 bottom quartile, 1⁄4 upper quartile, and middle quartile values for all inputs and outputs in the data, with sample results as shown in figure 5.

```
df_minmax.describe() # deskripsi data
```

	P0	Q0	V0	RHO	T0	RE	ALFA	BETA	CL	CD	CM25	CYAW	CROLL	CY
count	581.00	581.00	581.00	581.00	581.00	581.00	581.00	581.00	581.00	581.00	581.00	581.00	581.00	581.00
mean	0.53	0.47	0.51	0.44	0.49	0.40	0.52	0.50	0.49	0.38	0.51	0.50	0.51	0.51
std	0.25	0.19	0.17	0.22	0.21	0.20	0.28	0.18	0.28	0.25	0.29	0.30	0.29	0.30
min	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	0.40	0.32	0.38	0.21	0.35	0.24	0.33	0.50	0.25	0.18	0.37	0.38	0.38	0.38
50%	0.52	0.44	0.50	0.43	0.49	0.39	0.60	0.50	0.56	0.34	0.54	0.56	0.54	0.52
75%	0.66	0.59	0.62	0.57	0.61	0.55	0.74	0.50	0.75	0.51	0.62	0.62	0.62	0.62
max	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Figure 5. Description of data distribution in python.

Data.column, the results of this command displays index containing the names of data columns and the type of data that is usually the Object type.

Data.dtypes, this command displays the type of data from all data for each column.

Data.info(), this command displays more detailed info, including the number of rows and columns, the amount of NULL data, the type of data, and the memory use, as shown in **Figure 6.**

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 581 entries, 0 to 580
Data columns (total 15 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   P0      581 non-null    float64
 1   Q0      581 non-null    float64
 2   V0      581 non-null    float64
 3   MA      581 non-null    float64
 4   RHO     581 non-null    float64
 5   T0      581 non-null    float64
 6   RE      581 non-null    float64
 7   ALFA    581 non-null    float64
 8   BETA    581 non-null    float64
 9   CL      581 non-null    float64
 10  CD      581 non-null    float64
 11  CM25    581 non-null    float64
 12  CYAW    581 non-null    float64
 13  CROLL   581 non-null    float64
 14  CY      581 non-null    float64
dtypes: float64(15)
memory usage: 68.2 KB
```

Figure 6. Data.info() command results.

Sns.distplot(data), displays the distribution of each data value in its range, with results as shown in **Figure 7.**

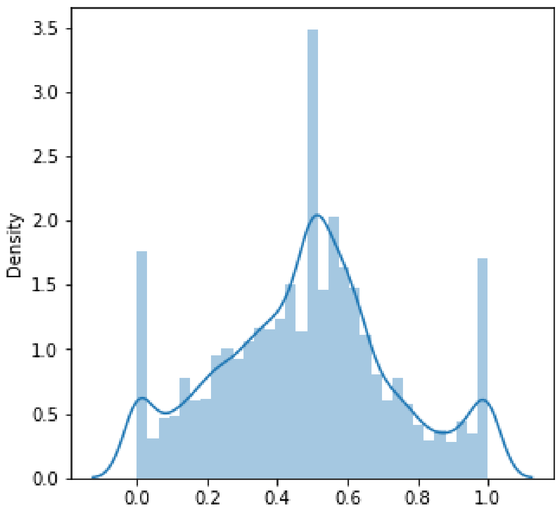

Figure 7. Graph result of distplot command.

Data.corr(), using facilities from the seaborn library to display correlations between data with the heatmap() format. This command displays a visual display of correlations between data, whether input or output. More detailed results can be seen in **Figure 8.**

***Corresponding Author | Fitra Hidiyanto ✉ fitr009@brin.go.id**

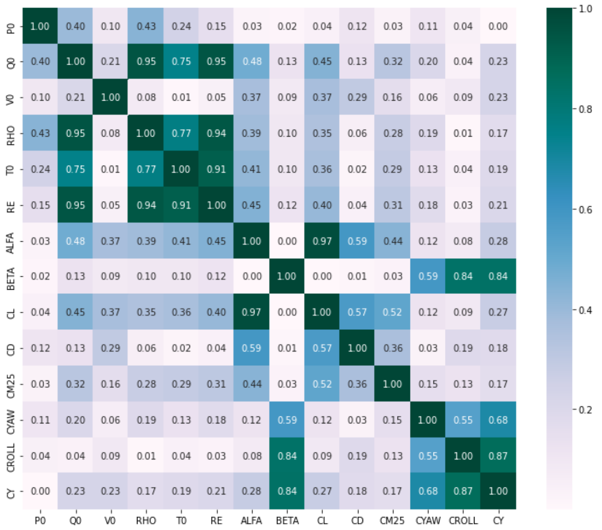

Figure 8. Heatmap correlation between input and output data.

Sns.boxplot(data=data), displays a boxplot graph from data, to detect outlier presences in all data for each column (input and target). It uses the features of the seaborn library with the results in **Figure 9.**

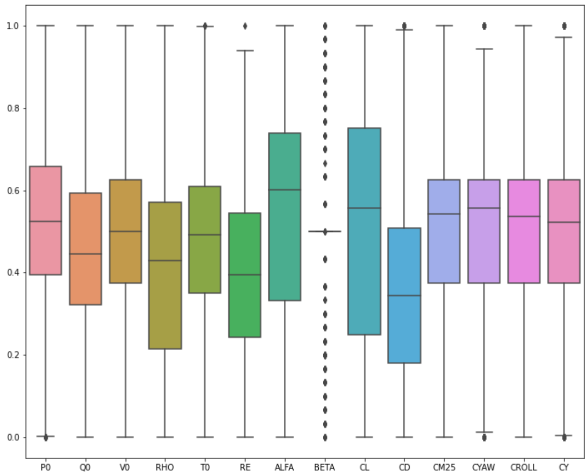

Figure 9. Box plot display of all data.

Feature Selection method

Large data dimensions can cause noise, irrelevant, and redundancy in data, thus producing an over-fitting model and errors in the machine learning algorithm. To overcome this problem, feature selection is used as a dimension reduction technique to clean noise, redundancy, and irrelevant data to improve algorithm performance [8].

This paper discusses feature selection using the Embedded method, which we consider the best method compared to the other two methods, i.e. the Filter method (Pearson correlation) and the Wrapped method (Forward, backward, stepwise). It takes into consideration the interaction of features like the Wrapper method, which is more accurate than the Filter method, to find the feature subset for the algorithm being trained and much less prone to over-fitting. We will also discuss the Random Forest method, whose decision tree is still included in embedded features [9].

Embedded methods (LassoCV and RidgeCV)

This embedded method takes advantage of the two methods (filter and wrapped) by processing the existing features and considering the cost of the process by paying attention to and taking the crucial features and affecting the model's performance.

Lasso (L1) regularization consists of adding a penalty to the different parameters of the machine-learning model to avoid over-fitting. Furthermore, of the various types of regularization, Lasso (L1) has the property of reducing some coefficients to zero. Therefore, these features can be removed from the model [10].

We can see in **Figure 10** below that five features were omitted due to the Lasso regression of these features being zero.

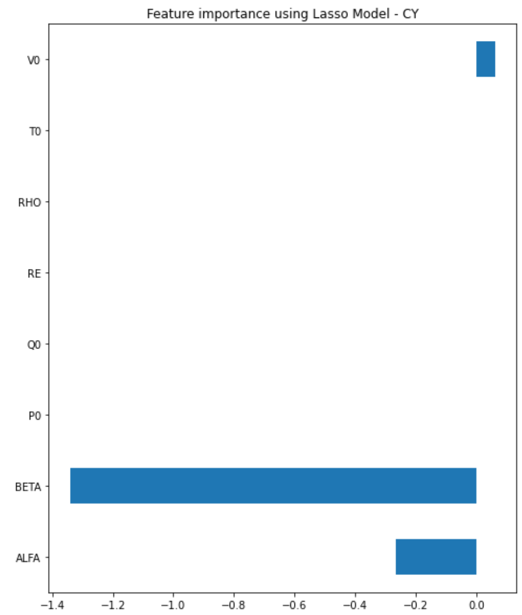

Figure 10. Example of Embedded results with LassoCV (L1) for $\mathbf{C_Y}$.

While the regression with RidgeCV (L2) tends to reduce the coefficients uniformly, L2 is valid when the input attribute has collinear/codependent characteristics. L2 makes the coefficient smaller. Decreasing this coefficient minimizes the effect of correlation between input attributes and makes the model more generalizable. RidgeCV (L2) works best when most attributes are relevant [11]. For the results of the RidgeCV method in **Figure 11.**

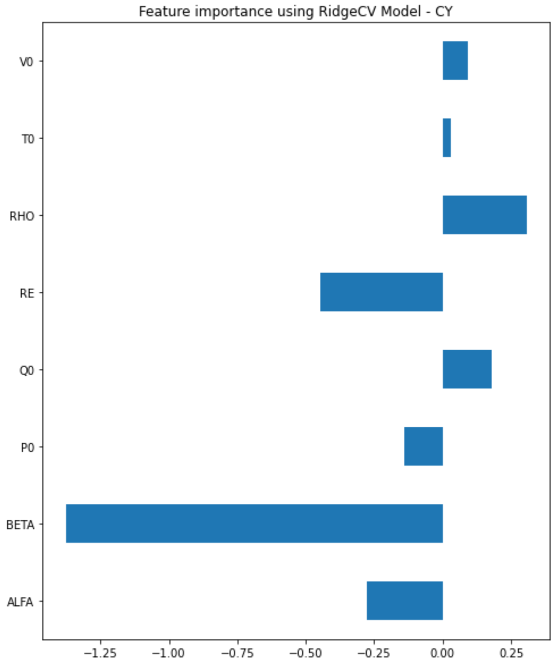

Figure 11. Example of Embedded results with RidgeCV (L2) for $\mathbf{C_Y}$.

Random Forest regression method

Random forest is one of the most popular machine learning algorithms because it provides good performance predictions, low overfitting, and is easy to read/understand. This Random forest uses a decision tree that is still included in the embedded method selection feature. This embedded method has high accuracy advantages, and generalization is better and easier to interpret [12].

A random forest is a meta estimator that fits some classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. If bootstrap=True (default), the sub-sample size is controlled with the max_samples parameter if bootstrap=True (default). Otherwise, the whole dataset is used to build each tree. Hyperparameters for this model are (1) the number of trees in the forest (n_estimators), (2) the maximum depth of the tree (max_depth), and (3) the number of features to consider when looking for the best split (max_features) [13]. The Random forest results for $\mathbf{C_{M25}}$ output can be seen in **Figure 12.**

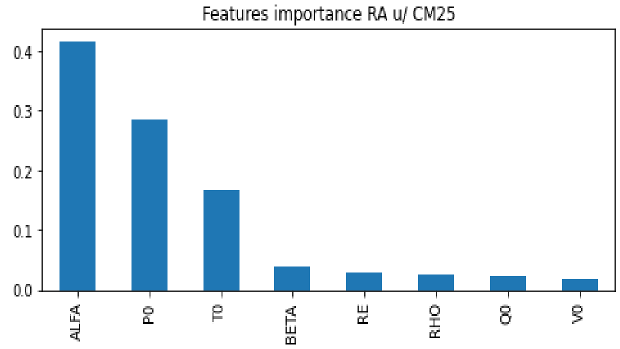

Figure 12. Example of Random Forest results for $\mathbf{C_{M25}}$.

RESULTS AND DISCUSSION

Feature selection result

Some feature selection results that we have run in this paper that show the correlation of inputs for each output can be seen in **Table 1** (LassoCV), **Table 2** (RidgeCV), and **Table 3** (Random Forest).

Table 1. Results of the Embedded Selection feature LassoCV method.

LassoCV	CL	CD	CM25	CYAW	CROLL	CY
ALFA	**0.94**	**0.62**	**0.37**	0.08	0.11	0.27
BETA	0	0.04	0	**0.92**	**1.35**	**1.34**
P0	0	0.02	0	0.05	0.03	0
Q0	0	0	0	0	0.04	0
RE	0	0	0.02	0	0.09	0
RHO	0	0	0.06	0.07	0	0
T0	0	0.35	0.05	0	0	0
V0	0	0.03	0	0	0	0.06

For brief description of this table parameter is described at previous section in introduction and methode section.

Table 2. Results of the Embedded Selection feature of the RidgeCV method.

RidgeCV	CL	CD	CM25	CYAW	CROLL	CY
ALFA	0.97	0.63	0.36	0.11	0.16	0.28
BETA	0	0.06	0.02	0.94	1.4	1.38
P0	0.28	0.26	0.52	0.07	0.37	0.14
Q0	0.27	0.21	0.44	0.06	0.44	0.18
RE	0.02	0.51	0.26	0.02	0.47	0.45
RHO	0.27	0.6	0.26	0.06	0.34	0.31
T0	0.43	0.65	1.1	0	0.42	0.03
V0	0.09	0.05	0.23	0.04	0.17	0.09

Table 3. Random Forest Selection Feature Results.

CL	ALFA	T0	P0	RE	BETA	Q0	V0	RHO
	0.984	0.008	0.004	0.001	0.001	0.001	0.001	0.001
CD	ALFA	T0	RE	BETA	P0	RHO	Q0	V0
	0.965	0.009	0.007	0.004	0.004	0.004	0.004	0.003
CM25	ALFA	P0	T0	BETA	RE	RHO	Q0	V0
	0.416	0.285	0.168	0.039	0.028	0.024	0.023	0.017
CYAW	BETA	T0	ALFA	P0	Q0	V0	RHO	RE
	0.440	0.141	0.141	0.111	0.061	0.051	0.034	0.022
CROLL	BETA	ALFA	P0	T0	Q0	RE	V0	RHO
	0.790	0.091	0.036	0.034	0.015	0.013	0.012	0.009
CY	BETA	ALFA	T0	P0	Q0	V0	RE	RHO
	0.842	0.114	0.014	0.012	0.006	0.005	0.004	0.002

From the results of the three embedded methods, we can see that the results of the LassoCV and Random Forest methods tend to be the same or similar for their dominant features. In contrast, the RidgeCV method has two main dominant features for several coefficients. However, for one, the most dominant features differ only for $\mathbf{C_{M25}}$ features compared to two other methods. For the LassoCV method, it can be seen that some features are worth 0, leaving only the dominant features. However, all features still have value for the Random Forest method even though the feature dominance score is much different from the main dominant feature.

Based on the results of these three methods, the Random Forest method was chosen for further modeling studies to yield three dominant features on all coefficients. It was chosen for its similarity in results with the other two methods and has a score for all its features (while LassoCV for the $\mathbf{C_L}$ coefficient only has one dominant feature). It also gave accurate feature selection results based on the data from the test results.

Some points that can still be discussed in this paper are whether it is true that the removal of an outlier will have a significant effect on the results of the feature selection. However, we have to evaluate which selection features method will be the most accurate by testing the final model using the unseen data.

CONCLUSION

Based on the results of the application of feature selection, it can be concluded that the data from the wind tunnel test results should be processed before being used in modelling using machine learning. Of the three feature selection methods that were applied, the Random Forest method was chosen because it is a widely used method with high accuracy. Based on this research result, it can differentiate dominant features from other features with high score differences. Furthermore, the selected features have shown good predictive values compared to the data from the test results.

Then the results of three features that most influence the aircraft coefficients with sorted scores result from random forest feature selection algorithm, namely for $\mathbf{C_L}$ are ALFA (0.984), T0 (0.008), P0 (0.004), on $\mathbf{C_D}$ are ALFA (0.965), T0 (0.009), RE (0.007), in $\mathbf{C_{M25}}$ are ALFA (0.416), P0 (0.285), T0 (0.168), in $\mathbf{C_{YAW}}$ are BETA (0.44), T0 .141), ALFA (0.141), in $\mathbf{C_{ROLL}}$ is BETA (0.79), ALFA (0.091), P0 (0.036), and in $\mathbf{C_Y}$ are BETA (0.842), ALFA (0.114) and T0 (0.014). We expect modeling aerodynamics coefficients with Machine Learning/AI can be faster and more accurate using the feature selection method.

ADDITIONAL INFORMATION

The script, normalization data, and results in this paper are specifically shared to be accessed by readers on Github at the address https://github.com/Dtscience80/Fiture-Selection-Analysis-of-Low-Speed-Wind-Tunnel-Data-Testing-. Hopefully, it will be helpful for all of our scientific development.

ACKNOWLEDGMENTS

This research is an independent research collaboration involving several parties. For that, the author would like to thank BRIN for providing facilities for research, Mrs. Dr. Hanni Defianti and Gilang Mantara Putra S. Kom. as a supporter, and all colleagues at BRIN who support the completion of this paper.

REFERENCES

[1] UPT – LAGG (2014). Program Manual Pengujian Model N219-B12 Power Off & Power On di Indonesia Low Speed Tunnel (ILST) UPT LAGG BPPT.

[2] Daryanto Y, Purwono J, Subagyo (2018). Wing configuration on Wind Tunnel Testing of an Unmanned Aircraft Vehicle. IOP Conf. Series: Journal of Physics: Conf. Series 1005 (2018) 012032 DOI: 10.1088/1742-6596/1005/1/012032. Page 5.

[3] Neeraj M, Maurya V, Kumar N (2020). A Review on machine learning (feature selection, classification and clustering) approaches of big data in different area of research. Journal of critical reviews. VOL 7, ISSUE 19, 2020. Page 2611.

[4] "Equation for the data processing of the ILST", national Lucht - EN Ruimtevaart laboratorium, National Aerospace Laboratory NLR The Netherlands. August 1987.

[5] Sahoo K, Samal AK, Pramanik J, Kumar Pani S (2019). Exploratory Data Analysis using python. International Journal of Innovative Technology and Exploring Engineering (IJITEE). ISSN: 2278-3075, Volume-8, Issue-12, October 2019. Page 4730.

[6] Henderi, Wahyuningsih T, Rahwanto E (2021). Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer. International Journal of Informatics and Information System. Vol. 4, No. 1, March 2021, pp. 13-20. Page 16.

[7] Nongthombam K, SharmaD (2021). Data Analysis using python. International Journal of Engineering Research & Technology (IJERT). ISSN: 2278-0181. Vol. 10 Issue 07, July-2021. Page 463-468.

[8] Venkatesh B, Anuradha J (2019). A Review of Feature Selection and Its Methods. Cybernetics And Information Technologies. Volume 19, no 1. March 2019. Online ISSN: 1314-4081. DOI: 10.2478/cait-2019-0001. Page 3.

[9] Abhigyan. Feature Selection For Dimensionality Reduction(Embedded Method) [Internet]. https://medium.com/. 2020 [cited 2022 Apr 8]. p. Analytics Vidhya. Available from: https://medium.com/analytics-vidhya/feature-selection-for-dimensionality-reduction-embedded-method-e05c74014aa.

[10] Hayati M, Muslim A (2021). Generalized Linear Mixed Model and Lasso Regularization for Statistical Downscaling. Enthuastic International Journal Of Statistics And Data Science. Volume 1, Issue 1, April 2021, pp. 38.

[11] Belli E (2020). Smoothly Adaptively Centered Ridge Estimator. arXiv:2011.00289v1 [stat.ME] 31 Oct 2020. Page 9.

[12] Divya P, Pavithra M, Jayalakhsmi S, Praveen Kumar P. (2021). Application of Random Forest Algorithm in Bio Informatics. International Journal of Information Technology Insights & Transformations. Vol. 5, Issue 1 – 2021. Page 19.

[13] Chaibi M, Benghoulam EM, Tarik L, Berrada M, El Hmaidi A (2022). Machine Learning Models Based on Random Forest Feature Selection and Bayesian Optimization for Predicting Daily Global Solar Radiation. International Journal of Renewable Energy Development (IJRED). 2022: 309-323. 309-323.