Universität der Bundeswehr München

Preliminary Results: Premixed source term prediction with ANNs

Maximilian Hansinger

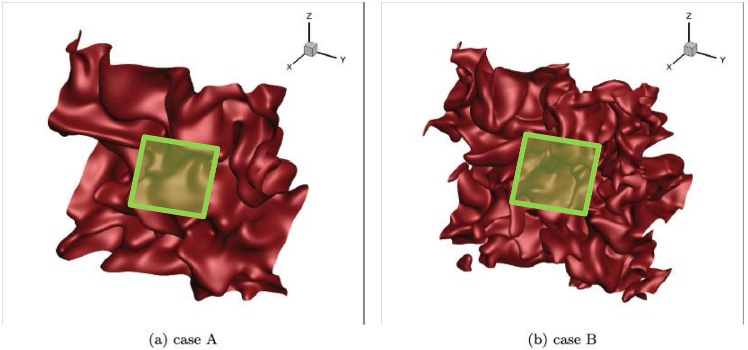
22. Dezember 2020

Idea: Predict $\overline{\omega}_{DNS}$ with filtered data

Planar premixed flame DNS data base: UPRIME=5, 9, 15

Extract filtered features: $\tau_c, c'_{SGS}, U'_{SGS}, mag(\nabla\widetilde{U}),\ mag(\nabla\tilde{c}), \Delta_{LES}, \bar{\dot{\omega}}_m, \tilde{c}, \bar{c}$

Train/Test split: Green region (schematically) is used as test set, rest for training.



Isofläche der Flamme bei c=0.85
(a) Ka = 5, (b) Ka = 15

Filter widths for **training**:

n = 4, 8, 16, 24, 28, 32, 40

Filter width **only for testing (not in training)**:

n = 20

Data preparation and training

1. For each DNS data set (UPRIME = 5, 9, 15) filter it with n = 4, 8, 16, 24, 28, 32, 40.
2. Compose a data base from all filter widths n.
3. Log transform the data base (values are highly skewed → closer to Normal distribution).
4. Compute the mean and standard deviation from the log-transformed data base and scale the data base for the training.
5. Train an ANN with UPRIME5 training data and another one with UPRIME15 training data.
6. Networks: 10 ResNet blocks a 200 Neurons (rather complex: 823k parameter).

Notes on the training and networks

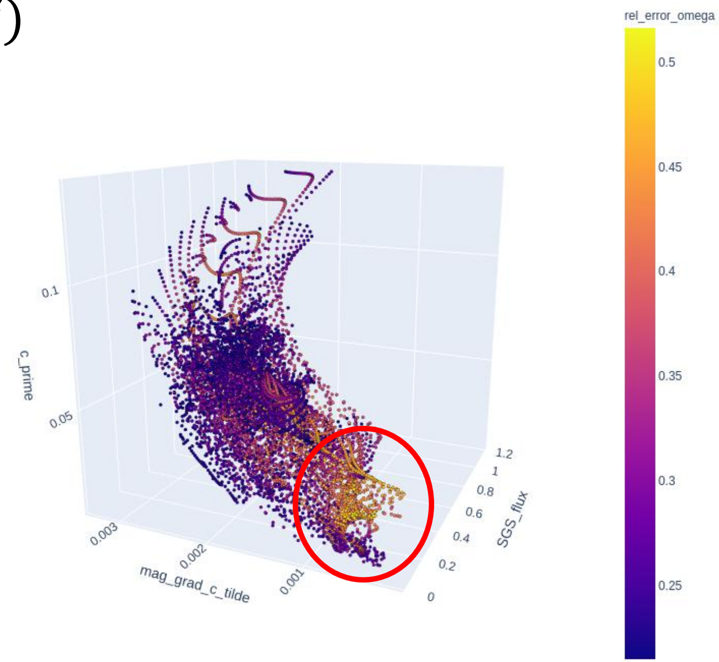
1. Loss function: MSE
2. Activation function: ReLU
3. Dropout rather decreased the training performance than improving it.
4. I used only one batch normalization layer in the last layer.
5. Choose a rather small batch size (128-512).
6. Learning rate: between 1e-3 to 1e-4

Preliminary data analysis

- Compute 3D scatter plot from: $\tau_c, {c'_{SG}}^S, ma^g(\nabla\tilde{c})$
- Color by relative Error between $\overline{\omega}_{DN}{}^S$ and $\overline{\omega}_m$

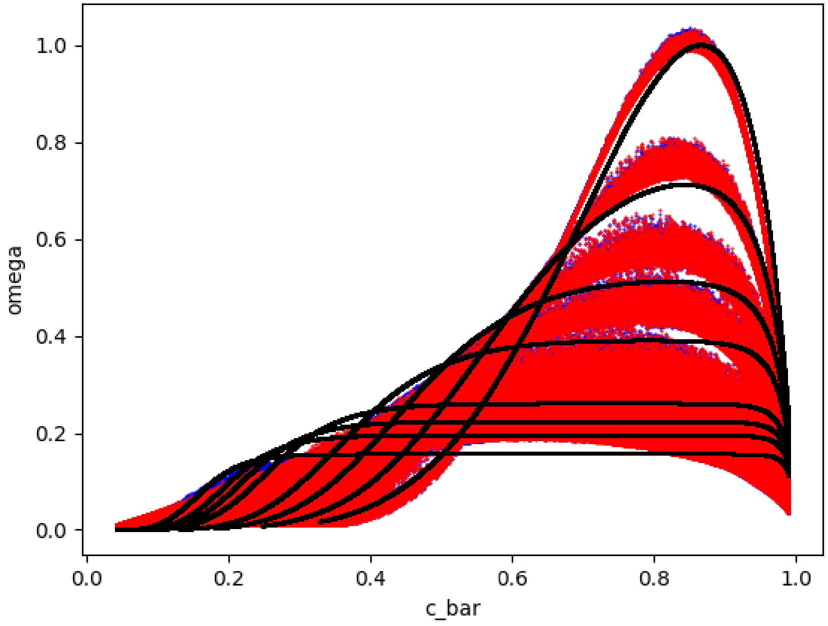
  → high error seems to be correlated with low $\tau_c, {c'_{SG}}^S, ma^g(\nabla\tilde{c})$ seem to

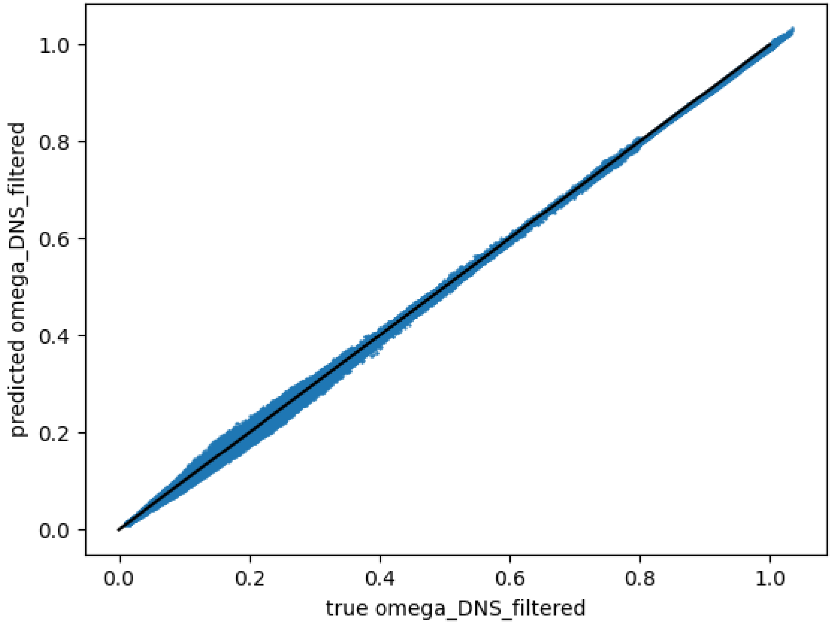
- I agree with Junsu: $\tau_c, {c'_{SG}}^S$ are difficult to model in LES. However, I think for model development and understanding the model they could be relevant.



Results: test data set UPRIME5

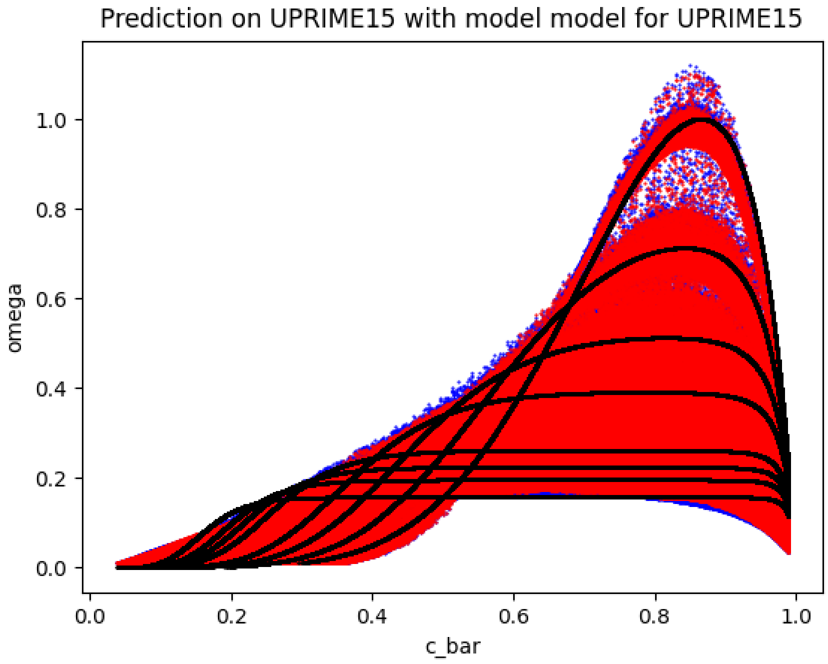
Red: prediction, blue: true value, black: Pfitzner model

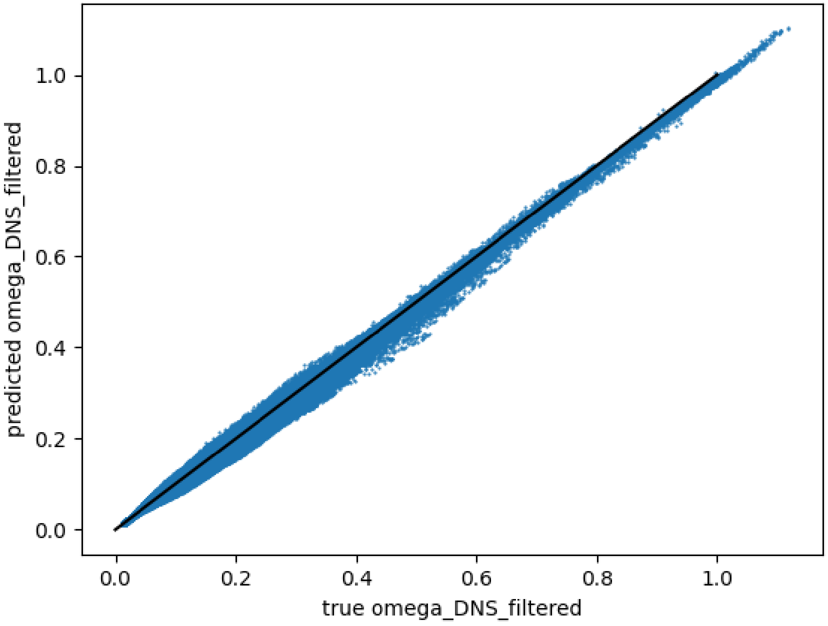




Results: test data set UPRIME15

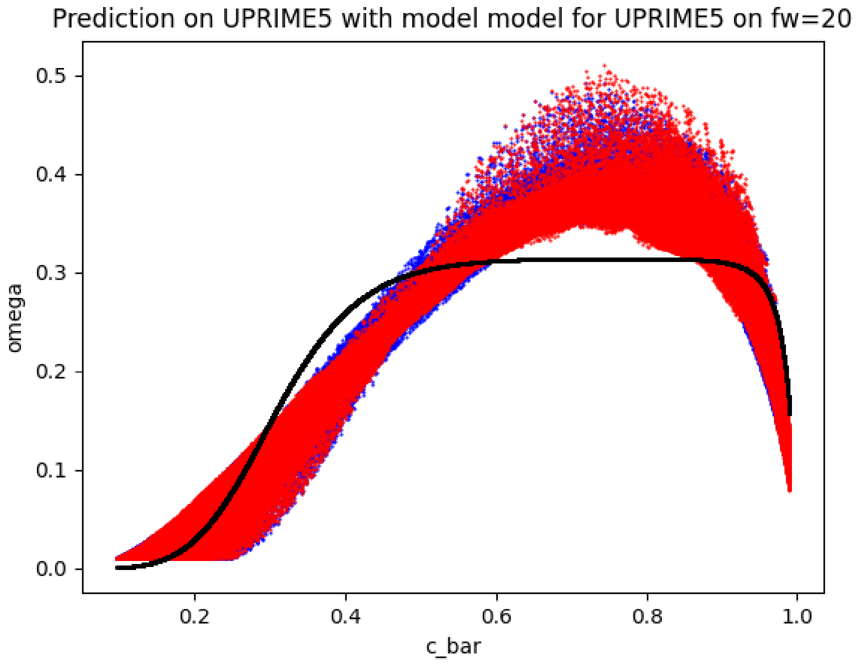
Red: prediction, blue: true value, black: Pfitzner model

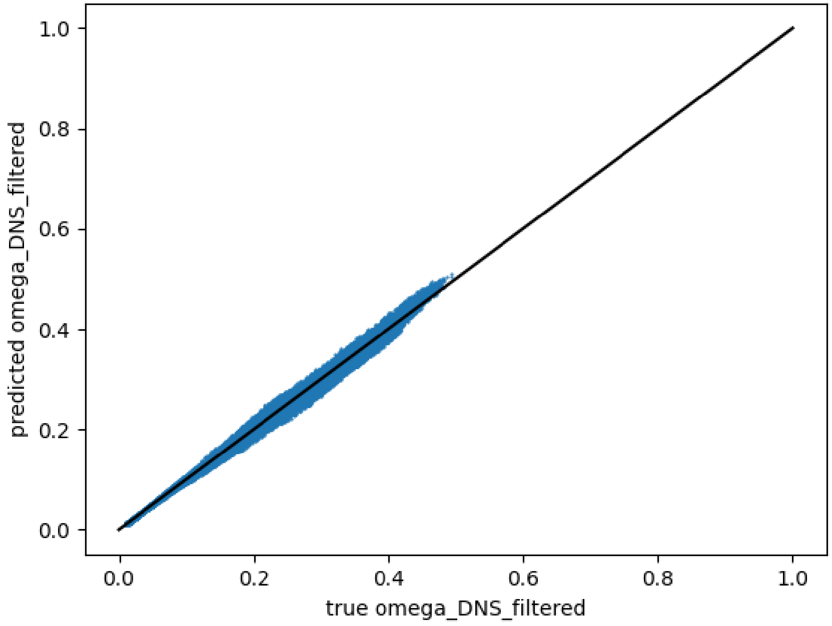




Results: keep out data set UPRIME5 (n=20)

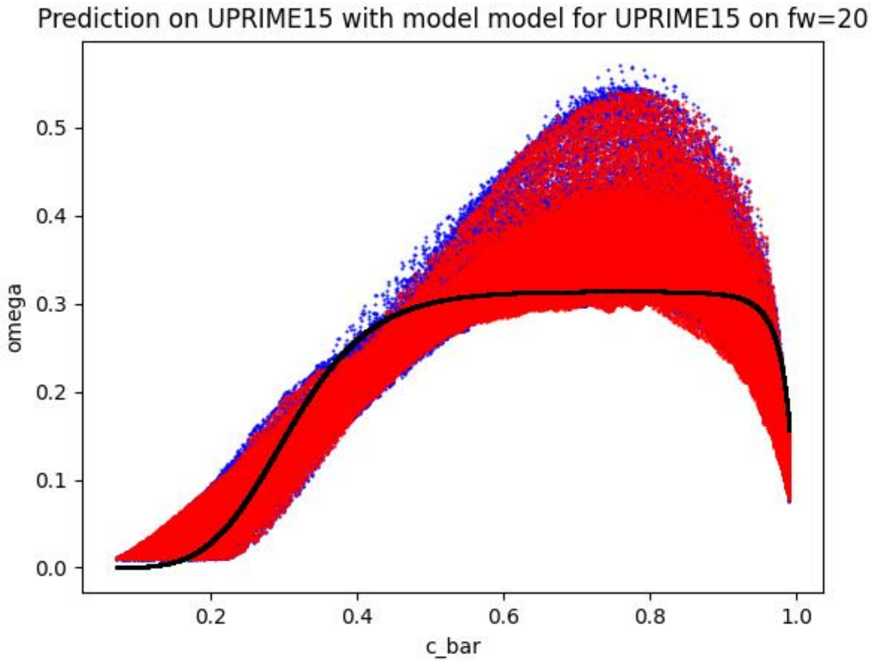
Red: prediction, blue: true value, black: Pfitzner model

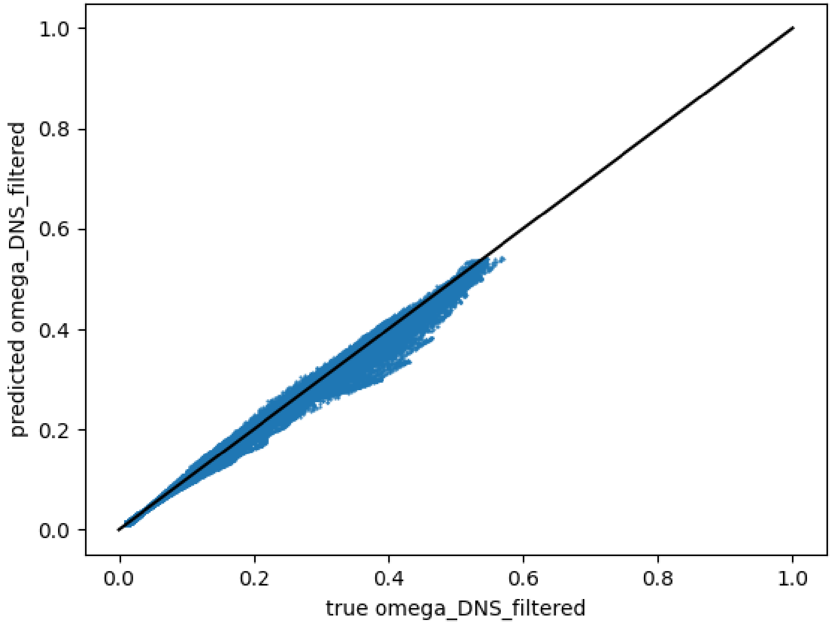




Results: keep out data set UPRIME15 (n=20)

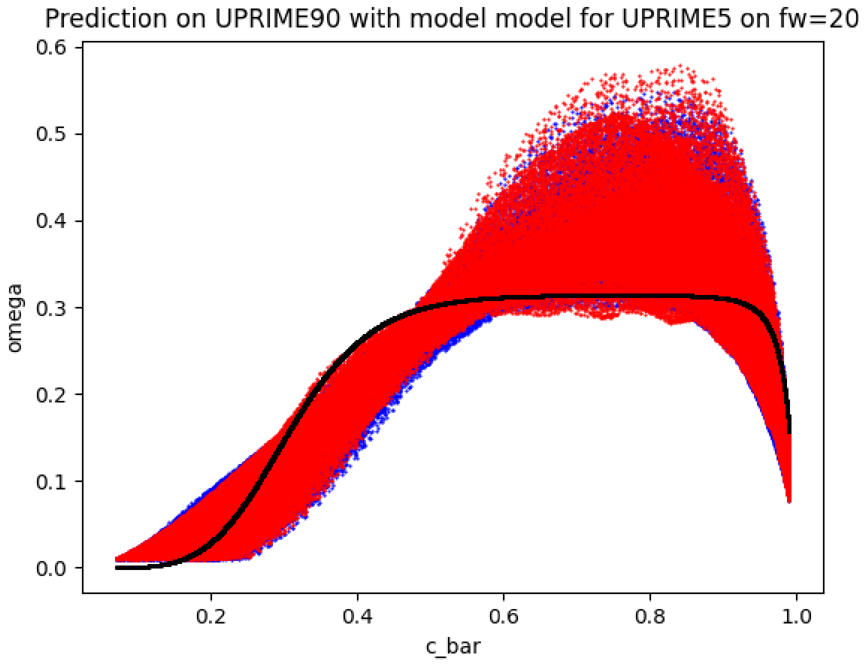
Red: prediction, blue: true value, black: Pfitzner model

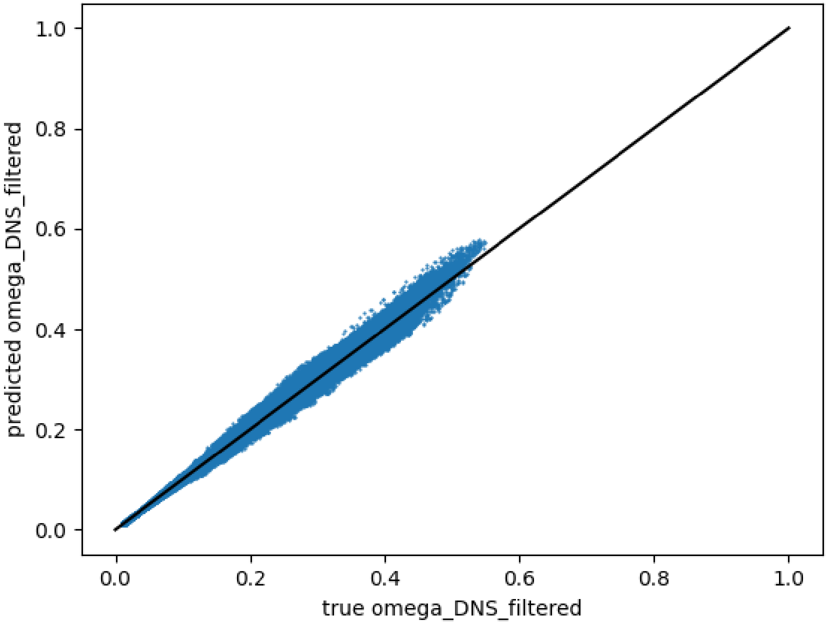




Results: keep out UPRIME9 with model trained for UPRIME5

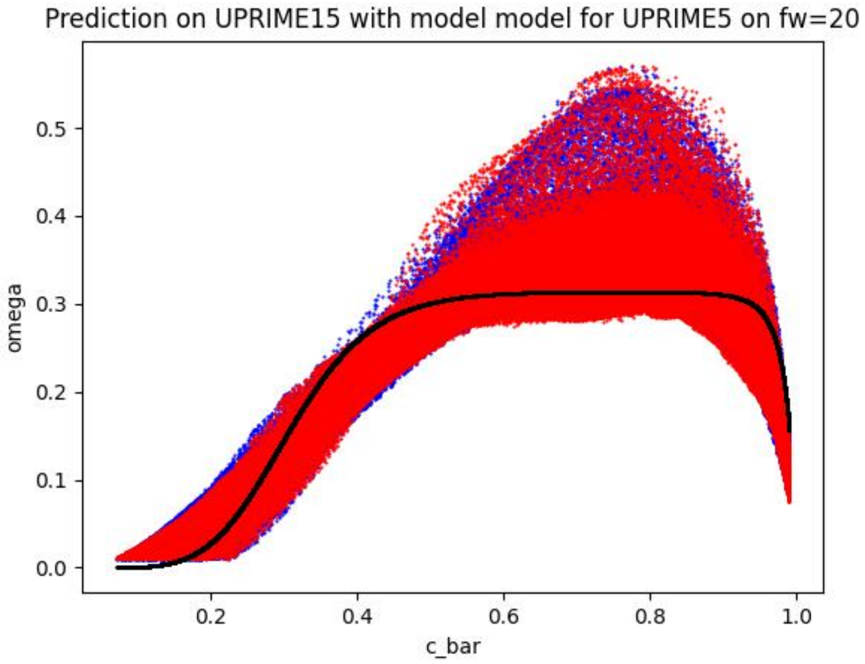
Red: prediction, blue: true value, black: Pfitzner model

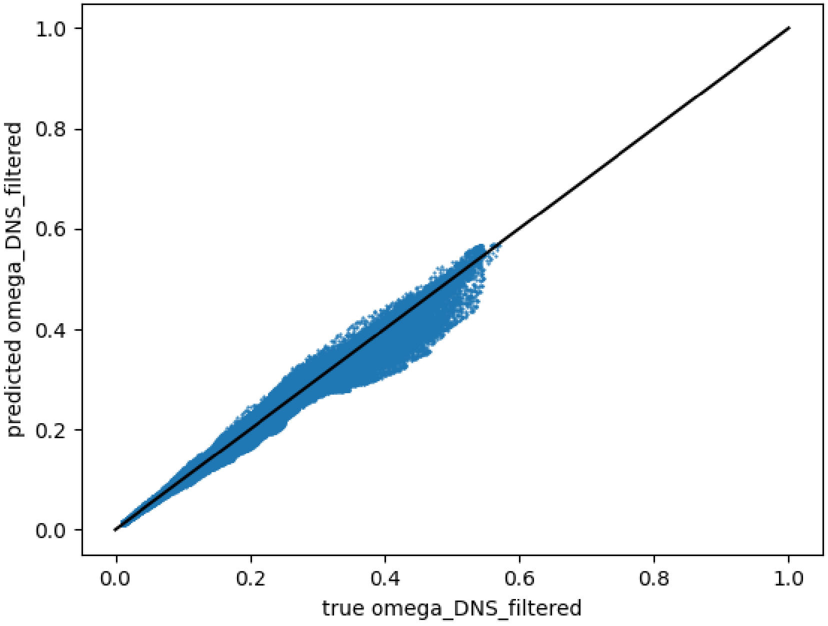




Results: keep out UPRIME15 \w model trained for UPRIME5

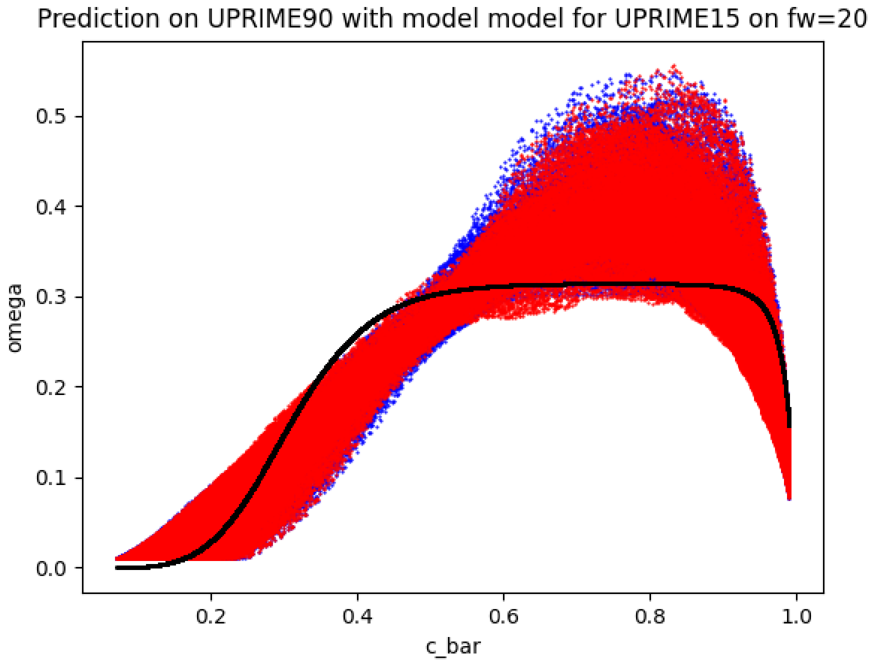
Red: prediction, blue: true value, black: Pfitzner model

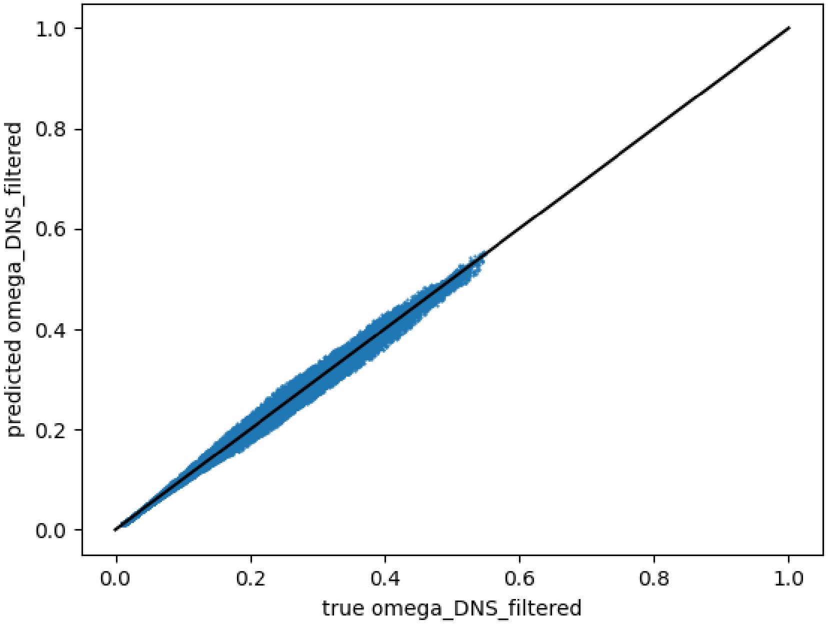




Results: keep out UPRIME9 \w model trained for UPRIME15

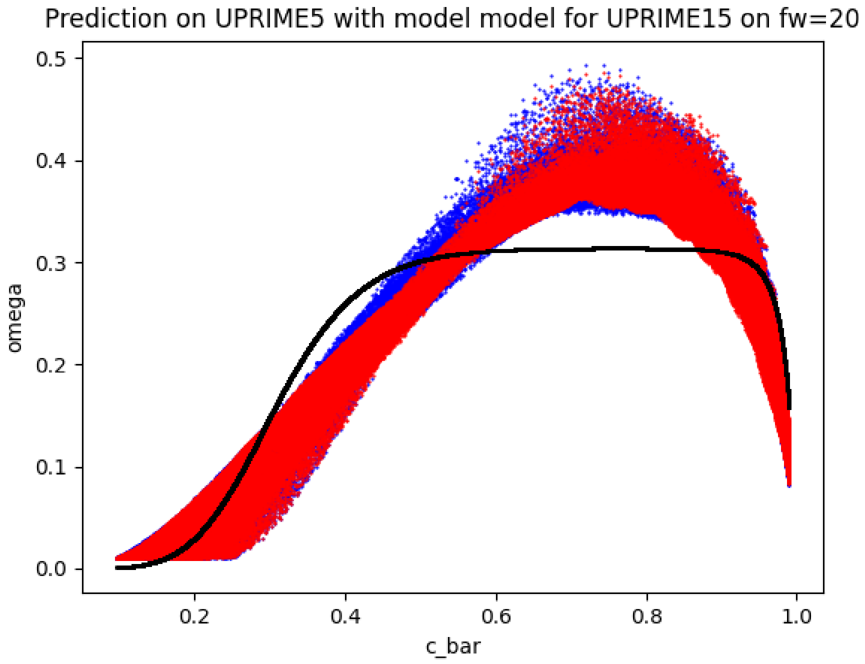
Red: prediction, blue: true value, black: Pfitzner model

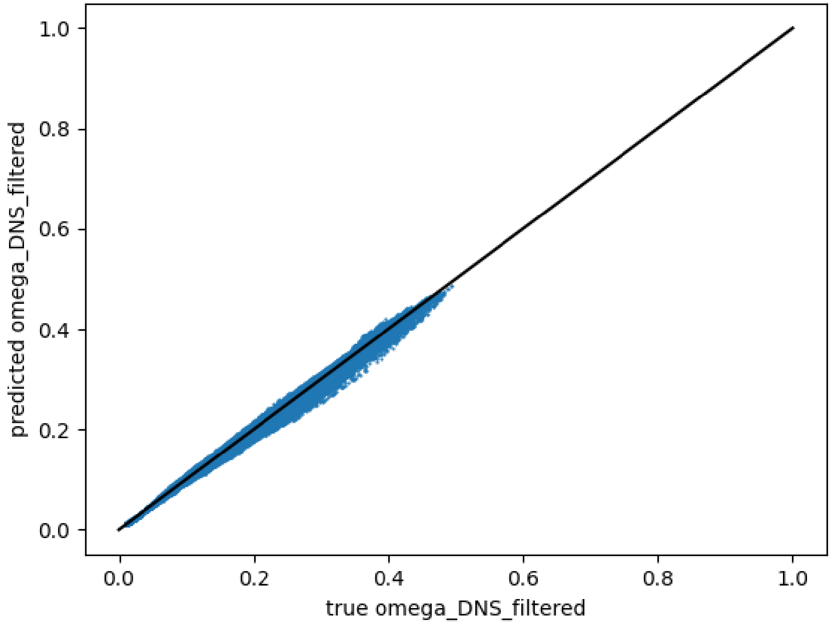




Results: keep out UPRIME5 \w model trained for UPRIME15

Red: prediction, blue: true value, black: Pfitzner model





Comments

1. Both the models UPRIME5 and UPRIME15 predict very well the cases and filter widths they were trained for on the test set.
2. UPRIME5 and UPRIME15 also predict very well filter widths they were not trained for (keep out data with n=20).
3. UPRIME5 and UPRIME15 also cope with cases at different turbulence levels, which they were not trained for. However, if the turbulence level is very different (lower or higher) the prediction performance decrease.
   Anyway, I did not expect that it would work so well…