HIGH SCHOOL EDITION
Journal of Student Research
Volume 13 Issue 4 (2024)

Utilizing Machine Learning to Predict Static Pressure Over a Wingtip

Bo Gardner

Saint Anns School, USA

ABSTRACT

Our goal was to find expected pressure around different wingtip shapes to predict vortice behavior. This project focused on aerodynamics, specifically the location where high and low-pressure air mixes over a lifting surface. High-pressure air mixes with the low-pressure air at the wingtip of a plane creating vortices that cause drag, which wastes fuel and slows down the aircraft. Not only is this bad for the environment, but it increases the cost of flight and affects the distance that larger planes can fly ahead of smaller planes due to wake turbulence. As planes have gotten lighter, faster, and safer, the issue of wingtip vortices and drag has continued to be a problem. The approach we used to answer this problem was to select an applicable data set using continuous machine learning models and later, discrete models to predict a pressure coefficient above the wing. We combined multiple datasets from the same research paper created by NASA to have numerous factors for the machine learning model to predict. As a result, we produced accurate static pressure predictions with 80% to 90% accuracy. Even more accurate were our model recall scores which were within 99%. As a result of the work done on this project, accurate predictions of expected pressure over an airfoil are achievable. With only a few input variables about speed and dimensions, an accurate static pressure can be found.

Introduction

As aviation technology advanced from the first powered aircraft, designers continued to strive to design faster vehicles. For military aircraft post World War 2, speed was a top priority, as it was for commercial aircraft with the dawn of the jet age. These high-speed planes are designed to transport people and cargo across the globe. However, the effects of wingtip vortices contribute to drag, threatening to slow down planes. By choosing to use more powerful engines, the core of the issue was fixed. Quicker planes meant increased fuel loss. During the 1970s, a design focus trend began that stepped back from increasingly fast planes, and instead looked at making smaller incremental improvements to efficiency and cost. More efficient engines, lighter materials, and other designs to help with aviation efficiency have since been developed. In 1977, NASA, USAF, and Boeing worked on a project to add winglets to a KC-135 aircraft (Fig 1). These winglets proved themselves with testing, as there was a 7% increase in lift-to-drag ratio. In the late 1990s, Boeing began using these winglets on existing aircraft to increase their efficiency. In the coming years, winglet designs were implemented into most future aircraft designs. In 2010, it was reported that Southwest Airlines 737-700 planes saved 100,000 gallons of fuel per year as a result of these winglets. Winglets work by creating a forward lift component on the tip of a wing. Without them, different pressure air from above and below mixes to create powerful vortices. This vortex then travels above the wing pushing the component of lift backward, thus creating drag. With a winglet that is angled inward, the air that flows over the winglet generates forward lift

Figure 1. KC-135A fitted with winglets. [9]NASA, O., 1979, KC-135A in flight - winglet study, In Flight.

For a winglet to be efficient, it must be designed to generate lift in a forward direction. However, a winglet is perpendicular to the fuselage and thus it generates lift towards the fuselage which is both useless and efficient. However, the wingtip vortices that are created change the direction of airflow over the wing, causing the direction of lift to be forward which is desirable. As a result, the goal should not be to fully remove the wingtip vortices, but rather to understand how the vortices will behave to construct a preferable winglet. The type of data used to predict the wingtip vortices created is continuous numerical data. Pressure around and over a lifting surface can be measured as a coefficient. Velocity at different points is also measured numerically. This makes predicting pressure possible using machine learning algorithms.

Background

In the past decade, much work has been done in the study of wingtip vortices. The most common approach used for these studies has typically been to experiment with wingtip shapes and angle of attack inside a wind tunnel, with flow visualization. These wind tunnels work by moving air at high speed over the object being tested. To gather data, forces can be measured with sensors attached to the object. Information about the airflow can be found using a variety of other techniques; using smoke injectors(Fig 2) and laser slits, 2d image planes visualizing airflow can be created. To measure pressure, additional sensors can be added to various sections of the wing.

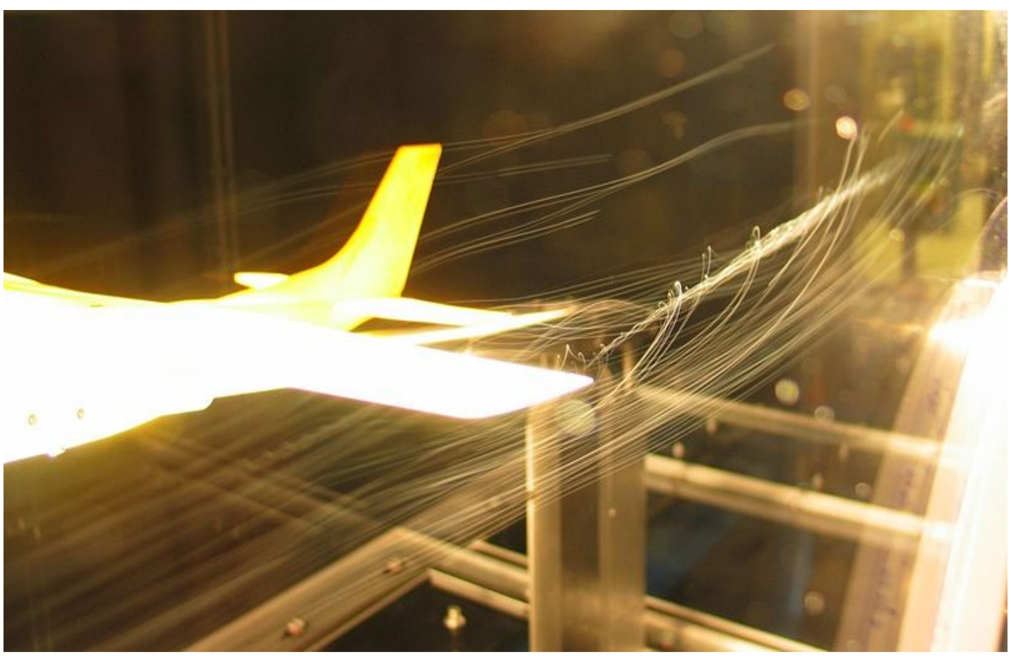

Figure 2. Flow visualization inside a wind tunnel demonstrating wingtip vortices formation. [10]Dale, B. F., 2007, Cessna 182 model-wingtip-vortex.

Experiments specific to wingtips have been done by exploring the addition of wingtips of different shapes. These experiments gather significant data for those specific shapes, but if attempting to use the information to control vortices or something equally specific, much trial and error is required. *An Experimental Study and Database for Tip Vortex Flow From An Airfoil(K.B.M.Q. Zaman, Amy F. Fagan, and Mina R. Mankbadi)* looked at different wingtips and winglet designs on a NACA0012 airfoil. Many images and numerical data were found from the experiments with information about vortex creation.

Using CFD software, it remains challenging to find the most optimal winglet and wingtip designs. Each test of the design is computationally heavy, making it difficult to test out a sizable amount of designs. Aircraft Winglet Design Increasing the aerodynamic efficiency of a wing *(HANLIN GONG ZHANG, ERIC AXTELIUS)* used CFD software to test different common winglet shapes. Due to the computational strain per simulation, only 9 shapes were tested. Many more specific designs that exist in between these could be more efficient.

For these reasons, implementing a machine learning algorithm to predict airflow has many advantages. With non-discrete data, all possible inputs can be predicted and instead of testing just a few shapes, all possibilities can be tested. Using machine learning for this has some drawbacks. Particularly for more complicated shapes, it is difficult to have an algorithm capable of predicting results, unless a vast amount of data is collected for training. This is the problem that the application of machine learning is aiming to fix.

Dataset

Our dataset was created by NASA. As a part of the research paper: *Turbulence Measurements in the Near Field of a Wingtip Vortex; Jim Chow, Greg Zilliac, and Peter Bradshaw),* a NACA 0012 wingtip was tested inside a wind tunnel at NASA Ames. (Fig 3) Using various methods, they measured for differences in speed and pressure around the wingtip.

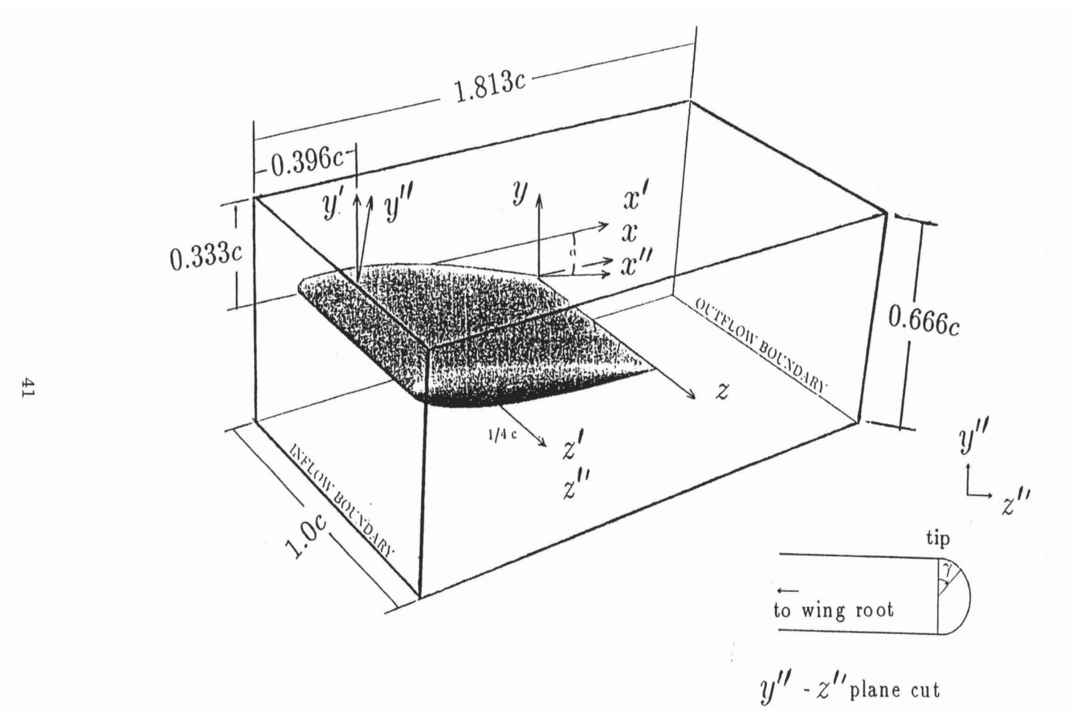

Figure 3. Schematic showing measurements for data collection. [7]Chow, J., Zilliac, G., and Bradshaw, P., 1997, "Turbulence Measurements in the Near Field of a Wingtip Vortex," NASA Technical Reports Server [Online]. Available: https://ntrs.nasa.gov/api/citations/19970011348/downloads/19970011348.pdf. [Accessed: Dec-2023].

Static pressure and pressure at the wingtip were measured using three wire probe apparatuses. X, Y, and Z. Coordinates were measured using force measurement devices. The NACA 0012 wingtip segment was placed inside an open circuit wind tunnel with a width of 32 inches and a height of 48 inches. (Fig 4)

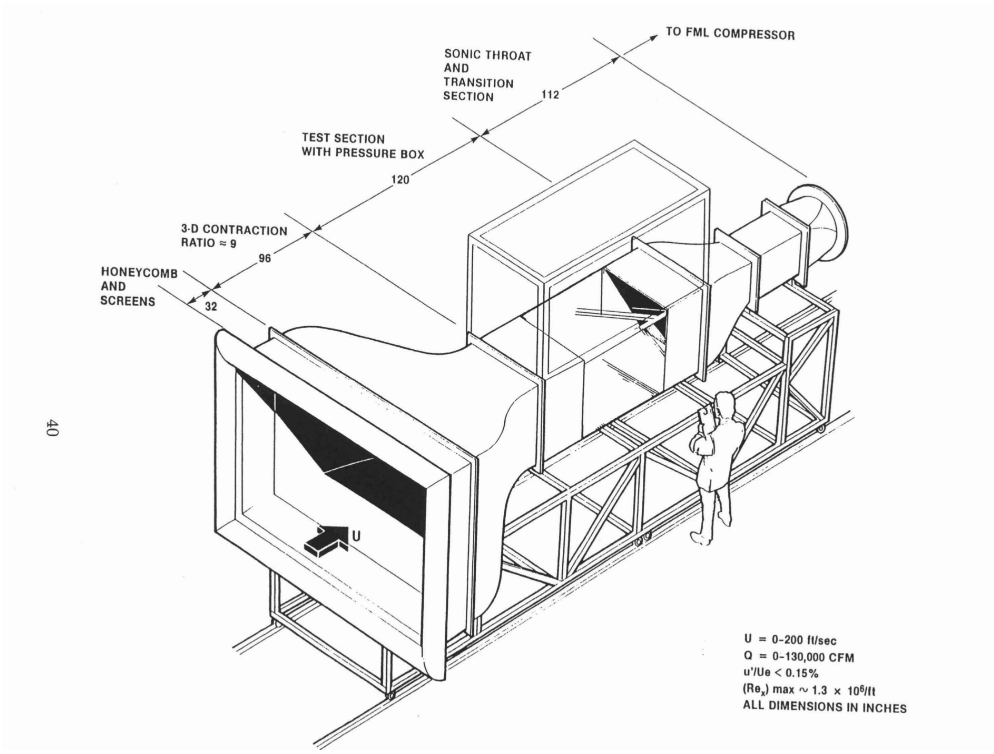

Figure 4. Diagram of wind tunnel design. [7]Chow, J., Zilliac, G., and Bradshaw, P., 1997, "Turbulence Measurements in the Near Field of a Wingtip Vortex," NASA Technical Reports Server [Online]. Available: https://ntrs.nasa.gov/api/citations/19970011348/downloads/19970011348.pdf. [Accessed: Dec-2023].

The data collected consisted of two separate .dat files. The first contained X, Y, and Z columns with measurements as well as speed data at different parts of the wingtip: u/uinf, v/uinf, and w/uinf. The other file contained the same X, Y, and Z data, and then Cp, Static, and Cp, total pressure measurements. Once combined into a single dataset, the data was continuous numerical data. 7,279 columns of data exist in the dataset. This was slightly smaller than ideal but provided us with enough data from which to work. (Fig 5)

	x	y	z	u/Uinf	v/Uinf	w/Uinf	Cp,total
0	-54.703	0.0	0.000	0.961	-0.007	-0.018	1.007
1	-54.703	0.0	1.000	0.955	-0.004	-0.021	1.006
2	-54.703	0.0	2.000	0.948	-0.005	-0.025	1.005
3	-54.703	0.0	3.000	0.942	-0.005	-0.027	1.004
4	-54.703	0.0	4.000	0.935	-0.005	-0.031	1.004
...	...	...	...	...	...	...	...
7274	32.297	23.0	20.807	0.937	-0.041	0.095	0.981
7275	32.297	23.0	22.551	0.936	-0.038	0.079	0.979
7276	32.297	23.0	24.341	0.936	-0.038	0.063	0.979
7277	32.297	23.0	26.163	0.935	-0.043	0.049	0.980
7278	32.297	23.0	28.000	0.935	-0.040	0.039	0.982

[7279 rows x 7 columns]

Figure 5. Screenshot of dataset preview.

Methodology / Models

The programming work was input using Google Colab. To use machine learning algorithms on the dataset, we first organized the data into one readable file. The two separate files were joined using a simple Python script. Because both datasets contained the same XYZ data, that column was dropped. To visualize the data, we used the Python library: Seaborn, to create visualizations. The first data visualized was Cp, Statics relationship with X, Y, and Z values. (Fig 6)

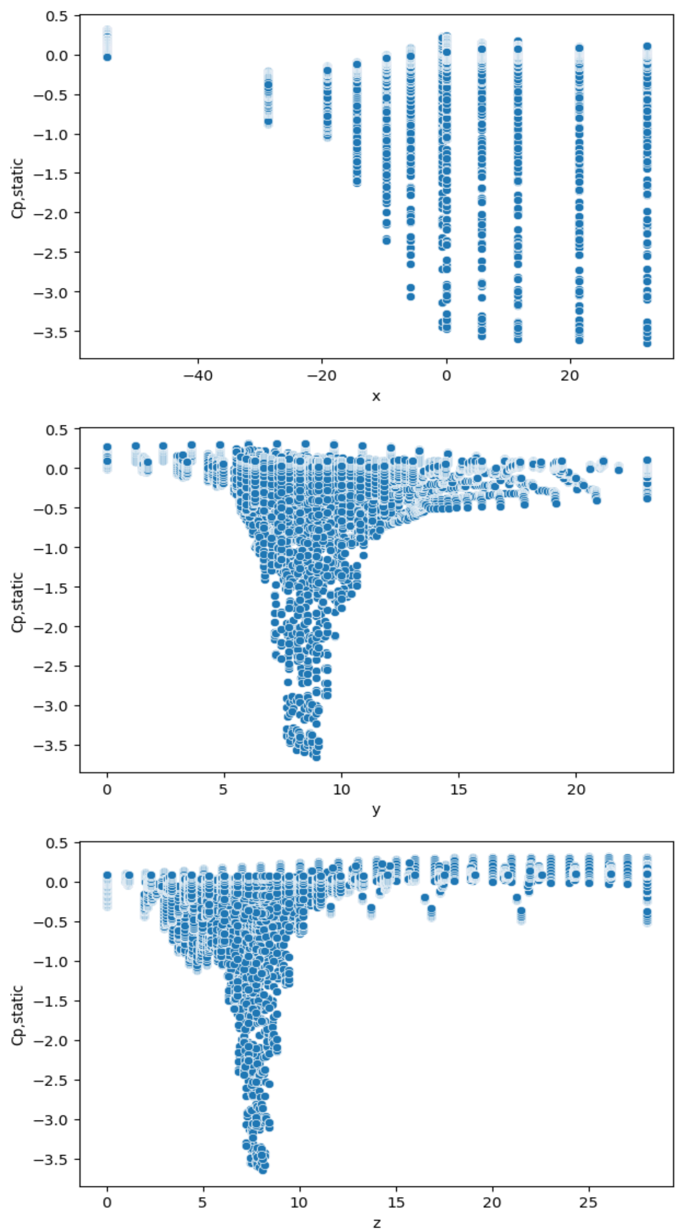

Figure 6. Cp Static values for X, Y, and Z values.

Data was also described using built-in Python "describe" functions. (Table 1) This showed mean, standard deviation, minimum values, percentiles, and maximum values for the entire data set.

Table 1. Results from the "describe" function. Df.describe values

	X	Y	Z	u/Uinf	v/Uinf	w/Uinf	Cp, static	Cp, total
Count	7279	7279	7279	7279	7279	7279	7279	7279

Mean	-4.960024	11.082556	10.000872	0.972867	-0.007056	0.06604	-0.298303	0.868165
Std	21.932157	5.421255	6.976357	0.164826	0.298030	0.315727	0.553720	0.299657
Min	-54.703000	0	0	0.0078	-0.971	-1.032	-3.649	-0.761
25%	-14.431000	7.632	5.415	0.928	-0.191	-0.035	-0.438	0.952
50%	-0.723000	10.052	7.761	0.968	-0.025	0.098	-0.125	0.996
75%	11.568000	14.2435	13	1.034	0.104	0.23	0.027	1.003
Max	32.297000	23	28	1.772	1.052	1.02	0.3130.313	1.053

Once we had a solid understanding of the data, we applied different machine-learning models. Data was split into training and testing data, as well as into the X vector and Y output. The Y output we aimed to find was Cp, Static, as it represents the behavior of the vortices over the wing, and all the rest of the data was put into the X vector. The data was split so that the testing data was 20% of all the data. Due to the limited amount of data, we chose to use as much data as possible for training, while still leaving enough for testing. After splitting the data, we had 5,823 rows of data for training and 1,456 rows for testing.

After preparing the data for the algorithms, we imported the Python library: scikit learn to apply these models. Throughout the entire research, we used both continuous numerical value models as well as discrete numerical value models. The first model applied was linear regression.

Linear regression works by finding a line of best fit for X vector data. This is done by taking training data and finding a line of best fit by minimizing the residual sum of squares. The second continuous model implemented was the random tree regression model. This model works by creating and comparing decision trees for the data that give the most accurate results. Specifically, it creates trees on smaller sections of the data and then compares them to find an average most efficient tree. For our implementation, we used the hyperparameter of a max depth of the trees being 2 levels. Another model that was applied to our data was support vector regression. This model works similarly to linear regression, but instead of finding a line of best fit, it finds a multidimensional hyperplane that represents the best fit for the data. We also applied a decision tree regressor model to our data which finds a decision tree that predicts the outputs. The last continuous data model that we used was the ridge regressor. We also experimented with using different hyperparameter values of alpha. We ended up graphing the accuracy from these different values to find the most optimal one. (See results section for outcome [Fig 10])

For our discrete model implementation, we first had to make our data contain only discrete Y outputs. To do this, we sorted Cp, Total pressure values into 2 groups: either 1 or 2. We sorted by finding the average value of all pressure values and sorting them into values above it and values below.(Fig 8)

```python
import numpy as np
pressure_c = []
# iterate over entries of Cp,total column and append 0 to pressure_c if entry < average, otherwise append 1
pressure_total = df['Cp,total'].tolist()

avg = np.mean(pressure_total)

for p in pressure_total:
  if p < avg:
    pressure_c.append(0)
  else:
    pressure_c.append(1)

df['CpTotClas'] = pressure_c
print(df)
print(sum(pressure_c))

y = df['CpTotClas']
print(X)
print(y)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,)
```

```
           x     y       z  u/Uinf  v/Uinf  w/Uinf  Cp,total  CpTotClas
0    -54.703   0.0   0.000   0.961  -0.007  -0.018     1.007          1
1    -54.703   0.0   1.000   0.955  -0.004  -0.021     1.006          1
2    -54.703   0.0   2.000   0.948  -0.005  -0.025     1.005          1
3    -54.703   0.0   3.000   0.942  -0.005  -0.027     1.004          1
4    -54.703   0.0   4.000   0.935  -0.005  -0.031     1.004          1
...      ...   ...     ...     ...     ...     ...       ...        ...
7274  32.297  23.0  20.807   0.937  -0.041   0.095     0.981          1
```

Figure 7. Screenshot of data splitting code

After sorting, we were able to implement discrete machine learning algorithms. We ended up only using 1 discrete method: logistic regression. Logistic regression works by finding a curve that fits in between discrete data points.

Results and Discussion

After implementing all of these models for our data, we evaluated their ability to predict pressure data using a variety of methods. For the continuous models, we used the scikit Python command: mean_absolute_error. This compares the Y testing data answers with the prediction Y values the model gave; the lower the result, the more accurate the models' predictions were.

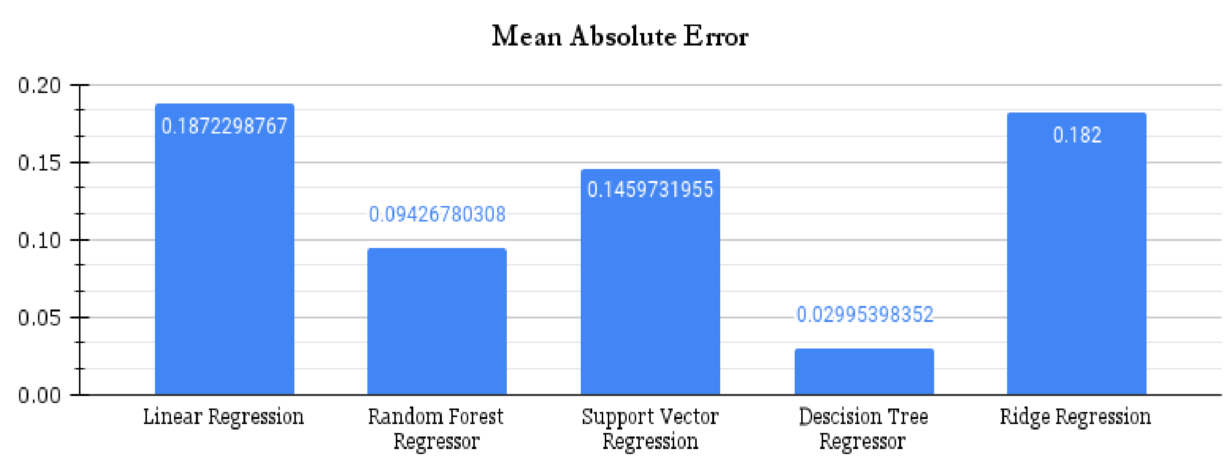

Figure 8. Mean absolute error results for the models tested.

For the continuous data ML models, the decision tree regressor performed the best, while the basic linear regression model performed the worst. The mean absolute error for all the models stayed under 0.2; working to predict static pressure based on only a few input variables, this is a satisfactory and positive result.

The second worst-performing model was the ridge regression model. To achieve slightly better results, we experimented with different hyperparameter values for alpha. Specifically, we ran through all alpha values between 0 and 1000. There was a visible correlation between the alpha values and the models' performance however, the scale was so small that the difference would be negligible. (Fig 10)

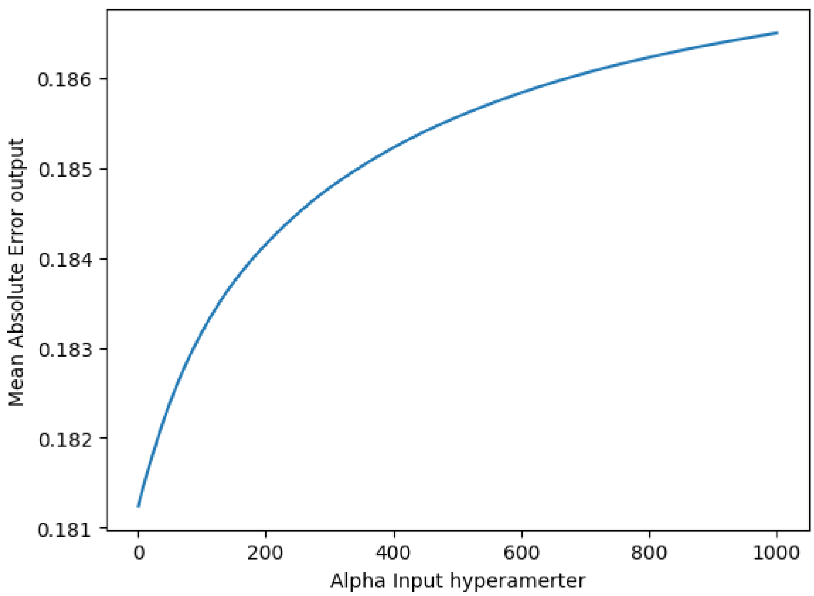

Figure 9. Graph of alpha values(x) and their effect on mean absolute error output(y)

The logistic regression model trained on the data performed well. To understand this performance, we utilized several different methods. We used scikit tools: accuracy, precision, recall, and F1 score commands, to compute the effectiveness of the model.

These scores are calculated using the following formulas: (Fig 11)

$$\text{Precision} = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Positive})}$$

$$\text{Recall} = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Negative})}$$

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{(\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative})} = \frac{\text{Total Number Correct}}{\text{Total Number of Samples}}$$

Figure 10. Formulas for precision, recall and accuracy scores.

$$F_1 = 2 * (Precision * Recall)/(Precision + Recall)$$

The formula for F1 score.

The precision of the model was 0.81, which is a favorable result. The recall score achieved by our model was 0.9982, and the F1 score was 0.893. After finding these results, we created a confusion matrix (Fig 13) to represent false positives, false negatives, correct positives, and correct negatives. This gave us a good visualization of the performance of the model.

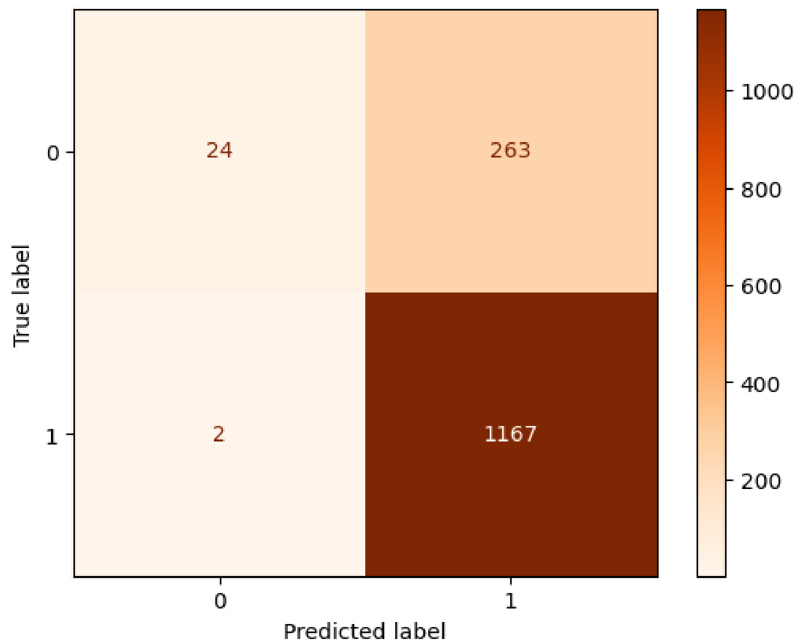

Figure 11.

As shown from these results we see that out of 1,169 values whose correct value was 1, the model incorrectly identified only 2. For predictions where the correct value was 0, the model did poorly, falsely labeling 263 out of 287 as 1, instead of 0. This is a major disadvantage of our logistic regression model.

Overall the models that used continuous data performed better than the logistic regression model, this is most likely because the data is a measurement of pressure, and so not inherently categorical. Our models did have some sources of error due to the size of the dataset. Considering the complexity of what the model is predicting, with only 7,000 approximate values, the quantity is not enough to train it comprehensively. Another issue is that the dataset we used is accurate for only the specific NACA0012 airfoil. If we were to test the models with a different airfoil, it would not be able to predict as well.

To solve some of these issues, future research could be achieved by collecting significantly more data using CFD (computational fluid dynamics) software, or wind tunnel testing over a long period. In addition, future data collection could be done for a vast amount of NACA airfoils, or even by documenting the dimensions of the chord length, camber line length, thickness, and profile. With these additional X vectors, the model could be trained to work on a variety of airfoil types.

Conclusions

As a result of our research, we were able to successfully apply and train both continuous and categorical/discrete data for the Turbulence Measurements in the Near Field of a Wingtip Vortex data collection created in 1997. Out of all the models used, the best performing was the decision tree regressor model. It performed with a mean absolute error of

0.0299. All of the models we trained were able to predict the static pressure of airflow over an airfoil to some degree. Knowing this information from speed and position data could be used to predict that path of wingtip vortices formed. Without the need to test each configuration, and with the ability to find precise values, optimal winglet positioning designs could be created. Since winglets' usefulness and performance is based on how they direct the wingtip vortices to create a forward lift component, this would be a helpful tool. If future research was to be done using machine learning to accomplish similar goals, collecting significantly more data, as well as testing on a huge variety of NACA airfoil designs could lead to a more versatile and applicable model. Our model works as a good demonstration of the usage of machine learning to predict airflow, but is not suitable for physical testing and usage.

Acknowledgments

We would like to thank Pablo Bonilla for guidance, and Inspirt AI for the opportunity to work with him and complete this project.

References

[1]Giuni, M., and Green, R. B., 2013, “Vortex formation on squared and rounded tip,” E Prints [Online]. Available: http://eprints.gla.ac.uk/77819/1/77819.pdf. [Accessed: 15-Mar-2024].
[2]Gongzhang, H., and Axtelius, E., 2020, “Aircraft Winglet Design,” Diva Portal [Online]. Available: http://www.diva-portal.org/smash/get/diva2:1440647/FULLTEXT01.pdf. [Accessed: 15-Mar-2024].
[3]Zilliac, G., 2021, “Flow Behind a NACA 0012 Wingtip.”
Dataset source
[4]Boldmethod, 2022, “This is how winglets work,” Boldmethod [Online]. Available: https://www.boldmethod.com/learn-to-fly/aerodynamics/how-winglets-reduce-drag-and-how-wingtip-vortices-form/. [Accessed: 15-Mar-2024].
[5]Scikit, “SciKit-Learn Documentation,” Scikit Learn [Online]. Available: https://scikit-learn.org/stable/. [Accessed: 15-Mar-2024].
[6]Spinoff, N., 2010, “Winglets save billions of dollars in fuel costs,” NASA Spinoff [Online]. Available: https://spinoff.nasa.gov/Spinoff2010/t_5.html. [Accessed: 15-Mar-2024].
[7]Chow, J., Zilliac, G., and Bradshaw, P., 1997, “Turbulence Measurements in the Near Field of a Wingtip Vortex,” NASA Technical Reports Server [Online]. Available: https://ntrs.nasa.gov/api/citations/19970011348/downloads/19970011348.pdf. [Accessed: Dec-2023].
[8]Zaman, K. B. M. Q., Fagan, A., and Makbandi, M. R., 2017, “An experimental study and database for Tip vortex flow ...,” NASA/TM [Online]. Available: https://ntrs.nasa.gov/api/citations/20180000918/downloads/20180000918.pdf. [Accessed: 13-Mar-2024].
[9]NASA, O., 1979, KC-135A in flight - winglet study, In Flight.
[10]Dale, B. F., 2007, Cessna 182 model-wingtip-vortex.