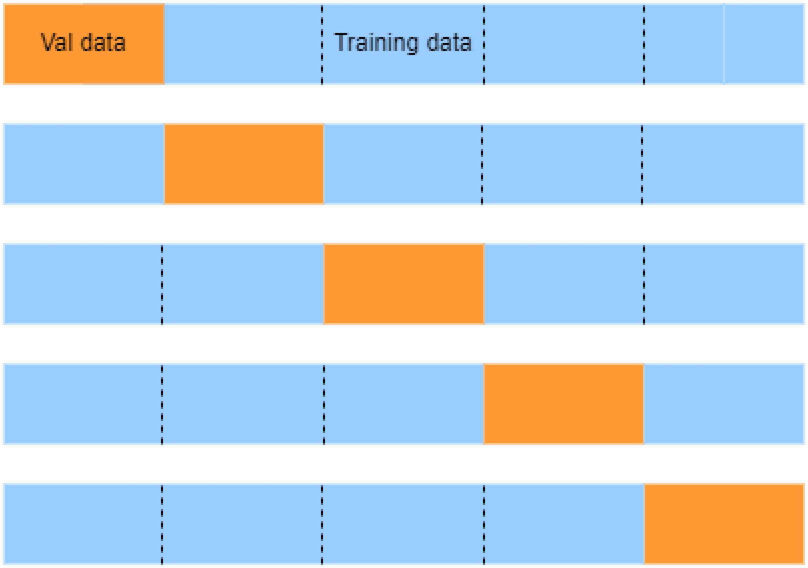
Cross Validation for Metamodels

Outline

- Recall: Cross-Validation
- What are metamodels
- Metamodels example
- Why is it important?
- Code example

Quick cross validation recap

- K-fold cross validation
  - Test data is already put aside
  - K blocks of data
  - Rotate the validation block



- Increase usage of the data
- Decrease splitting bias
- Train an ensemble of models

Other possible methods

- Naïve and fast cross-validation
- Leave-one-out cross-validation
- Fast leave-one-out cross-validation of a linear model
- Corrected leave-one-out
- K-fold cross-validation
- Fast K-Fold cross-validation of a linear model

- Source: https://openturns.github.io/openturns/latest/user_manual/response_surface/_generated/openturns.MetaModelValidation.html

Metamodels

What is a **model**?

- A model is an abstraction of phenomena in the real world

$$y = f(\mathbf{x})$$

What is a **metamodel**?

- The idea is to find a proper approximation that is accurate, cheap and fast to evaluate.

$$\hat{y}(\mathbf{x}) \approx y(\mathbf{x})$$

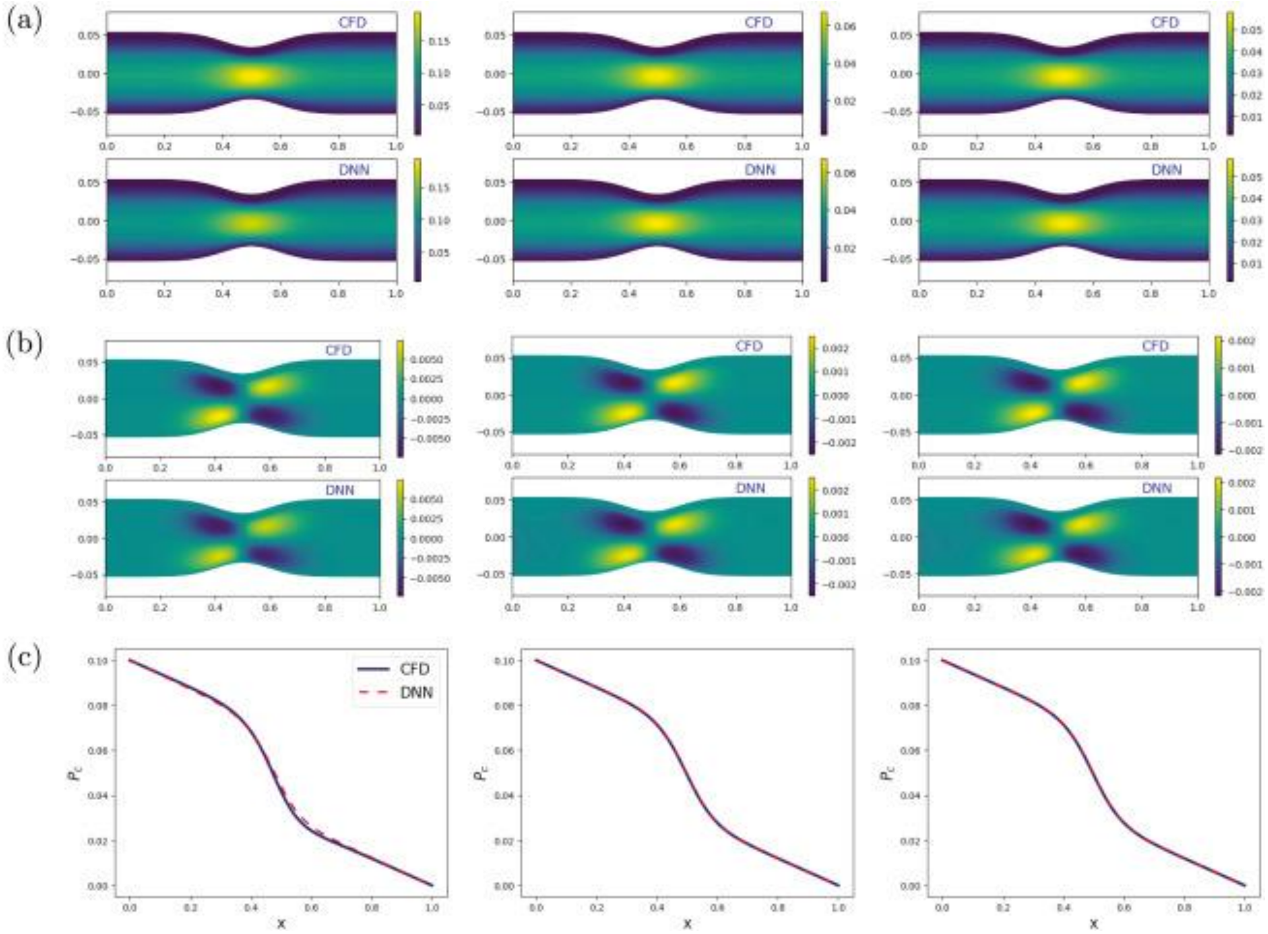
- Source: https://link.springer.com/article/10.1007/s00158-015-1366-1

How to construct such model?

1. **Definition of Design Space:** select design variables and bounds of the design space
2. **Experimental Design:** perform a design of experiments (DOE) to select a set of sampling points
3. **Models Evaluation:** evaluating the true model or simulation at each of the sampled points
4. **Metamodels Creation:** by using any available procedure, like: PRS, Polynomial Response Surface, RBF, KRG, NN, SVR, etc.
5. **Metamodel Validation:** error metric or cross validation

Example of Metamodels

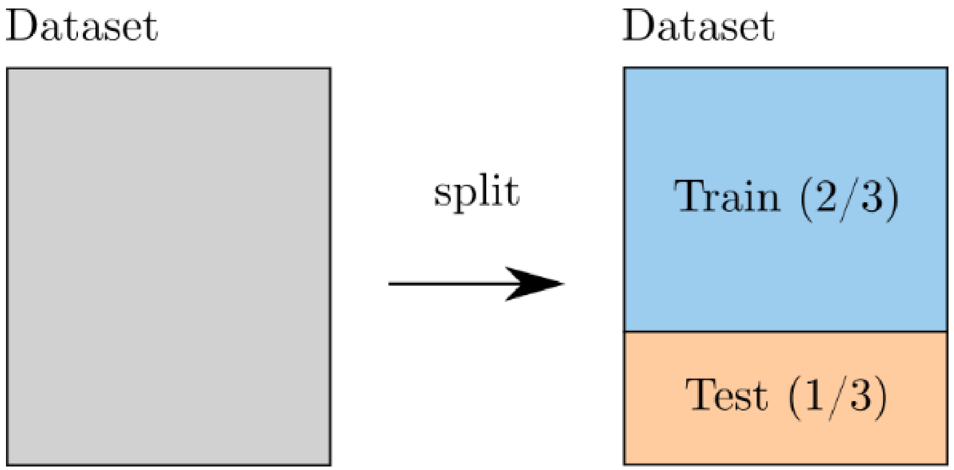
- LabelFree-DNN-Surrogate
    - Simulation-free, physics-constrained deep learning. A surrogate for the CFD model.
- DL model gives quick solutions for fluid flow problems without the grid-based computations of CFD.
- Embedding the governing eq. (Navier-Stokes) and boundary/initial conditions into the loss function of DNN.
- Loss function: ensure the model's prediction satisfy the Navier-Stokes eq.



- Source: https://www.sciencedirect.com/science/article/pii/S004578251930622X#fig4

Validation in context of metamodel

- In ordinary validation of a metamodel, we hold some observations from the sample and train the model on the remaining observations.
- Then, we test the metamodel against the held-out observations.



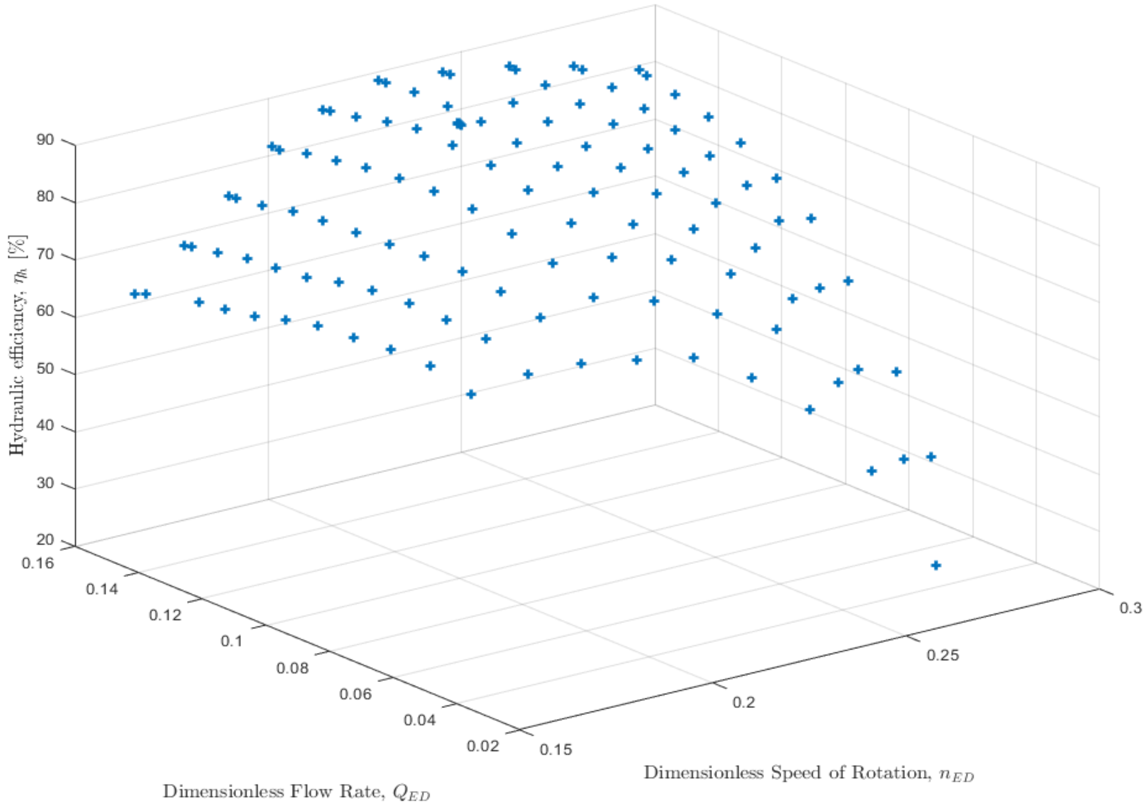
- Source: https://openturns.github.io/openturns/latest/user_manual/response_surface/_generated/openturns.MetaModelValidation.html

Importance of Cross-Validation for Metamodels

- Accuracy estimation
- Overfitting prevention
- Model selection
- Hyperparameter tuning
- Error and uncertainty quantification
- Efficient use of limited data

Code example

- Want a model for efficiency
- Collected efficiency measurements of a reversible pump-turbine



Code example

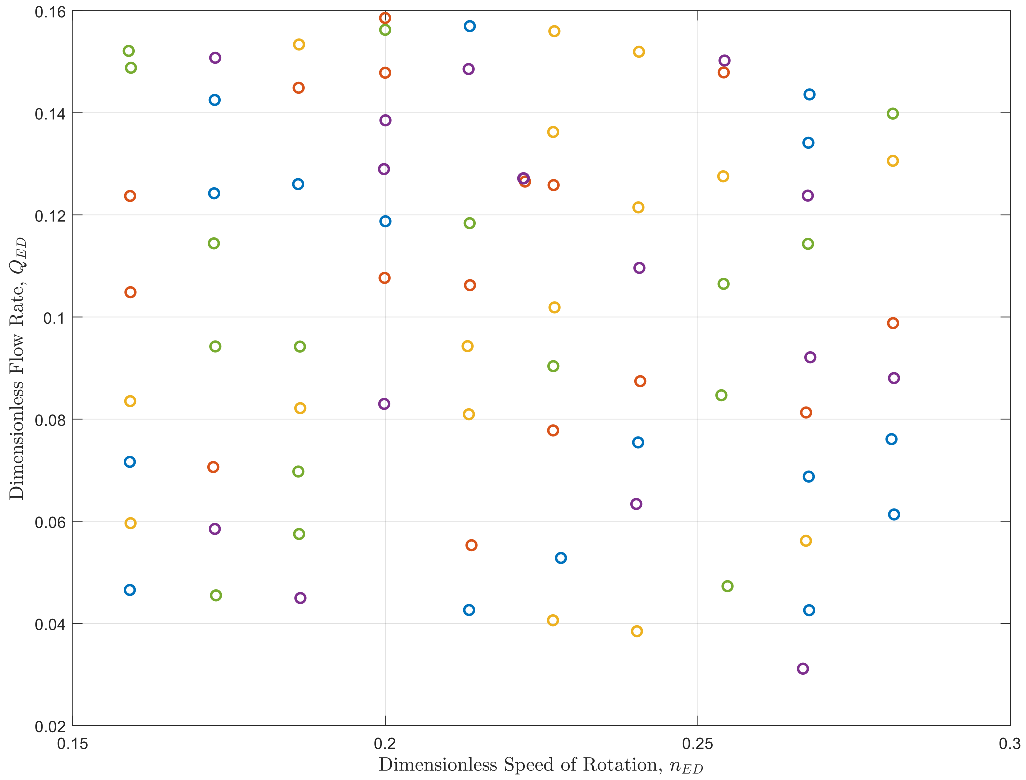
- Divided into training set and test set

```
%% Divide Data into Training and Test Sets
% Define number of bins to ensure domain coverage
numBins = 8;

% Determine bin edges for nED and QED to distribute data evenly
[~, edges_nED] = histcounts(nED, numBins);
[~, edges_QED] = histcounts(QED, numBins);

% Initialize training and test index lists
trainIndices = [];
testIndices = [];

% Loop over each bin in nED and QED to proportionally split data
for i = 1:numBins
    binIndices_nED = find(nED >= edges_nED(i) & nED < edges_nED(i+1));
    for j = 1:numBins
        binIndices_QED = find(QED >= edges_QED(j) & QED < edges_QED(j+1));

        % Find indices in both current nED and QED bins
        binIndices = intersect(binIndices_nED, binIndices_QED);
        numBinData = length(binIndices);

        % Split data in bin if there are values present
        if numBinData > 0
            numTrainBin = round(2 * numBinData / 3);
            shuffledBinIndices = binIndices(randperm(numBinData));

            % Assign data points to training and test sets
            trainIndices = [trainIndices; shuffledBinIndices(1:numTrainBin)];
            testIndices = [testIndices; shuffledBinIndices(numTrainBin+1:end)];
        end
    end
end

% Separate training and test data using determined indices
nED_train = nED(trainIndices);
QED_train = QED(trainIndices);
eta_train = eta(trainIndices);
nED_test = nED(testIndices);
QED_test = QED(testIndices);
eta_test = eta(testIndices);
```

Code example

- Did a 5-fold cross validation of the training data
- Training RMSE: 0.5624

```matlab
%% 5-Fold Cross-Validation on Training Set
% Combine training data into a matrix for partitioning
trainData = [nED_train, QED_train, eta_train];

% Create 5-fold cross-validation partition
cv = cvpartition(size(trainData, 1), 'KFold', 5);
validationErrors = zeros(cv.NumTestSets, 1);  % Array for errors

figure()
hold on
% Iterate over each fold to train and validate
for i = 1:cv.NumTestSets
    trainIdx = cv.training(i);
    testIdx = cv.test(i);

    % Extract training and validation sets for this fold
    nED_train_fold = nED_train(trainIdx);
    QED_train_fold = QED_train(trainIdx);
    eta_train_fold = eta_train(trainIdx);

    nED_val_fold = nED_train(testIdx);
    QED_val_fold = QED_train(testIdx);
    eta_val_fold = eta_train(testIdx);

    % Train a Gaussian Process model on the fold's training set
    cvmodel = fitrgp([nED_train_fold, QED_train_fold], eta_train_fold);

    % Predict on the validation set and calculate validation error
    eta_pred = predict(cvmodel, [nED_val_fold, QED_val_fold]);
    validationErrors(i) = mean(sqrt((eta_val_fold - eta_pred).^2));  % RMSE

    % Plot the validation data points for visual validation
    plot(nED_val_fold, QED_val_fold, 'o', 'LineWidth', 1.5)
end
```



Code example

- Tested the model against the test data
- Test RMSE: 0.3541

```
%% Fit Model on Training Set and Predict on Test Set
% Fit the Gaussian Process model on the full training data
model = fitrgp([nED_train, QED_train], eta_train);

% Predict eta values on the test set and compute error
eta_model_pred = predict(model, [nED_test, QED_test]);
averageModelError = mean(sqrt((eta_test - eta_model_pred).^2));
```