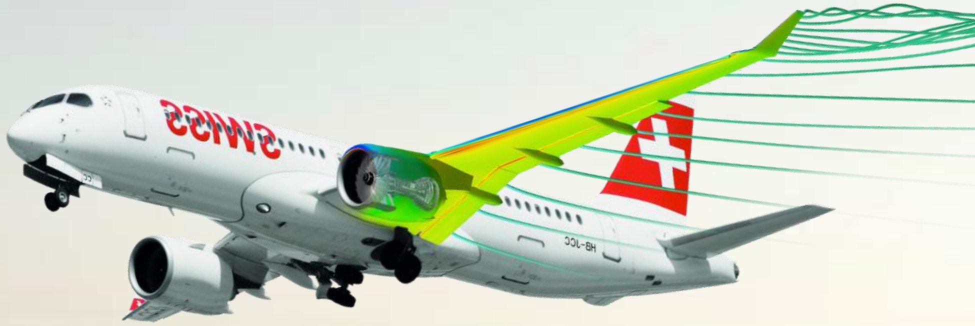
ZHAW - Zurich University of Applied Sciences

EASA MODEL-SI Project
Machine Learning models

Andrea Vaiuso

June 2024, Winterthur, CH



Bayesian Neural Network

Skiron Digital Twin

Simulation

Input

Output

Flow Conditions → Digital Model → Physical Response

Skiron Sim. Flow Condition: Input

Input data label	Data Type	Description
aoa	Float32	Angle Of Attack
aos	Float32	Angle Of Sideslip
u_inf	Float32	Freestream Velocity
PP	Int32	Pusher Propeller RPM value
FR	Int32	Front Right Propeller RPM value
FL	Int32	Front Left Propeller RPM value
RR	Int32	Rear Right Propeller RPM value
RL	Int32	Rear Left Propeller RPM value

Skiron Sim. Physical Response: Output

School of Engineering
ZAV Centre for Aviation

Output data label	Data Type	Description
T_PP	Float32	Thrust value of Pusher Propeller
Q_PP	Float32	Torque value of Pusher Propeller
T_FR	Float32	Thrust value of Front Right Propeller
Q_FR	Float32	Torque value of Front Right Propeller
T_FL	Float32	Thrust value of Front Left Propeller
Q_FL	Float32	Torque value of Front Left Propeller
T_RR	Float32	Thrust value of Rear Right Propeller
Q_RR	Float32	Torque value of Rear Right Propeller
T_RL	Float32	Thrust value of Rear Left Propeller
Q_RL	Float32	Torque value of Rear Left Propeller

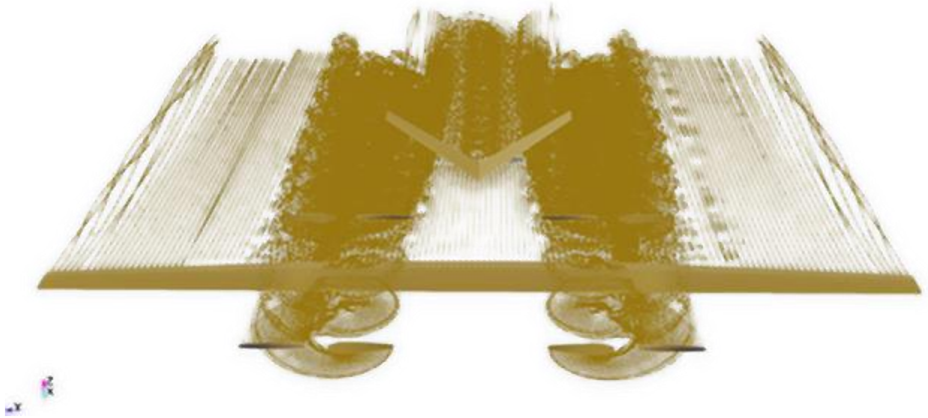
Artificial Intelligence Approach

Machine Learning (ML) models are a special type of data-driven model that can predict complex phenomena after a **training phase** on samples.

The prediction stage is **significantly faster** than the training phase and does not require substantial computational resources. These models are suitable for **real-time simulation** systems, a task that is not feasible for complex simulation models such as CFD.

We need good samples to train a ML model

Samples



High Fidelity CFD

One week per sample
High accuracy

Medium Fidelity Simulations

A few hour per sample
Good accuracy in certain flow conditions

Low Fidelity Methods

A few seconds per sample
Low accuracy

AI-Based Multi Fidelity Model

Data Fusion techniques aim to build a model by using data of varying fidelities.

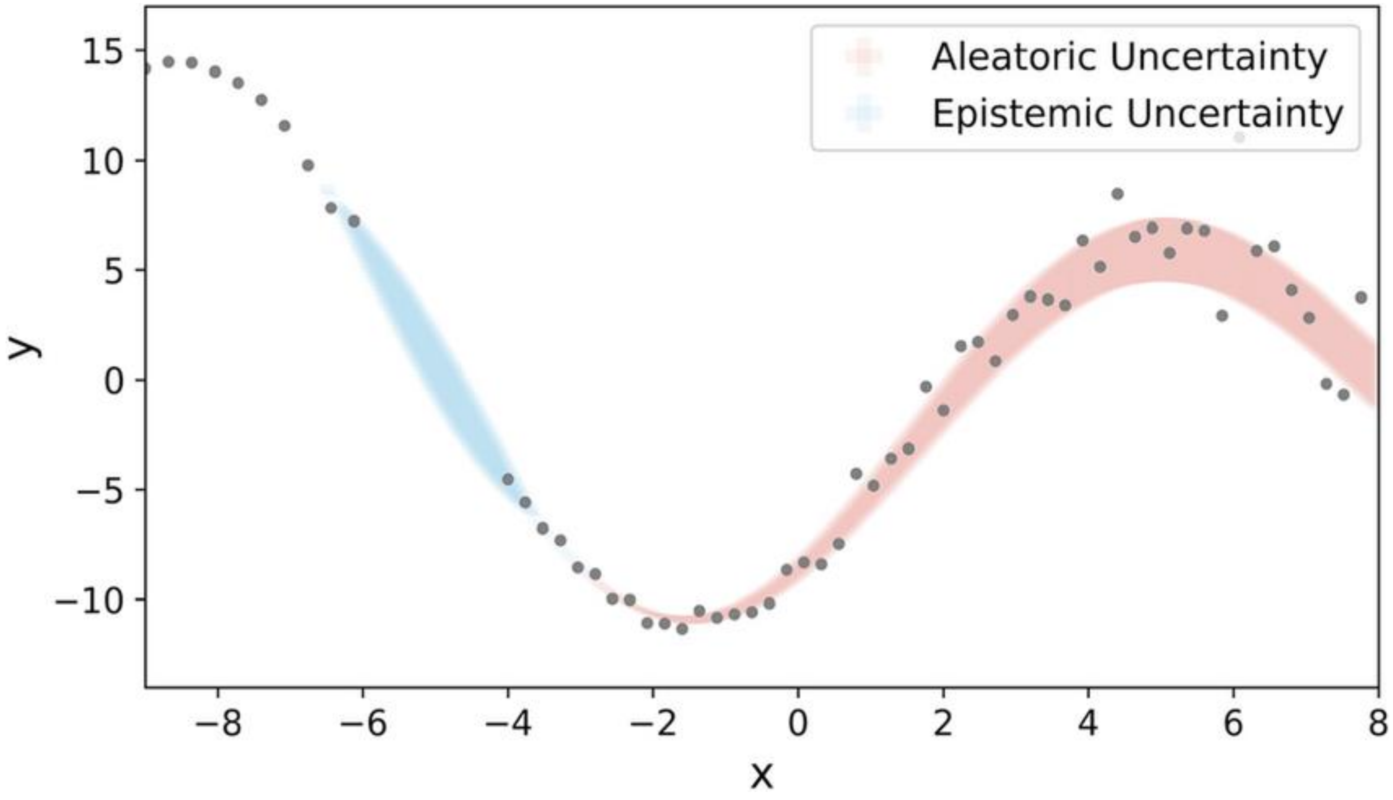
This technique can be used to **improve the accuracy** of the models in **predicting physical behaviors**, including aerodynamic loads, aeroelasticity, flight dynamics, etc.

Data Fusion **reduces the need for many high-fidelity samples**. This approach trains the model on many low-fidelity samples, then fine-tunes it with a few high-fidelity ones.

Uncertainty Quantification

Uncertainty quantification is crucial for understanding the reliability of the model's predictions.

- **Aleatoric uncertainty** is referred to uncertainty of the data
- **Epistemic uncertainty** is referred to uncertainty of the model
- **Predictive uncertainty** is the sum of aleatoric and epistemic uncertainty



A physical simulation model should also provide information about uncertainty of the generated data

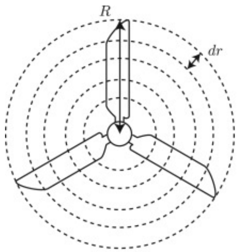
Dataset

We run some high- and low-fidelity simulation to generate the dataset for training the ML-based model.

In the end, the dataset was composed by several samples with different fidelities, computed from different flow conditions.

Input flow conditions were chosen in a way that could represent better typical Skiron-X maneuvers and behaviors.

Initial dataset:



2006 Low-Fidelity Samples
Blade Element Method

64 High-Fidelity Samples
DUST simulations

First Approach: Co-Kriging

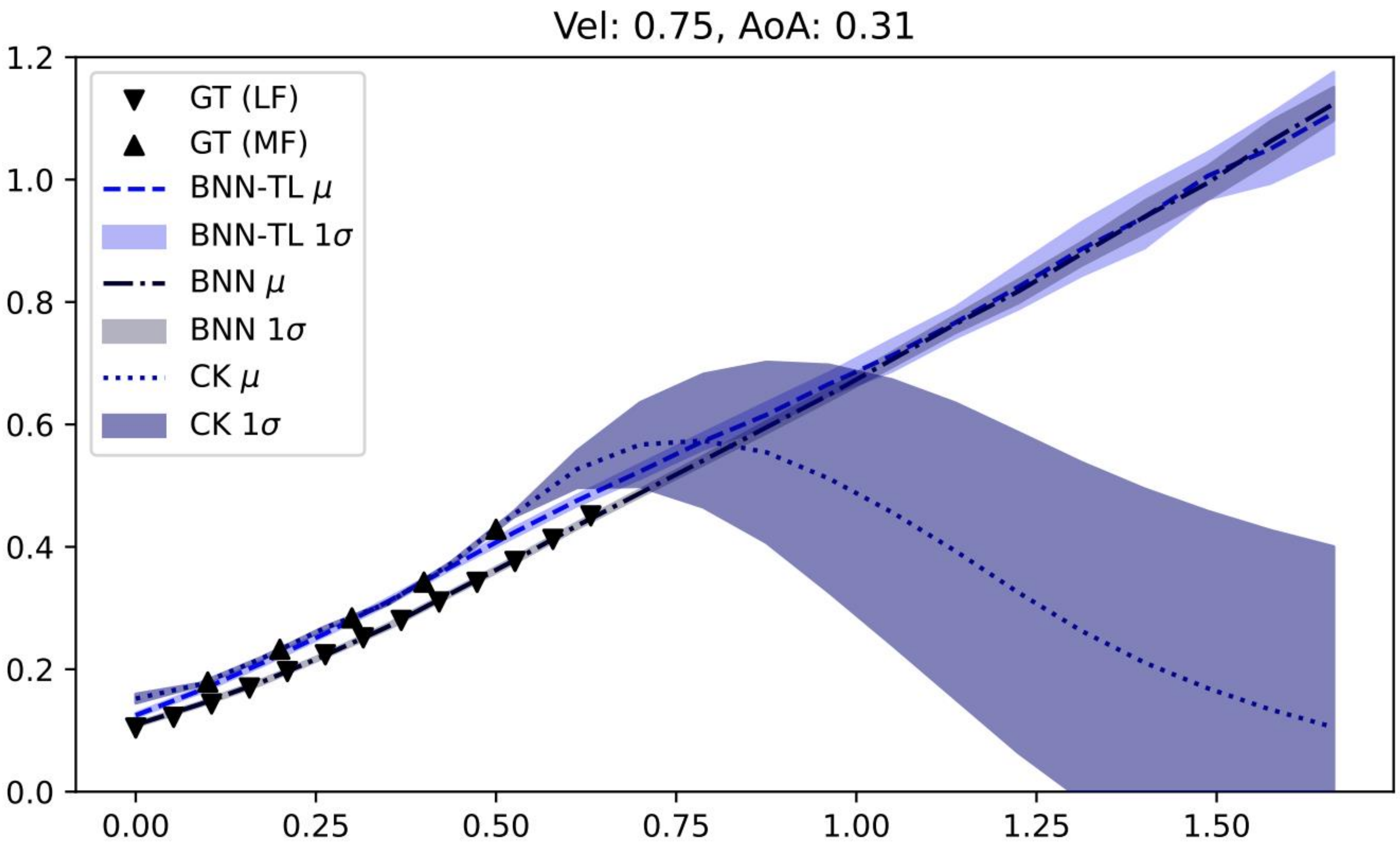
Co-Kriging model is a state-of-the-art data-driven method for Data Fusion and uncertainty quantification.

Co-Kriging can be used to both predicting physical behaviors based on multi-fidelity data and generating **predictive uncertainty**.

This data-driven method is based on the **Gaussian Process** (GP), a probabilistic regression method that is used to make predictions on an augmented dataset, where a low fidelity model creates alternative values for higher fidelity samples. Finally, GP is used to make the final prediction.

First Approach: Co-Kriging

Compared to more complex machine learning methods based on neural networks, Co-Kriging has shown **poor generalization capabilities** outside the training input space and in highly nonlinear regions of the input.

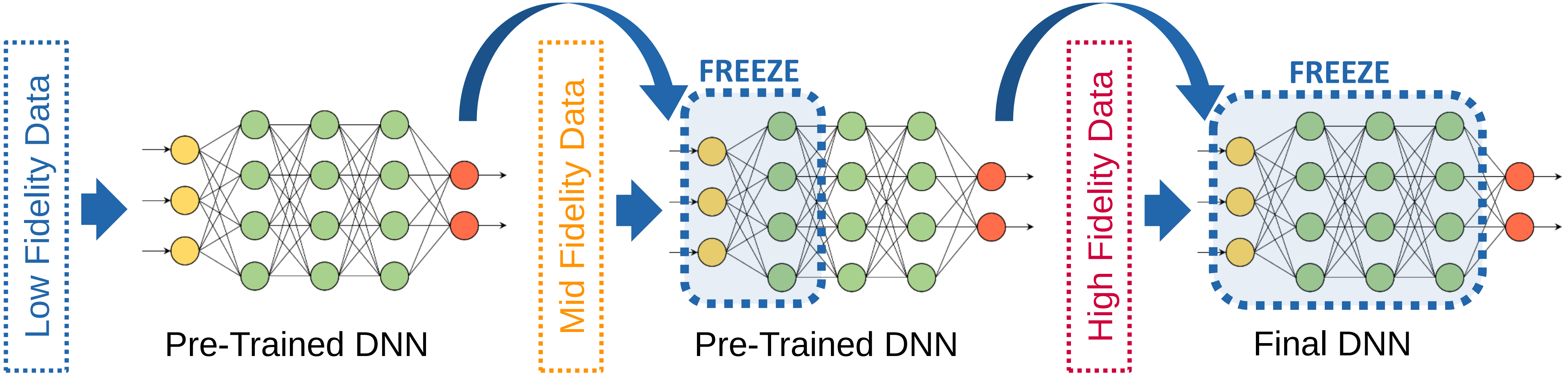


Second Approach: Neural Network with TL

School of Engineering

ZAV Centre for Aviation

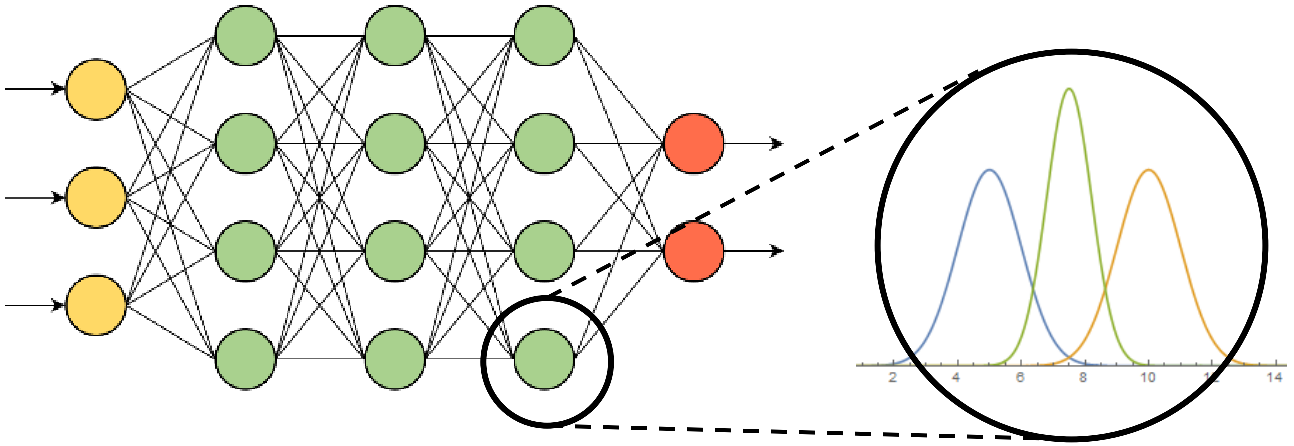
We created an optimized **Dense Neural Network** that uses **Transfer Learning (TL)** for Data Fusion.



We get improved accuracy for output predictions with feasible training time and real-time predictions. No uncertainty quantification available.

Third Approach: BNN with TL

We decided to fuse the **Bayesian Neural Network (BNN)** approach **with TL** to get both Data Fusion and uncertainty quantification.



BNN generates a **nondeterministic model**. Using Monte Carlo sampling, it’s possible to calculate the mean and the standard deviation of the model’s predictions.

Third Approach: BNN with TL

School of Engineering
ZAV Centre for Aviation

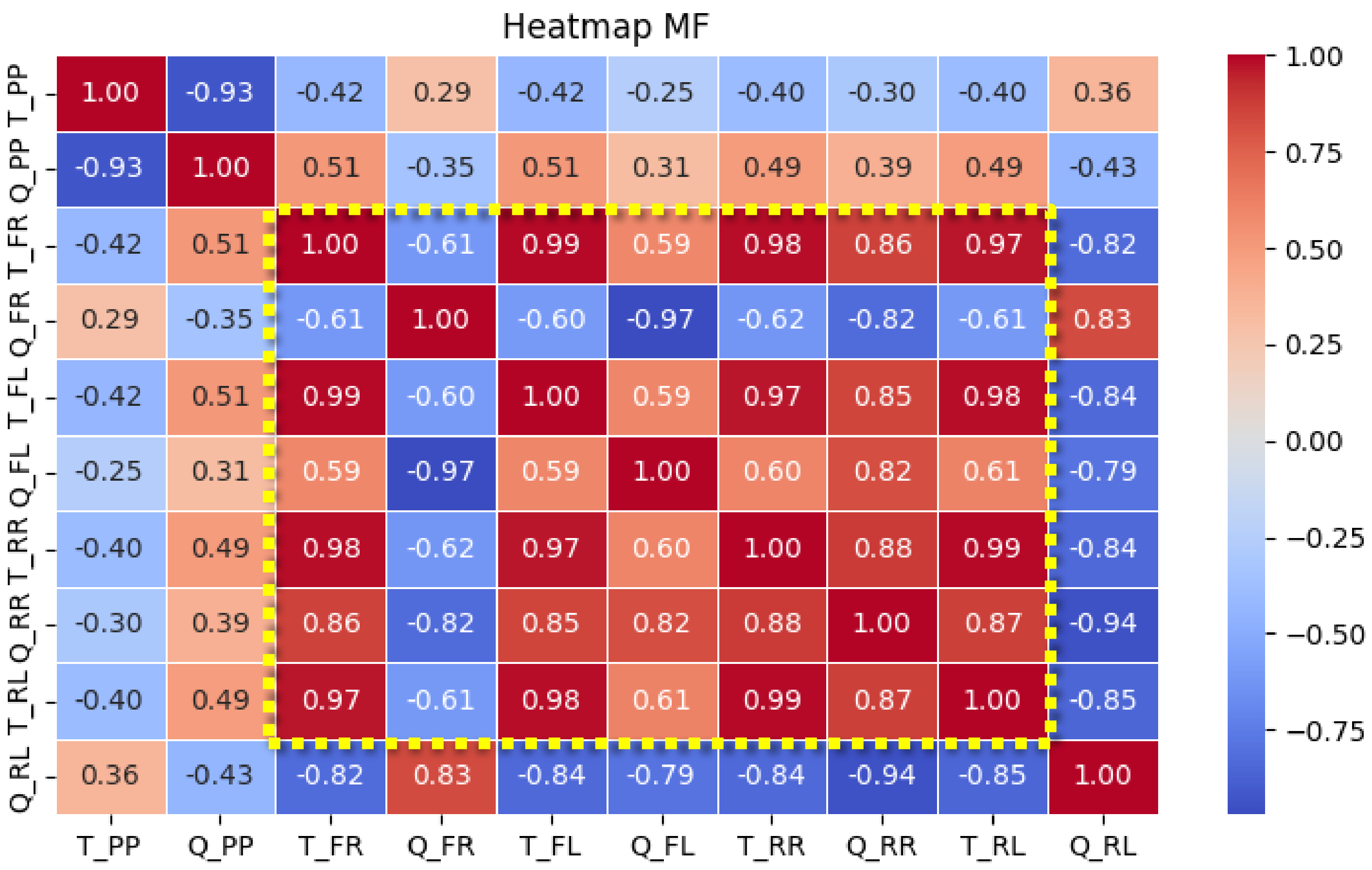
The biggest challenge was the building process of a complete **PyTorch library** that correctly uses Bayesian Layers for Transfer Learning. Ultimately, all network hyperparameters, including the number of frozen layers, were selected through **Bayesian Hyperparameter Optimization**.

We created an open-source PyTorch BNN-TL library, adhering to modular and object-oriented programming principles.

After trying various methods, we calculated the model's error and uncertainty using two custom percentage functions.

BNN test

Despite low percentage error on test-set, after conducting manual checks, we discovered that all the models were failing in predicting correctly the output with some particular flow conditions. Strong correlations in the mid-fidelity dataset were causing the model to learn incorrect patterns with inconsistent data associations.



Data Augmentation

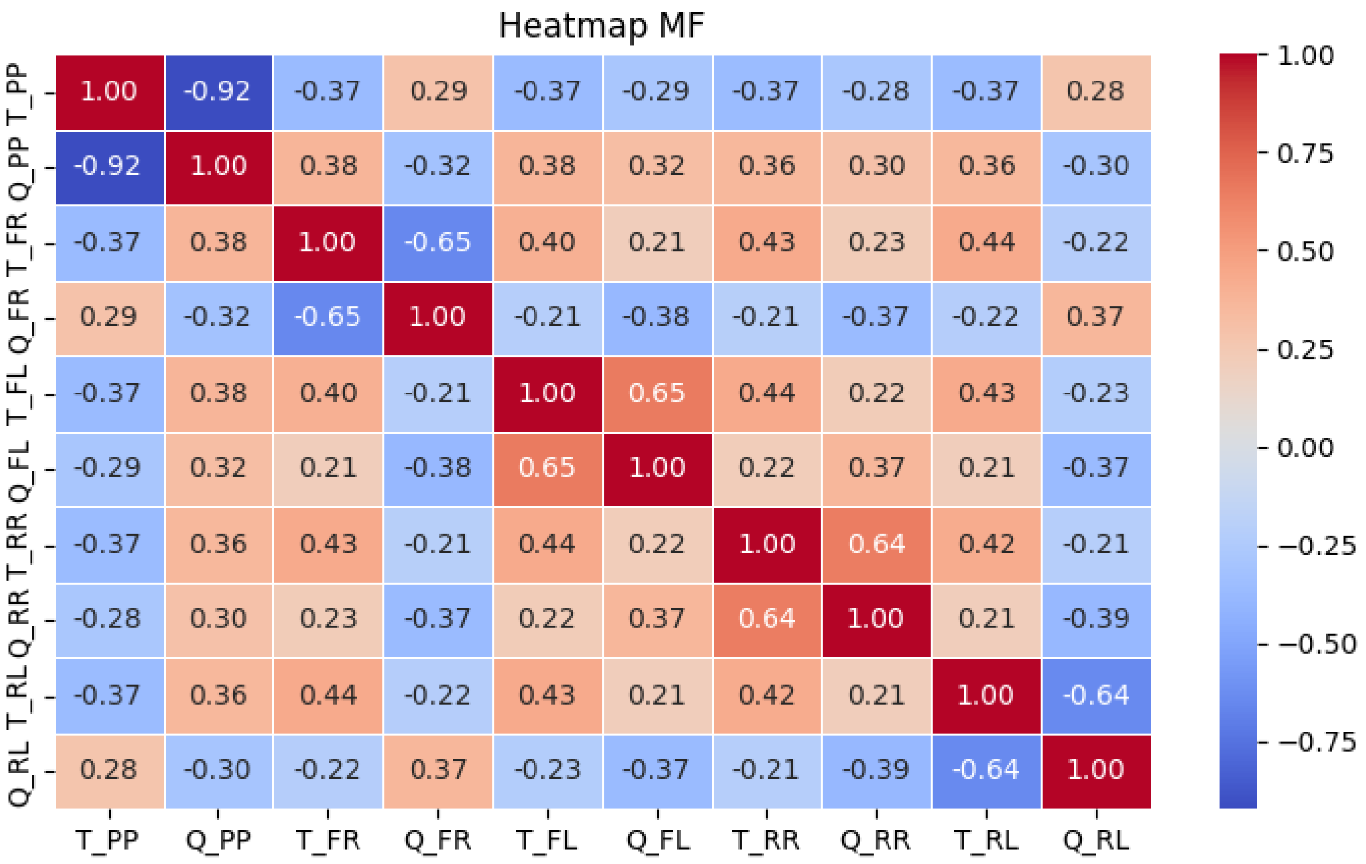
This means that our knowledge about physical behavior of Skiron-X was insufficient to get a good set of representative samples.

To reduce data correlation, we implemented a **data augmentation** process by:

1. Executed 52 new DUST simulation based on the **uncertainty prediction** output of the Bayesian Network (adaptive sampling).
2. Randomly adding zero RPM values along with corresponding **zero values** for thrust and torque.
3. Generating new values by using **cubic interpolation** under fixed flow conditions.
4. Generating new values by **varying the RPM of individual blades** and incrementing the thrust according to established curves.

Final Model

In the end, we successfully expanded the mid-fidelity dataset and reduced the correlation between variables. The final model passed our tests and exhibited an even lower percentage error compared to models trained on separate datasets.

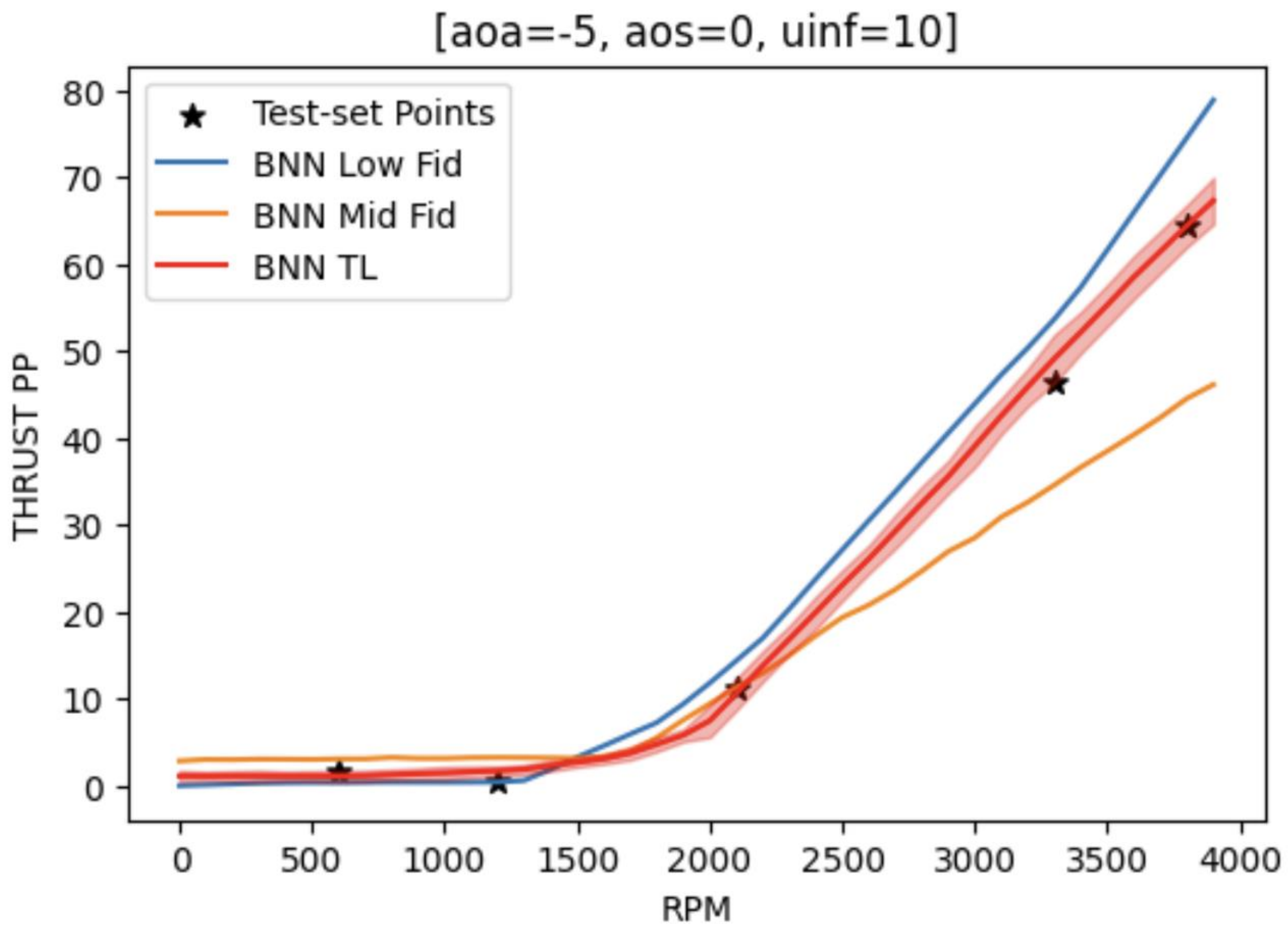


52 (real) **+ 888** (syntethic)
new samples with **Data Augmentation**

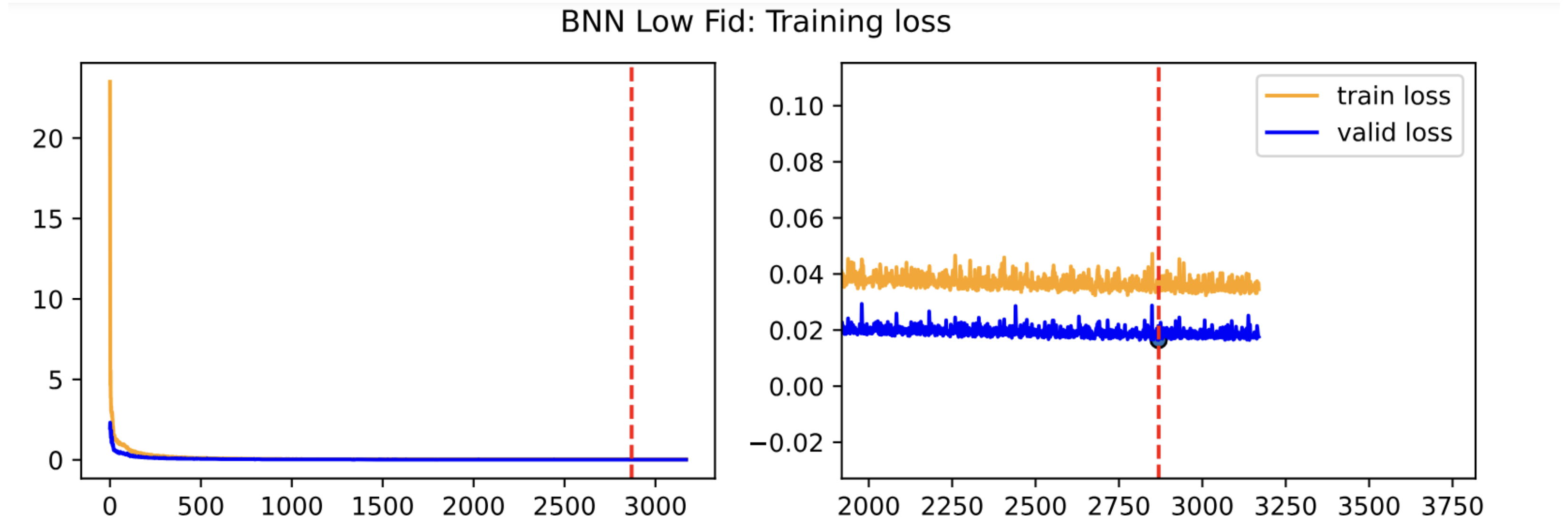
1004 Final dataset (70% train, 15% validation, 15% test)

Model Integration

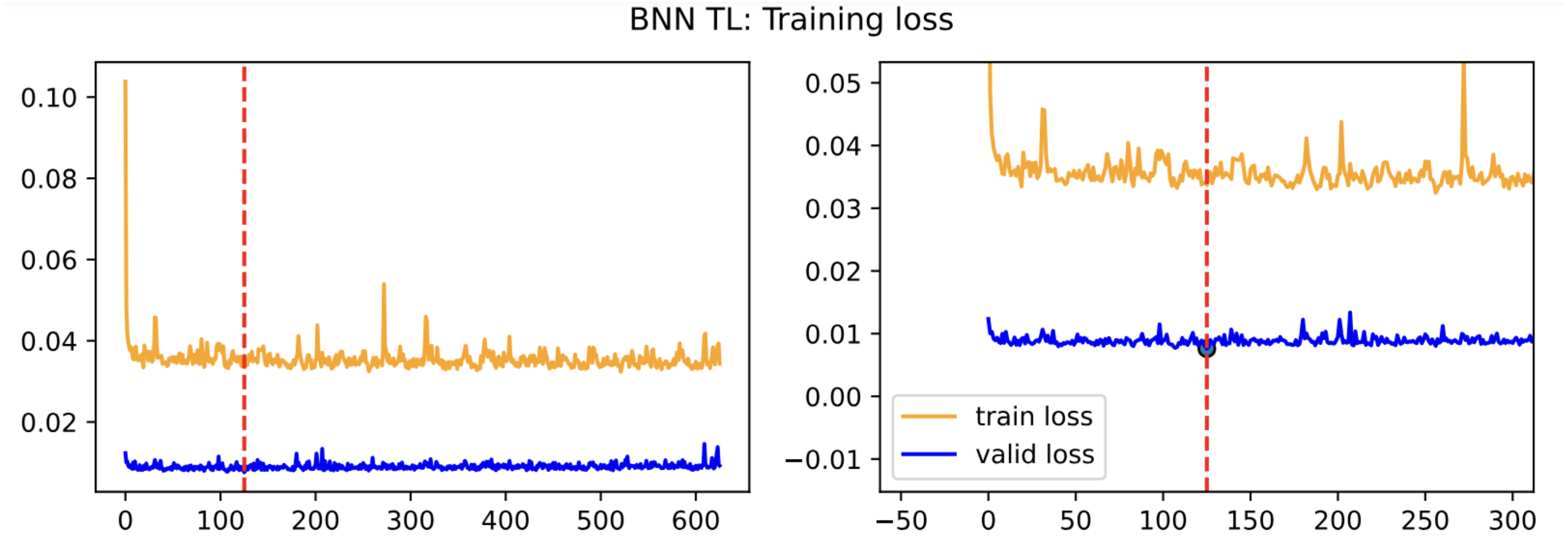
BNN with TL achieves good accuracy and consistently outperforms single models trained on separate datasets. The predictive uncertainty quantification provided valuable insights into the model's convergence capability (epistemic) and data coherence (aleatoric). The model was ultimately integrated into the system.



Pre Train Loss



Transfer Leraning Loss



Results

Model	Training Time	Prediction Time	Average Perc. Error	Average Perc. Confidence	N. Params
BNN LF	~41 min	~0.02 sec	8.5%	94.2%	174932
BNN MF	~35 min	~0.02 sec	6.4%	92.1%	19092
DF: BNN TL	~55 min	~0.02 sec	1.8%	97.3%	174932
DF: CoKriging	~4 min	~0.005 sec	5.9%	88.2%	-

Benchmarks on NVIDIA Quadro P2000 GPU

Final model hyperparameters:

- **Input dimension**: 8
- **Number of layers**: 5
- **Units per layer**: 224, 144, 160, 112, 96
- **Optimization Function**: LeakyReLU
- **Prior Standard Deviation**: 0.0351
- **Prior Mu**: 0
- **Output dimension**: 10
- **Learning rate**: 0.0016
- **Learing rate during TL**: 0.0081
- **N. parameters**: 174932

Clusteringdata analisys

Clustering

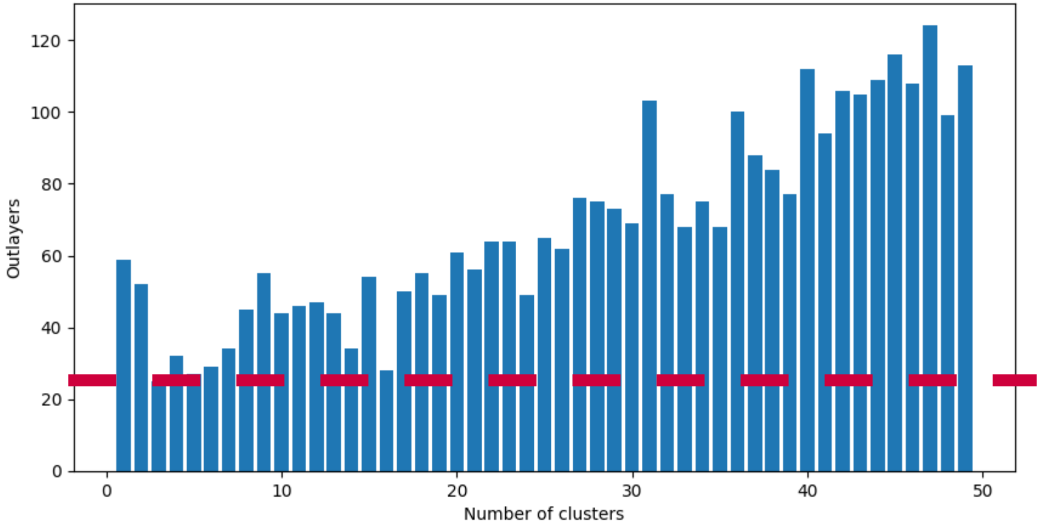
Artificial intelligence were exploited to provide a **tool for data analysis**. **Clustering** algorithms are data driven unsupervised learning methods that can learn statistical pattern from the training data.

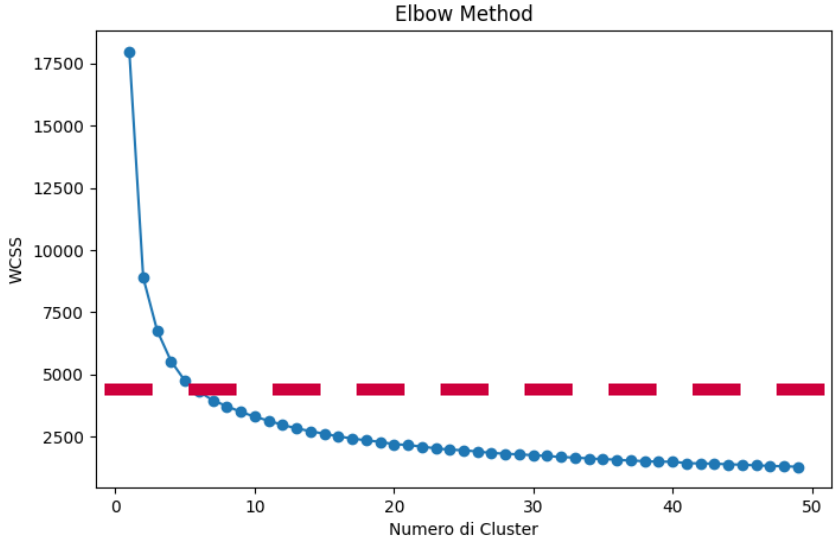
Clustering was applied to samples obtained from **aeroelastic simulations**. The goal of this analysis is to find and identify **particular patterns of flow conditions** that can be considered **dangerous**.

This preliminary analysis was conducted using the simple clustering method **k-means** by using a fast time series feature extracting method.

Number of cluster

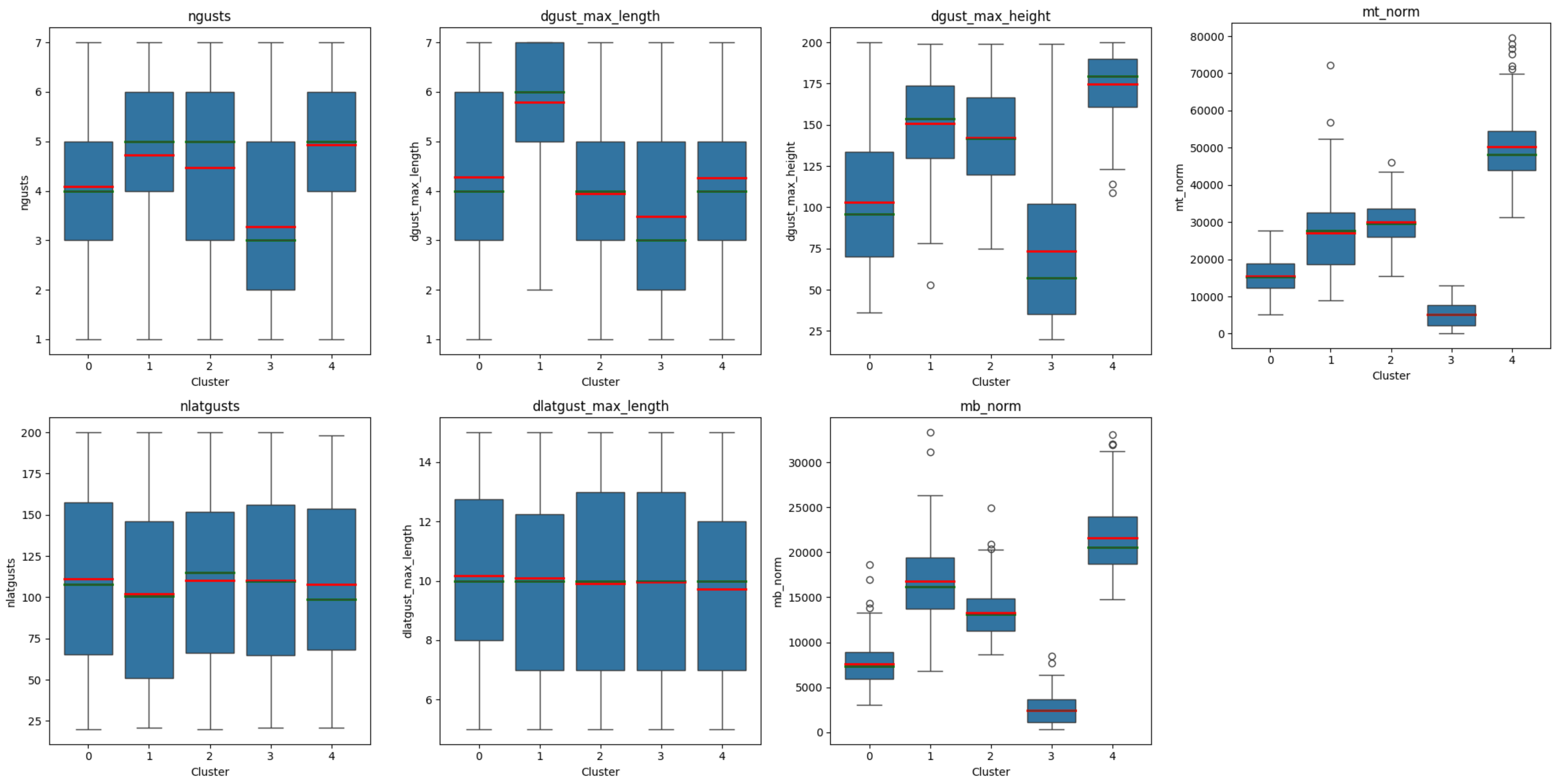
To get a good number of clusters with K-Means, it’s important to minimize the number of outliers in the data. Then, use the elbow method by calculating the inertia for a range of k, and identify the point where the reduction in inertia starts to slow down significantly. This point represents the optimal number of clusters.





Preliminary cluster results

School of Engineering
ZAV Centre for Aviation



Conclusions

Data Fusion techniques:

- Leveraging samples of varying fidelities, it’s possible to reduce the reliance on high-fidelity datasets and thus saving considerable computational resources and time.

Implementation of uncertainty quantification:

- Enhances the reliability and interpretability of our models

Applied k-means clustering to aeroelastic simulation data:

- Identified patterns in flow conditions that may indicate potential risks

ML can contribute to the field of aircraft certification process by providing a more efficient and accurate means of simulating complex aerodynamic phenomena and automatizing and optimizing analysis.

Thank you!
Questions?

zh
aw
School of Engineering
ZAV Centre for Aviation