Predicting Aerodynamic Characteristics of Non-Conventional Airfoil Based on NACA Series Airfoil Data

Final Project Report

EL 5004 Intelligence System

by

Febrian Akhmad Baehaqi

23620002

DEPARTMENT OF AEROSPACE ENGINEERING

FACULTY OF MECHANICAL AND AEROSPACE ENGINEERING

INSTITUT TEKNOLOGI BANDUNG

2021

Table of Contents

1. Introduction

1.1 Background

The prediction of the aerodynamic characteristics of an airfoil is very useful in the design phase of the aircraft. Not only on the aircraft, airfoil also used in wide variety of engineering design, such as wind turbine, propeller, compressor, power turbine and high-speed cars. One needs to do the experiment to get the accurate results of the aerodynamics characteristics of the airfoil. This experimental result can be expensive and take a lot of times and effort to get the good results.

Luckily, NASA has done many experimental setups and results for the NACA series airfoil for a lot of airfoil configurations. This NACA series experiment results can be used to develop a machine learning model that predict the aerodynamic characteristics of the other airfoils, especially the non-conventional airfoil. A large set of experimental data of the NACA airfoil series can be trained using machine learning algorithm so that the model can predict other airfoil configurations without doing any expensive experimental setup.

This process will speed up the airfoil analysis for the non-conventional airfoil using the model produced by training the NACA datasets. Expensive experimental setup can be replaced by a good machine learning model which will save so much time and resources in the airfoil analysis. Hence, the design and the optimization process can be speed up and create a much more efficient design at a rapid time.

The artificial neural network (ANN) is a good machine learning method to fit and predict the aerodynamic characteristics of the airfoils. ANN used because its capability to do the nonlinear mapping using the activation function since the airfoil aerodynamic characteristics analysis is very nonlinear field to be analysed. Hence, it is a good fit to be used to make the airfoil analysis machine learning model.

1.2 Objective

The objective of this final project is to make a model that can predict the aerodynamic characteristics of an airfoil based on the given airfoil database which are the NACA series airfoil using artificial neural network algorithm.

2. Basic Theory and Literature Study

2.1 Airfoil

Airfoil is a structure of curved surfaces designed to produce the most favourable ratio of lift to drag. It is basically the cross section or a 2D version of a wing. Airfoil designed as a streamlined body/surface which produce lift (L) which is the force perpendicular to the direction of motion and drag which is the force parallel to the direction of motion. These two forces are so called the aerodynamic forces and the dimensionless form of them is the aerodynamic characteristics of an airfoil which are called the coefficient of lift (C_l) and the coefficient of drag (C_d).

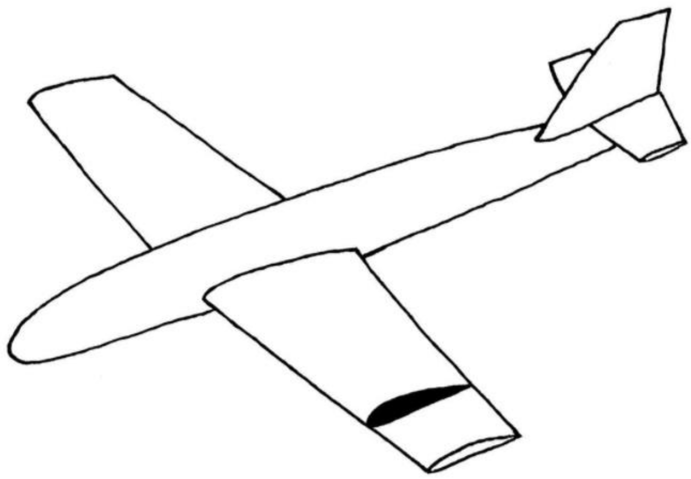

Figure 1. Airfoil representation in a wing of an aircraft

Some terminology in the airfoil science are described below

a) Chord: the distance between the leading edge and the trailing edge
b) Upper Surface: The suction surface of the airfoil
c) Lower Surface: The pressure surface of the airfoil
d) Angle of attack: The angle formed between a reference line on a body and the oncoming flow.
e) Camber: The asymmetry between upper and lower surface.

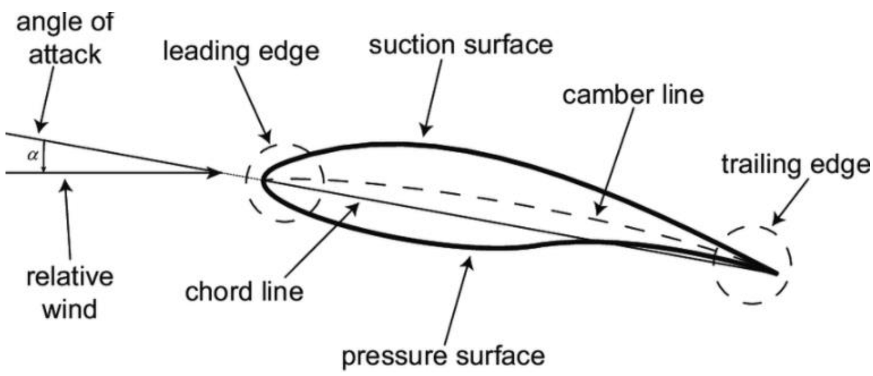

Figure 2. Airfoil terminology

Airfoils are expressed with coordinate value x and y normalized by its chord so the coordinated scaled from 0 to 1.

NACA 2412

[Stations and ordinates given in percent of airfoil chord]

Upper surface		Lower surface	
Station	Ordinate	Station	Ordinate
0	-----	0	0
1.25	2.15	1.25	−1.65
2.5	2.99	2.5	−2.27
5.0	4.13	5.0	−3.01
7.5	4.96	7.5	−3.46
10	5.63	10	−3.75
15	6.61	15	−4.10
20	7.26	20	−4.23
25	7.67	25	−4.22
30	7.88	30	−4.12
40	7.80	40	−3.80
50	7.24	50	−3.34
60	6.36	60	−2.76
70	5.18	70	−2.14
80	3.75	80	−1.50
90	2.08	90	−.82
95	1.14	95	−.48
100	(.13)	100	(−.13)
100	-----	100	0

L. E. radius: 1.58
Slope of radius through L. E.: 0.10

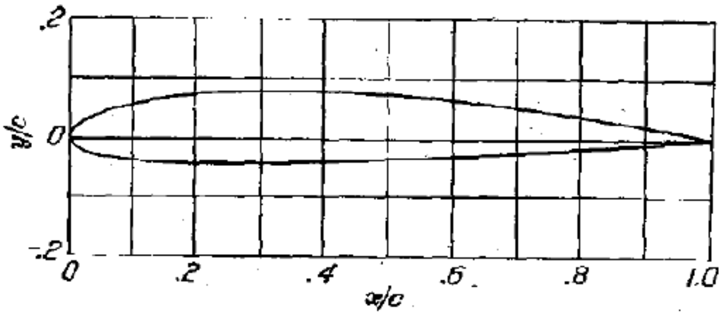

Figure 3.Airfoil coordinates system

The characteristic of the airfoil sometimes represented as the C_l graph which plot C_l to angle of attack α and drag polar plot plot C_l to C_d. The example of these plots can be seen on Figure 4 and Figure 5.

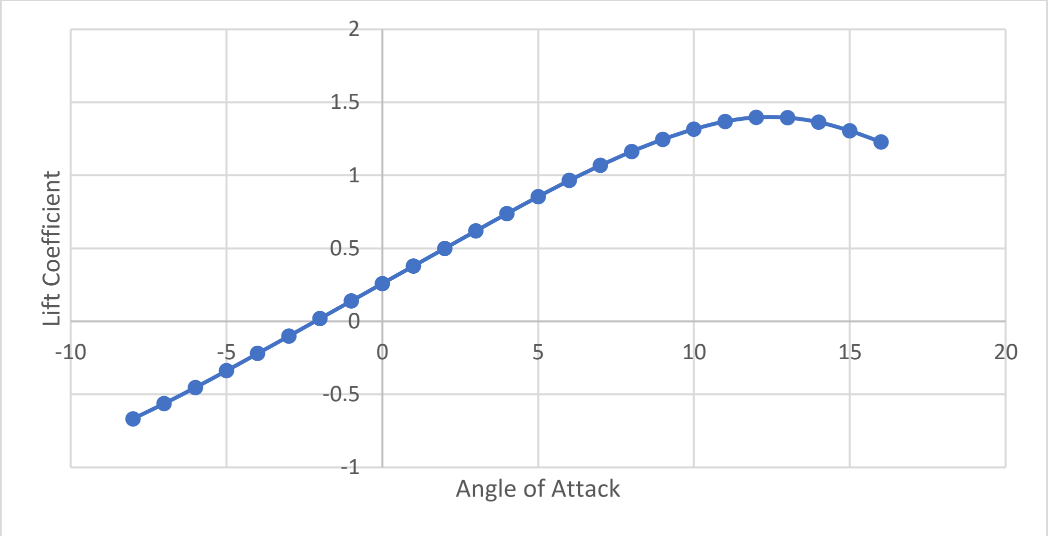

Figure 4. Cl graph of NACA 2412 Airfoil

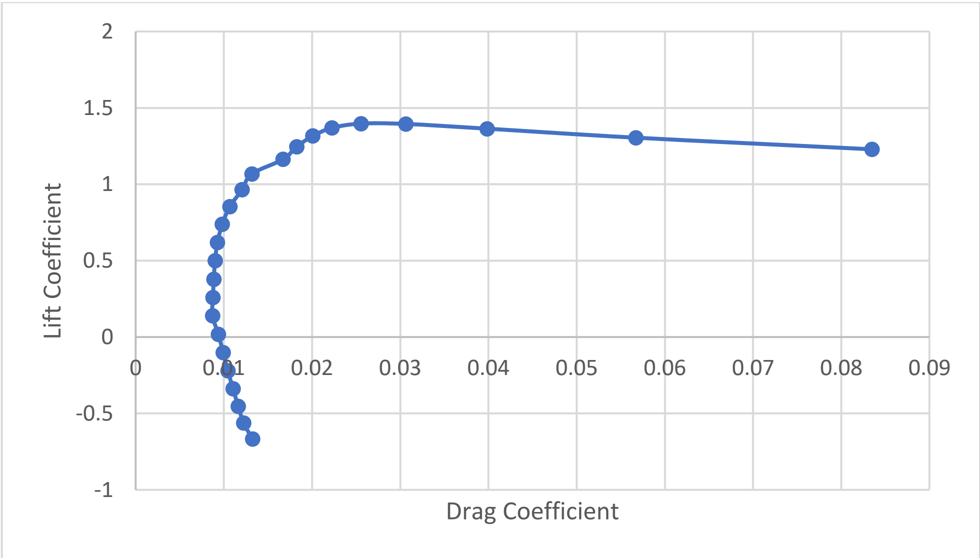

Figure 5. Drag Polar of NACA 2412 Airfoil

2.2 NACA Airfoil

The NACA airfoils are airfoil shapes for aircraft wings developed by National Advisory Committee for Aeronautics (NACA), now National Aeronautics and Space Administration (NASA). The airfoils are described using a series of digits that can be entered into equations to generate the airfoil shape. There are 3 most common types of the NACA airfoil series which are NACA 4 series, NACA 5 series, and NACA 6 series. The NACA airfoils established during the late 1920s and into the 1930s and the complete catalog of 78 airfoils appeared in the NACA's annual report in 1933. Some of the NACA airfoil shaped can be seen in Figure 6.

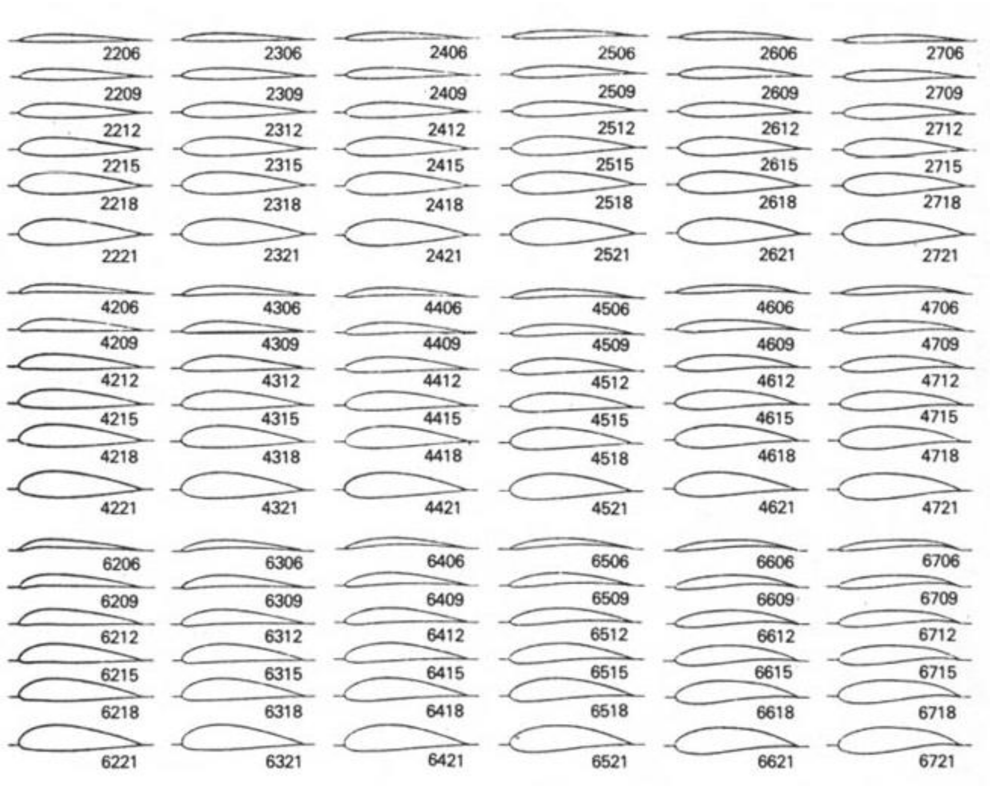

Figure 6. NACA Series Airfoil

2.3 Selig Airfoil

The Selig Airfoil will be used as the non-conventional airfoil to be tested to the model. It is a set of airfoil created by UIUC Applied Aerodynamics Group in Illinois, United States. The Selig Airfoil has various shapes airfoil and labelled as the low Reynolds number airfoils. The name of the airfoil starts with S letter followed by some digits.

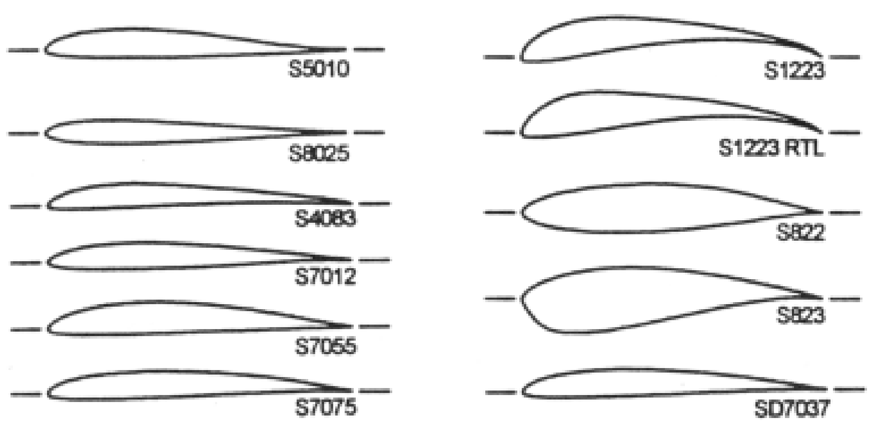

Figure 7. Selig Airfoils

2.4 Artificial Neural Network

Artificial Neural Network (ANN) is a machine learning algorithm that designed to simulate the way human brain analyses and processes information. It can spot patterns in data very well that makes it an optimal solution for classifying, clustering, and making prediction and regression. Its capability to process huge amount of data, feature extraction, and achieving the best performance with large amount of data makes it more popular nowadays. The self-learning capabilities of ANN enable them to produce better results as more data are analysed.

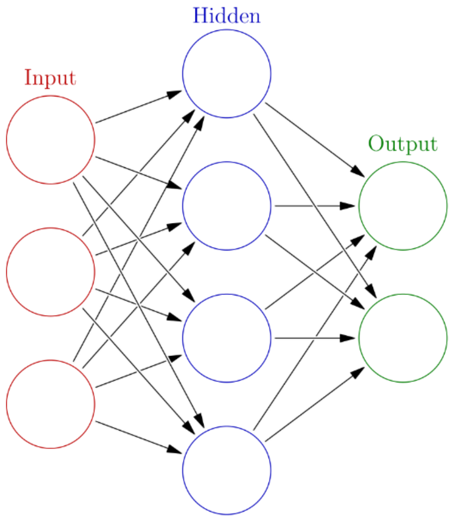

Figure 8. Neural network

ANN has an input layer and output layer, between these two layers there are other hidden layers that perform mathematical computations that help determine the decision or action the machine should take. ANN must be trained before it can produce any solution. The data is processed by each hidden layer and then move on to the next based on connections that are weighted.

3. Methodology

The artificial neural network is used because its capabilities to solve nonlinear problem as the airfoil analysis is a highly nonlinear field. TensorFlow will be used to create the ANN model. In general, there is no rule of thumb to know what the best ANN architecture is to use for a particular problem. Hence, the architecture of the ANN will be the same as composed in the Ref. [2] with one input layer, six hidden layer, and one output layer. This architecture is chosen based on their trial and error after many experiments and found that this is the match architecture to use in airfoil characteristics prediction. The number of units for each hidden layer is 100. The activation function for each layer can be seen in Figure 9.

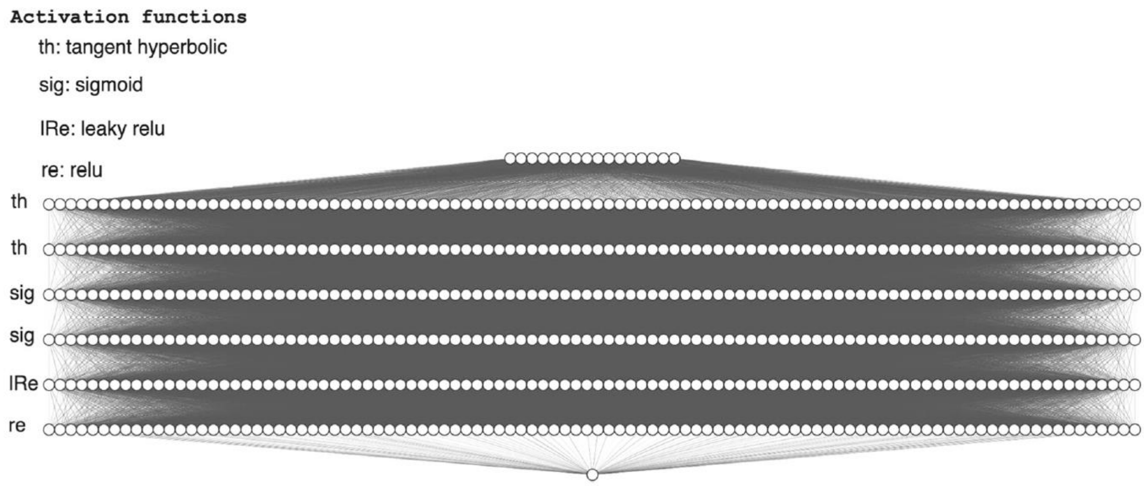

Figure 9. The ANN architecture that will be used [2]

The training data will use the NACA 4 series, 5 series and 6 series data with 70 available airfoil configurations with 16 variations of angle of attack and 5 variations of Reynolds Number. The datasets used is having about 5200 sample with 90% training sample and 10% test sample. Later, a new dataset will be used to test the model in predicting non-conventional airfoils. The conventional airfoil is the Selig airfoil which consists of 6 configurations.

The loss function used to this model is the mean squared error (MSE). MSE is the most popular used evaluation criteria in linear regression model. It calulates the sum over all of the squared differences between the predicted value and true values and then divided by the number of the batch size. The MSE can be expressed as follows.

$$MSE = \frac{1}{n}\sum\left(y_{target_i} - y_{predict_i}\right)^2$$

Another evalution criteria that will be used to analyze the accuracy of the model is the mean absolute error (MAE). It is the sum of the prediction and true values differences divided by the number of the batch size. In graphical representation, MAE is the vertical distance of a prediction point to the $x = y$ line as in Figure 10. MAE can be expressed as follows.

$$MAE = \frac{1}{n}\sum \left| y_{target_i} - y_{predict_i} \right|$$

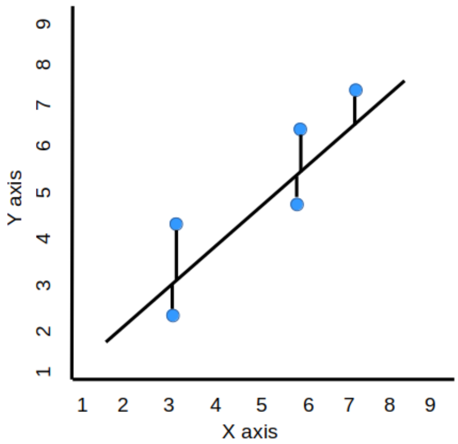

Figure 10. Mean absolute error representation.

The main loss function is the MSE, and the MAE is only used due to its easy understanding in representing the errors in geometrical representation. The early stopping method is also utilized to save the running time where the training process will be stopped if the errors are not getting better in 200 epochs.

4. Results and Analysis

4.1 Training, Validation and Testing

Training, validation and testing of the model is conducted using the NACA airfoil datasets consist of 70 airfoil configurations, 16 variations of angle of attack, 5 variations of Reynolds Number and about 5200 samples. The samples are splitted into 72% training set, 18% validation set and 10% test set.

The datasets have been trained with initial 1000 epochs, but the mean square errors get converged and did not improve after about 600 epochs, so the training is stopped at about 600 epochs. The error convergency can be seen in Figure 11 and Figure 12 for the mean square error and the mean absolute error, respectively. From those figures, the errors are starting to get converged after 100 epochs. Both training error and validation error gives a low value. The validation errors are slightly above the training error which indicates a good fitting of the model. The validation error does not diverge overtime which also indicates that there is no indication of overfitting. The final mean square error (MSE) and mean absolute error (MAE) are:

$$MSE = 0.000447$$

$$MAE = 0.013771$$

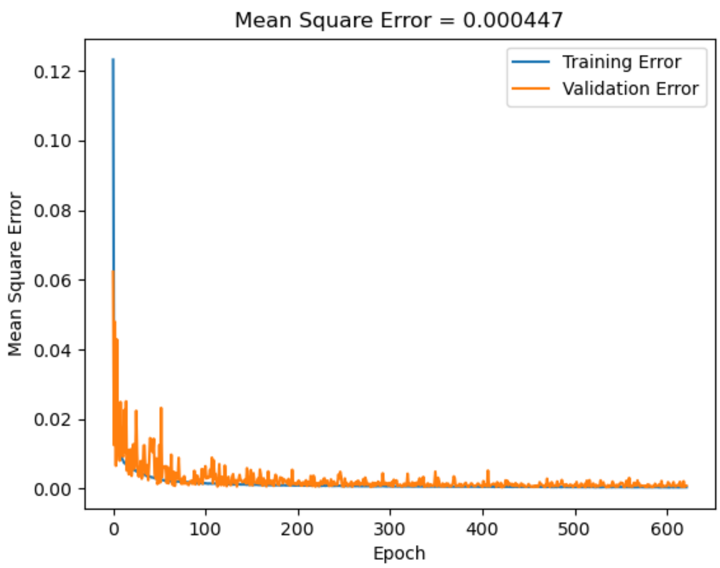

Figure 11. Mean square error convergency

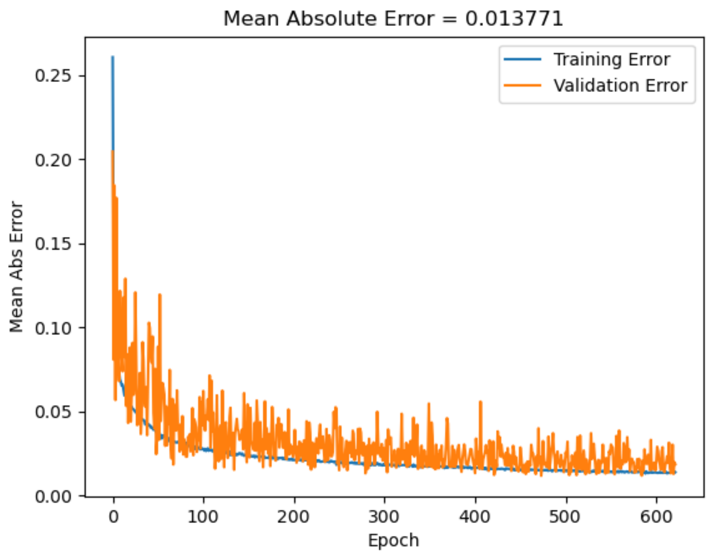

Figure 12. Mean absolute error convergency

The test dataset has been predicted by the model. The difference between the actual value and the predicted value by the model using the test datasets with about 500 sample can be seen in Figure 13. The MAE and MSE are slightly bigger than the training MAE and MSE which means the model is a good fit.

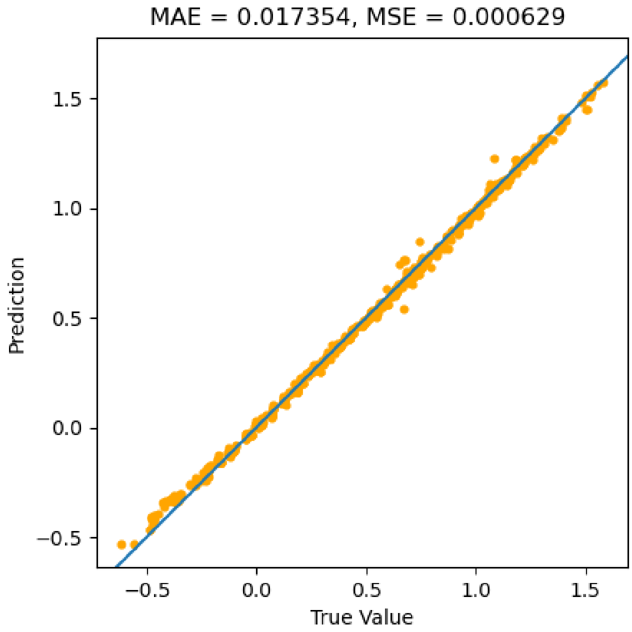

Figure 13. Test dataset prediction

4.2 Selig Airfoil

The Selig dataset use 6 different airfoil configurations, 3 variety of Reynolds Number and 8 variety of angle of attack. The Selig datasets contains of 150 samples in total. The prediction of the aerodynamics characteristics is conducted using the model trained by the NACA airfoil datasets. The Selig dataset prediction can be seen in Figure 14.

From Figure 14, the prediction is more scattered from the true value compared to the previous test dataset prediction. The errors are also bigger than the training error and the dataset error. But overall, the model is quite good in predicting the characteristics of the Selig airfoil.

The errors can be bigger relatively to the test errors caused by several things. First, the Selig airfoil is designed for low-speed conditions and the dataset is also in low speed. Whereas the

NACA airfoil is for more general purpose and the NACA dataset is in medium speed conditions. These differences can make the accuracy of the machine learning model gets lower. Second, some of the Selig airfoil have high thickness and more cambered whereas the NACA datasets generally have medium thickness and less cambered.

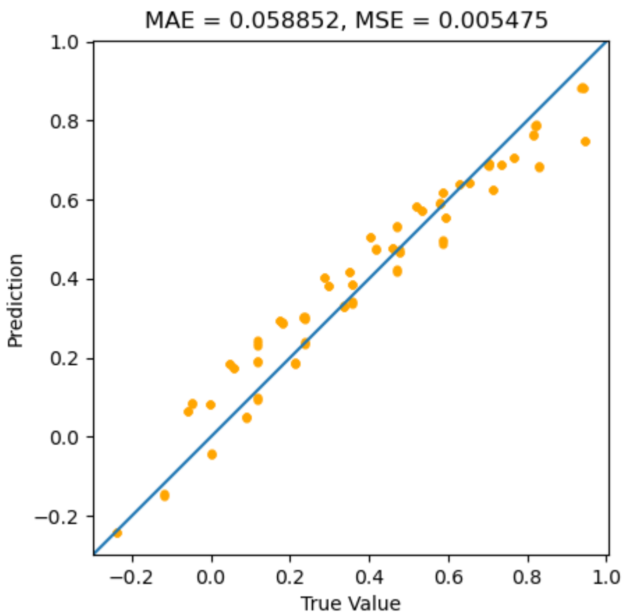

Figure 14. Selig dataset prediction

Some ways that can be done to improve the model. One of which is to use greater dataset of NACA airfoil which covers the high and low Reynolds Number, high and low angle of attack and more various configurations, so the model can catch the real physics in the airfoil analysis and produce more general predictions that can used for any other airfoils.

5. Conclusion

From the results and analysis above, it can be concluded that:

- The artificial neural network can be used to predict the aerodynamic characteristics of airfoil.
- The errors of the neural network model for predicting the aerodynamic characteristics are good enough, and slightly higher when predicting other datasets of airfoil series.
- The accuracy can be improved by adding more datasets that covers a wide variety of variables.

References

[1] Abbot, I. H., von Doenhoff, A. E., Stivers, Jr., L. S., *Summary of Airfoil Data*, National Advisory Committee for Aeronautics, Report No. 824, Langley Field, Virginia, 1945.

[2] Bouhlel, M. A., He, S., Martins, J. R. R. A., "*Scalable gradient-enhanced artificial neural networks for airfoil shape design in the subsonic and transonic regimes,"* Structural and Multidisciplinary Optimization, Springer, Germany, 2020.

[3] Selig, M. S., Guglielmo, J. J., Broeren, A. P., Giguere, P., *Summary of Low-Speed Airfoil Data*, Department of Aeronautical and Astronautical Engineering, University of Illinois at Urbana-Champaign, Illinois, Unites States, 1995.

Attachement

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Dense

Datasets=pd.read_csv('Datasets2.csv',sep=';')
Airfoil=pd.read_csv('Airfoil Datasets.csv',sep=';')
df=Airfoil.merge(Datasets, on='Airfoil')

X = df.iloc[:, 1:104]
y = df.iloc[:, 104]

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X = scaler.fit_transform(X)

Xselig=X[5209:, :]
yselig=y[5209:]

X=X[:5208, :]
y=y[:5208]

X_train,  X_test,  y_train,  y_test  =  train_test_split(X,  y,  test_size=0.1,
   random_state=0)

def build_model():
    model = Sequential()
    model.add(Dense(16, activation='tanh', input_shape=[X_train.shape[1]]))
    model.add(Dense(100, activation='tanh'))
    model.add(Dense(100, activation='tanh'))
    model.add(Dense(100, activation='sigmoid'))
    model.add(Dense(100, activation='sigmoid'))
    model.add(Dense(100, activation=tf.keras.layers.LeakyReLU()))
    model.add(Dense(100, activation='relu'))
    model.add(Dense(1))
    optimizers=tf.keras.optimizers.RMSprop(0.001)

    model.compile(loss='mse', optimizer=optimizers, metrics=['mae', 'mse'])
    return model

model=build_model()

class PrintDot(keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs):
        if epoch % 100 == 0: print('')
        print('.',end='')

EPOCHS=1000

early_stop=keras.callbacks.EarlyStopping(monitor='val_loss',patience=200)

history = model.fit(X_train, y_train, epochs=EPOCHS,
                    validation_split=0.2, verbose=0,
```

```
                    callbacks=[early_stop, PrintDot()])

def plot_history(history):
    hist=pd.DataFrame(history.history)
    hist['epoch']=history.epoch

    plt.figure(1)
    plt.xlabel('Epoch')
    plt.ylabel('Mean Abs Error')
    plt.plot(hist['epoch'],hist['mae'],
             label = 'Training Error')
    plt.plot(hist['epoch'],hist['val_mae'],
             label = 'Validation Error')
    plt.title('Mean Absolute Error = {:5.6f}'.format(hist['mae'][len(hist)-1]))
    plt.legend()

    plt.figure(2)
    plt.xlabel('Epoch')
    plt.ylabel('Mean Square Error')
    plt.plot(hist['epoch'],hist['mse'],
             label = 'Training Error')
    plt.plot(hist['epoch'],hist['val_mse'],
             label = 'Validation Error')
    plt.title('Mean Square Error = {:5.6f}'.format(hist['mse'][len(hist)-1]))
    plt.legend()

plot_history(history)

loss, mae, mse = model.evaluate(X_test, y_test, verbose=0)

print("\nTesting Mean Abs Error: {:5.6f} ".format(mae))

test_predictions = model.predict(X_test).flatten()
plt.figure(3)
plt.scatter(y_test, test_predictions,s=12,c='orange')
plt.xlabel('True Value')
plt.ylabel('Prediction')
maePred=(abs(test_predictions-y_test)).sum()/len(test_predictions)
msePred=((test_predictions-y_test)**2).sum()/len(test_predictions)
plt.title("MAE = {:5.6f}, MSE = {:5.6f} ".format(maePred,msePred))
plt.axis('equal')
plt.axis('square')
plt.plot([-100,100],[-100,100])

selig_prediction=model.predict(Xselig).flatten()
plt.figure(4)
plt.scatter(yselig, selig_prediction,s=12,c='orange')
plt.xlabel('True Value')
plt.ylabel('Prediction')
maeSelig=(abs(selig_prediction-yselig)).sum()/len(selig_prediction)
mseSelig=((selig_prediction-yselig)**2).sum()/len(selig_prediction)
plt.title("MAE = {:5.6f}, MSE = {:5.6f} ".format(maeSelig,mseSelig))
plt.axis('equal')
plt.axis('square')
plt.plot([-100,100],[-100,100])
```