Variance Decomposition of MEDLI2 Reconstructed Heating Using Neural Networks

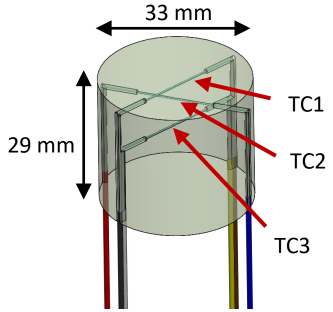
Hannah Alpert

Mars 2020 Project

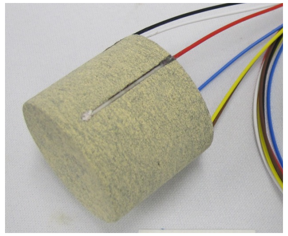
Contents

MEDLI2 Inverse Heating Reconstruction

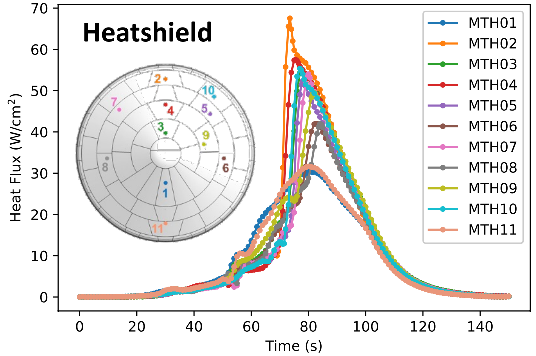
- **Mars 2020 aeroshell had an instrumentation suite called MEDLI2**
  - Included plugs (MISPs) made of the heatshield and backshell TPS materials
    - 1-3 thermocouples embedded within each plug
    - Flush mounted into heatshield (MTH) and backshell (MTB) in 11 locations on the heatshield and 6 locations on the backshell

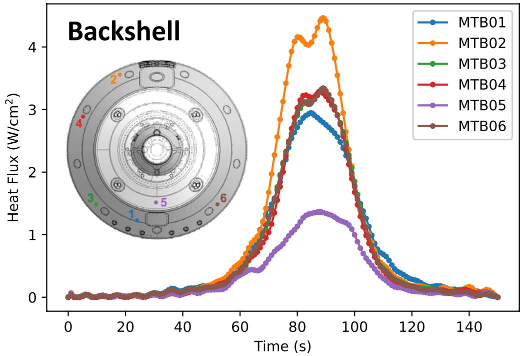


Heatshield PICA Plug

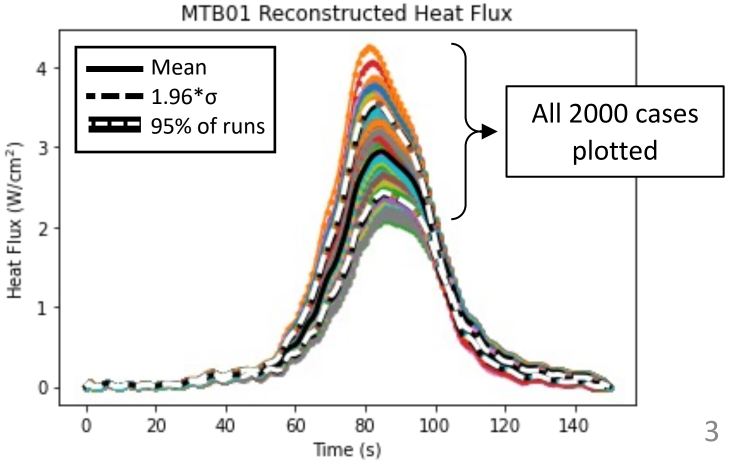


- **Used data from thermocouples to figure out aerothermal environments during entry**
  - Fully Implicit Ablation and Thermal Response (FIAT) Model take aerothermal environment as input and outputs in-depth temperature throughout TPS material
  - FIAT_Opt runs through different environments until the output closely matches flight TC data (40 min per run)
  - Determined surface heating profiles for all MTH and MTBs





- **Accounted for uncertainties in material properties using Monte Carlo simulations**
  - Assumed distribution for each property based on flight-lot material testing
  - Ran 2000 FIAT_Opt iterations and calculated 95% confidence intervals



- ***What is the sensitivity of the reconstructed heat flux and heat load to the uncertainty in each material property?***

Variance Decomposition

- **Variance decomposition** is a method used to **quantify** how much the output of a model can be attributed to **uncertainty in each of the model input factors**
- **Sobol indices** can be used to **calculate the sensitivity of the output** to each input factor
  - Total Sensitivity: $S_{T_i} = 1 - \frac{V_{X_{\sim i}}\left(E_{X_i}(Y|X_{\sim i})\right)}{V(Y)} = \frac{1}{N}\frac{\sum_{j=1}^{N} f(A)_j \left(f(A)_j - f(A_B^{(i)})_j\right)}{V(Y)}$

$$V(Y) = \frac{1}{N-1}\sum_{j=1}^{N}\left(f(A)_j - f_0\right)^2$$

- **A** and **B** are (*N x K*) sampling matrices where K is the # of factors and N is the # of samples
  - $\mathbf{A_B^{(i)}}$ is matrix **A** where the ith column is replaced with the ith column from **B**

$$A = \begin{array}{c} \begin{array}{ccc} \rho & C_p & k \end{array} \\ \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \\ A_{41} & A_{42} & A_{43} \\ A_{51} & A_{52} & A_{53} \end{bmatrix} \end{array} \Big\downarrow N \qquad \mathbf{B} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \\ B_{41} & B_{42} & B_{43} \\ B_{51} & B_{52} & B_{53} \end{bmatrix} \begin{array}{l} \to f(B)_1 \\ \to f(B)_2 \\ \to f(B)_3 \\ \to f(B)_4 \\ \to f(B)_5 \end{array} \qquad A_B^{(1)} = \begin{bmatrix} B_{11} & A_{12} & A_{13} \\ B_{21} & A_{22} & A_{23} \\ B_{31} & A_{32} & A_{33} \\ B_{41} & A_{42} & A_{43} \\ B_{51} & A_{52} & A_{53} \end{bmatrix} \begin{array}{l} \to f(A_B^{(1)})_1 \\ \to f(A_B^{(1)})_2 \\ \to f(A_B^{(1)})_3 \\ \to f(A_B^{(1)})_4 \\ \to f(A_B^{(1)})_5 \end{array}$$

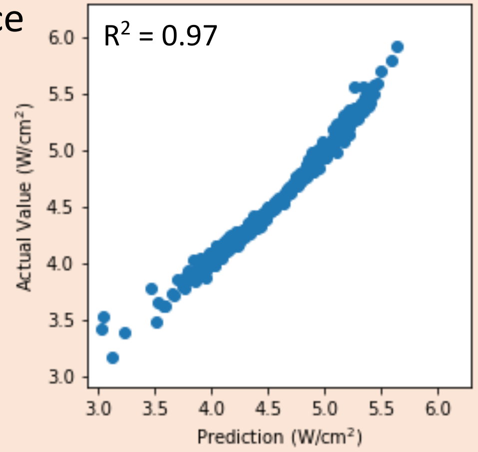
- For total sensitivity, need to to compute output for **A** and all $\mathbf{A_B^{(i)}}$ matrices
  - This entails running FIAT_Opt $N \times (K + 1)$ times
- Initial test case showed that Sobol indices do not converge until ~30,000 iterations
  - With each *FIAT_Opt* run taking 40 min on a single CPU and access to 20 CPUs at once → 41 days per MISP

Machine Learning Approach

- **Using a surrogate or predictive model in place of FIAT_Opt enables us to drastically reduce computation time, making variance decomposition feasible**
- **Need a *training set*, a *validation set*, and a *test set***
  - Training set: a subset of the data on which many models (with differently tuned parameters) were trained
  - Validation set: a <u>different</u> subset of the data used to assess the models and choose a “final” model
  - Test set: a <u>different</u> subset of data on which the performance metrics of the final model were evaluated
- **Trained 3 types of machine learning models**

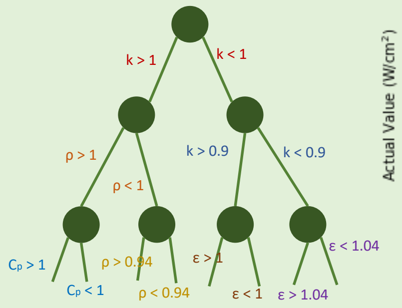
Ridge Regression with Cross-Validation

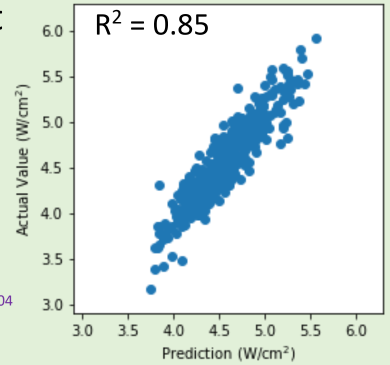
- Similar to multivariate linear regression, with an added regularization term to reduce overfitting



Random Forest Regression

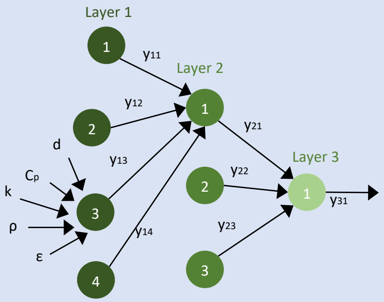
- Ensemble learning method in which several decision trees are created, split a specified number of times based on one variable each time, until every branch only has a specified number of cases left

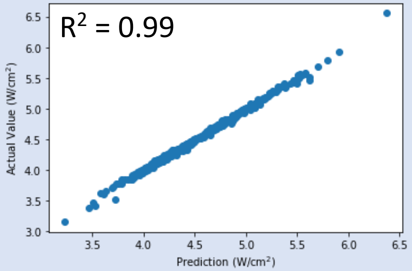




Deep Neural Network (DNN)

- Network comprised of several nodes that take in inputs, create a non-linear function, and feed that into the next node





$$y_{11} = Relu[m_{111}(mass) + m_{211}(vel) + m_{311}(efpa) + m_{411}(r) + b_{11}]$$
$$y_{31} = [m_{131}(y_{21}) + m_{231}(y_{22}) + m_{331}(y_{23}) + b_{31}]$$

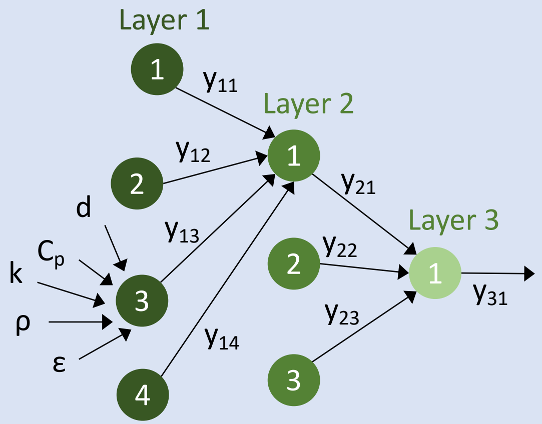
Node #
Variable #
Layer #

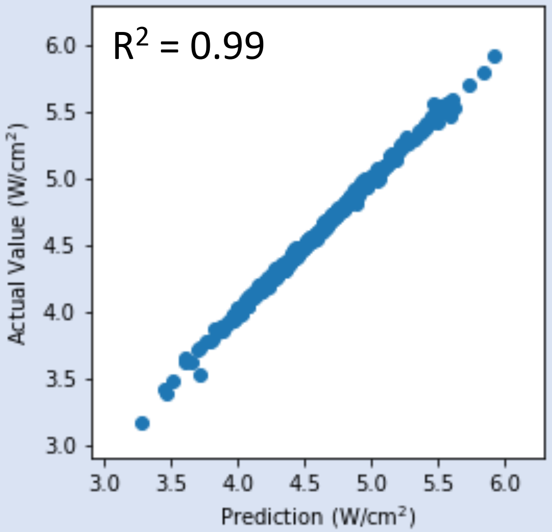
Machine Learning Approach: Deep Neural Network

- **DNN outperformed random forest and ridge regression models on validation set in early testing so further tuned DNN hyperparameters to attain final model**
- **Training a DNN is a stochastic process**
  - Weights of neural network are randomly initialized pre-training
  - Order in which samples from training set are "seen" by model can influence final weights
- **Trained 100 DNNs with the same architecture and hyperparameters to evaluate variation in final model performance caused by stochasticity**

Deep Neural Network (DNN)

- Network comprised of several nodes that take in inputs, create a non-linear function, and feed that into the next node





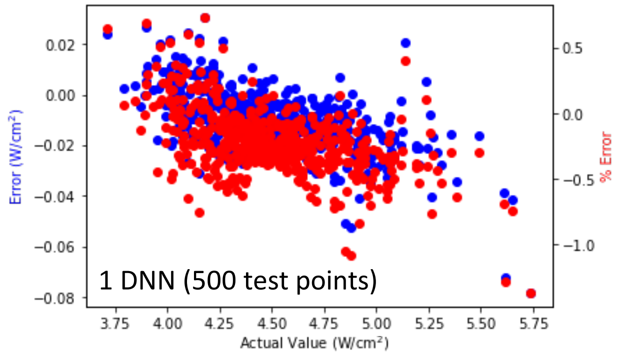
$$y_{11} = Relu[m_{111}(mass) + m_{211}(vel) + m_{311}(efpa) + m_{411}(r) + b_{11}]$$
$$y_{31} = [m_{131}(y_{21}) + m_{231}(y_{22}) + m_{331}(y_{23}) + b_{31}]$$

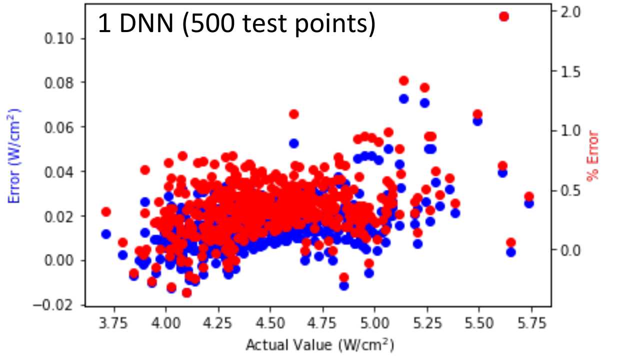
Node #
Layer #
Variable #

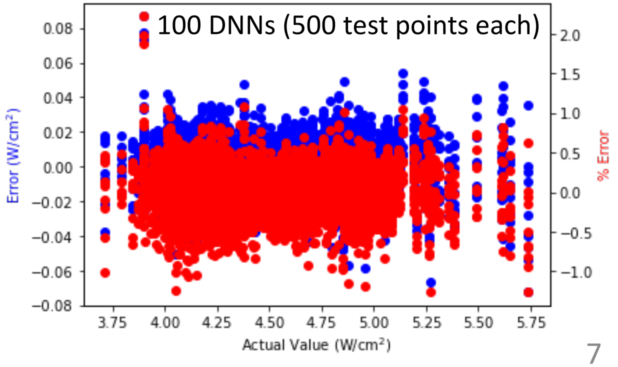
Neural Network Performance

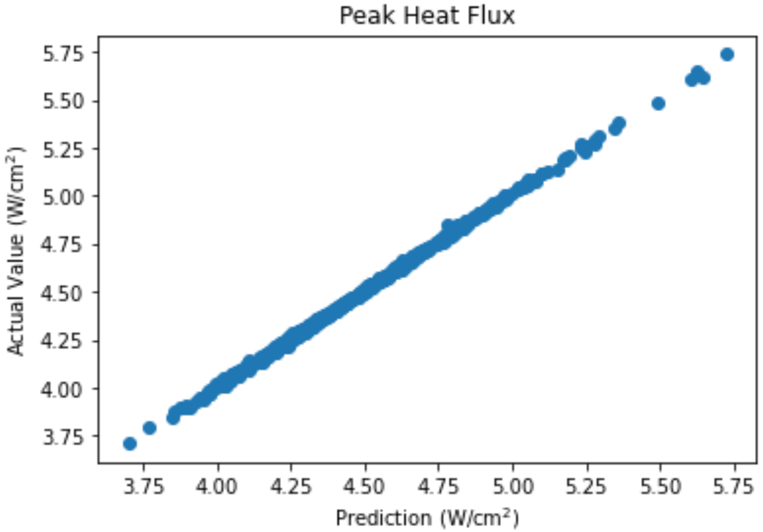
- **Accurate and consistent peak heat flux and heat load estimates**
  - Evaluated mean absolute error (MAE) and mean absolute percent error (MAPE) to assess accuracy and their standard deviations to assess consistency
  - 100 DNNs have different biases but across all DNNs the bias is fairly uniform

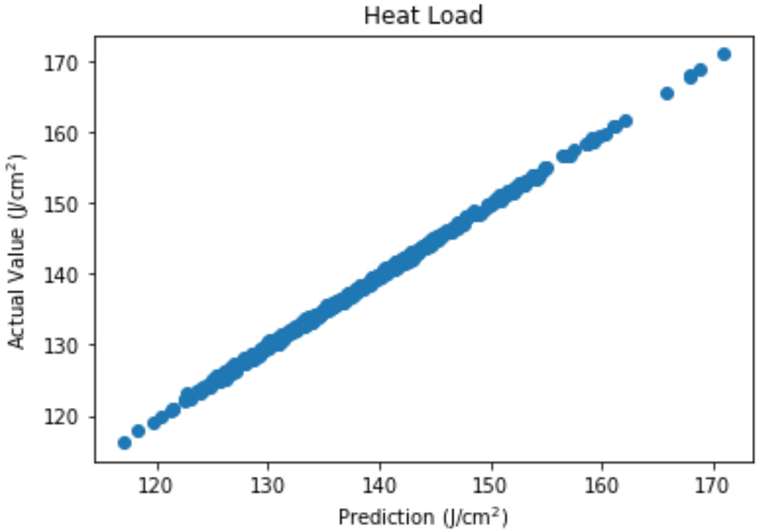
	Average MAE	Std MAE	Average MAPE	Std MAPE
Peak Heat Flux	0.010 W/cm^2	0.004 W/cm^2	0.21%	0.08%
Heat Load	0.358 J/cm^2	0.179 J/cm^2	0.25%	0.12%











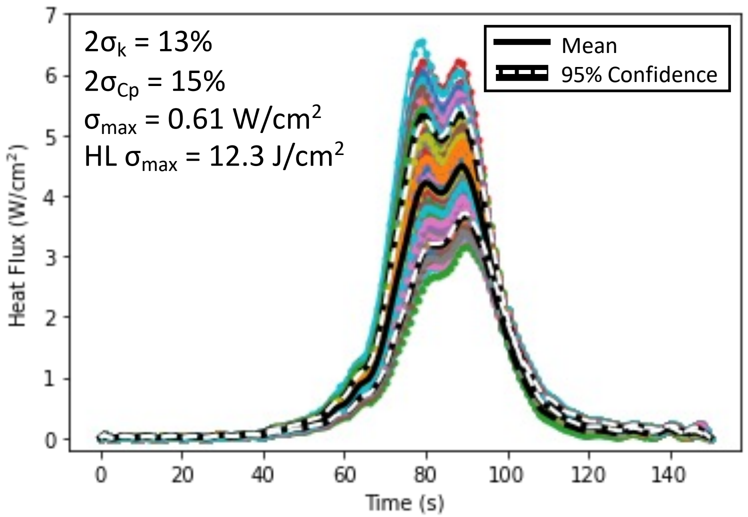
Sensitivity Analysis

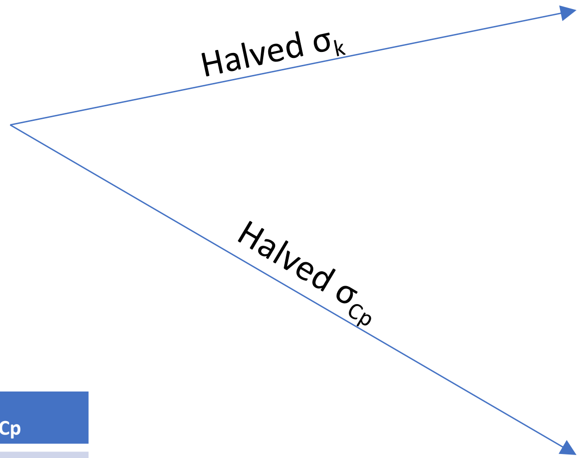
- **Instead of running FIAT_Opt, used neural network to predict peak heating for each line of A and $A_B^{(i)}$ matrices, reducing computation time *significantly* (10000x)**
- **Peak heat flux**
  - C_p drives uncertainty, 2.2x greater than *k*
  - Standard deviation on the Sobol index for C_p = 1% of the mean
- **Heat load**
  - C_p drives uncertainty, 2.8x greater than *k*
  - Standard deviation on the Sobol index for C_p = 1.4% of the mean

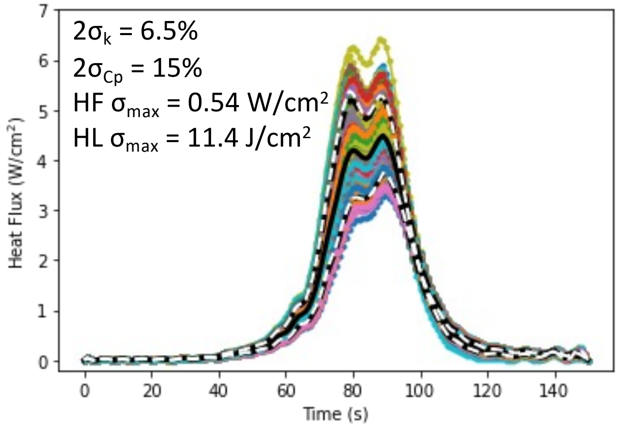
	Density	Heat Capacity	Thermal Conductivity	Virgin Emissivity	Char Emissivity	TC Depth
	ρ	C_p	k	ε_v	ε_c	d
Peak Heat Flux						
Mean	0.014	0.550	0.251	0.122	0.058	0.071
Std Dev	0.001	0.005	0.004	0.003	0.002	0.002
Heat Load						
Mean	0.018	0.589	0.212	0.090	0.056	0.075
Std Dev	0.005	0.008	0.008	0.005	0.004	0.002

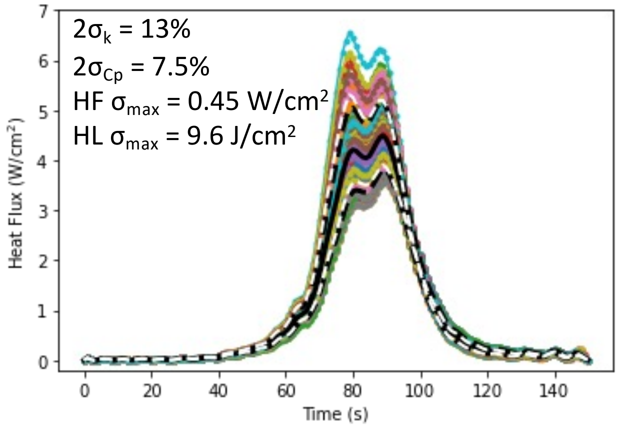
Sensitivity Analysis: Sanity Check

- **Confirmed results from sensitivity analysis by running Monte Carlo analysis with different uncertainty distributions**
- **Using $2\sigma_k$ of 13% and $2\sigma_{Cp}$ of 15%, maximum standard deviation is 0.61 W/cm^2**
- **Halving $2\sigma_k$ reduces max std dev to 0.54 W/cm^2, while halving $2\sigma_{Cp}$ reduces it to 0.45 W/cm^2**









	Nominal	Halved σ_k	Halved σ_{Cp}
Peak Heat Flux Std Dev (W/cm^2)	0.61	0.54	0.45
Heat Load Std Dev (J/cm^2)	12.3	11.4	9.6

Summary

- **Used 2000 Monte Carlo iterations to train a neural network that can be used in lieu of FIAT_Opt for significantly (10000x) faster computation time**
  - Tried three different machine learning models and found that the deep neural network had better results than ridge regression and random forest regression
  - DNNs were highly accurate and consistent in predicting peak heat flux and heat load
- **Found that heat capacity is the biggest drivers of uncertainty in reconstructed peak heating and heat load, followed by thermal conductivity**
- **Results can be leveraged to provide requirements for material property measurements needed to improve the accuracy of surface heating prediction and ultimately lead to the reduction of design margins in the future**
- **Next steps – conduct sensitivity analysis on all MISPs and see how Sobol indices vary across material (SLA-561V vs. PICA) and aeroshell location**

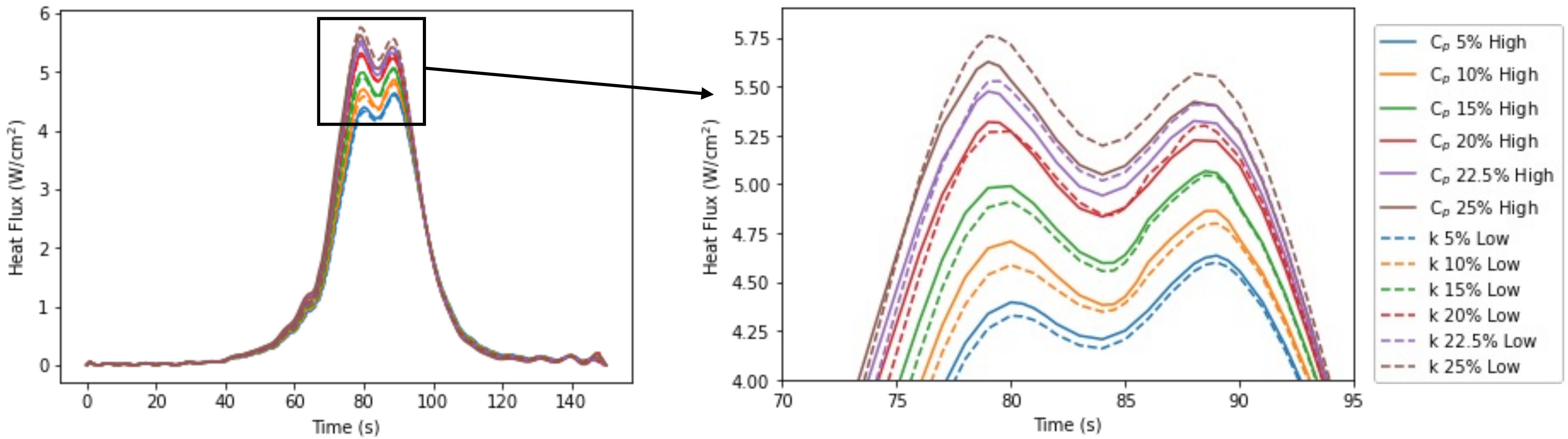
References

1. **Hwang, H., et al., "Mars 2020 Entry, Descent, and Landing Instrumentation 2 (MEDLI2)," in *AIAA Thermophysics Conference*, 2016.**
2. **Y. Chen and F. Milos, "Ablation and Thermal Response Program for Spacecraft Heatshield Analysis," *Journal of Spacecraft and Rockets*, vol. 36, no. 3, pp. 475–483, 1999.**
3. **Alpert, H., et al., "Inverse Estimation of Mars 2020 Entry Aeroheating Environments Using MEDLI2 Flight Data," in *AIAA SciTech Forum*. AIAA, 2022.**
4. **Monk, J., et al., "MEDLI2 Material Response Model Develop- ment and Validation," in *AIAA SciTech Forum*, 2022.**
5. **Saltelli, A., et al., "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index," *Computer Physics Communications*, 181, 259-270, 2010.**
6. **Moradi, A., et al., "A variance decomposition approach to uncertainty quantification and sensitivity analysis of the Johnson and Ettinger model," *J. of the Air and Waste Management Assoc.,* 65(2), 154-164, 2015.**

Increasing/Decreasing Value k and C_p

- **Increased C_p and decreased k by 5-25%**
- **C_p has more of an impact than k if the change is <20%, but k has more of an impact if change is >20%**

	Normal	C_p						k					
		5%	10%	15%	20%	22.5%	25%	5%	10%	15%	20%	22.5%	25%
Peak HF (W/cm^2)	4.42	4.63	4.86	5.07	5.32	5.47	5.63	4.59	4.80	5.04	5.30	5.53	5.76
% Change	--	4.9	10.0	14.6	20.3	23.8	27.3	4.1	8.6	14.1	19.9	25.04	30.3



“Final” Neural Network

- Created 500 new data points as test set
- Train on 2000 data points
- Take 40,000 samples and retrain NN 100 times to evaluate variation

		ST rho	ST Cp	ST k	ST Ev	ST Ec	ST d	ST tot	MAE	MAPE	Spearman
	0	0.015908	0.549609	0.247301	0.123660	0.056883	0.071511	1.064871	0.006723	0.151422	0.999618
	1	0.013619	0.556066	0.253507	0.125288	0.058927	0.068316	1.075723	0.012140	0.268377	0.999546
	2	0.014160	0.554432	0.245160	0.123995	0.057877	0.070664	1.066288	0.007188	0.160813	0.999648
	3	0.015499	0.548050	0.261644	0.118490	0.055802	0.068164	1.067649	0.015221	0.329658	0.999549
	4	0.015067	0.552800	0.254145	0.124329	0.057546	0.066583	1.070470	0.021985	0.484463	0.999498
	...	...	...	...	...	...	...	...	...	...	...
	99	0.013767	0.547917	0.247505	0.123669	0.058069	0.073720	1.064646	0.015431	0.340632	0.999587
Mean	100	0.014263	0.549719	0.251085	0.122433	0.057878	0.070858	1.066235	0.009538	0.212193	0.999640
Std Dev	101	0.000926	0.004506	0.003538	0.002540	0.001746	0.001771	0.003558	0.003626	0.079607	0.000059
Max	102	0.016683	0.562068	0.261644	0.128745	0.062079	0.074462	1.075723	0.028132	0.620876	0.999748
Min	103	0.012036	0.537770	0.243547	0.117074	0.053558	0.066418	1.056405	0.005698	0.126624	0.999477

S_{Tk}

Mean: 0.251
Std Dev: 0.004
Max: 0.261
Min: 0.245

S_{TCp}

Mean: 0.550
Std Dev: 0.005
Max: 0.562
Min: 0.538

MAE: 0.010 W/cm^2
MAPE: 0.21%