A General Drag Model for Assemblies of Non-Spherical Particles Created With Artificial Neural Networks (ANN)

Project Award Number DE-FE0031894

PI: Zhi-Gang Feng, Ph.D.
Mechanical Engineering
University of Texas at San Antonio (UTSA)
Presented by: Sergio A. Molina

April 28, 2023

Implementing Multi-Layered Neural Network to Estimate Drag Coefficients in Spherocylinder Particles

Project Award Number DE-FE0031894

PI: Zhi-Gang Feng, Ph.D.
Mechanical Engineering
University of Texas at San Antonio (UTSA)
Presented by: Sergio A. Molina

April 28, 2023

Overview

Introduction

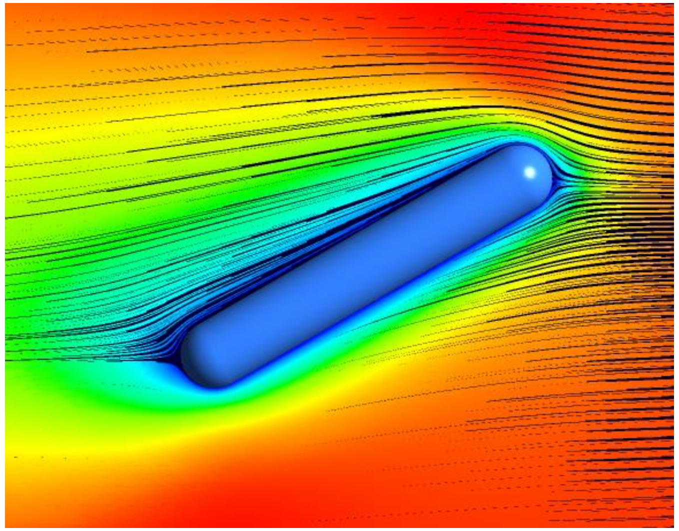
Neural Networks

Methodology

Designing Multi-Layered Neural Network (MLNN)

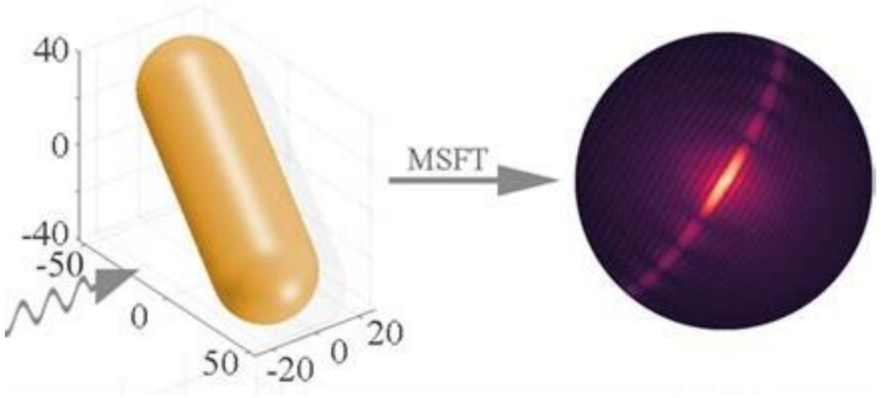
Results and Discussion

Conclusions

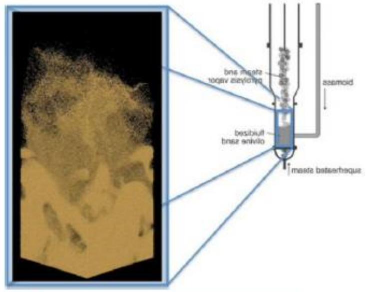


Motivation

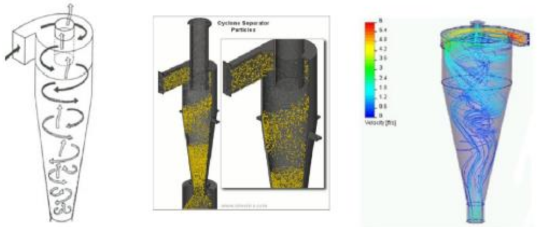
- Research of hydrodynamic forces on non-spherical particles is of upmost importance
- Most particles are non-spherical in nature
- knowledge of forces is critical for determining particle trajectories
- Applications widely range from biological systems to industrial processes
- Examples are separation process, coal combustion, and dispersion of pollutants



Dispersion of Pollutants



Coal Combustion



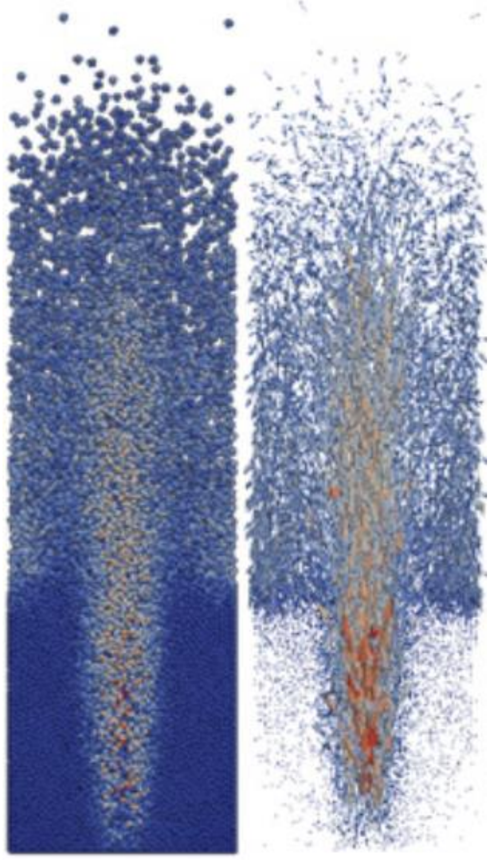
Separation Process (cyclone)

Technical background

- Newton’s 2nd Law of Motion for a particle:
  - $m\frac{d\vec{\boldsymbol{V}}}{dt} = \sum\vec{\boldsymbol{F}}$

- Particle drag determines the movement of particles in particulate flows

- Key to the modeling and understanding of all phenomena associated with the momentum, heat and mass transfer to the surroundings in all particulate processes (e.g., the process in a fluidized bed reactor)



Technical background (continued)

- The studies on the non-spherical particle drag in the literature are very limited

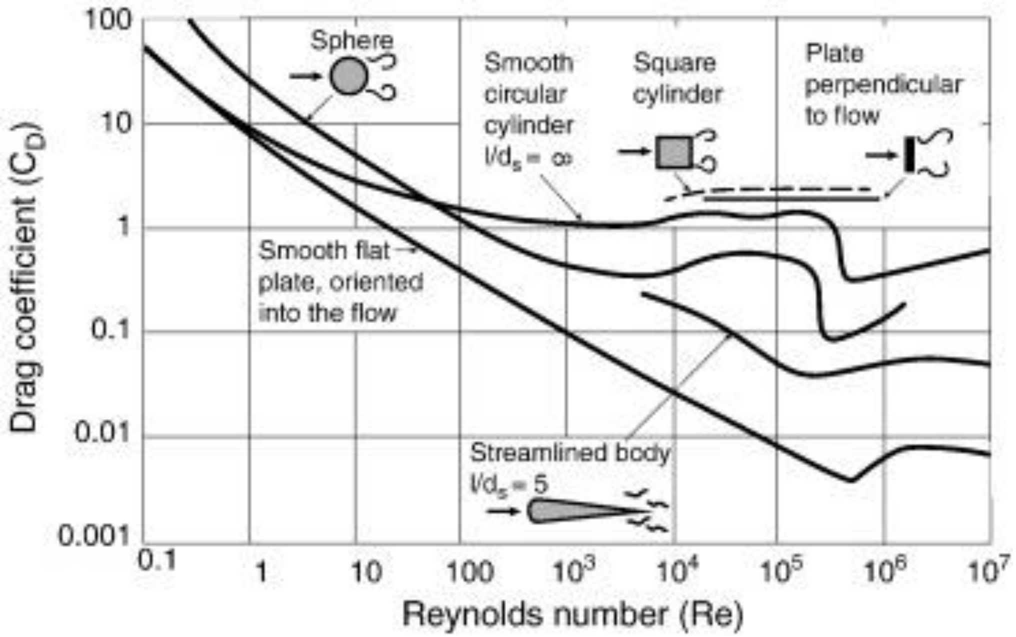
- Most simulation packages currently use the drag models of spherical particles

Gidaspow drag correlation

$$\beta_{gm} = \begin{cases} \dfrac{3}{4} C_D \dfrac{\rho_g \varepsilon_g \varepsilon_m \left| \mathbf{u}_g - \mathbf{u}_m \right|}{d_{pm}} \varepsilon_g^{-2.65} & \varepsilon_g \geq 0.8 \\ \dfrac{150 \varepsilon_s \left(1 - \varepsilon_g\right) \mu_g}{\varepsilon_g d_{pm}^2} + \dfrac{1.75 \rho_g \varepsilon_m \left| \mathbf{u}_g - \mathbf{u}_m \right|}{d_{pm}} & \varepsilon_g < 0.8 \end{cases}$$

$$C_D = \begin{cases} 24/\mathrm{Re}\left(1 + 0.15\,\mathrm{Re}^{0.687}\right) & \mathrm{Re} < 1000 \\ 0.44 & \mathrm{Re} \geq 1000 \end{cases}$$

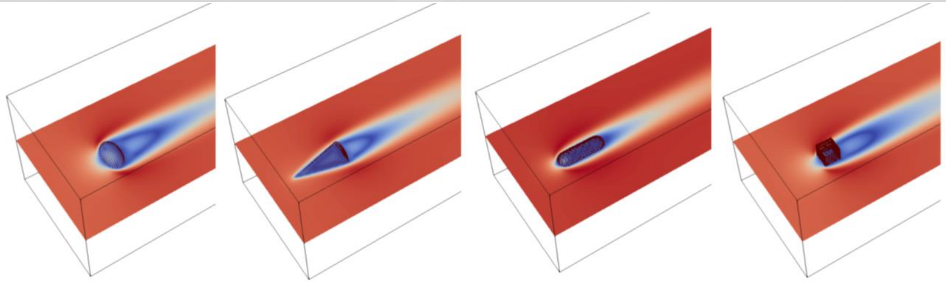
$$\mathrm{Re} = \frac{\rho_g \varepsilon_g \left| \mathbf{u}_g - \mathbf{u}_m \right| d_{pm}}{\mu_g}$$



Introduction

Non-spherical particles being studied

- Ellipsoid, cone, spherocylinder, cube, etc.

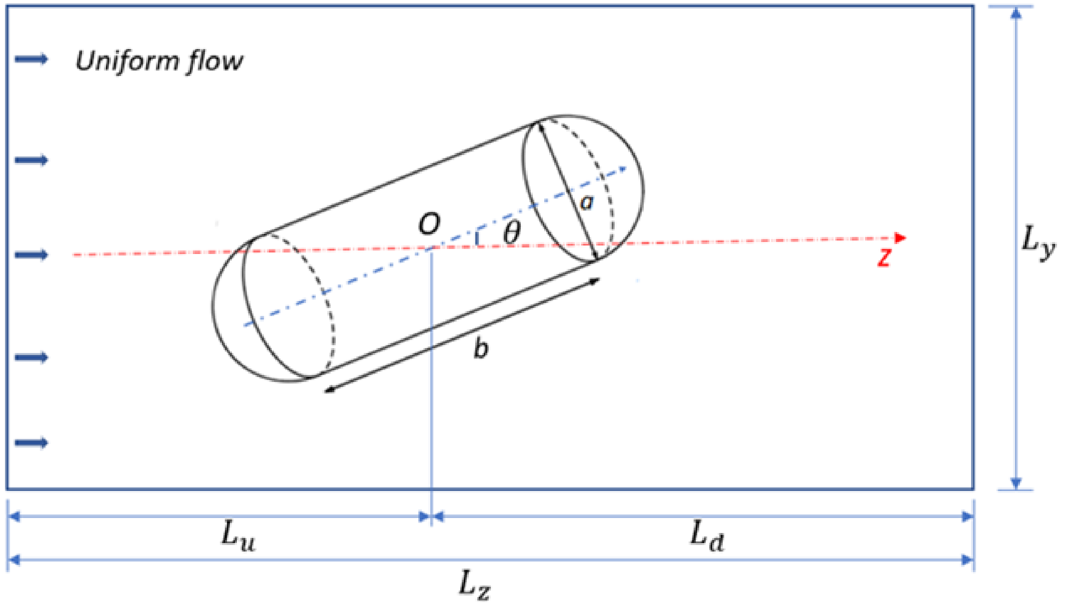


An example: Spherocylinder

- A very common shape, very few studies in literature

Uniform flow over a Spherocylinder

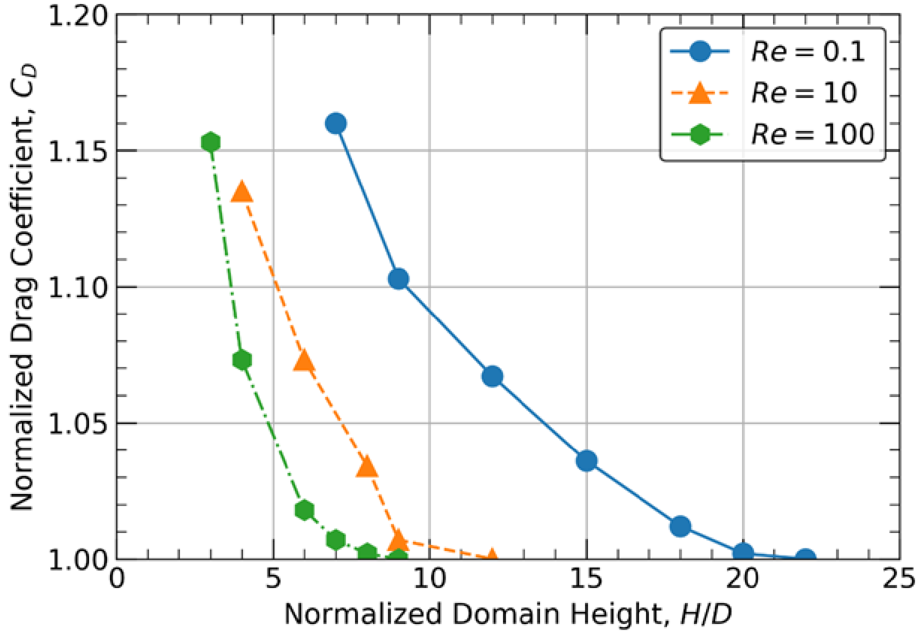
- **Three Dimensionless Parameters - Inputs**
  - Reynolds Number: $Re = \frac{\rho U D_e}{\mu}$
  - Aspect ratio: $\beta = \frac{a+b}{a}$
  - Incident angle: θ



- **Three Coefficients - Outputs**
  - Drag coefficient: C_D
  - Lift coefficient: C_L
  - Torque coefficient: C_T

Numerical Simulations

- Direct Numerical Simulation (DNS) Method



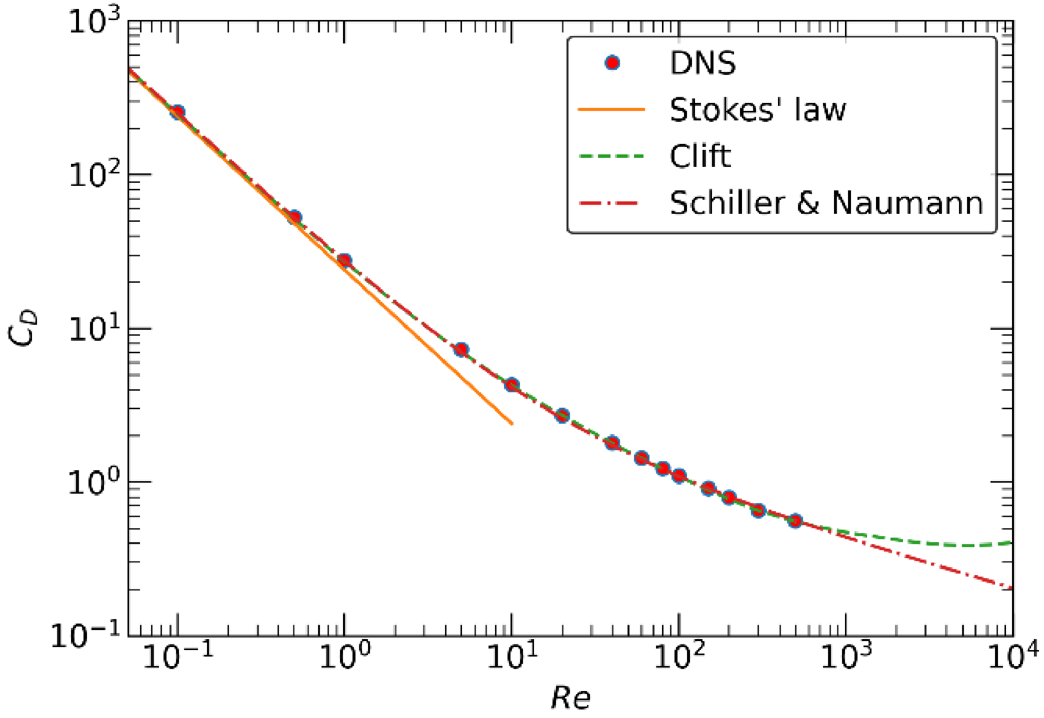
Affect of the domain size to the drag

Re	Grid Resolution (D/h)	Domain Size (L/D)
$0.1 \leq Re \leq 5$	10	$18 \times 18 \times 18$
$5 < Re \leq 200$	20	$9 \times 9 \times 20$
$200 < Re$	30	$8 \times 8 \times 24$

Selection of grid resolution and grid size in the simulations

Validations

Drag Coefficient of a Sphere



Spherocylinder at $\beta = 6$ and $\theta = \pi/3$

Re=10	C_D	C_L	C_T
Zastawny et al.	5.00	0.85	1.2
Ouchene	6.60	1.20	1.50
Present	6.92	1.23	1.57

Re=300	C_D	C_L	C_T
Zastawny et al.	1.25	0.56	0.6
Ouchene	1.49	0.56	0.84
Present	1.40	0.53	0.82

General Coefficient of Drag Coefficient

- A general correlation for the drag coefficient was developed
  - **Aspect Ratio:** $1 \leq \beta \leq 6$,
  - **Orientation Angle:** $0^0 \leq \theta \leq 90^0$
  - **Reynolds Number:** $0.1 \leq Re \leq 300$
  - ***Not able to determine accurate correlations for lift and torque coefficients***

Drag coefficient of a spherocylinder

$$C_{D,\theta} = C_{D,\theta=0^\circ} + (C_{D,\theta=0^\circ,90^\circ} - C_{D,\theta=0^\circ})sin^n\theta$$

$$n = 2 - (0.72 - 0.062\beta)(1 - e^{-(0.012-0.0034\beta+0.00038\beta^2)Re}$$

$$C_{D,\theta=0^\circ} = (a_0 + a_1\beta^{0.5} + a_2\beta + a_3\beta^{1.5} + 0.15\beta^2)C_{Ds}$$

$$C_{D,\theta=90^\circ} = (b_0 + b_1\beta^{0.5} + b_2\beta + b_3\beta^{1.5} + 0.15\beta^2)C_{Ds}$$

$$a_0 = 2.460 + 0.203\sqrt{Re} - 0.00613Re.$$

$$a_1 = -3.461 - 0.324\sqrt{Re} + 0.00912Re.$$

$$a_2 = 2.957 + 0.151\sqrt{Re} - 0.00420Re.$$

$$a_3 = -1.084 - 0.0252\sqrt{Re} + 0.000699Re.$$

$$b_0 = 2.107 + 0.00357\sqrt{Re} - 0.00304Re.$$

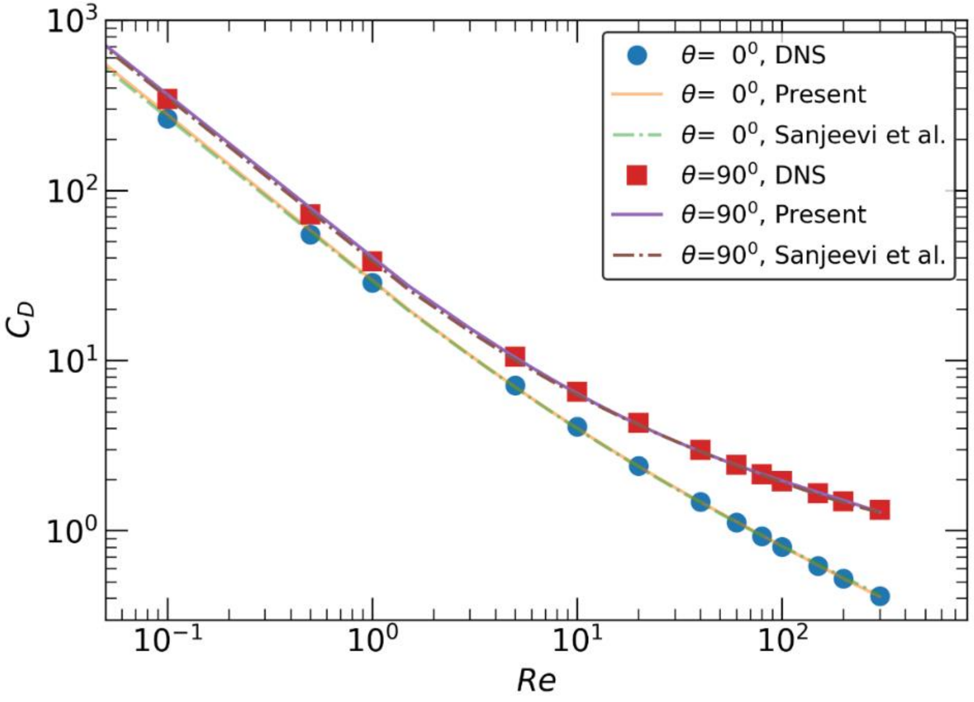
$$b_1 = -3.037 - 0.0487\sqrt{Re} + 0.00575Re.$$

$$b_2 = 2.872 + 0.0605\sqrt{Re} - 0.00388Re.$$

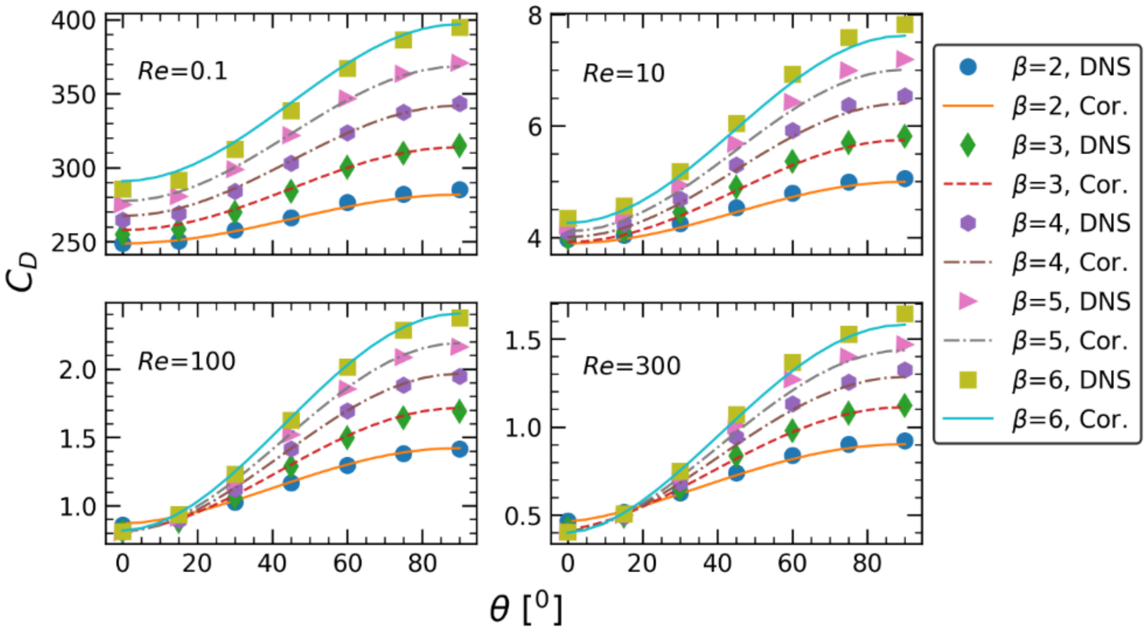
$$b_3 = -1.070 - 0.0106\sqrt{Re} + 0.000641Re.$$

Results of General Drag Correlation*

Comparisons of Correlations
(Drag coefficient at $\theta = 0^o$ and 90^0 for $\beta = 4$)



Drag coefficients for a spherocylinder in terms of (θ, β, Re)



*Feng et al., “A General and Accurate Correlation for the Drag on Spherocylinders”, to be submitted to IJMF, 2023.

Limitations of Correlation Methods

- Mainly limited to two variables, very difficult to extend to three variables
- Very difficult to accurately correlate complex non-linear relationship
- Very sensitive to outliers, leading to skewed inaccurate results
- Overfitting issue, may not perform well when applied to new data
- Cannot account for all variables

Problem Statement

- The process of determining accurate coefficient of drag, lift, and torque estimates for non-spherical particles is often time-consuming requiring specialized skill-sets and expensive software

Objective Statement

- To develop an efficient Multi-Layered Neural Network (MLNN) that accurately predicts the coefficient of drag, lift, and torque for Spherocylinder particles within Reynolds Numbers ranging from 0.1 – 300, Aspect Ratios from 1 – 6, and Incident Angles ranging from 0° – 90°

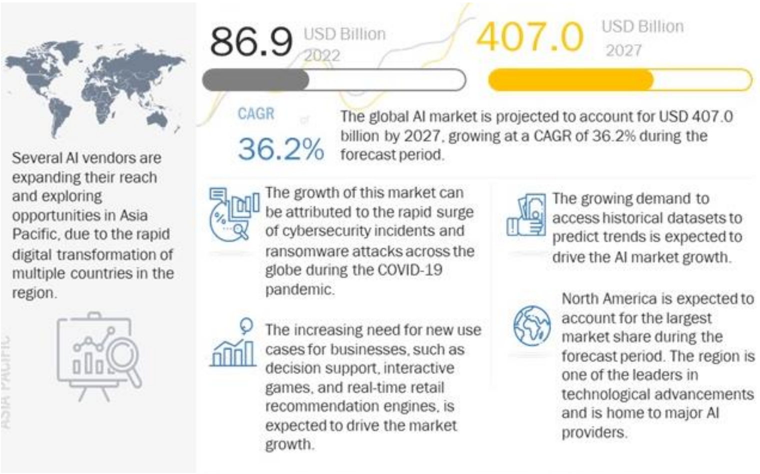
- Specifically, to produce a regression neural network model, that may be loaded into Python, and enable users to input various Reynolds Numbers, Aspect Ratios, and Incident Angles within the constraints

Growth in Artificial Intelligence

- Artificial Intelligence (AI) is a branch of computer science that focuses on simulation of intelligent behavior within computers

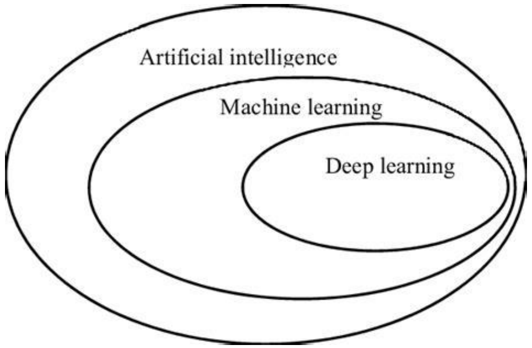
- The field of has experienced a tremendous surge in growth
  - Caused by an increase in computational power and increased data availability
  - Global AI market size is expected to reach $309.6 billion by 2026 with a CAGR of 39.7% from 2021 to 2026

- The amount of data created worldwide is projected to increase from 64.2 zettabytes in 2020 to 181 zettabytes in 2025



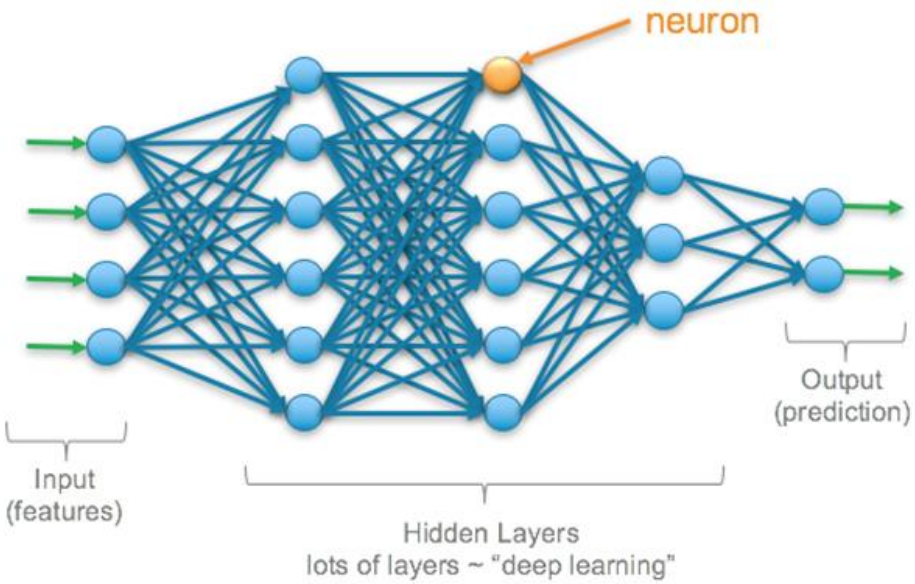
Deep Learning Architecture

- Deep Learning and Machine Learning are subfields of AI



- Structure of a simple neural network:
  - Input layer
  - Intermediate layers (hidden)
  - Output layer

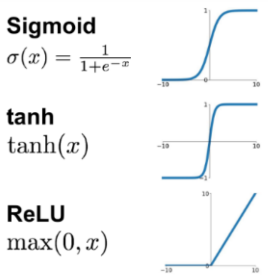
- Neural networks serve either classification or regression applications

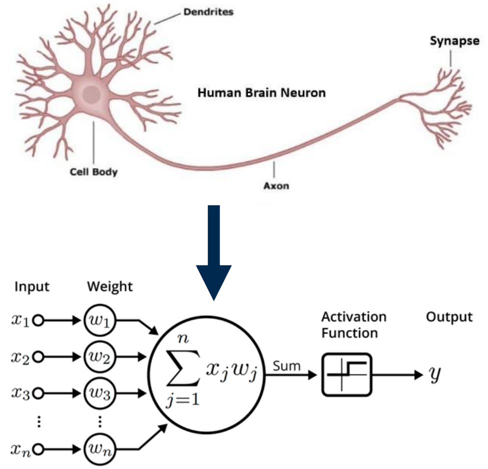


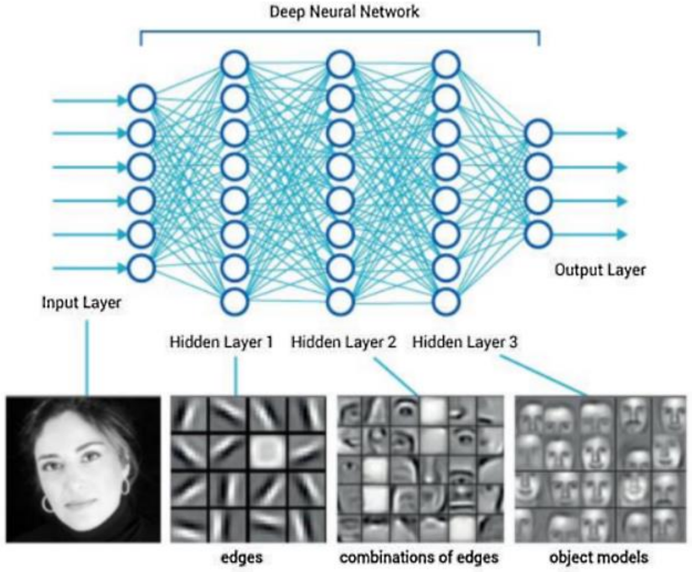
Neural Networks

Artificial Neural Networks

- Neural networks digitally resemble neural activity of the human brain via activation functions
- Goal is to develop a multi-layered network that can be trained and tested to recognize unique patterns
- Examples of common neural networks applications
  - Convolutional Neural Networks (CNN) for facial recognition
  - Multi-Layered Neural Networks (MLNN) for estimating real estate property appraisals via non-linear regression
  - Long-Short Term Memory (LSTM) for future stock price prediction

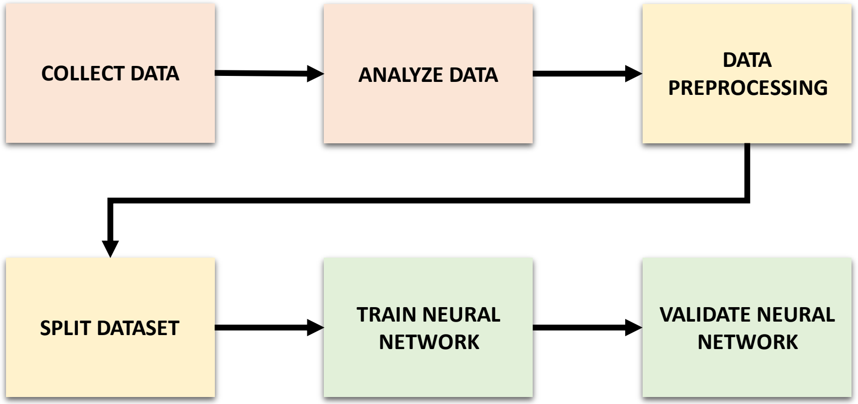






Approach

- Collect data from team members via DNS study
- Leverage statistical tools for analyzing dataset
- Determine if data preprocessing is necessary
- Split the data into training, testing, and validation sets
- Train the proposed neural network
- Validate the proposed neural network



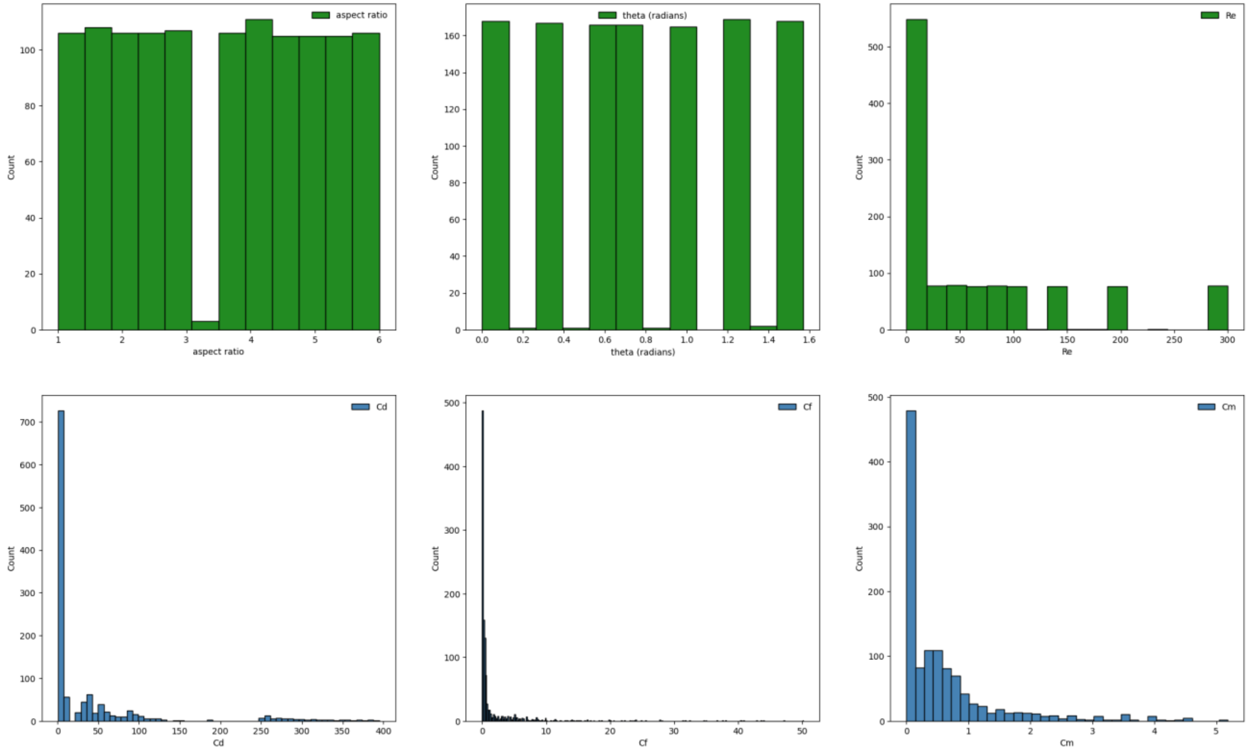
Methodology

Collect Data

- The team provided +1200 data points generated via Direct Numerical Simulations (DNS)
  - The simulation was setup with a spherocylinder particle within a continuous flow

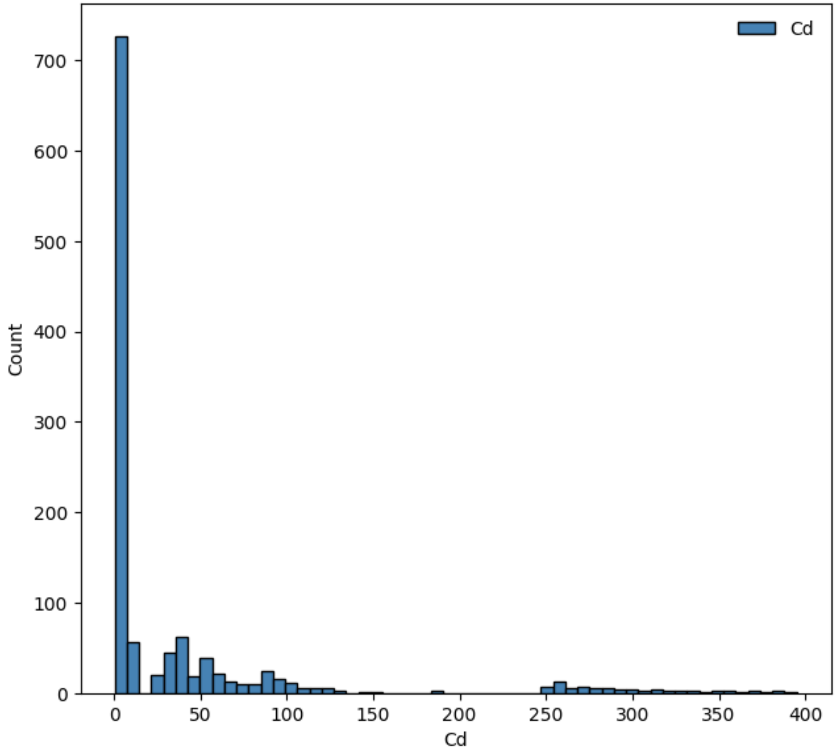
- **Input features (discretized)**
  - **Aspect Ratio, β:** [1.0 – 6.0]
  - **Reynolds Number, Re:** [0.1 – 300]
  - **Angle of Incident, θ :** [0° – 90°]

- **Output features**
  - **Coefficient of drag, lift, and torque**



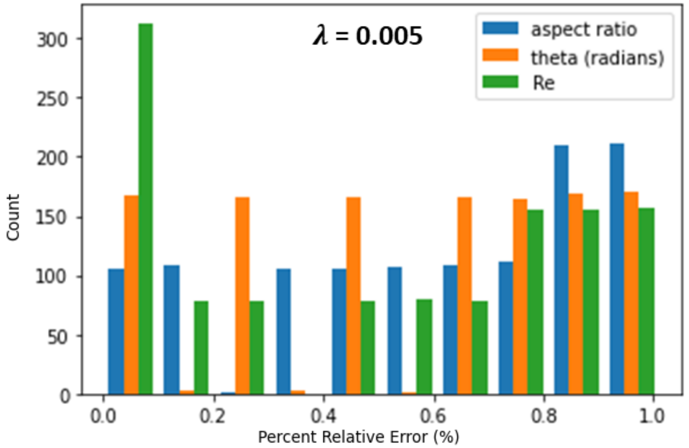
Distributions in Data

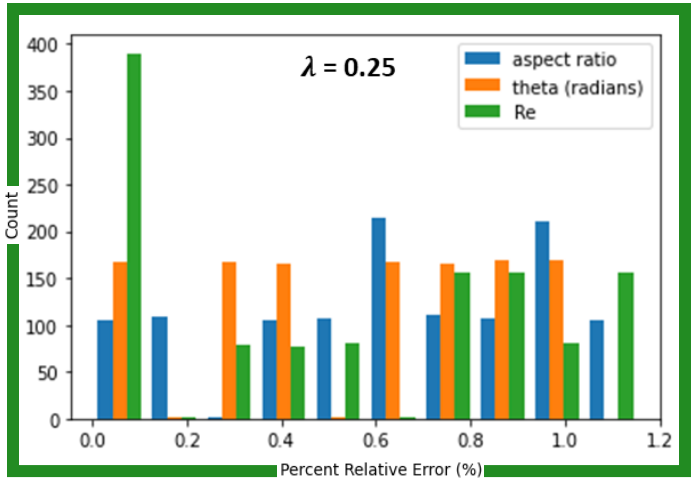
- The output label data is right-skewed, exponentially distributed
- Skewed distributions lead to model learning bias due to over-representation
- The range of values is large which can also result learning bias towards larger values
  - **Coefficient of Drag, C_D:** [0 – 400]
  - **Coefficient of Lift, C_L:** [0 – 60]
  - **Coefficient of Torque, C_T:** [0 – 6]

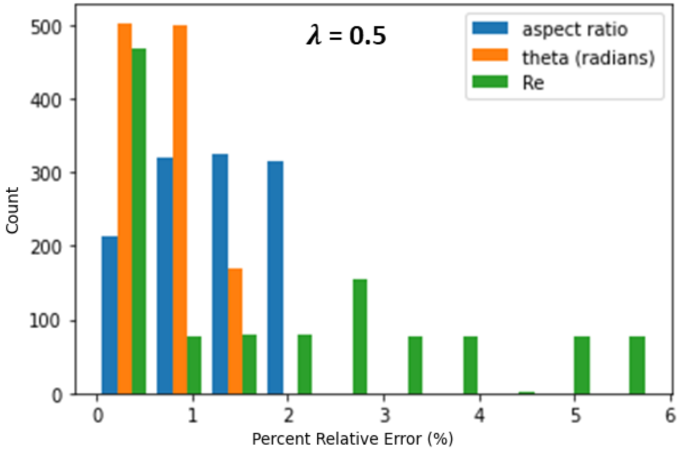


Data Preprocessing

- Data transformation via Box-Cox transformation
- Minimizes cases of overrepresentation
  - $x_{trans} = \begin{cases} \dfrac{x^{\lambda}}{\lambda}, & \lambda \neq 0 \\ \ln(x+1), & \lambda = 0 \end{cases}$
  - $y_{trans} = \begin{cases} \dfrac{y^{\lambda}}{\lambda}, & \lambda \neq 0 \\ \ln(y+1), & \lambda = 0 \end{cases}$
  - **λ set to 0.25**





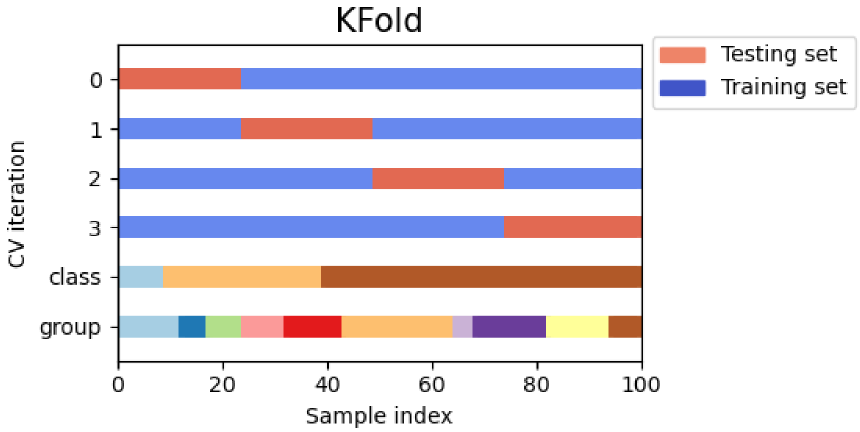


Split Dataset

- Training a neural network requires splitting the dataset
  - Common split ratios are randomly shuffled to 80%, 20% for training and testing, respectively

- To avoid underfitting or overfitting, attention that sufficient data representation is present within the training and test splits

- K-fold Cross Validation leveraged to avoid issues with overfitting
  - Common values for k range from 3 to 10
  - K set to 5 provided 75% of data for training and 25% for testing
  - Select the best performing k-fold



Designing Multi-Layered Neural Network (MLNN)

Establishing Baseline Single Layer Neural Network (SLNN)

- Nodes – 5
- Epoch – 1000
- Cost Function –Mean Squared Logarithmic Error
- Activation Function (hidden layers) – ReLU
- Batch Size – 255
- K-Folds – 5

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 3)                 12

 dense_1 (Dense)             (None, 5)                 20

 dense_2 (Dense)             (None, 3)                 18

=================================================================
Total params: 50
Trainable params: 50
Non-trainable params: 0
_________________________________________________________________
```

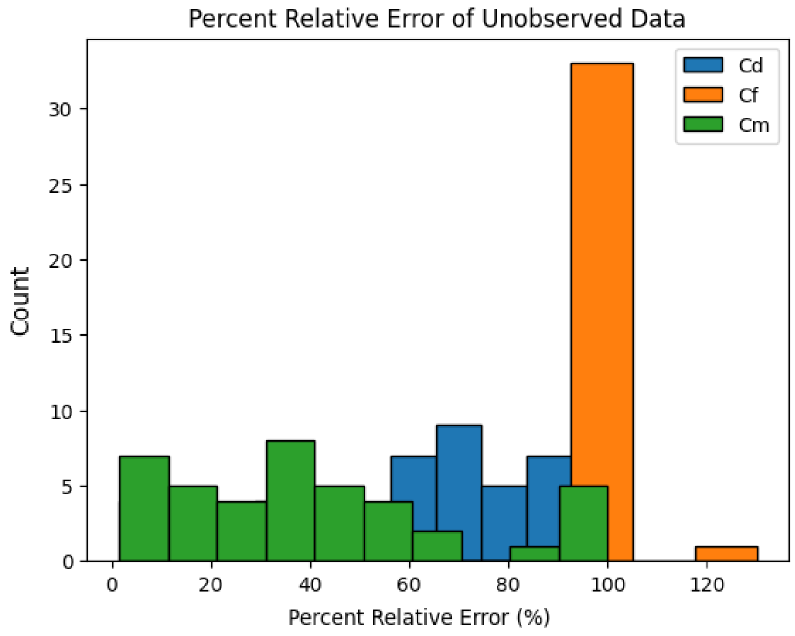


Table 1. Baseline SLNN Performance Results on Observed Data

Learning Rate	Batch Size	Layers	Train Time	RMSE	R^2 Cd	R^2 Cf	R^2 Cm
0.0001	225	1	5m 15s	26.55	0.63	0.59	0.61

Table 2. Baseline SLNN Performance Results on Unobserved Data

Learning Rate	Batch Size	Layers	Train Time	RMSE U.D.	R^2 U.D. Cd	R^2 U.D. Cf	R^2 U.D. Cm
0.0001	225	1	5m 15s	6.51	0.92	0.82	0.54

Designing Multi-Layered Neural Network (MLNN)

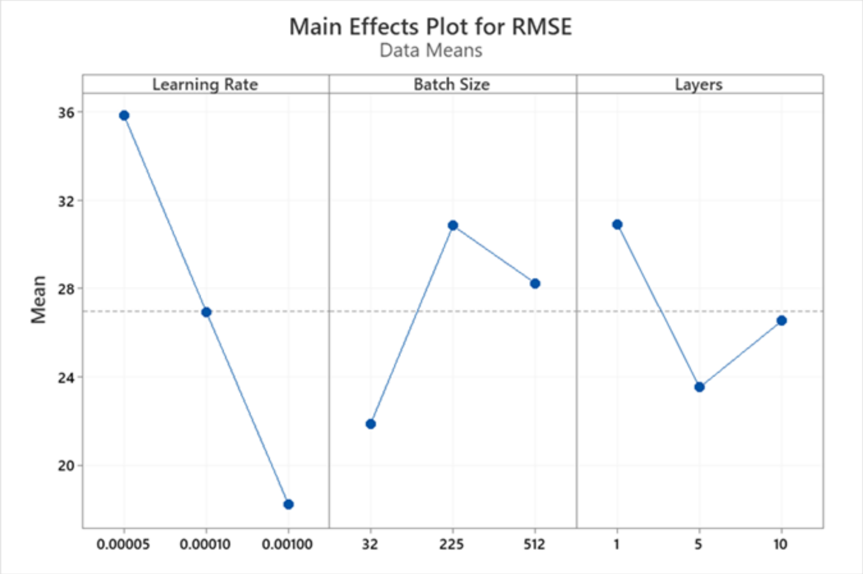
Design of Experiments (DOE)

- Scope of DOE
  - Determine the critical hyper parameter “knobs”
  - 3 factors with 3 levels

- Learning rate is among the most influential hyper parameters for the model’s RMSE

- Lower Batch sizes tend to add regularization during training via introduction of variation
  - May prevent the model from overfitting to the training data during the optimization process

Hyper Parameters	Values
Learning Rate	0.00005
	0.0001
	0.001
Layers	1
	5
	10
Batch Size	32
	225
	512



Designing Multi-Layered Neural Network (MLNN)

Best MLNN from DOE

- Nodes – 5
- Learning Rate – 0.001
- Batch Size – 32
- Epoch – 1000
- Cost Function –Mean Squared Logarithmic Error
- Activation Function (hidden layers) – ReLU

```
Model: "sequential_60"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_420 (Dense)           (None, 3)                 12

 dense_421 (Dense)           (None, 50)                200

 dense_422 (Dense)           (None, 150)               7650

 dense_423 (Dense)           (None, 350)               52850

 dense_424 (Dense)           (None, 500)               175500

 dense_425 (Dense)           (None, 350)               175350

 dense_426 (Dense)           (None, 50)                17550

 dense_427 (Dense)           (None, 3)                 153

=================================================================
Total params: 429,265
Trainable params: 429,265
Non-trainable params: 0
```

Table 5. MLNN Best Performance Results on Observed Data

Learning Rate	Batch Size	Layers	Train Time	RMSE	R^2 Cd	R^2 Cf	R^2 Cm
0.001	32	5	12m 57s	5.5	0.98	0.98	0.98

Table 6. MLNN Best Performance Results on Unobserved Data

Learning Rate	Batch Size	Layers	RMSE UD	R^2 UD Cd	R^2 UD Cf	R^2 UD Cm
0.001	32	5	2.1	0.99	0.88	0.94

Designing Multi-Layered Neural Network (MLNN)

Final MLNN

- Nodes – 75
- Learning Rate – 0.001
- Batch Size – 32
- Epoch – 1000
- Cost Function –Mean Squared Logarithmic Error
- Activation Function (hidden layers) – ReLU

```
Model: "sequential_375"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_2625 (Dense)          (None, 3)                 12

 dense_2626 (Dense)          (None, 75)                300

 dense_2627 (Dense)          (None, 75)                5700

 dense_2628 (Dense)          (None, 75)                5700

 dense_2629 (Dense)          (None, 75)                5700

 dense_2630 (Dense)          (None, 75)                5700

 dense_2631 (Dense)          (None, 3)                 228

=================================================================
Total params: 23,340
Trainable params: 23,340
Non-trainable params: 0
_________________________________________________________________
```

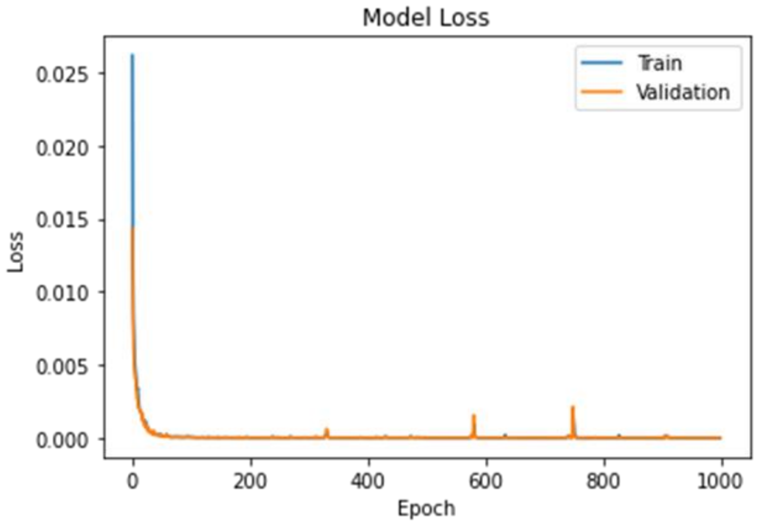


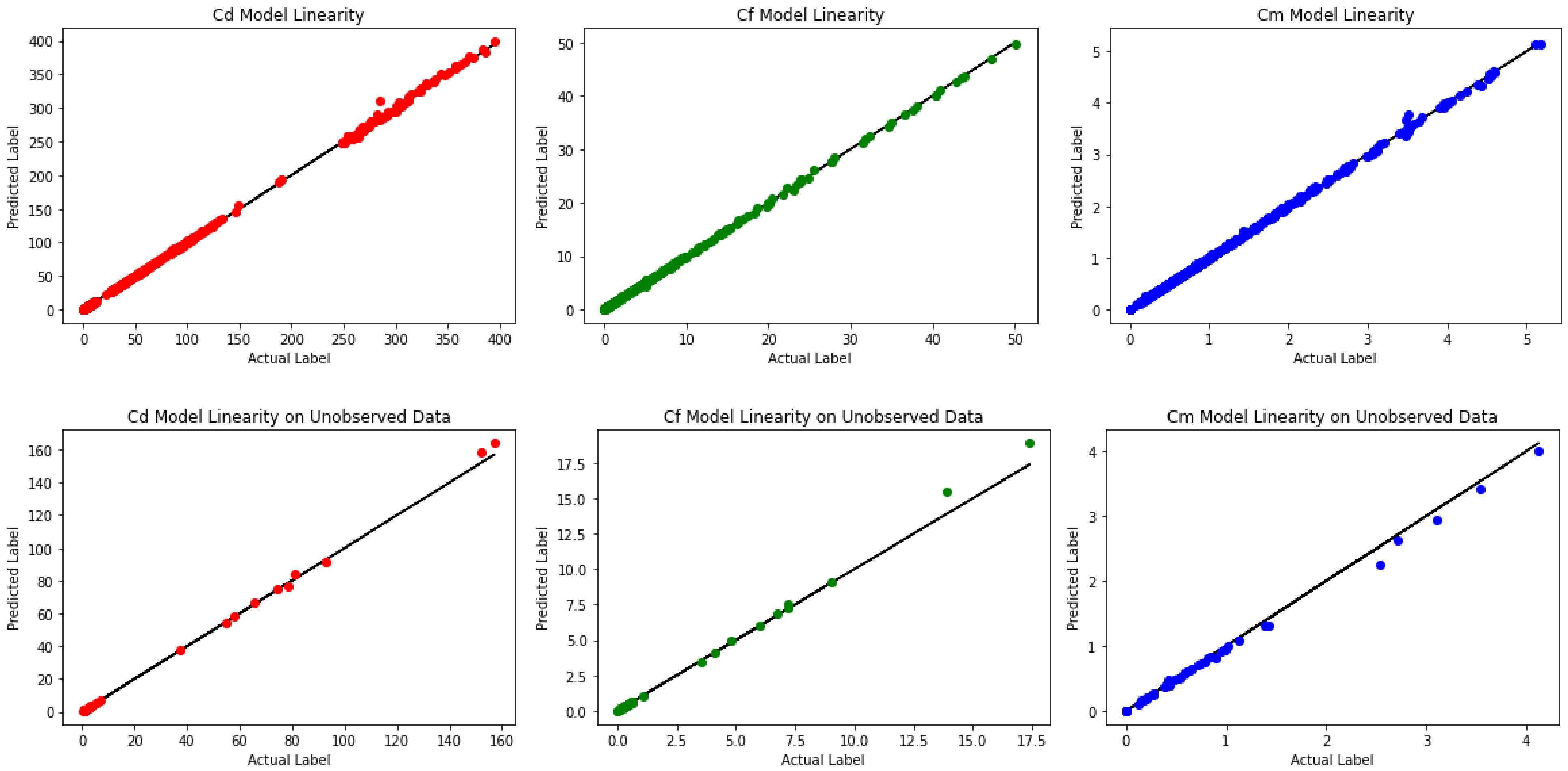
Table 7. MLNN Best Performance Results on Observed Data

Learning Rate	Batch Size	Layers	Train Time	RMSE	R^2 Cd	R^2 Cf	R^2 Cm
0.0001	225	3	15m 15s	0.69	0.999	0.999	0.999

Table 8. MLNN Best Performance Results on Unobserved Data

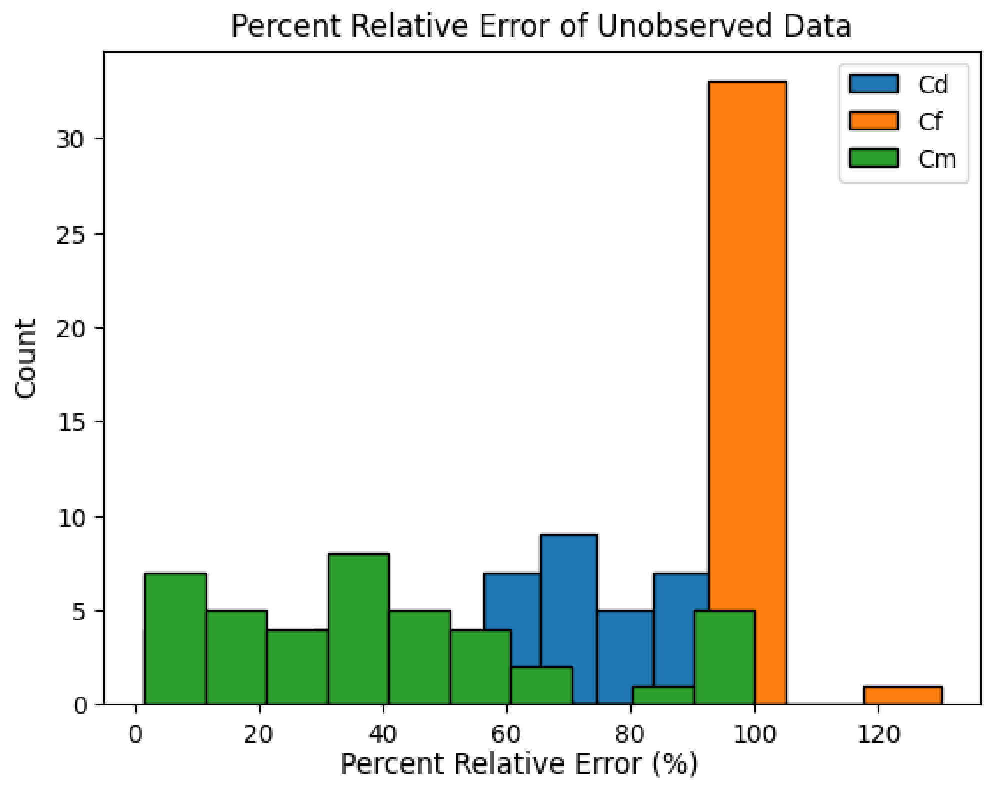
Learning Rate	Batch Size	Layers	Train Time	RMSE U.D.	R^2 U.D. Cd	R^2 U.D. Cf	R^2 U.D. Cm
0.0001	225	3	15m 15s	0.999	0.999	0.999	0.999

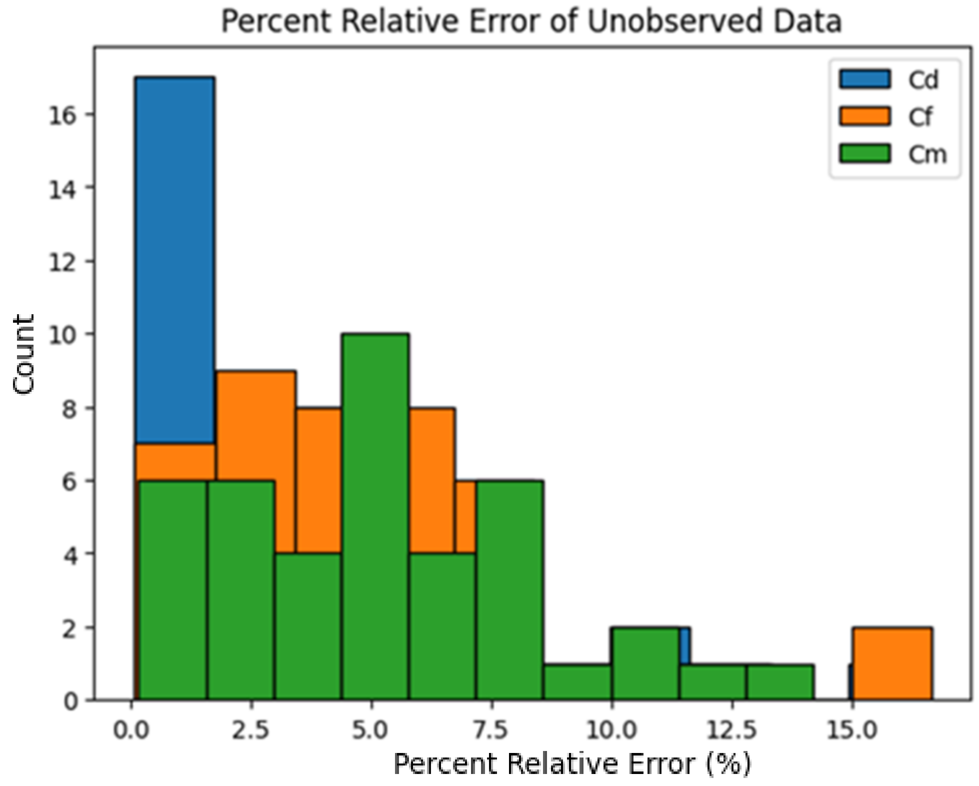
Final MLNN Model Performance



Baseline SLNN and Final MLNN Comparisons

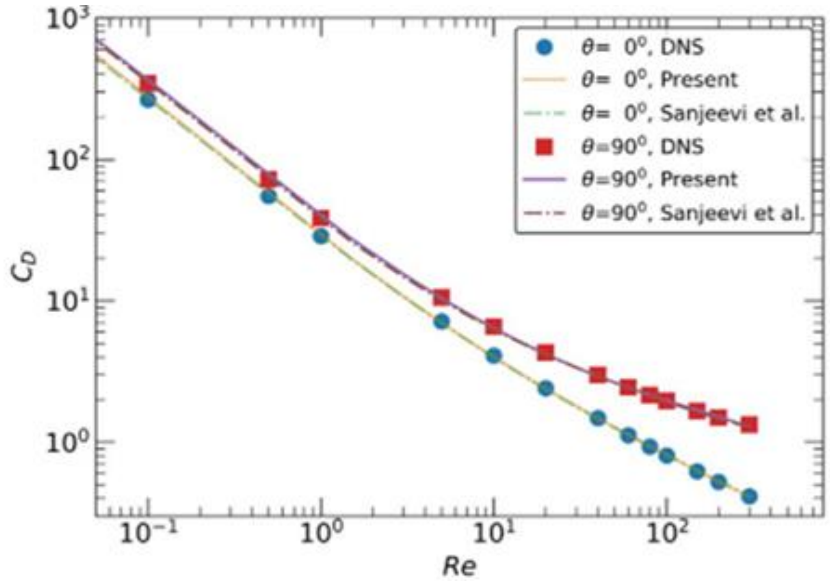
- Tuning hyper parameters via DOE significantly reduced the ***percent relative error***
- Percent relative error significantly reduced from 120% to 15%

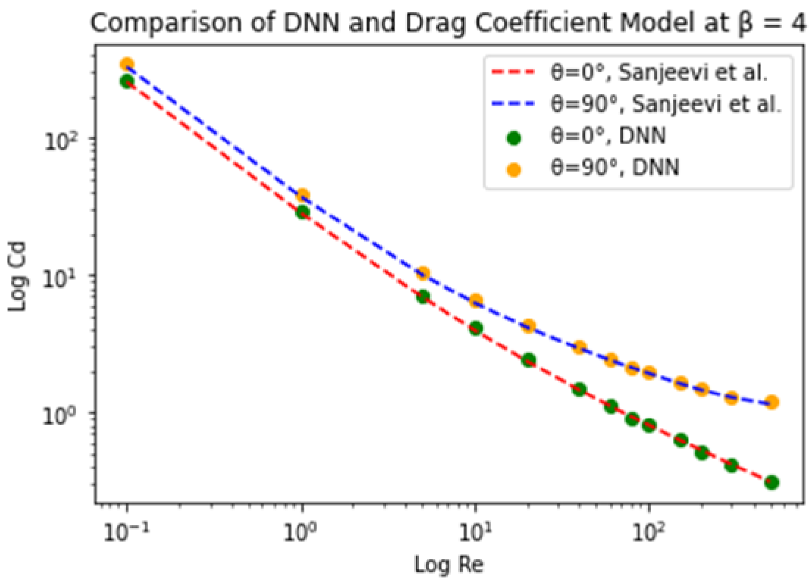




MLNN Comparison with Sanjeevi et al. Drag Correlation

- Aspect Ratio of 4.0 is referenced as in literature
- MLNN achieved a correlation coefficient of 99.6%
- MLNN coefficient of drag estimates fit the correlation well

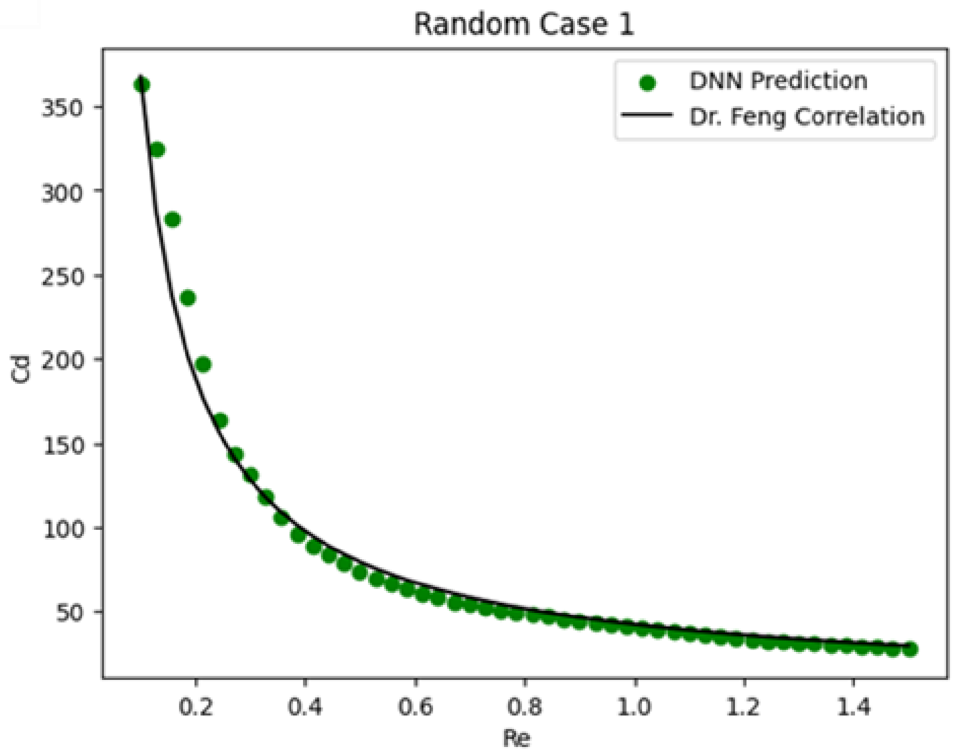




MLNN Comparisons to Mathematical Correlations

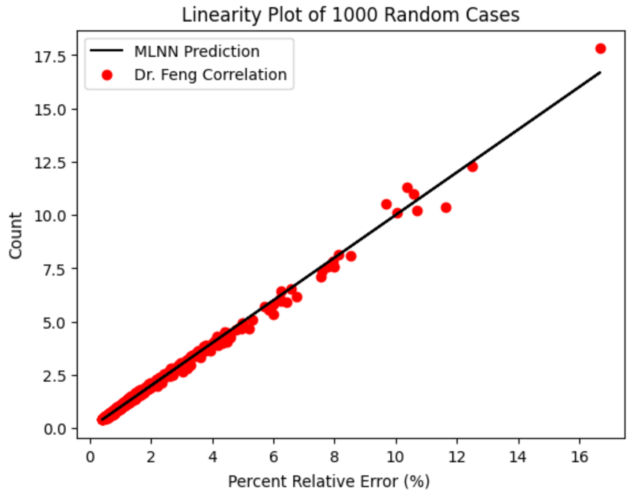
MLNN Comparison with General Drag Correlation

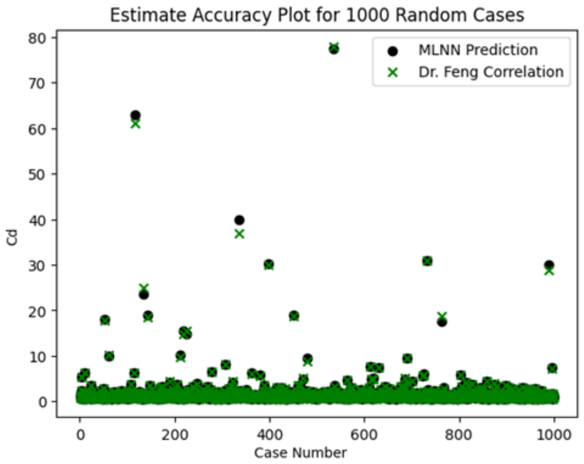
- Case 1 values selected at random with aspect ratio of 1.6, incident angle of 34.4°
- The values for Reynolds Number vary from 0.1 – 1.6
- MLNN achieved a Case 1 **correlation coefficient** of **99.3%**
- MLNN coefficient of drag estimates fit the correlation well

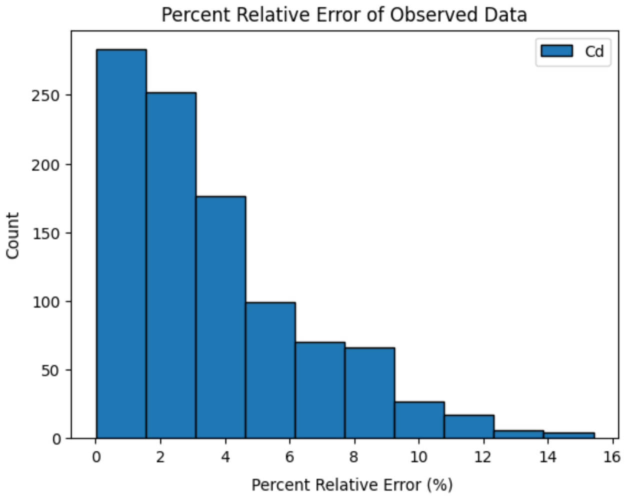


MLNN Comparison with General Drag Correlation

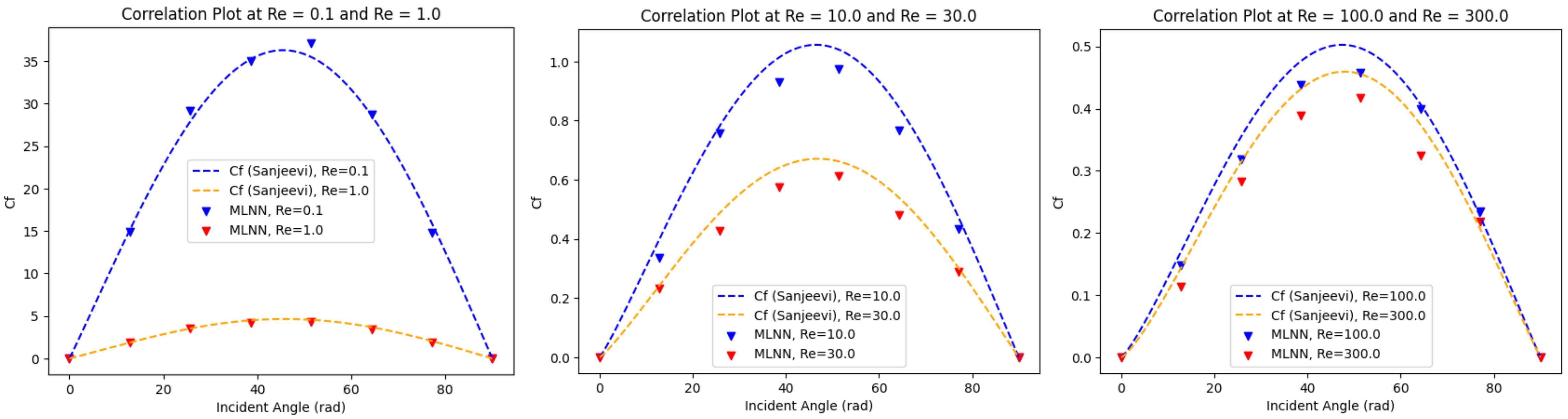
- 1000 random cases for aspect ratio, incident angle, and Reynolds number
- MLNN achieved a **correlation coefficient** of **99.9%**
- MLNN coefficient of drag estimates fit the correlation extremely well







MLNN Comparison with Sanjeevi et al. Lift Correlation



Conclusions

Summary

- A wide-range of applications from describing separation processes to cellular biology benefit from this research

- A general correlation for the drag coefficient for a Spherocylinder was developed using traditional numerical approaches

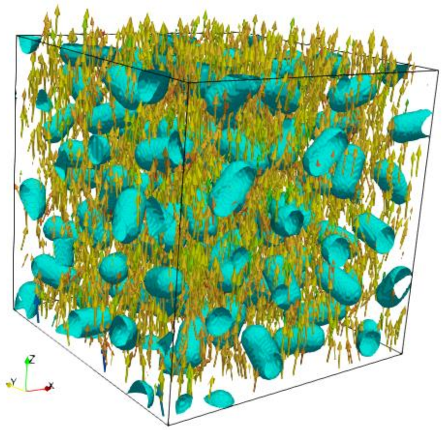
- A MLNN was developed for estimating the coefficients of drag, lift, and torque of Spherocylinders using modern neural network regression methods

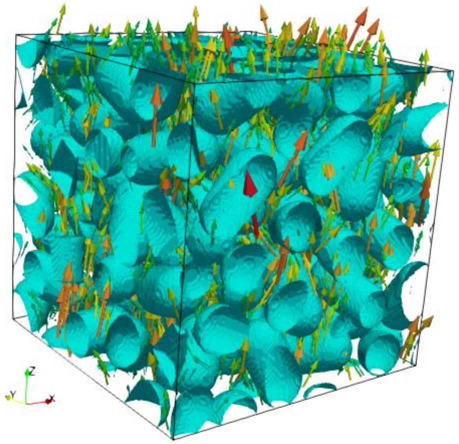
- Future improvements to the model's performance, shape selection, or input feature constraints are strongly encouraged

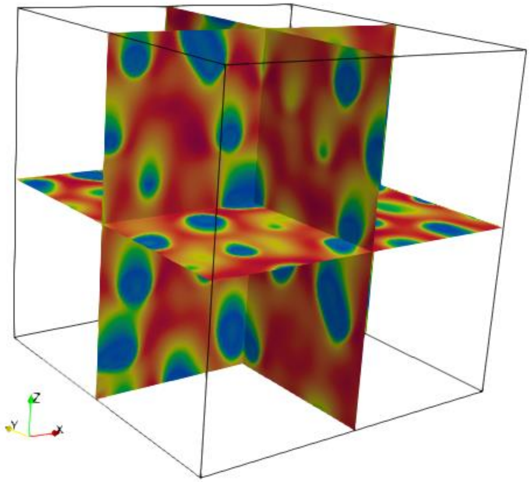
Current and Future Work

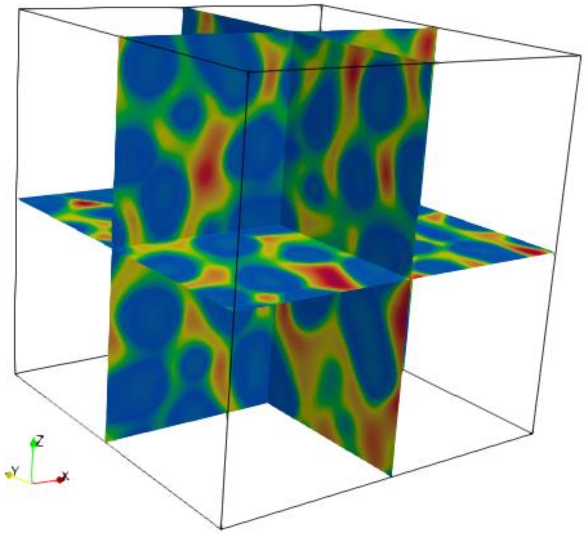
- From a single Spherocylinder particle to an assembly of particles
  - Solid fractions
  - Particle configurations, etc.

- Develop neural network for other non-spherical particles
  - Ellipsoid, short-cylinders, etc.









Acknowledgments

Contributors to this project

- **Professor:** Zhi-Gang Feng, Ph.D. Zhigang.Feng@utsa.edu

- **Graduate students:** Joshua Corner, Daniel Hinojosa, Jack Smith, and Sergio A. Molina Sergio.Molina@utsa.edu

- **Undergraduate student:** Andres Leon Islas, Mahnoor Bokhari, James Moseley, and Joshua Beltran

- **UTSA Multiphase Flow Simulation Laboratory**

UTSA Engineering
MULTIPHASE FLOW SIMULATION LABORATORY

Acknowledgements

- Special thanks to the U.S. DOE for sponsoring this research!

U.S. DEPARTMENT OF ENERGY
Office of Fossil Energy

Thank you for your time and attention!