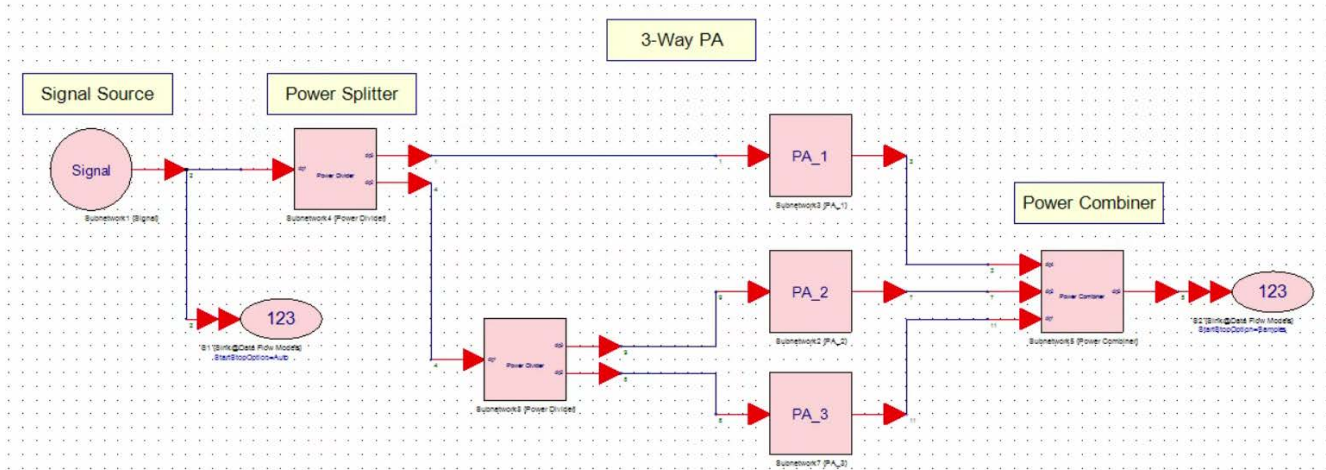


Supplementary Information for Final Lab

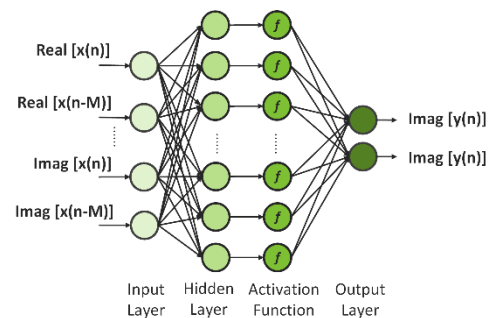
- **Practical Application Scenario – Nonlinear System with 3-Way PA**



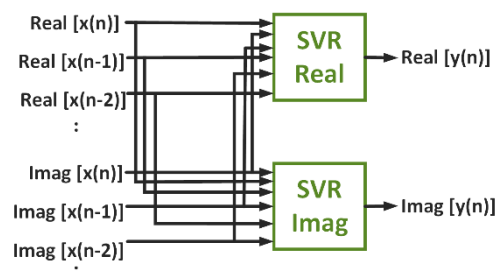
- **Behavioral Modeling Based on Machine Learning**

Behavioral modeling plays a critical role in predicting nonlinearity. Many models have been proposed, such as Volterra, memory polynomial (MP) models. To further enhance modeling accuracy, models based on machine learning are proposed, such as real-valued time-delay neural network (RVTDDNN), time-delay support vector regression (SVR) and so on.

Real-Valued Time-Delay Neural Network

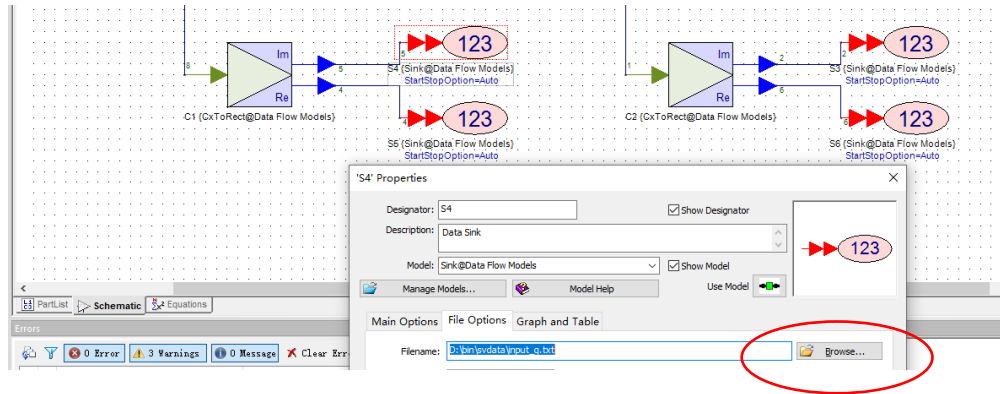


Time-Delay Support Vector Regression

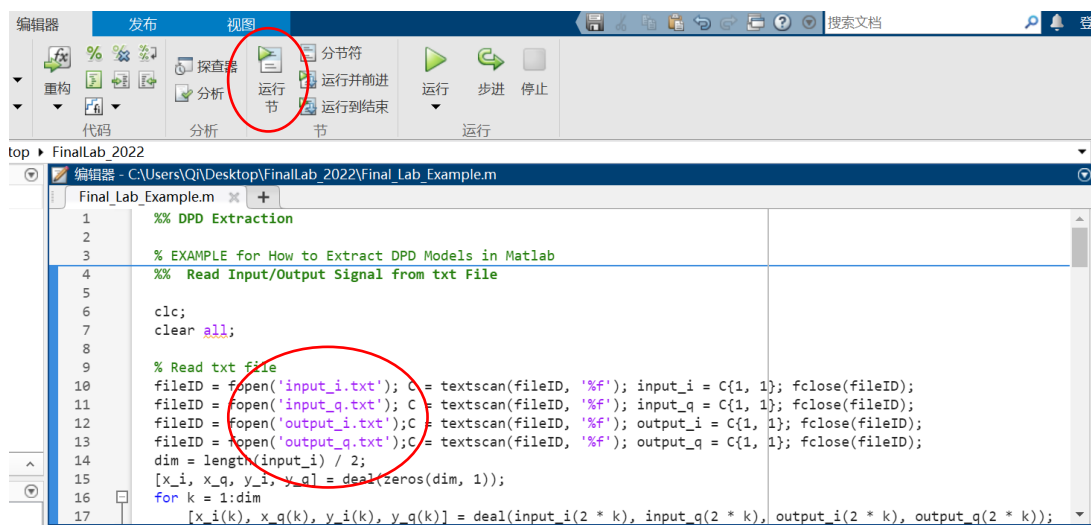


- **Obtain Input/Output Signals in *SystemVue***

Open "DPD Extraction" schematic, **Sink** is used to collect input signals and output signals into txt files for extracting DPD coefficients in Matlab. Choosing **file path** by click Browse, then clicking **Run Analyses**.



- **DPD Extraction in *Matlab***
- Read Input/Output Signal from txt File, Click **Run Section**.

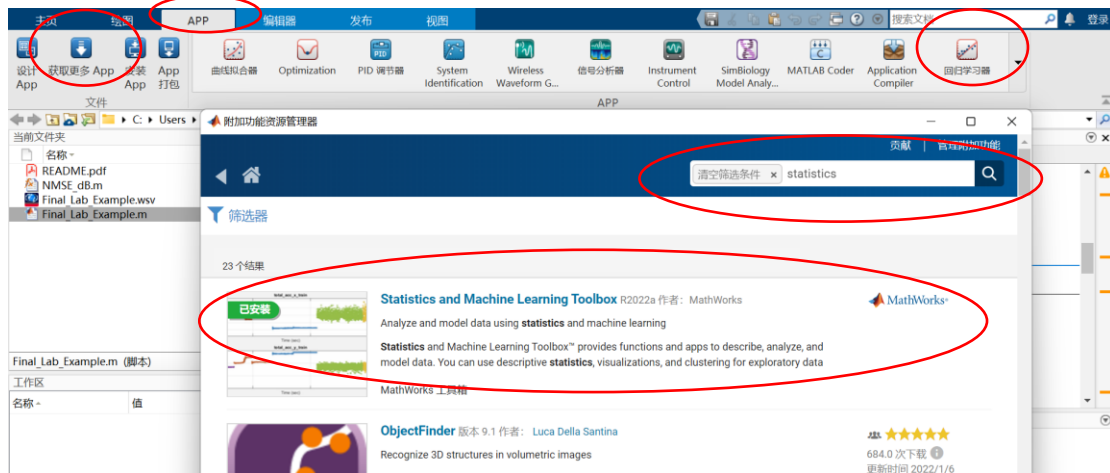


Using the function of **feedforwardnet**, which created a single hidden layer neural network. The number of neuron in hidden layer can be adjusted. (What other parameters of the model can be changed and what impact will changing them have?)

- Set Input/Output Matrix;
- Adjust Parameters and Train Models;
- Check Modeling Performance and Save Models;

- **Example II: Regression Learner**

If you have not installed **Regression Learner**. Click **Get More Apps**, search **Statistics and Machine Learning Toolbox** and install it.

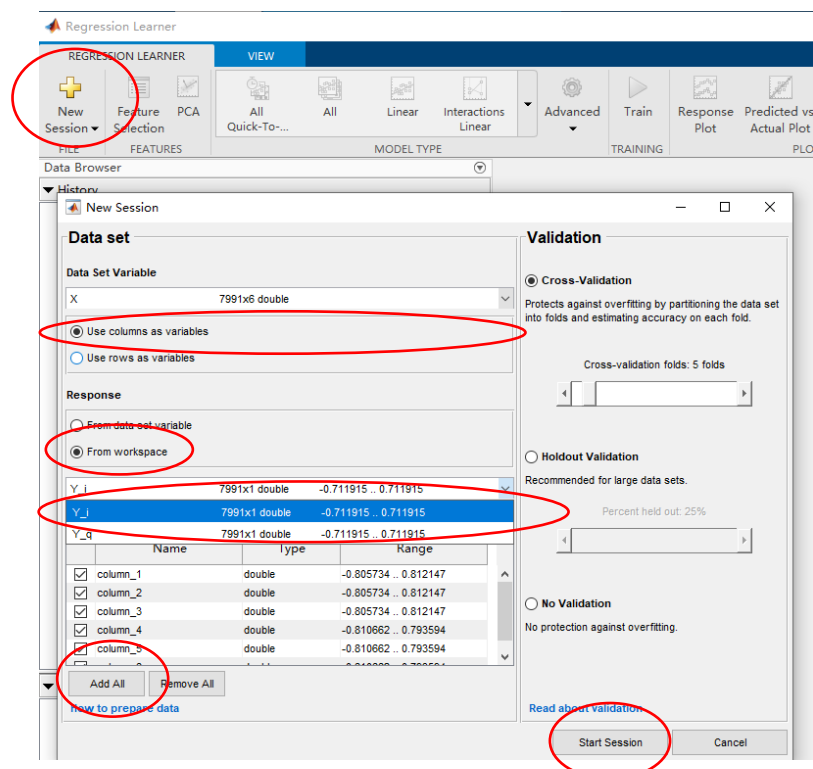


- Set Input/Output Matrix;
- Train Models in App and Save to Workspace;
- Check Modeling Performance and Save Models;

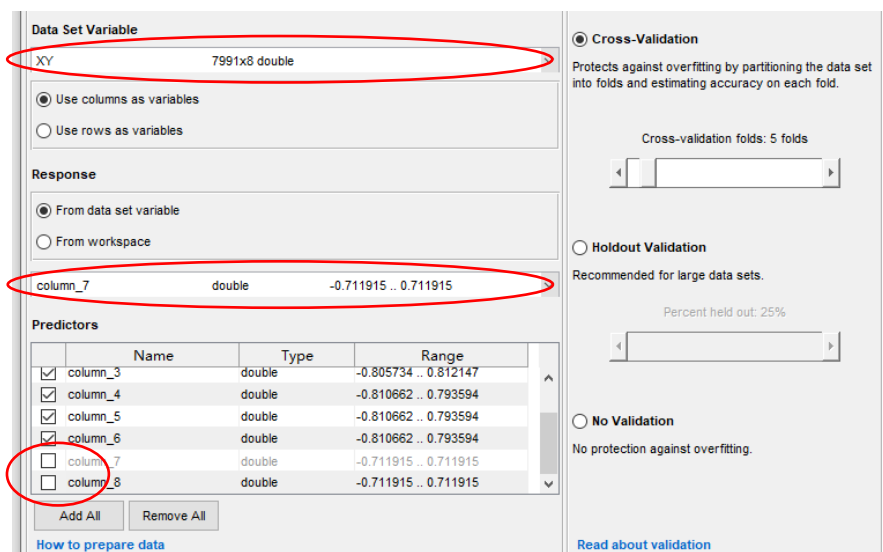
- **Train Models in App**

Train models in **Regression Learner**. Click New Session and select data set variable. Click Start Session. (Note: Two models need to be trained, one output is Y_i , the other output is Y_q .)

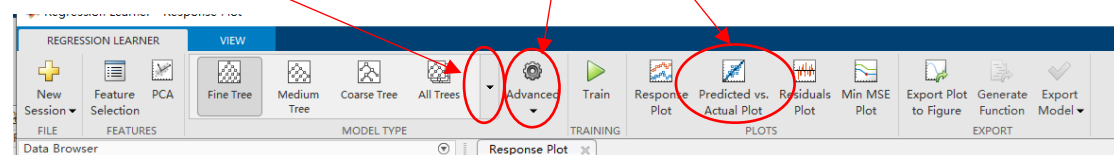
For R2020a or Newer Versions:



For R2019b or Earlier Version

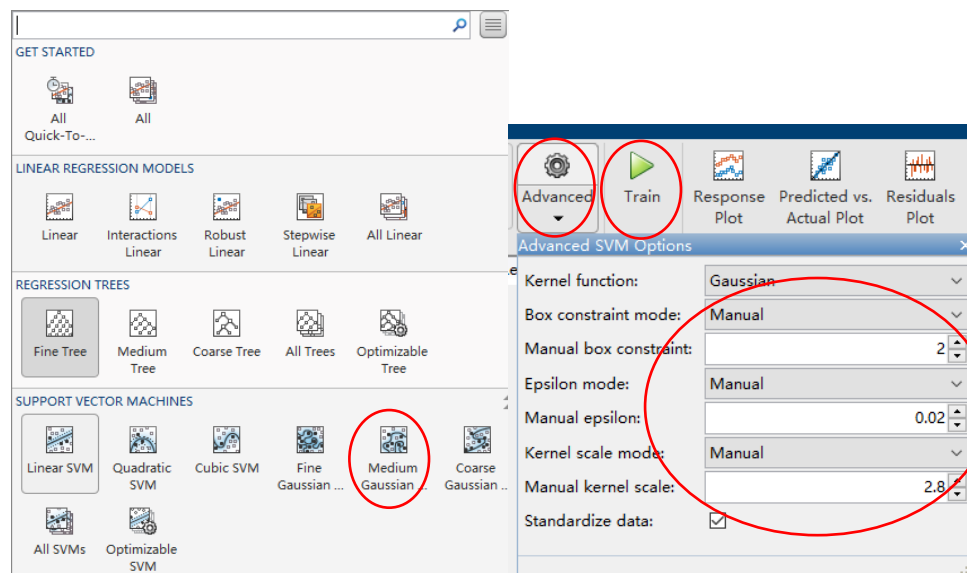


Choose different model. Adjust advanced options. Plot.

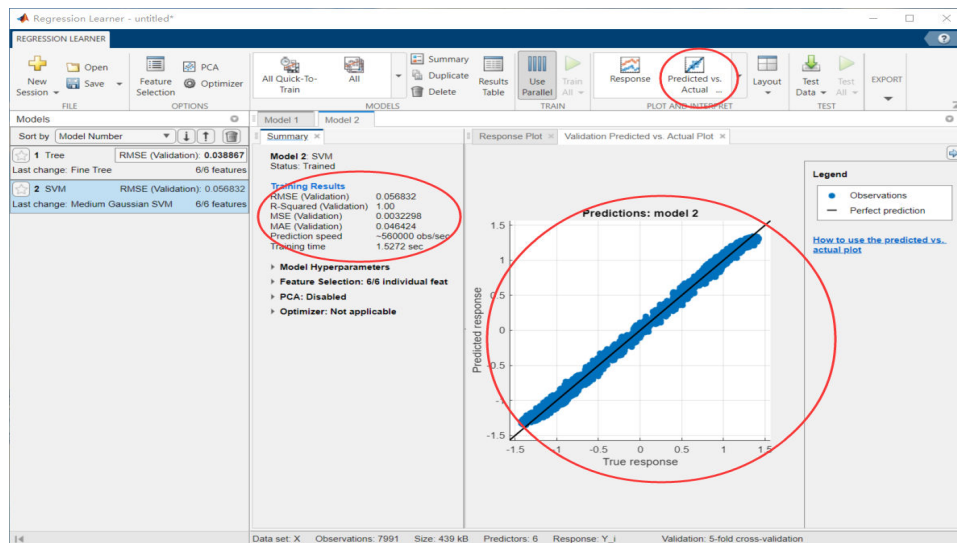


AN EXAMPLE using SVM

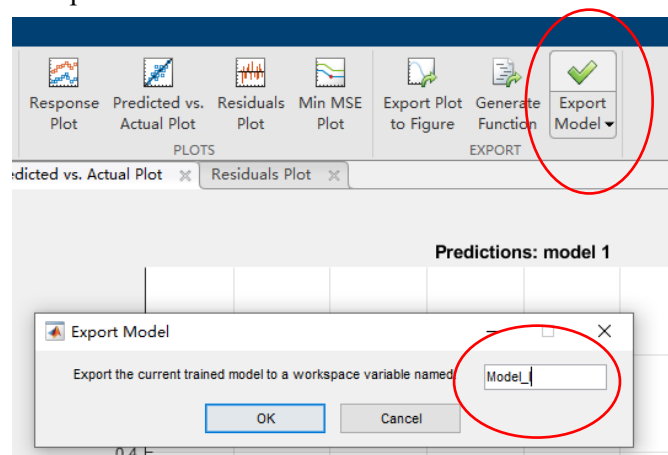
Select **Medium Gaussian SVM**, adjust advanced options as following graph. Click Train.



Check by RMSE and diagrams.



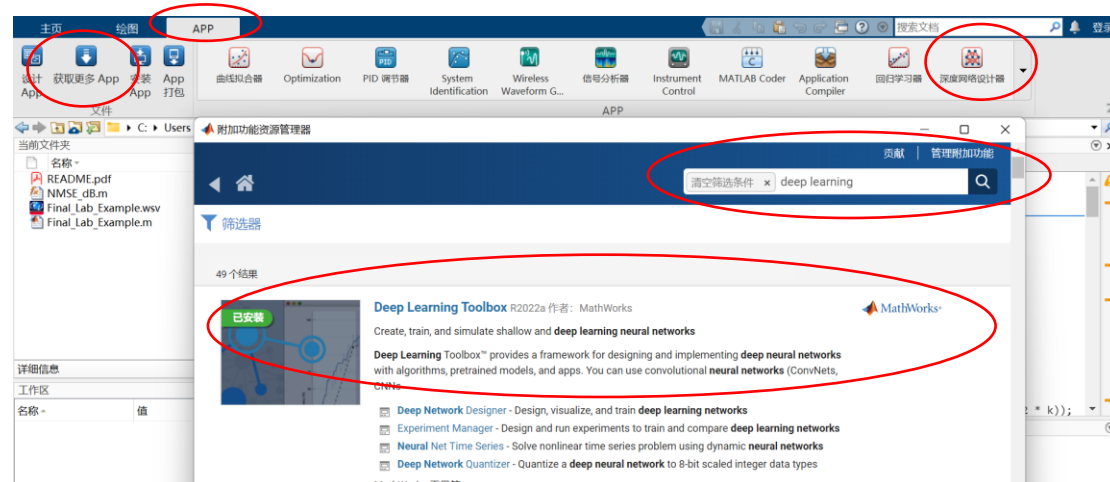
Export Model to Workspace.



Repeat the above steps, train another model of Y_q .

• Example III: Deep Network Designer

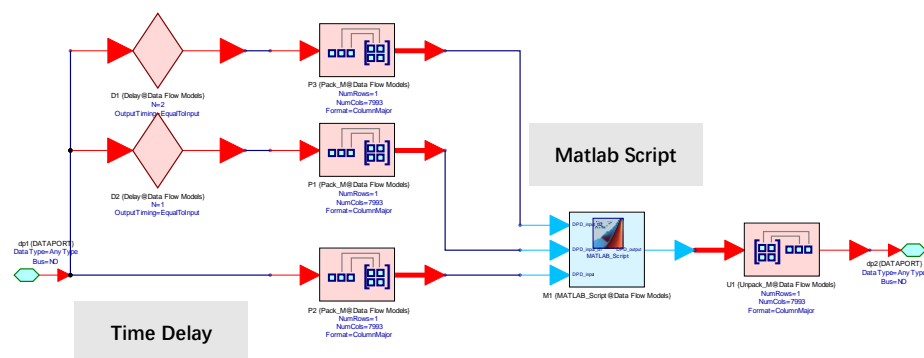
Using **Deep Network Designer**, which creates deep neural network. If you have not installed **Deep Network Designer**. Click **Get More Apps**, search **Deep Learning Toolbox** and install.



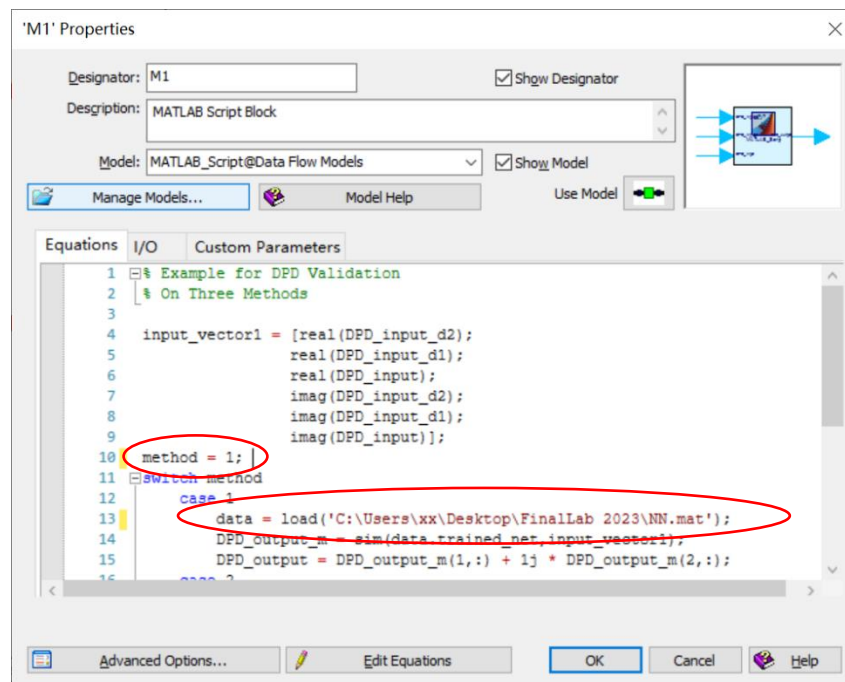
- Design Network and Set Input/Output Matrix;
- Adjust Parameters and Train Models;
- Check Modeling Performance and Save Models;

• DPD Validation in *SystemVue*

Open "DPD Validation" schematic, and open **DPD(Model)** schematic.



Set **method = 1** and change the file path in **Matlab Script**. Observe DPD results by the NMSE.



In total, we provide examples of 3 methods.

- method=1 Feedforward Neural Network;
- method=2 Regression Learner;
- method=3 Deep Network Designer;

Run **Example II & III** in *Matlab* as before first, and then change the number of **method** to 2 or 3 and the file path in Matlab script in "DPD Validation" schematic in *SystemVue*.

```
10 method = 1;
11 switch method
12     case 1
13         data = load('C:\Users\zhe\Desktop\FinalLab 2023\NN.mat');
14         DPD_output_m = sim(data.trained_net, input_vector1);
15         DPD_output = DPD_output_m(1,:) + 1j * DPD_output_m(2,:);
16     case 2
17         data = load('C:\Users\zhe\Desktop\FinalLab 2023\Model.mat');
18         input_vector = input_vector1;
19         DPD_output_i = data.Model_I.predictFcn(input_vector);
20         DPD_output_q = data.Model_Q.predictFcn(input_vector);
21         DPD_output = DPD_output_i(:) + 1j * DPD_output_q(:);
22         DPD_output = DPD_output.';
23     case 3
24         data = load('C:\Users\zhe\Desktop\FinalLab 2023\DeepNN.mat');
25         DPD_output_m = predict(data.DeepNN, input_vector1);
```

Tips

- We recommend that you use the **Copyrighted Version** of **MATLAB**, otherwise you may encounter problems. Most errors are caused by the wrong file path of the model.
- Verification of **DPD performance** in **SystemVue** may take a long time. When optimizing the parameters of the model, you can refer to the **modeling performance** calculated in **MATLAB**. In general, the two results are very close.
- Different models have different performance, so **try more**. Instead of using the example method, we would like to see your **own method**, such as, GAN, LSTM, and so on.
- The input of the model can be changed. Using more input information may get better performance. Which means you can change the **input matrix**.
- Have difficulties? Ask for help in the [MATLAB official Help](#) or ask **TAs**.
- Good luck to get excellent results.