Prediction of Transition and Separation Location on Airfoils Using Machine Learning

Lu Changqing, Aerospace Engineering (2019), *University of Michigan*

***Abstract*—In this report, both neuron network model and decision tree model are used to predict the transition location of airfoils. The results from both regression (the exact location prediction) and classification (chordwise location bin prediction) are compared between two models and the model with simple configuration and good accuracy was chosen to predict the drag coefficient c_f distribution along the airfoil chord with good validation results.**

***Index Terms*—machine learning, transition point, separation point, airfoil, decision tree**

I. Introduction

In fluid dynamics, the study on transition and separation locations on an airfoil given a certain flow condition is rather important since they are related to the design of aircraft or aerodynamic components that provide aerodynamic forces and moments. However, the accurate transition or separation locations are computationally expensive to determine, which leads to expensive design iteration or fluid dynamics analysis.

Machine learning is now a widely applied scientific study of algorithms and statistical models where computers build mathematical models based on sample data and make predictions without being explicitly programmed.

In the following part of the report, different methods of machine learning are presented to show the results when they are applied to predict the transition and separation location on an airfoil when the geometry of the airfoils and flow conditions are input as features.

II. Features and Samples

A. Features

As stated in Section I, the input features are related to the geometry of the airfoils and the flow conditions. Given one airfoil with certain geometry, flow conditions are changed for different transition and separation location results. For airfoil shape, certain range of the geometry features are determined. Table I shows the range of interest, where c is the chord length of the airfoil. For the flow conditions, we want the range of interest to be as wide as possible and reasonable at the same time. Table II shows the range of the flow condition. Note that when implementing different machine learning method, the range of angle of attack might be different during the process and the range shown below is the widest range of interest. In order to produce better models during the machine learning process, all of the features are normed into a range of $[0, 1]$.

Geometry	Flow Condition
thickness t	angle of attack α
camber t_c	Reynolds number Re
max camber location x_c	

Geometry	Range
thickness t	$5\% \sim 40\%c$
camber t_c	$0\% \sim 8\%c$
max camber location x_c	$10\% \sim 50\%c$

Table I: Geometry Feature Range

Table II: Flow Condition Range

Flow Condition	Range
angle of attack α	$-12^\circ \sim 18^\circ$
Reynolds number Re	$10^5 \sim 10^7$

B. Samples

Samples for training contains the features and the transition and separation location along the chord. The generation of samples are completed using *Xfoil*. *Xfoil* are also used to output the drag coefficient c_f curves of the airfoil.

Latin hypercube sampling method is used to choose uniformly distributed points within the given feature range.

III. Neuron Network Model - Regression

A. Model

Single vector machine is a classic machine learning method implemented by separating the space within feature space using hyperplanes and then predict output given input features. The python package `tensorflow.keras` is rather useful since it is a high-level interface to build machine learning models and do training.

For the method, several models for regression were implemented to see the influences of number of neurons or number of layers on the prediction results. A basic model containing 3 layers and 64 neurons on each layer was first implemented (Figure 1).

Figure 1: Model Example

```python
def create_model():
    model = tf.keras.Sequential()
    model.add(layers.Dense(64,activation = 'relu',input_shape =(4,)))
    model.add(layers.Dense(64,activation = 'relu'))
    model.add(layers.Dense(64,activation = 'relu'))
    model.add(layers.Dense(1))
    model.compile(optimizer = tf.train.RMSPropOptimizer(0.00001),loss = 'mse',
    return model
```

B. Results

For the training data, Latin hypercube sampling method gives 60 airfoil geometries at a constant maximum camber position at $40\%c$. The training data geometry features are shown in Figure 2. During the training process, total 9405 set of training data was used. 90% of the training data were used for training while 10% of the training data were used for validation. And `tensorflow.keras` was coded to stop the training process if the mean absolute error of the validation data stops decreasing, which is a way to prevent overfitting.

Figure 3 shows the an example training process which monitors the mean square error and mean absolute error.

Using the basic model shown in Figure 1, we used a set of test data to see whether the model was validated. The feature range of the test data is shown in Table III. Note that the maximum camber position of the set is set at a constant value.

Figure 2: Training Set (Geometry)

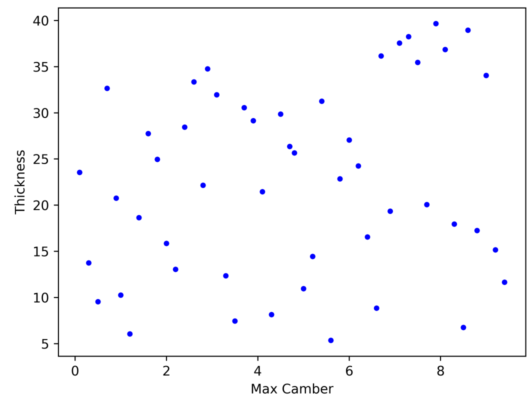


Figure 3: Mean Absolute Error(MAE) (left) and Mean Square Error(MSE) (right) decrease during the training process.

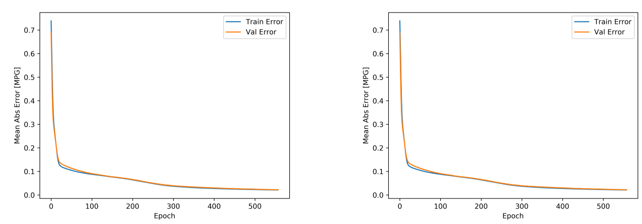


Table III: Test Set: 10 Airfoils

Feature	**Range**
thickness t	$25\%c \sim 30\%c$
max camber t_c	$4\%c \sim 8\%c$
max camber position x_c	$40\%c$

With the <u>basic model</u> trained, we can compare the prediction and the actual transition locations.

Changing the model in Figure 1 by decreasing one layer, the validation results of the test data are shown in Figure 6 and Figure 7.

C. Analysis

From Figure 5 and Figure 4, we can see that the predictions work for most of the data points since the data points are scattered along the $y = x$ line. More accurate predictions are made for the

Figure 4: Basic Model Transition Location Prediction vs. Actual - Top Surface

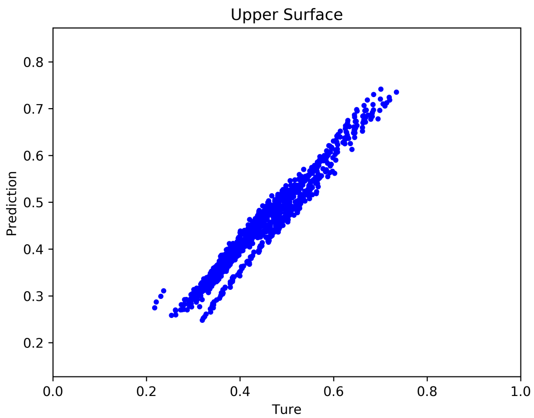


Figure 5: Basic Model Transition Location Prediction vs. Actual - Bottom Surface

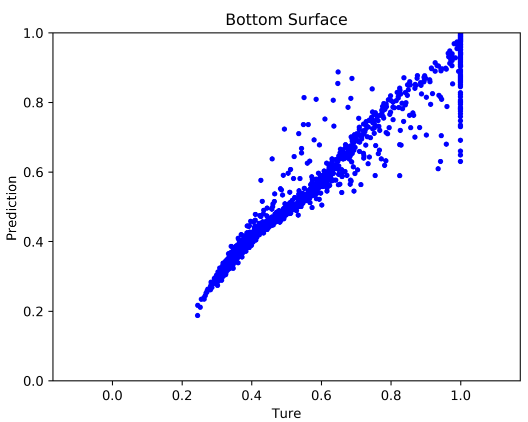


Figure 6: Decreasing One Layer: Transition Location Prediction vs. Actual - Top Surface

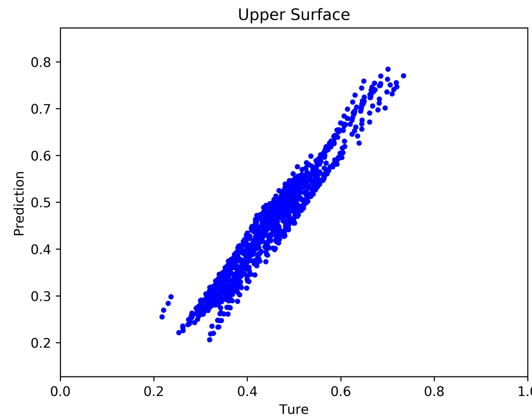
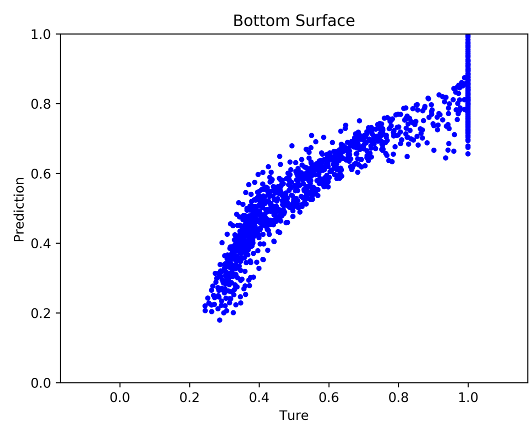


Figure 7: Decreasing One Layer: Transition Location Prediction vs. Actual - Bottom Surface



top surface. Since most airfoil transition location we are interested in are on the top surface given a positive angle of attack, the `tensorflow.keras` single vector machine approach to predict transition locations showed some promising results.

However, compare the results coming from two models, one with three hidden layers and one with two hidden layers, we can see that especially for the bottom layer, the prediction deviates from the true value once the actual transition locations are near the trailing edge of the airfoil. Therefore, to create a simple model with only a few layers and a small number of neurons while having accurate predictions is rather difficult given a small number of features we have considered.

IV. Neuron Network Model - Multi-Class Classification

A. Basic Idea

Sometimes for simple analysis purpose, we do not have to accurately predict the transition location with a ± 0.0001 accuracy. We only need to know on which part of the airfoil the transition happen. Therefore, we can establish a classification machine learning problem.

By sectioning the chord-wise location into 8 parts, we can classify the transition location results into 9 classes, which are, $1/8 \sim 1/4, 1/4 \sim 3/8, 3/8 \sim 1/2, 1/2 \sim 5/8, 5/8 \sim 3/4, 3/4 \sim 1$ of the chord length and no transition.

B. Sample and Model

From the results of the single vector machine regression approach, we can see that it is important to choose a set of training data such that the transition locations of the airfoils of choice are well-spread chord-wise. With a well-spread training set, the trained model can give more accurate prediction since the training process is not biased due to lack of training samples.

For the classification approach, we tried two set of training data to train the classification model. The geometry features of the airfoils in both sets are chosen using Latin hypercube sampling method. The geometry range of the airfoils are the same as Figure 2 with a constant maximum camber position location at $40\%c$.

For sample space 1, we have a total number of samples: 2259.

Table IV: Sample Space 1 Classification

Class	Top Surface	Bottom Surface
1	291	247
2	211	299
3	390	380
4	566	203
5	335	131
6	210	96
7	148	72
8	108	754
9	0	77

For sample space 2, we have a total number of samples: 24483.

Table V: Sample Space 2 Classification

Class	Top Surface	Bottom Surface
1	3454	2579
2	1866	2822
3	3997	4382
4	6457	2562
5	3954	1555
6	2119	1195
7	1399	879
8	1237	7878
9	0	711

As we can see from Table IV and Table V, the transition location distribution of both sets are consistent. It is interesting to see that transition happens on top surfaces for all geometry and flow conditions. And a large part of the transitions happening on bottom surface are within $7/8c \sim c$ chord-wise.

Given that when doing regression, the number of neurons on each layer is too large compared to the number of features, which is not ideal because the objective is to find a simple model with good prediction accuracy. With two sets of samples listed above, we also changed the training model configuration to see whether a simple while effective model can be found for this classification problem.

C. Results and Analysis

1) Sample Space 1 Results: For sample space 1, a training model consisting of 4 hidden layers with 20 neurons on each layer was trained. With a 90% and 10% training and validation split, as the training goes on, the accuracy of training sample increases to around 97% while the validation accuracy stays around 30%, which means the model trained does not work well with new data points within the geometry feature space.

2) Sample Space 2 Results: For sample space 2, we tried out three models, whose configurations are shown in Table VI.

Table VI: Model Configuration for Classification - Sample Space 2

Model	Layer	Neuron
1	3	12
2	3	8
3	3	6

After training, we use the data from sample space 1 to validate the prediction accuracy.

Table VII: Classification Accuracy Results - Sample Space 2

Model	Top Surface	Bottom Surface
1	91.4%	90.4%
2	$\mathbf{92.3\%}$	$\mathbf{90.6\%}$
3	88.6%	90.2%

As we can see from Table VII, model 2 clearly has the best performance with a comparatively simple model (only 8 neurons per layer) and the highest accuracy of all three models. Therefore, it is possible to find an effective and simple model when the prediction of transition locations is set up as a multi-class classification problem.

V. Decision Tree

A. Basic Idea

Decision tree is another machine learning approach. Instead of building a model to classify the features into multiple classes in one decision, the decision tree model focuses on one specific feature and classify the data points into two classes in one decision.

Using a python package `sktlearn`, we can set up the decision tree model rather easily and train with sample data faster than using the classical neuron network model in `tensorflow.keras`.

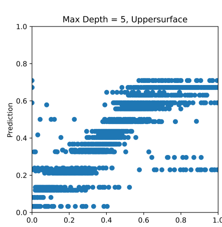
It turns out that we can also train the decision tree model with regression problem. Therefore, both classification and regression problems are set up to compare the decision tree with neuron network model approach.

B. Regression

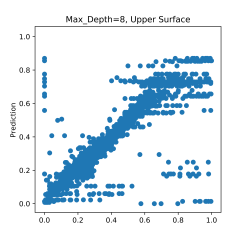
Using the sample space 2 as the training set and sample space 1 as the validation set, we trained the decision tree model with different maximum depth to compare the model accuracy.

Figure 8: Top Surface Transition Position Prediction Using Decision Tree Model with Varying Maximum Depth

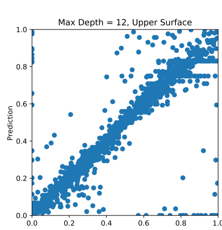
(a) max depth = 5



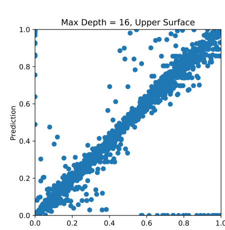
(b) max depth = 8



(c) max depth = 12



(d) max depth = 16



Note that the number of maximum depth is a parameter showing the maximum allowable complexity of the decision tree model. And from Figure 8, we can see that for this specific regression problem, a maximum depth of 12 has good prediction accuracy. Compare with the regression results shown in Section III, the decision tree model actually produces better results and the models generally take less time to train.

C. Multi-Class Classification

Based on the choice of the maximum depth in Section V-B, we use the decision tree model for the multi-class classification problem.

After training, the accuracy of prediction are shown in Table VIII.

Table VIII: Decision Tree Classification Results

Max depth	**Top Surface**	**Bottom Surface**
12	83.98%	84.64%
16	86.14%	84.24%

Compare the classification results from decision tree model with the results in Table VII, we can see that the neuron network model predicts better for the classification problem.

VI. c_f Prediction

A. Basic Idea

Both separation location and transition location is related to the friction coefficient distribution on the surfaces of the airfoils. Therefore, to predict separation and transition location, we can also predict the c_f distribution as the target and then calculate the transition and separation location from the c_f distribution. In this way, only one model is needed for the c_f distribution compared to the fact that two separate models are needed if we want to predict separation and transition location respectively.

B. c_f data compression

Given a limited number of features (thickness, camber, maximum camber position, Reynolds number and angle of attack), we cannot expect accurate prediction of c_f on each node chordwise.

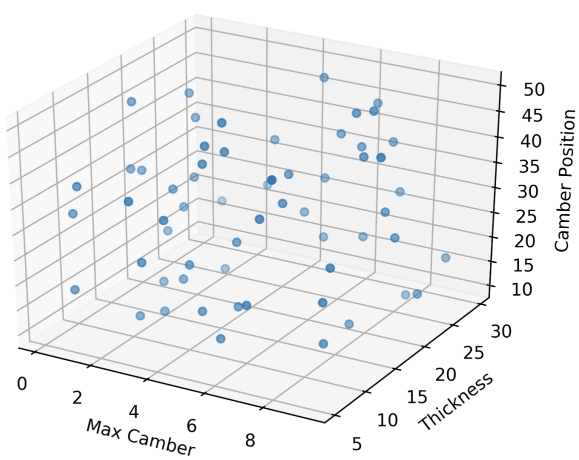
The $Xfoil$ gives c_f values at a limited number of chordwise node. And by single value decomposition (SVD), we can get a set of basis vectors such that these c_f curves (vector) are linear composition of these basis vectors. With this basis vector space, we can now represent any c_f curves by calculating the linear composition coefficients. These coefficients are calculated by projection c_f curves onto each basis vector.

Ideally, only part of the basis vectors are dominant in representing the c_f curves. The coefficients of these dominant basis vectors are the ones that contain the most important information about the c_f distribution over the airfoil. By training a machine learning model for each dominant coefficient, we can predict the coefficients of these dominant basis vectors and reconstruct the c_f curves. From the c_f curves, transition and separation locations are easy to determine.

C. Sample Space and Validation Space

Note that before the prediction of c_f curves, a geometry feature - maximum camber location is a constant at $40\%c$. Now for the new sample space, we vary the maximum camber location from $10\%c$ to $50\%c$. Figure 9 shows the geometry feature of each airfoil sample we select.

Figure 9: Sample Space Geometry



Using $Xfoil$, we can input self-generated airfoil geometry files as well as airfoil files from other sources. Therefore, we will choose two sets of data to validate the model we trained.

1) Validation set 1 is the 10% of the airfoil shown in Figure 9.
2) Validation set 2 is from the airfoil database. We chose airfoils randomly and generate a few cases for $Xfoil$ to compute the c_f distribution.

D. Bottom Surface c_f

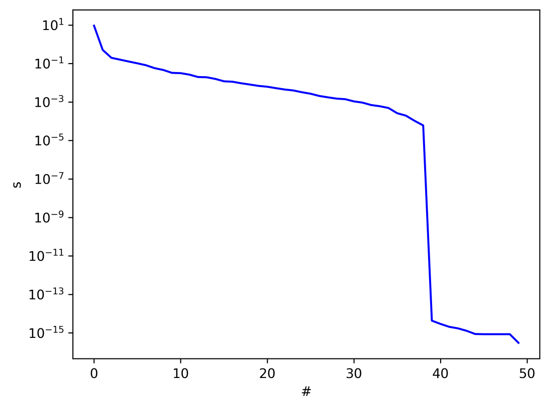
Single Value Decomposition

From the c_f curve results generated by sample space shown in Figure 9, we chose randomly $N = 500$ curves from all the samples to do single value decomposition (Equation 1),

$$\mathbf{M}_{c_f} = \mathbf{USV}^{-1} \tag{1}$$

where $\mathbf{M}_{c_f}$ is the 50×500 matrix containing the c_f curves on 50 chordwise stations. Figure 10 shows the diagonal values of s in the diagonal matrix $\mathbf{S}$.

Figure 10: Single Value Decomposition - s



In Equation 1, the orthornormal matrix $\mathbf{U}$ contains the basis vectors. As we can see from the values of s in Figure 10, s decreases rapidly around the 38th basis vector.

As we can see from Figure 11, as the dominance of the basis vector decreases (s decreases), the chordwise oscillation of the unit vector increases. Basis vectors with little dominance, the chordwise oscillation is high (Figure 12).

Coefficient Prediction

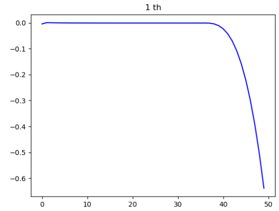
After generating the basis vectors, we can know do the projection of c_f vectors onto the basis vectors and get the corresponding coefficients results.

Based on the comparison between the neuron network model and decision tree model in Section III and Section V-B, we have concluded that for regression problems, decision tree model has high accuracy and the training process takes less time. Therefore, we used decision tree model with maximum depth of 25 for the coefficient regression problem. The training features are camber position, thickness, maximum camber position, angle of attack and Reynolds number.

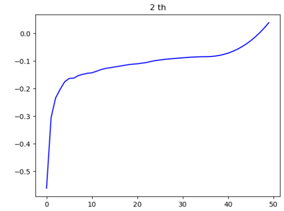
As we can see from Figure 14, as the dominance of the basis vectors decreases, the corresponding co-

Figure 11: Basis Vector 1-10

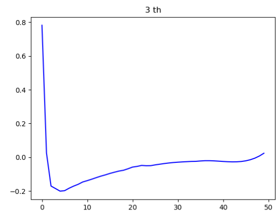
(a) Basis Vector - 1



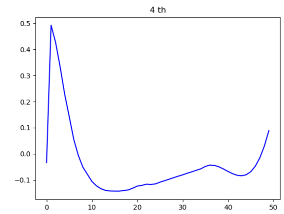
(b) Basis Vector - 2



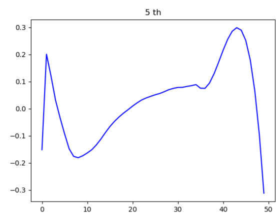
(c) Basis Vector - 3



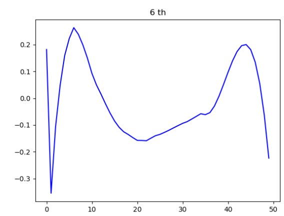
(d) Basis Vector - 4



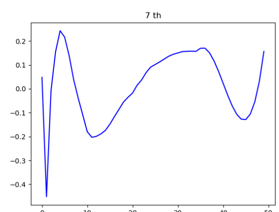
(e) Basis Vector - 5



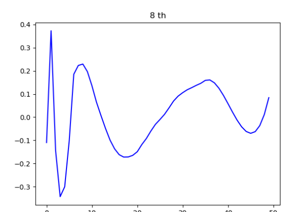
(f) Basis Vector - 6



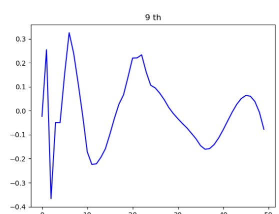
(g) Basis Vector - 7



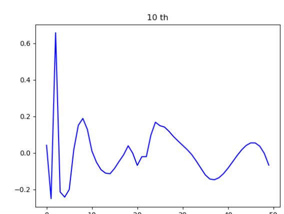
(h) Basis Vector - 8



(i) Basis Vector - 9



(j) Basis Vector - 10



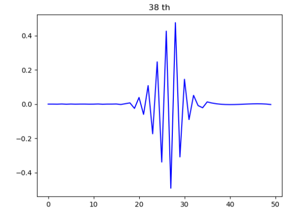
efficients are rather small. For basis vectors $11-18$, the coefficients are order of 10^{-3}. For basis vectors $21-28$, the coefficients are order of 10^{-4}.

And for the coefficients of the dominant basis vectors, the prediction versus true value gives good results (Figure 13), which means that the predictions of the c_f curves would be accurate.

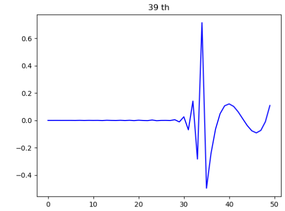
c_f Curves Reconstruction and Validation

Figure 12: Basis Vector 38-41

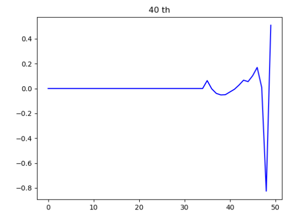
(a) Basis Vector - 38



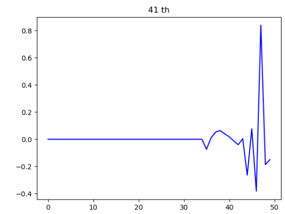
(b) Basis Vector - 39



(c) Basis Vector - 40



(d) Basis Vector - 41



After the prediction of the coefficients, we can reconstruct the c_f curves given the predicted coefficients and basis vectors. Since we have 10% of the airfoils are generated for validation purpose. We first see some examples of the reconstructed c_f curves and compare them with the results given by $Xfoil$.

Figure 15 shows the comparison where the red curve gives the c_f results from $Xfoil$ and the blue curve gives the prediction. As we can see, the blue curve basically captures the trend of the red curves except for some extreme cases on the leading edge and trailing edge. Figure 15 only shows part of the results (case chosen randomly).

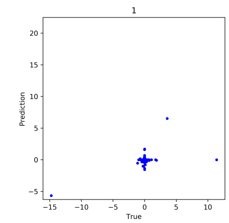
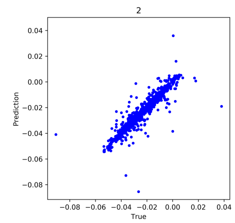
c_f Curves Prediction (Extrapolation)

After validate the prediction results in the sample space (Figure 9), we can try using the model to predict the airfoils that are not designed according to the NACA series calculation. Since we only have five features, which cannot accurately represent all the geometric characteristics and the flow condition, we expect the extrapolation prediction results poorer than the validation examples.

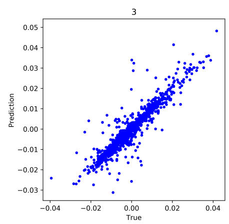
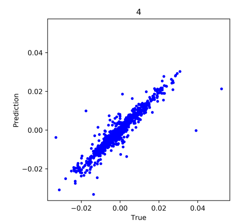
As we can see from Figure 16, the prediction results for some cases are acceptable while some of the results are poor. This confirms our expectation and means that there are still other important features that are not included while generating the model. For example, the maximum thickness chordwise position is a geometric feature that are not considered in our model. However, the extrapolation results show that machine learning might be a

Figure 13: Coefficient 1-10 Prediction vs. True

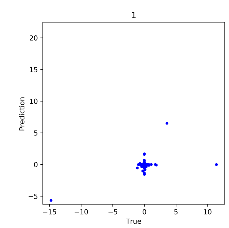
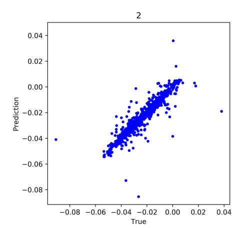
(a) Coefficient - 1 (b) Coefficient - 2

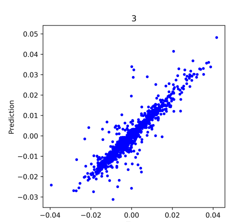
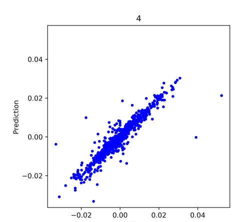
(c) Coefficient - 3 (d) Coefficient - 4

(e) Coefficient - 5 (f) Coefficient - 6

(g) Coefficient - 7 (h) Coefficient - 8

(i) Coefficient - 9 (j) Coefficient - 10

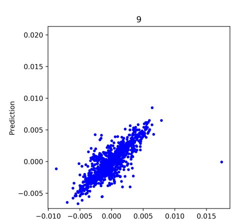
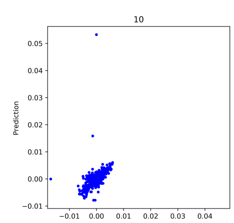
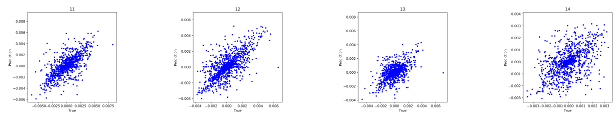



Figure 14: Coefficient 11-18 Prediction vs. True

(a) 11 (b) 12 (c) 13 (d) 14



(e) 15 (f) 16 (g) 17 (h) 18

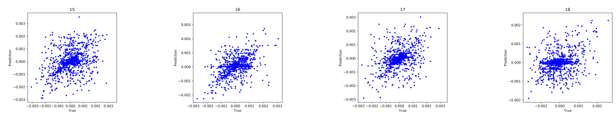
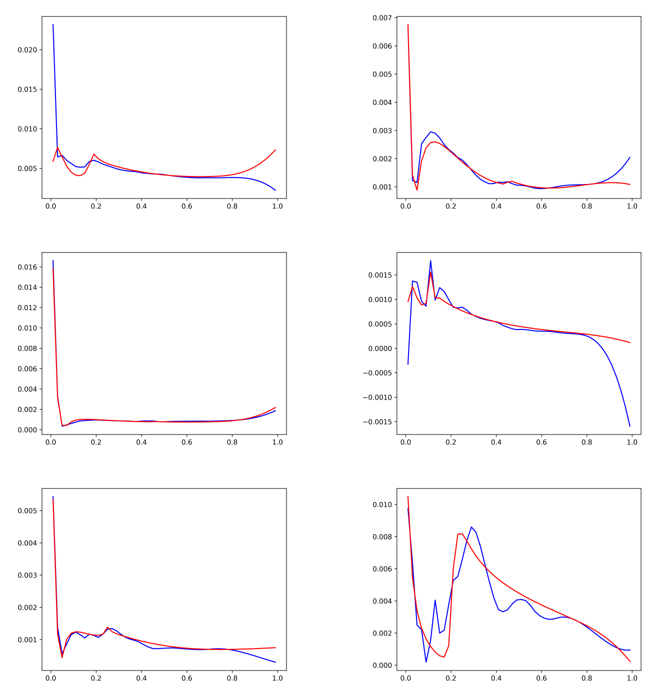


Figure 15: c_f Curves Reconstruction Validation

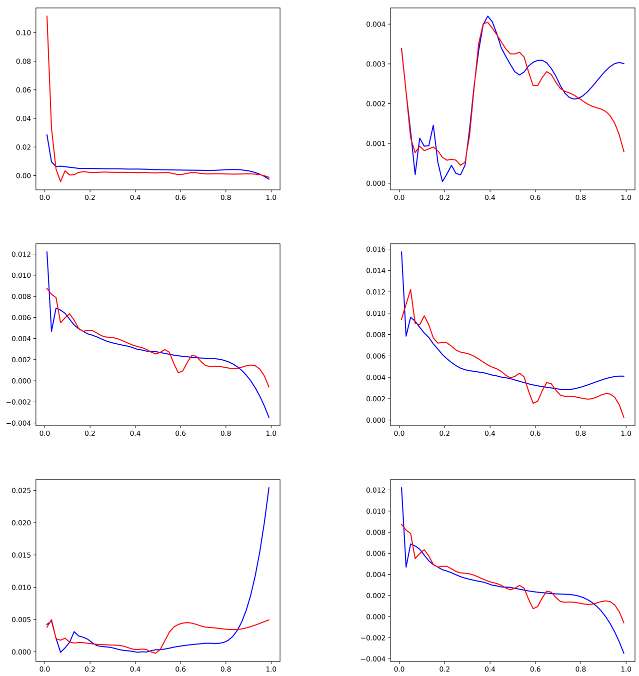


promising approach to approximate the c_f curves given the airfoil shape and flow condition is within a certain range.

VII. Limitation

1) The number of geometry features of the airfoils are limited to maximum camber, thickness and maximum camber position, which can not entirely represent the geometric characteristics of a general airfoil. And the airfoil geometry file are calculated according to NACA series while many airfoils do not follow the NACA series rules, which would lead to poor prediction results when doing prediction extrapolation.
2) The geometry differences of thick airfoil and thin airfoil are not considered during training. A rather general sample space are chosen. However, if the model is trained for more specific airfoils with specific geoemtry parameter range, the prediction would be more accurate.
3) Similarly for the flow conditions, the angle of attack range is a wide range of 30° and the Reynolds number range is between 10^5 and

Figure 16: c_f Curves Prediction Extrapolation



10^7. If model is trained for more specific flow condition range, the prediction would be more accurate.

VIII. Conclusion

In conclusion, by constructing some simple neuron network model and decision tree model, machine learning is proved to be useful in prediction of the transition location on airfoils as well as the drag coefficient c_f distribution over the airfoils. While extrapolation into airfoil database shows some poor prediction results, by classifying sample space into different geometry range and train models respectively, the prediction results would be improved. Addition of geometry parameters would also improve the prediction.