Building Apps in MATLAB: App Designer for Pdm

Kantika Wongkasem
Application Engineer
Ascendas Systems Co.,Ltd

R2024b

MathWorks®
Authorized Reseller

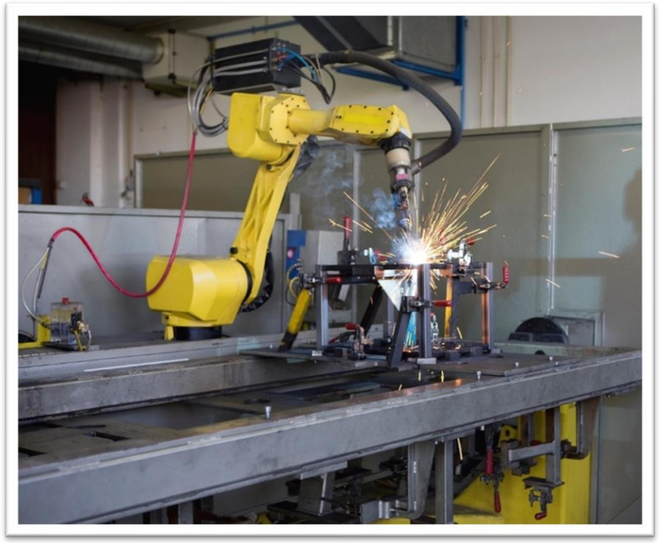
Agenda

- **Recap Predictive Maintenance Concepts**
  - Preprocessing data
  - Modelling (Application predictive)
  - Deployment
- **Creating with App Designer**
  - Overview
  - Creating User Interface
  - Adding Callbacks and Properties

Journey 2: Data, data, everywhere…

Goal | Data | Approach | Result

- **Anomaly Detection:** Detect when the machine deviates from normal operation.
- Avoid surprises. Address anomalies before catastrophic failure occurs.



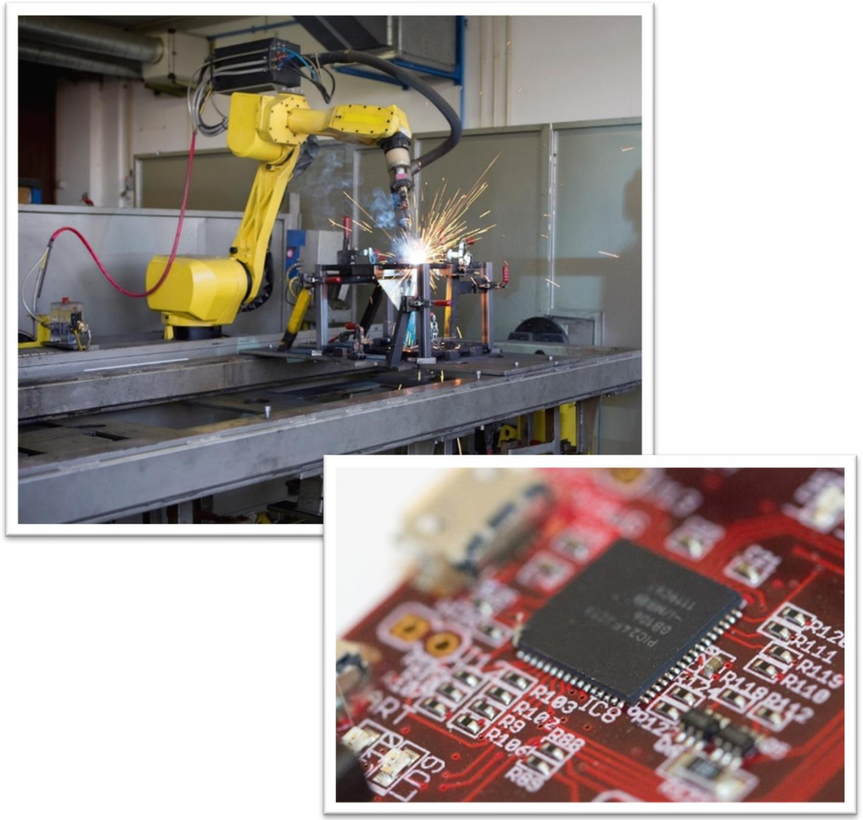
Currently

- Routine monthly maintenance
- Not many failures
- But when failures do happen…

Journey 2: Data, data, everywhere…

Goal | Data | Approach | Result

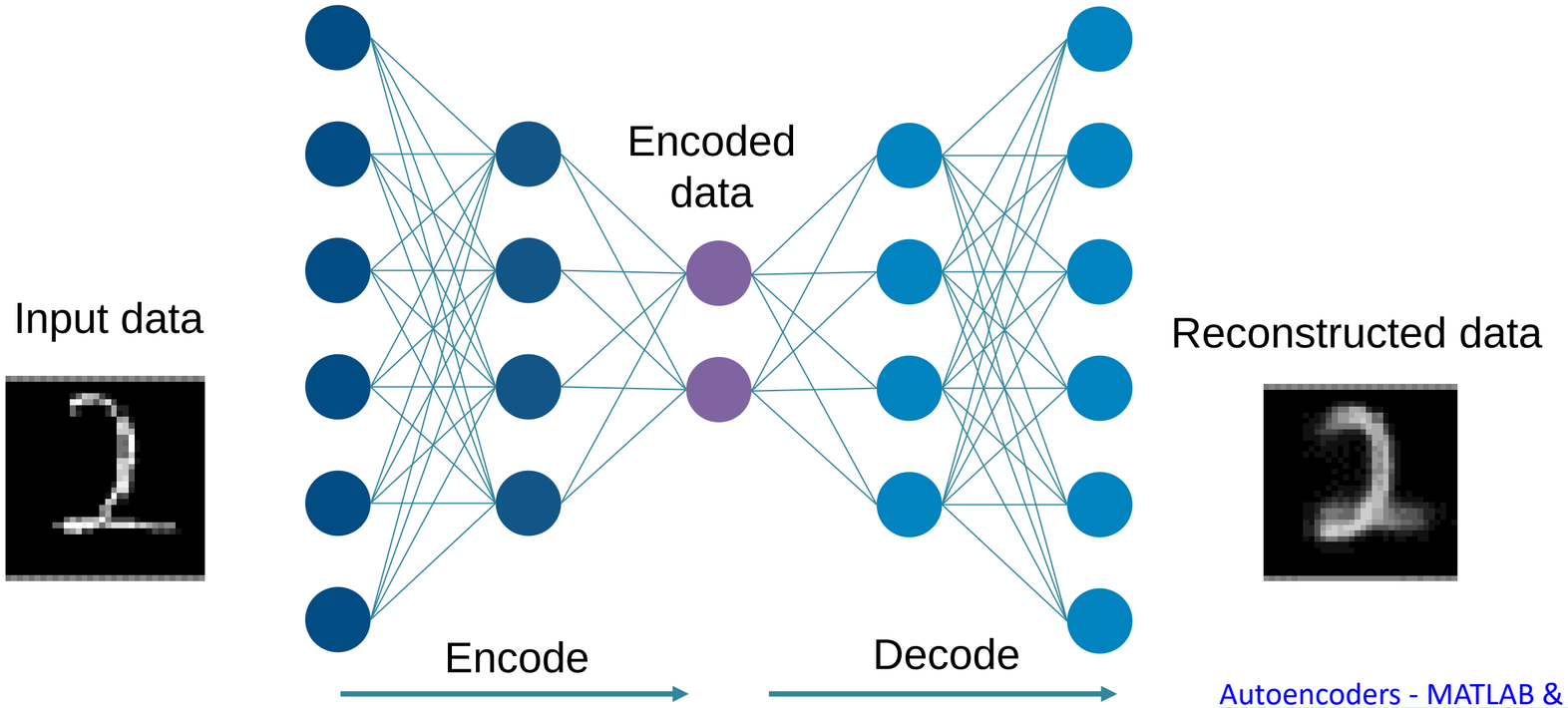
- Vibration data from 3-axis accelerometers
- Labeled “before” and “after” maintenance
  - “After” data = Normal ✓
  - “Before” data = Not sure ?
- Some data tagged as “abnormal” by maintenance crews



Journey 2: Data, data, everywhere…

Goal | Data | Approach | Result

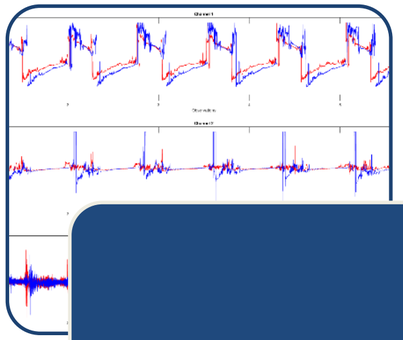
Autoencoder



[Autoencoders - MATLAB & Simulink]

Journey 2: Data, data, everywhere…

Goal | Data | Approach | Result

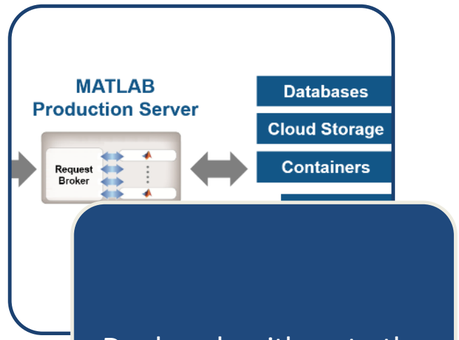


Access data from files

Extract and rank features with Diagnostic Feature Designer App

```
he biLSTM network layers
s = [ sequenceInputLayer(featureDimension, 'Name',
ilstmLayer(16, 'Name', 'bilstm1')
eluLayer('Name', 'relu1')
ilstmLayer(32, 'Name', 'bilstm2')
eluLayer('Name', 'relu2')
ilstmLayer(16, 'Name', 'bilstm3')
eluLayer('Name', 'relu3')
ullyConnectedLayer(featureDimension, 'Name', 'fc')
egressionLayer('Name', 'out') ];

 Training
ns = tra
Plots',
MiniBatc
MaxEpoch

trainNe
```

Train autoencoder on normal data



Deploy algorithms to the cloud

MATLAB R2021a

HOME PLOTS APPS LIVE EDITOR INSERT VIEW

Search Documentation

New Open Save Compare Print Export FILE

Go To Find Bookmark NAVIGATE

Text Title B I U M TEXT

Code Control Task Refactor CODE

Run Section Section Break Run and Advance Run to End SECTION

Run Step Stop RUN

C: ▸ Work ▸ Local Demos ▸ anomaly-detection-using-autoencoders ▸

Live Editor - C:\Work\Local Demos\anomaly-detection-using-autoencoders\Part01_DataPrepFeatureExtraction.mlx

Part01_DataPrepFeatureExtraction.mlx

Part 1: Data Preparation and Feature Extraction

Industrial Machinery Anomaly Detection

Table of Contents

Load Data

```
1  load("IndustrialMachineData.mat")
```

Visualize Data Before and After Maintenance

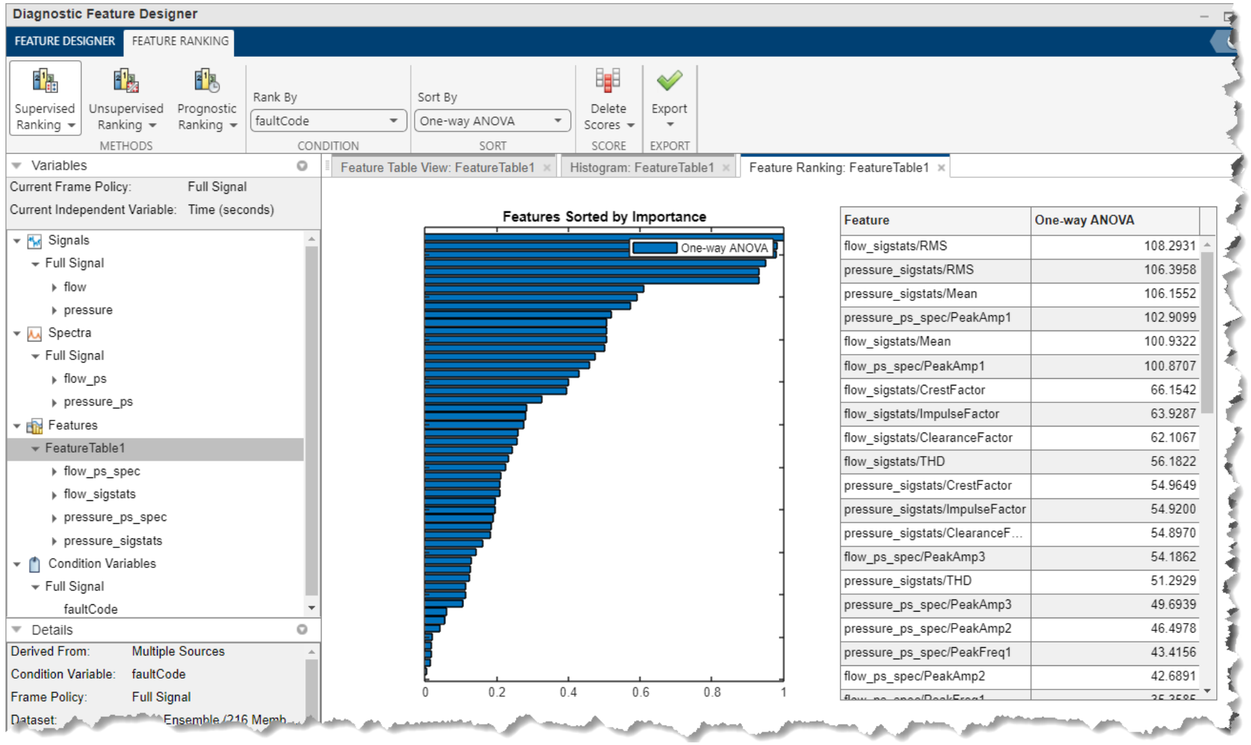
Visualize data before and after maintenance across channels for one member of the ensemble

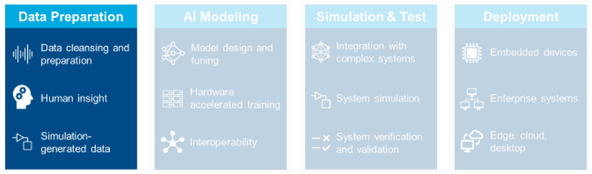
Project - Anomaly Detection Industrial Machine

UTF-8 LF script

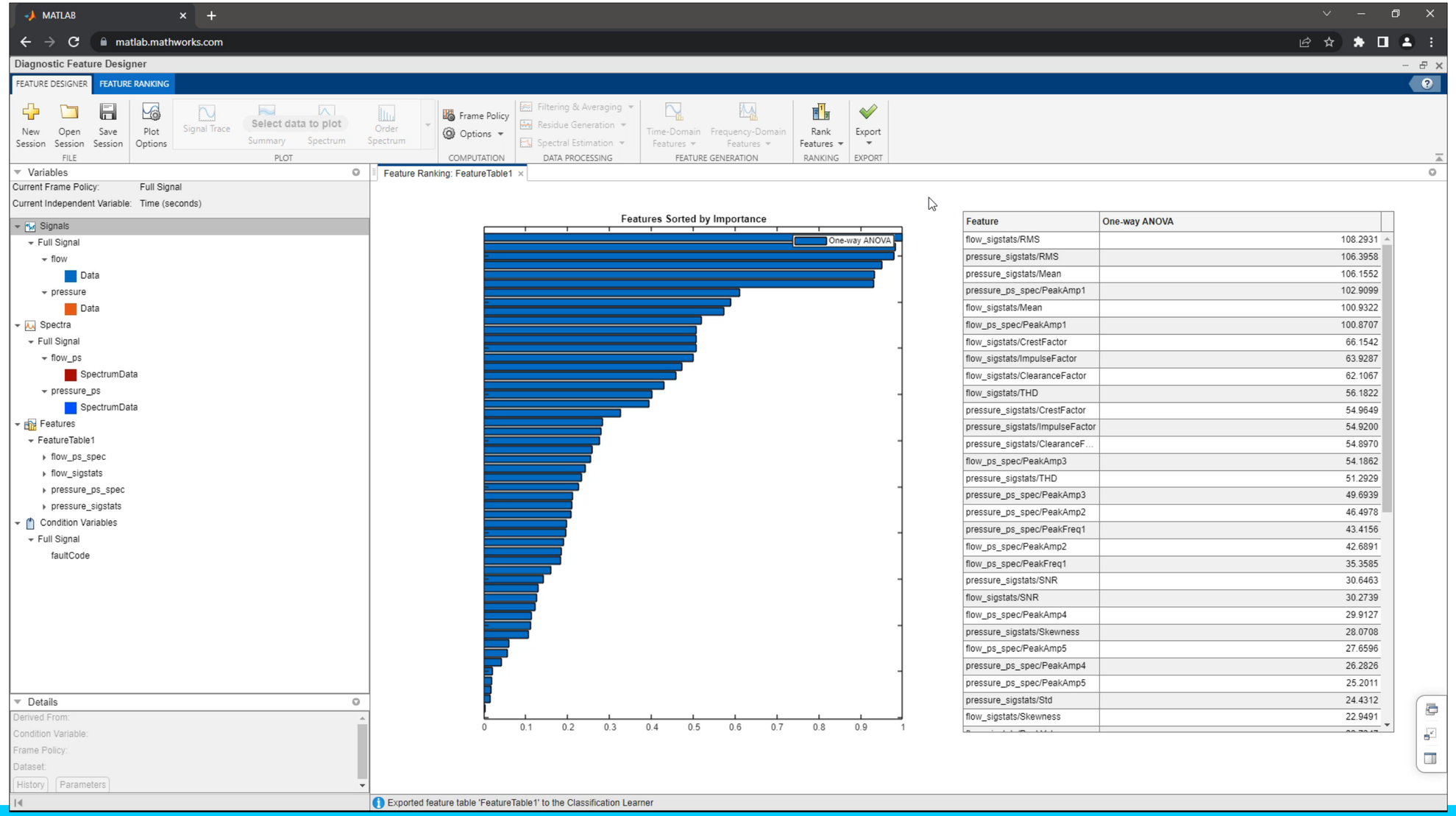
Overview: Feature Extraction

- Purpose
  - Interactively extract the best features for training a machine learning model for fault classification
- What you will learn
  - Extract features both in the time and frequency domain
  - Rank the extracted features and use them to train machine learning models





Next – Train Classification Models using Classification Learner app



Feature	One-way ANOVA
flow_sigstats/RMS	108.2931
pressure_sigstats/RMS	106.3958
pressure_sigstats/Mean	106.1552
pressure_ps_spec/PeakAmp1	102.9099
flow_sigstats/Mean	100.9322
flow_ps_spec/PeakAmp1	100.8707
flow_sigstats/CrestFactor	66.1542
flow_sigstats/ImpulseFactor	63.9287
flow_sigstats/ClearanceFactor	62.1067
flow_sigstats/THD	56.1822
pressure_sigstats/CrestFactor	54.9649
pressure_sigstats/ImpulseFactor	54.9200
pressure_sigstats/ClearanceF…	54.8970
flow_ps_spec/PeakAmp3	54.1862
pressure_sigstats/THD	51.2929
pressure_ps_spec/PeakAmp3	49.6939
pressure_ps_spec/PeakAmp2	46.4978
pressure_ps_spec/PeakFreq1	43.4156
flow_ps_spec/PeakAmp2	42.6891
flow_ps_spec/PeakFreq1	35.3585
pressure_sigstats/SNR	30.6463
flow_sigstats/SNR	30.2739
flow_ps_spec/PeakAmp4	29.9127
pressure_sigstats/Skewness	28.0708
flow_ps_spec/PeakAmp5	27.6596
pressure_ps_spec/PeakAmp4	26.2826
pressure_ps_spec/PeakAmp5	25.2011
pressure_sigstats/Std	24.4312
flow_sigstats/Skewness	22.9491

LG Energy Solution Uses Deep Learning for Predictive Maintenance on Industrial Cutter

Challenge

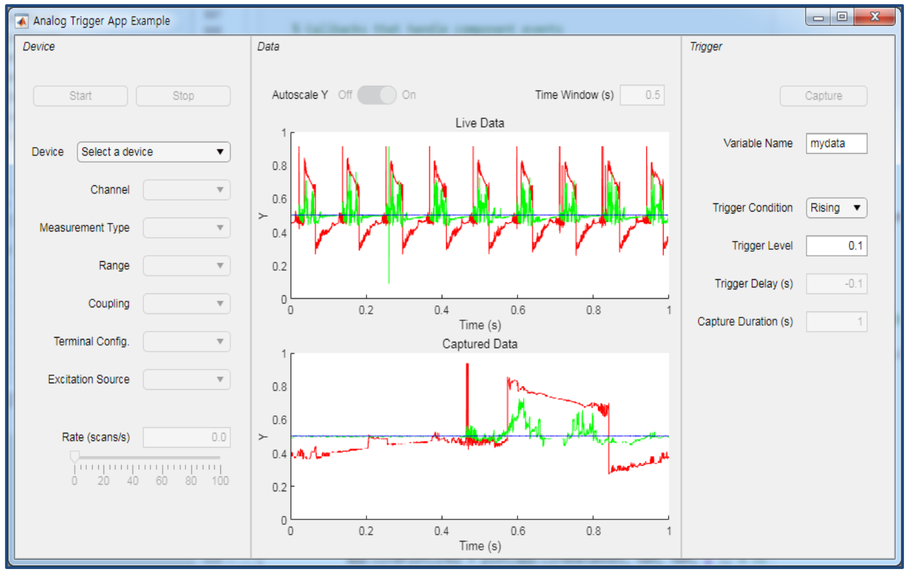
Eliminate reliance on the site engineer's opinion for scheduling predictive maintenance, which can sometimes be too conservative

Solution

Develop a condition monitoring system and deploy standalone executable that acquires raw data from an NI device, makes predictions, and displays the result.

Key Outcomes

- Generated features and trained various AI models using interactive apps
- Developed entire workflow from data acquisition to deployment
- Accelerated prototyping using MathWorks engineering support



Condition monitoring system using deep learning

"Three advantages of MATLAB that led our project to success: app-based AI development workflow, compatibility with third-party hardware, and short test cycle with rapid prototyping."

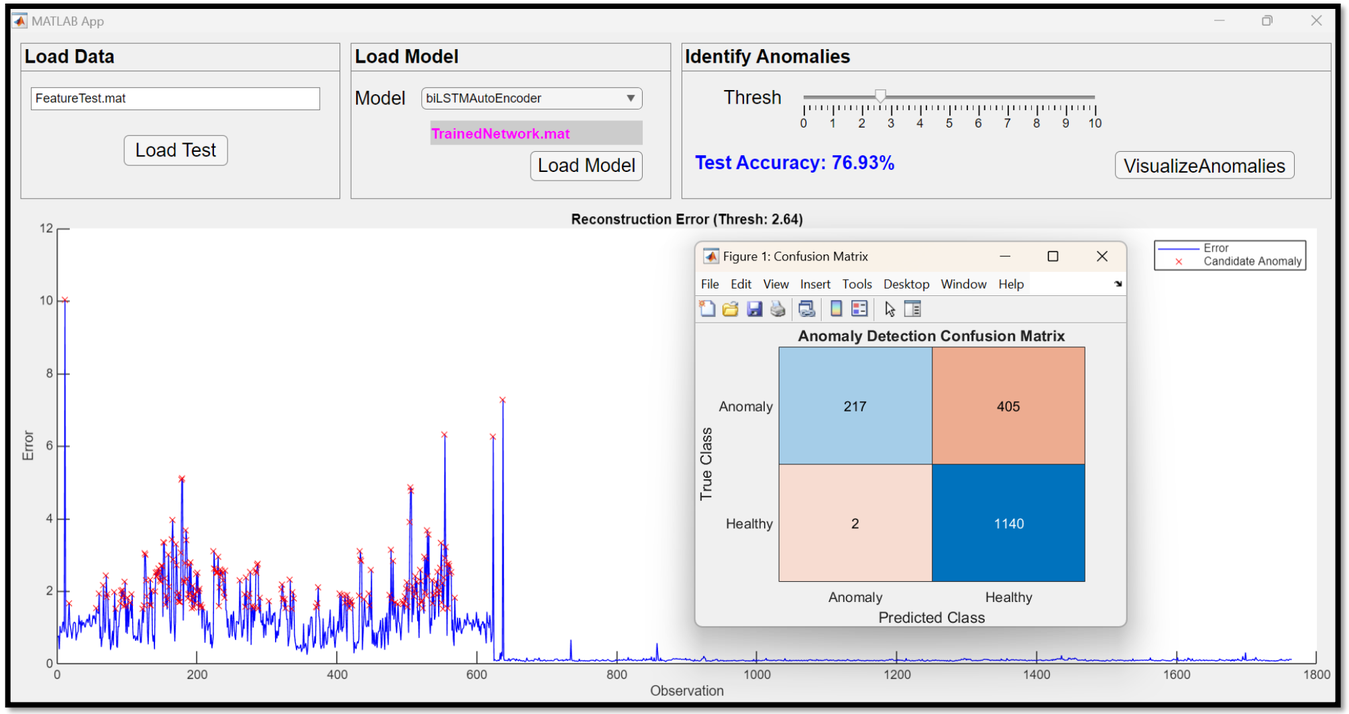
– Junghoon Lee, LG Energy Solution

» Explore MATLAB for deep learning

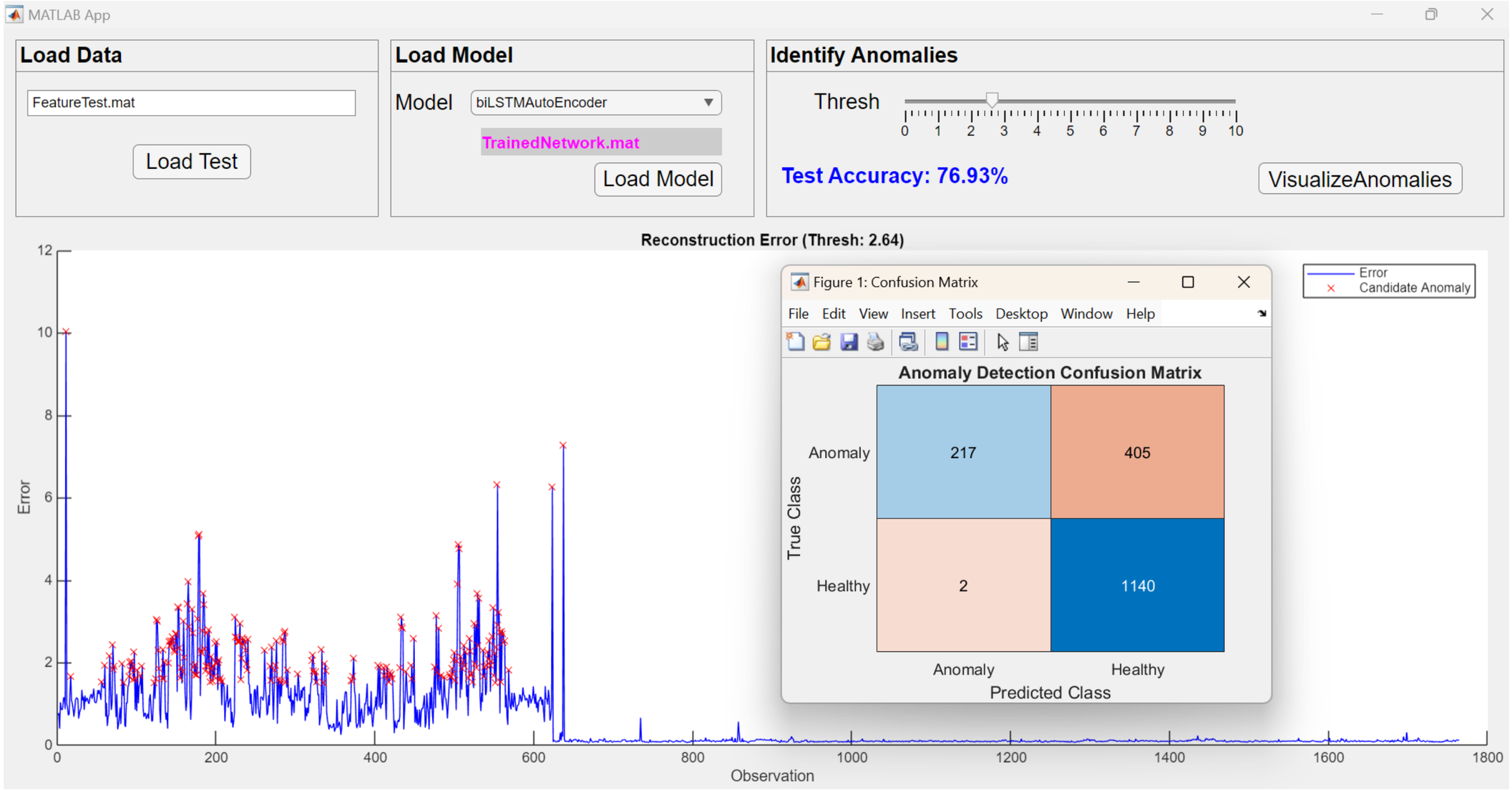
Agenda

- **Creating with App Designer**
  - Overview
  - Creating User Interface
  - Adding Callbacks and Properties

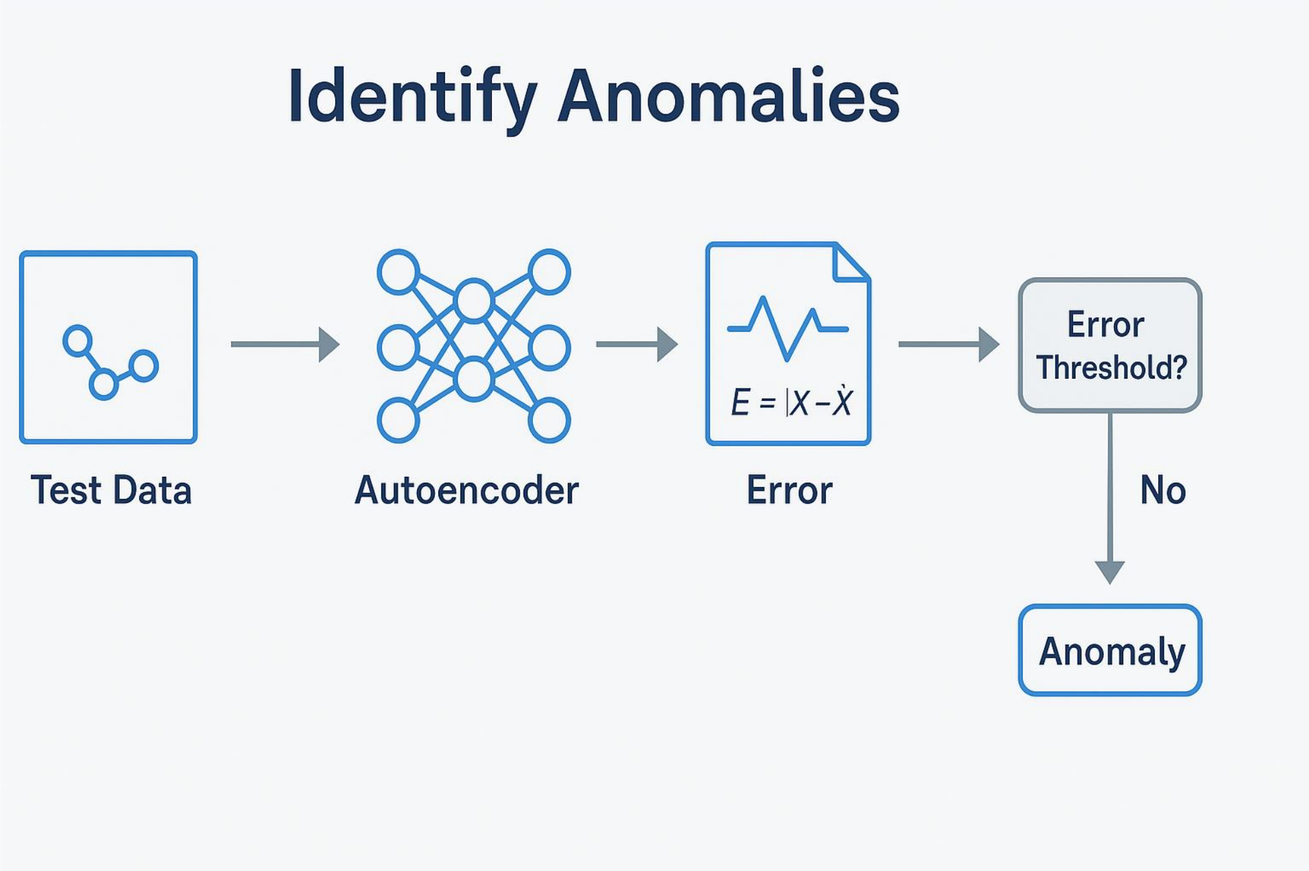
Create → Package → Share

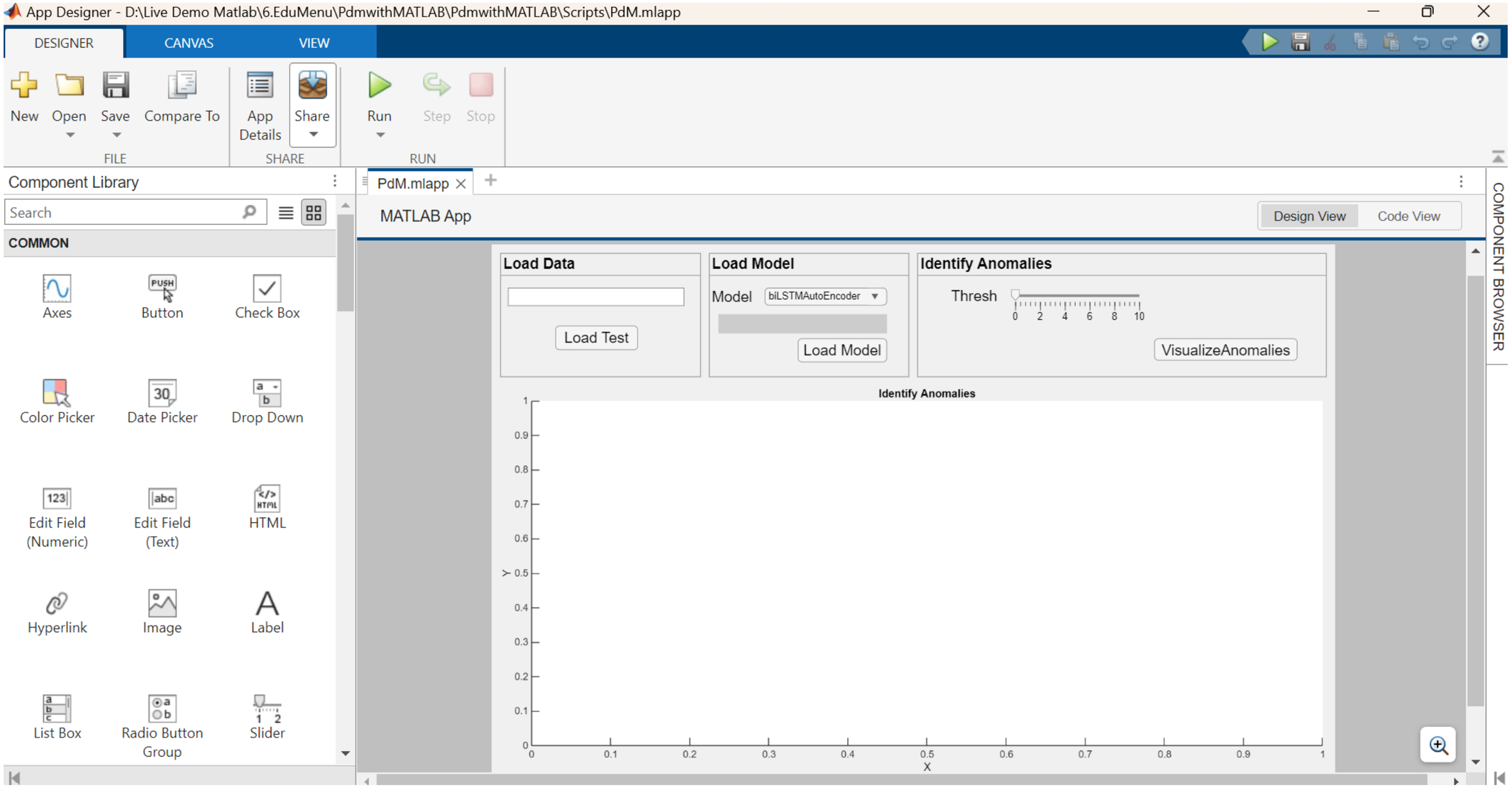


Demo- Anomaly Detection App Designer

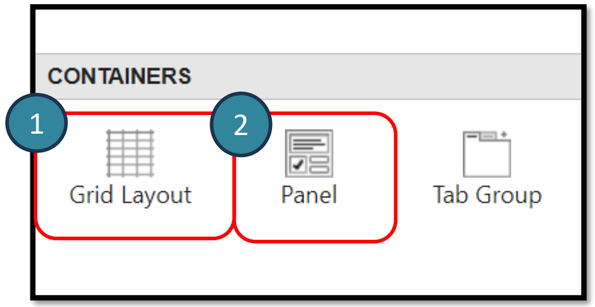


Concept for this App

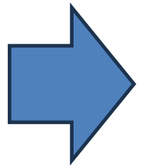


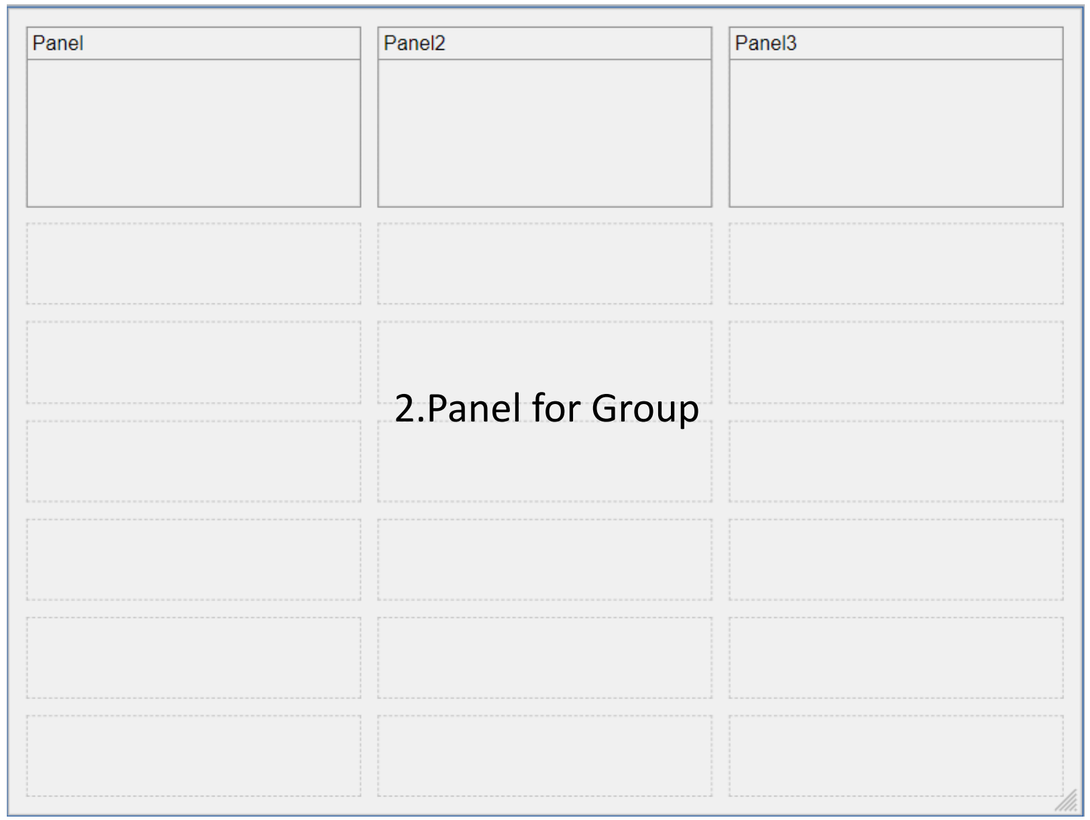

App Designer - D:\Live Demo Matlab\6.EduMenu\PdmwithMATLAB\PdmwithMATLAB\Scripts\PdM.mlapp
DESIGNER
CANVAS
VIEW
New
Open
Save
Compare To
FILE
App Details
Share
SHARE
Run
Step
Stop
RUN
Component Library
Search
COMMON
Axes
Button
Check Box
Color Picker
Date Picker
Drop Down
Edit Field (Numeric)
Edit Field (Text)
HTML
Hyperlink
Image
Label
List Box
Radio Button Group
Slider
PdM.mlapp
MATLAB App
Design View
Code View
Load Data
Load Test
Load Model
Model
biLSTMAutoEncoder
Load Model
Identify Anomalies
Thresh
VisualizeAnomalies
Identify Anomalies
COMPONENT BROWSER

Step 1

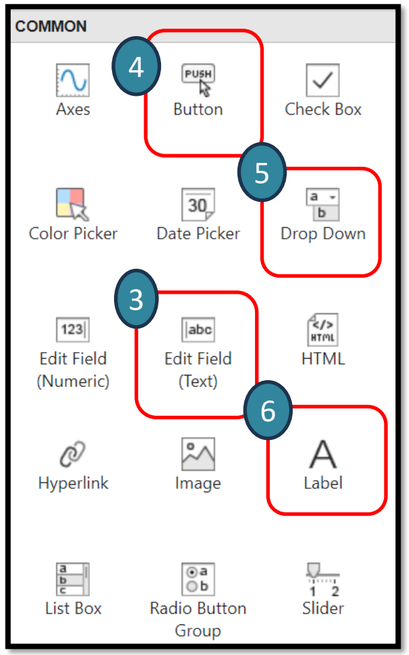

CONTAINERS
1
2
Grid Layout
Panel
Tab Group

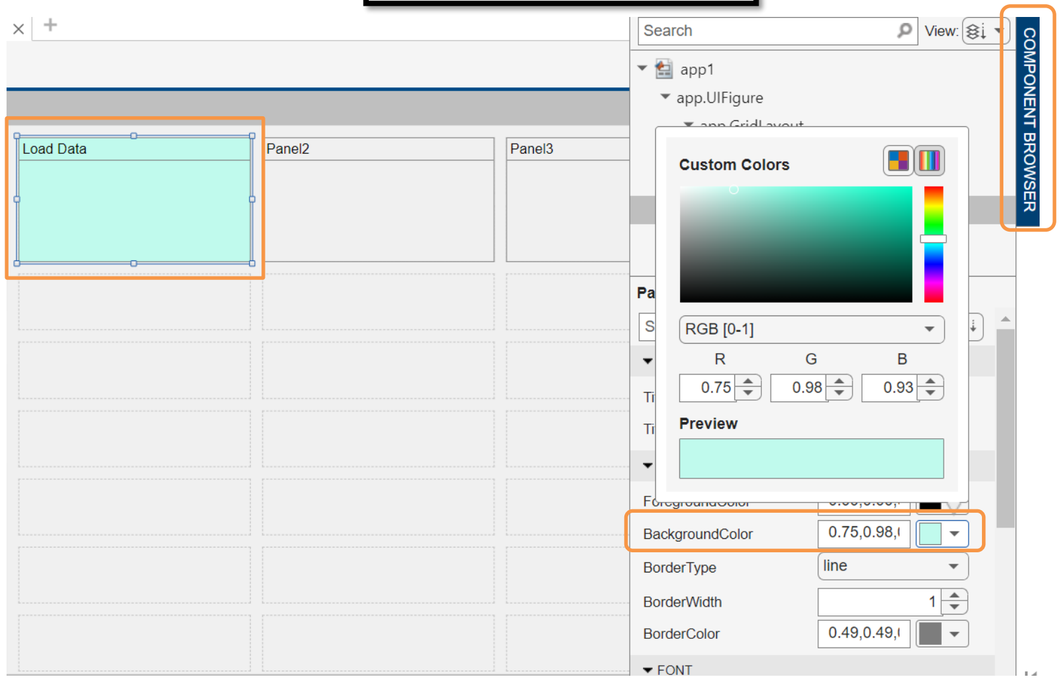

1.Grid Layout

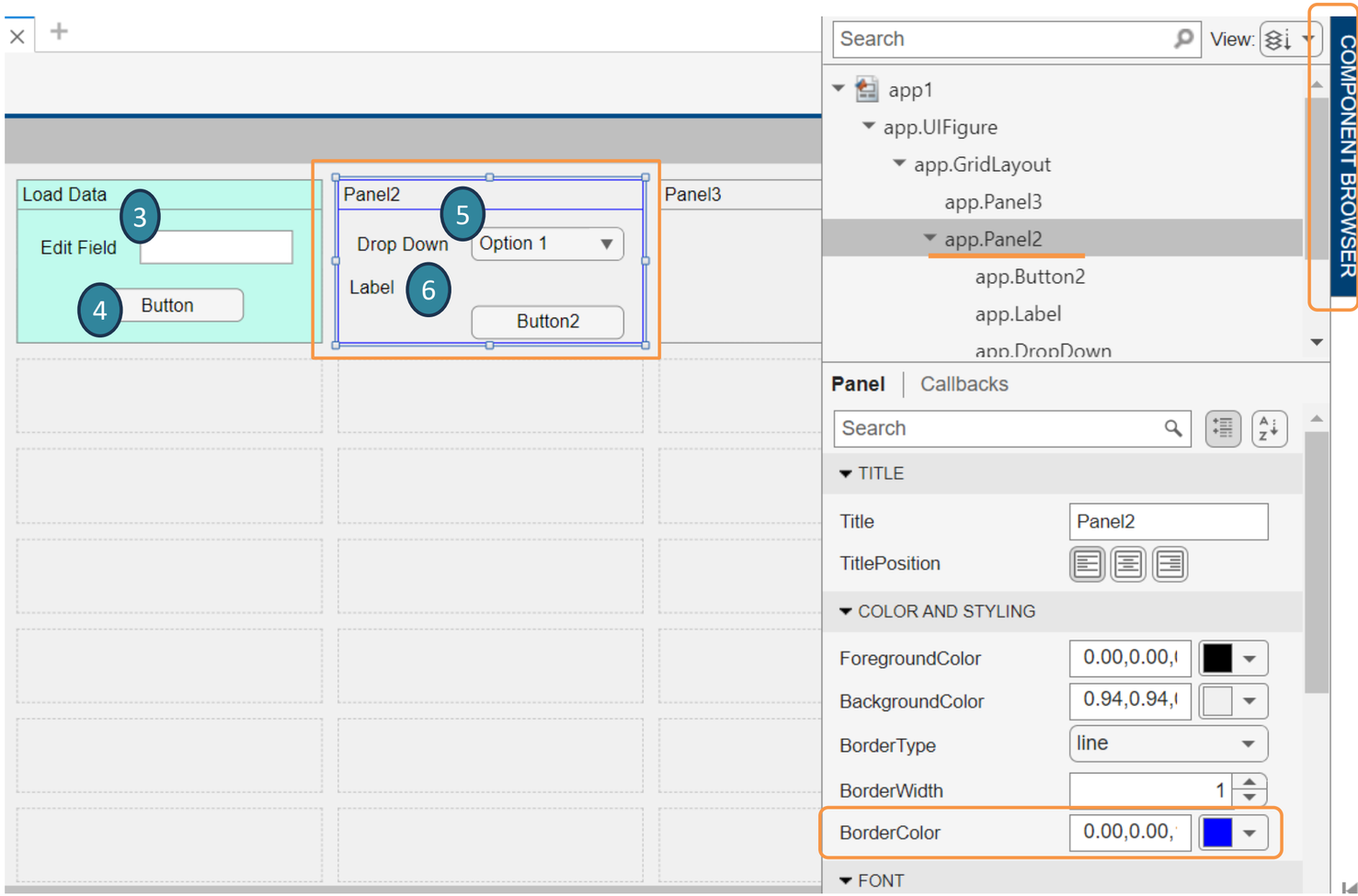



Panel
Panel2
Panel3
2.Panel for Group

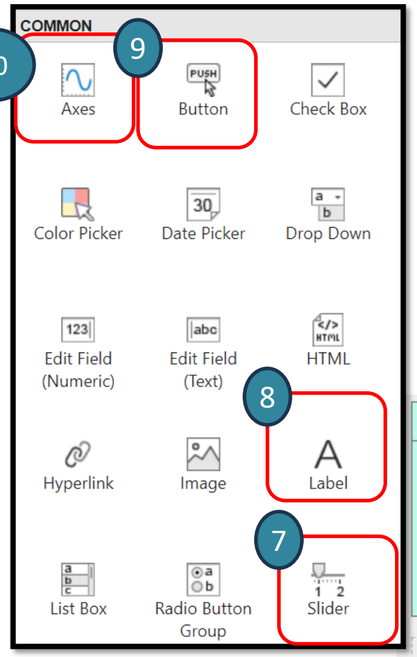
Step 2

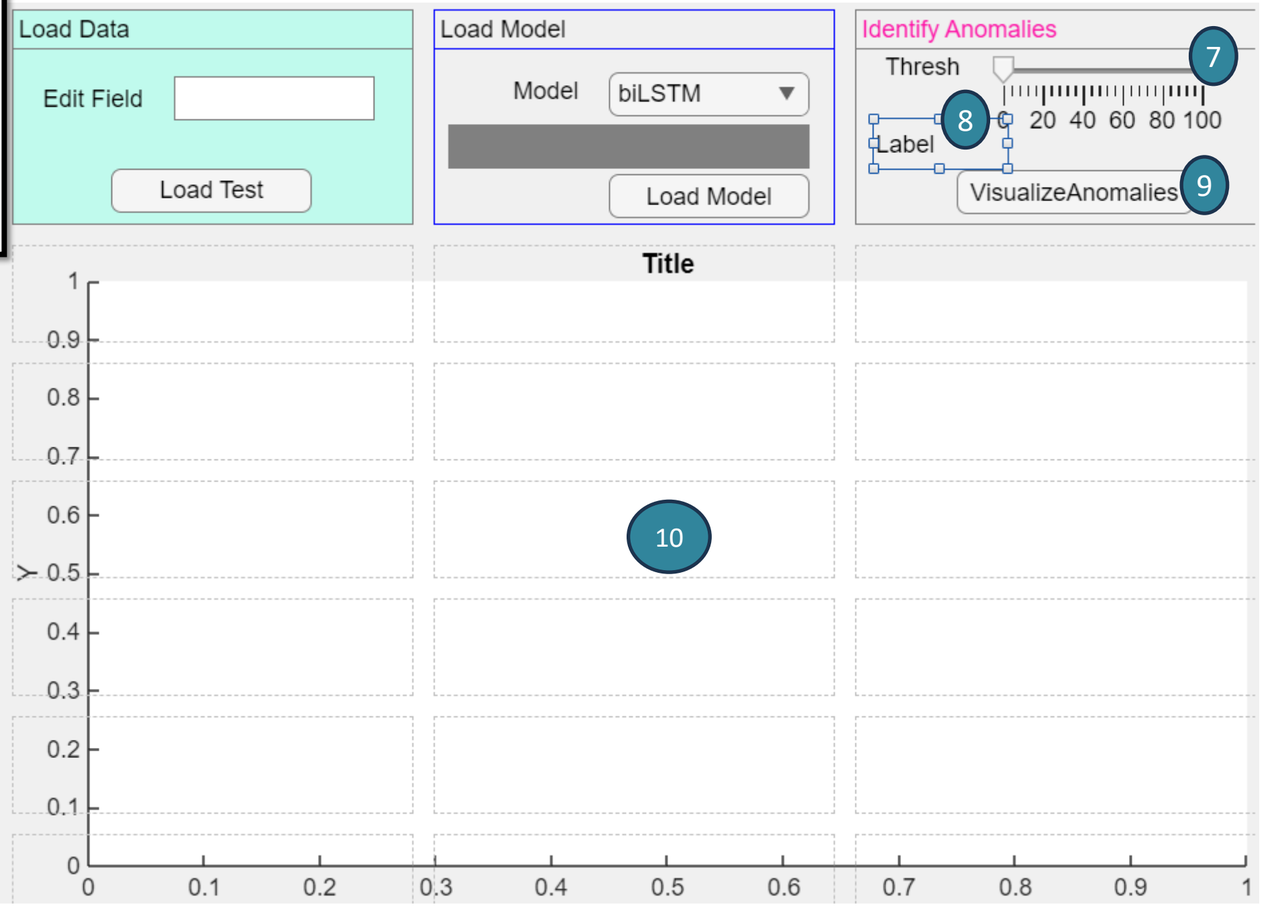

COMMON
Axes
Button
Check Box
Color Picker
Date Picker
Drop Down
Edit Field (Numeric)
Edit Field (Text)
HTML
Hyperlink
Image
Label
List Box
Radio Button Group
Slider
4
5
3
6

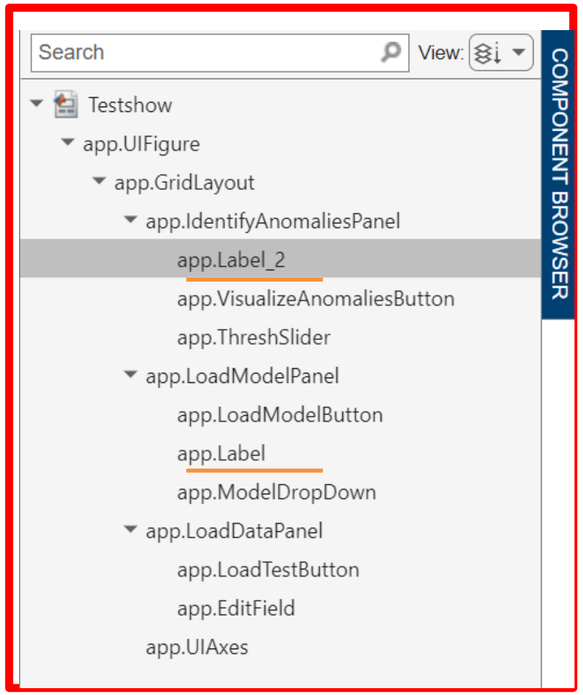

Load Data
Panel2
Panel3
Search
View:
app1
app.UIFigure
COMPONENT BROWSER
Custom Colors
RGB [0-1]
R
G
B
0.75
0.98
0.93
Preview
BackgroundColor
0.75,0.98,
BorderType
line
BorderWidth
1
BorderColor
0.49,0.49,
FONT


Load Data
Edit Field
Button
Panel2
Drop Down
Option 1
Label
Button2
Panel3
3
4
5
6
Search
View:
app1
app.UIFigure
app.GridLayout
app.Panel3
app.Panel2
app.Button2
app.Label
app.DropDown
COMPONENT BROWSER
Panel
Callbacks
TITLE
Title
Panel2
TitlePosition
COLOR AND STYLING
ForegroundColor
0.00,0.00,
BackgroundColor
0.94,0.94,
BorderType
line
BorderWidth
1
BorderColor
0.00,0.00,
FONT

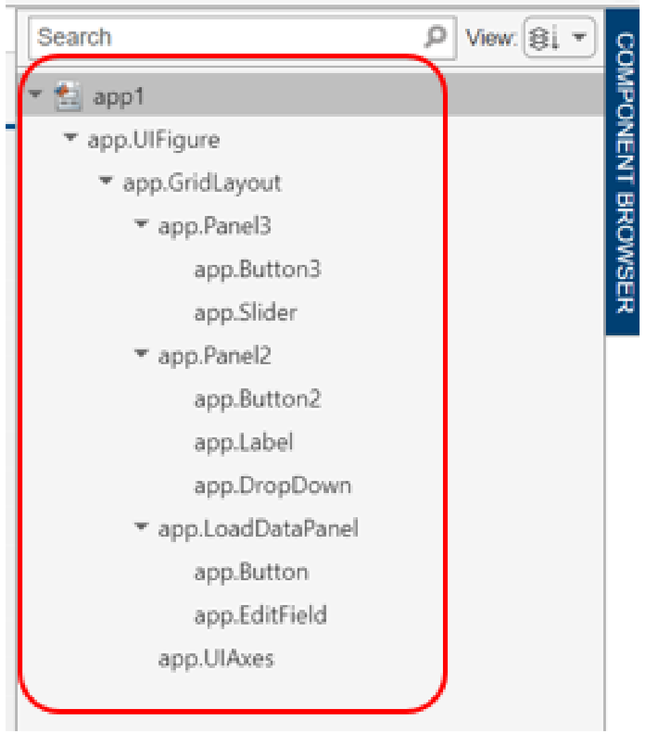
Step3

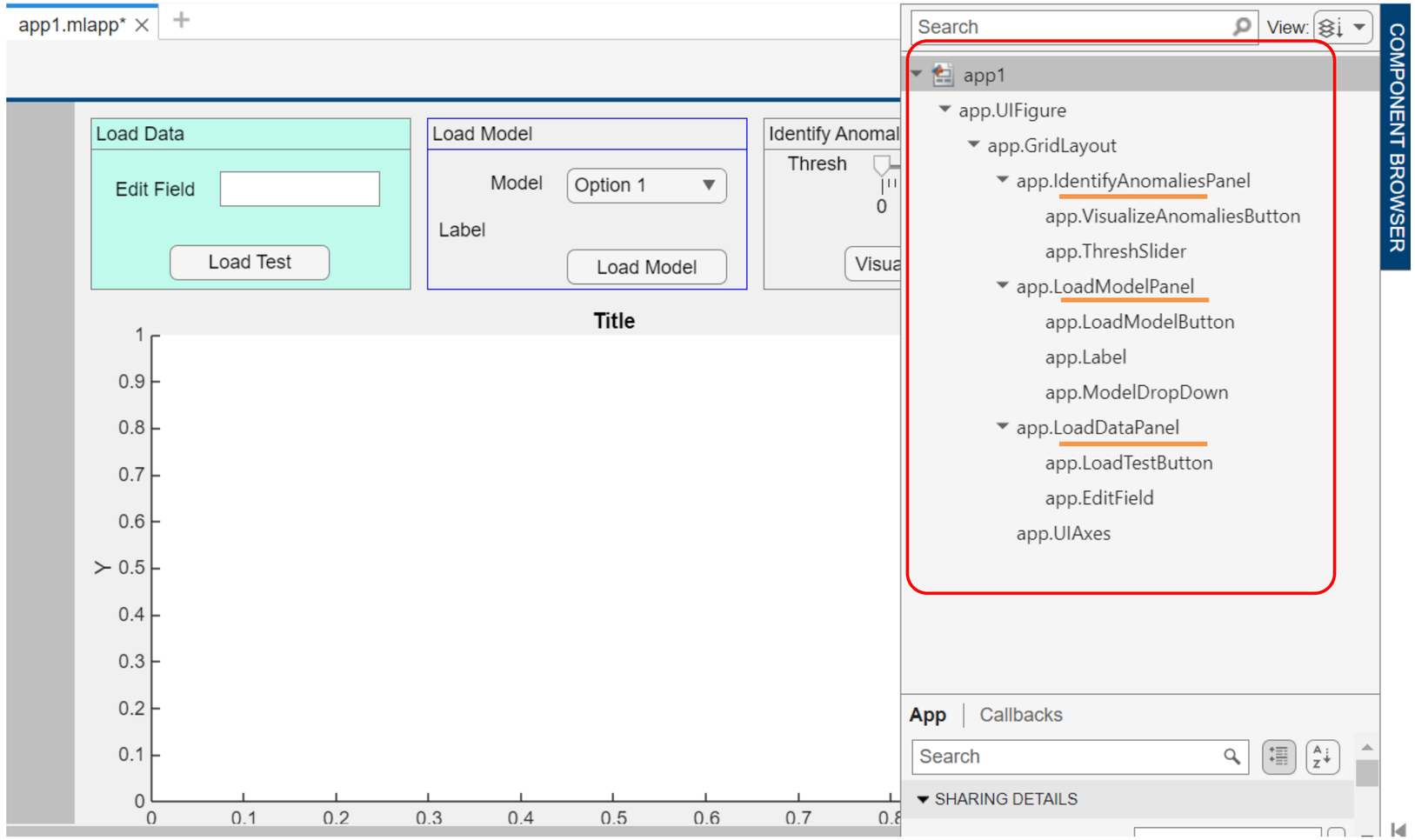

COMMON
10
9
Axes
Button
Check Box
Color Picker
Date Picker
Drop Down
Edit Field (Numeric)
Edit Field (Text)
HTML
8
Hyperlink
Image
Label
7
List Box
Radio Button Group
Slider


Load Data
Edit Field
Load Test
Load Model
Model
biLSTM
Load Model
Identify Anomalies
Thresh
7
8
Label
20 40 60 80 100
VisualizeAnomalies
9
Title
10
Y
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

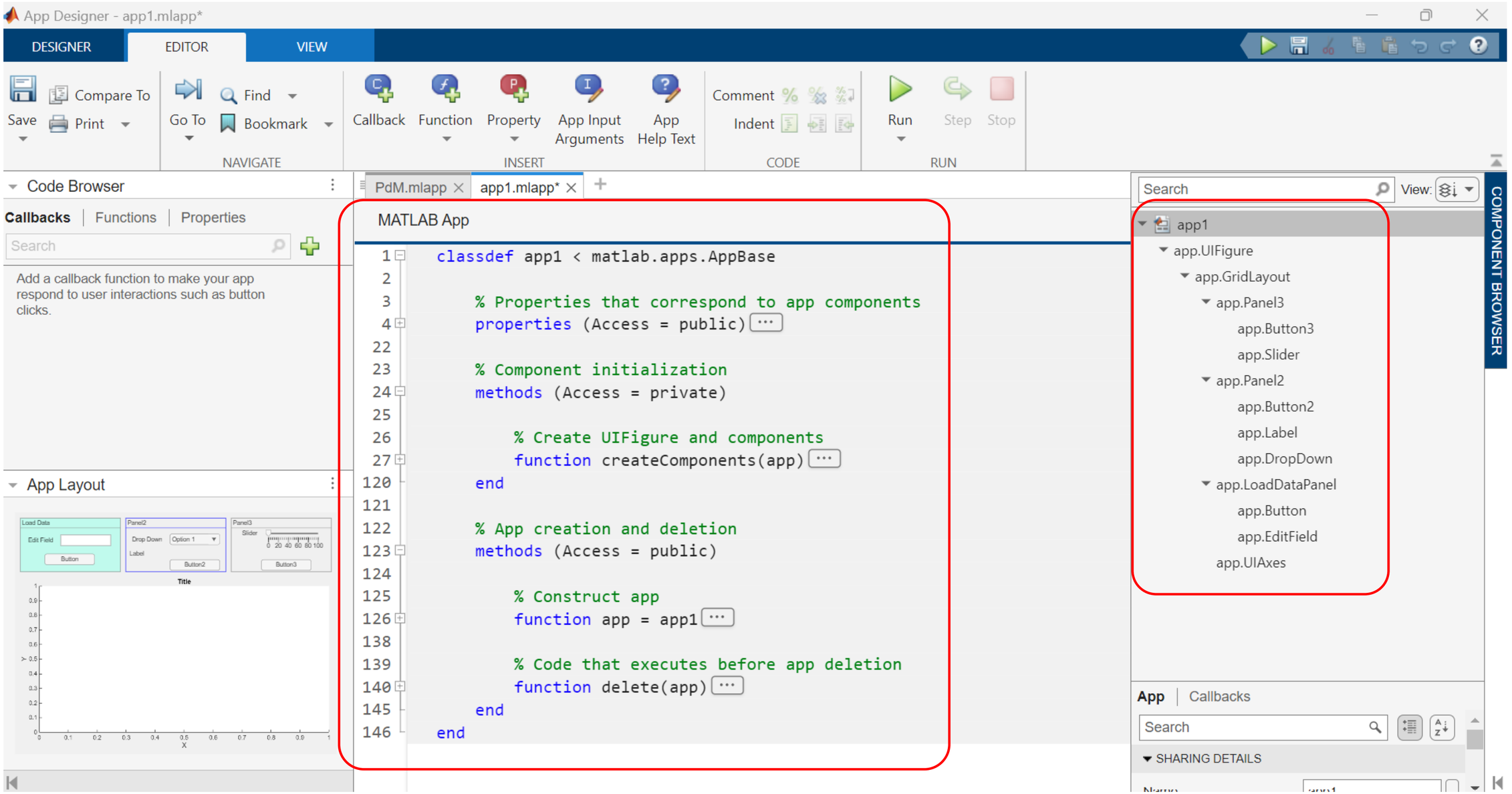

Search
View:
COMPONENT BROWSER
Testshow
app.UIFigure
app.GridLayout
app.IdentifyAnomaliesPanel
app.Label_2
app.VisualizeAnomaliesButton
app.ThreshSlider
app.LoadModelPanel
app.LoadModelButton
app.Label
app.ModelDropDown
app.LoadDataPanel
app.LoadTestButton
app.EditField
app.UIAxes

Component Browser

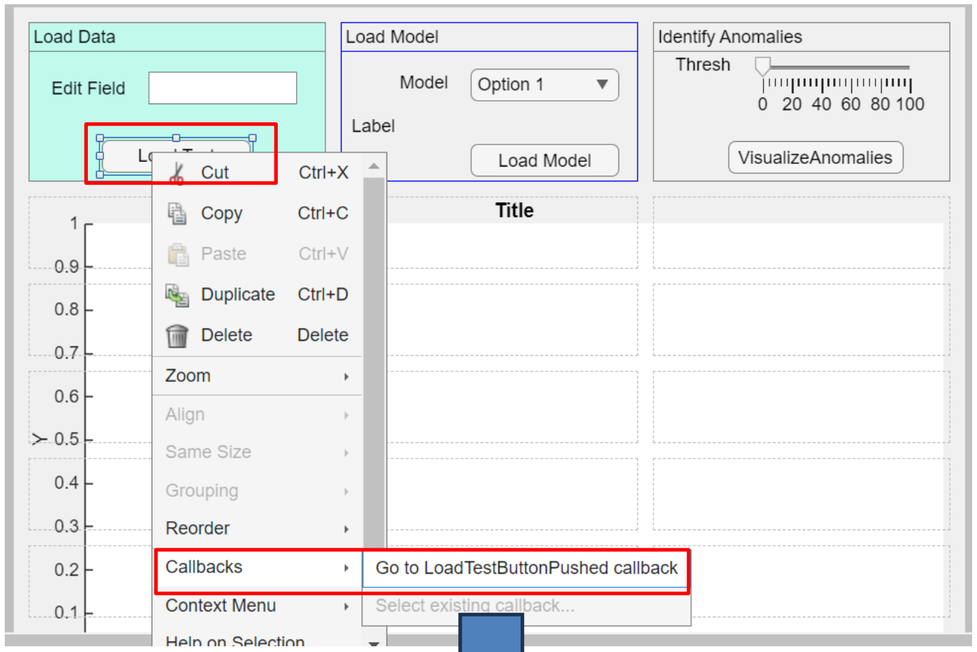




Code View


App Designer - app1.mlapp*
DESIGNER
EDITOR
VIEW
Save
Compare To
Print
Go To
Find
Bookmark
NAVIGATE
Callback
Function
Property
App Input Arguments
App Help Text
INSERT
Comment
Indent
CODE
Run
Step
Stop
RUN
Code Browser
Callbacks
Functions
Properties
Search
Add a callback function to make your app respond to user interactions such as button clicks.
App Layout
PdM.mlapp
app1.mlapp*
MATLAB App
1 classdef app1 < matlab.apps.AppBase
2
3 % Properties that correspond to app components
4 properties (Access = public) ...
22
23 % Component initialization
24 methods (Access = private)
25
26 % Create UIFigure and components
27 function createComponents(app) ...
120 end
121
122 % App creation and deletion
123 methods (Access = public)
124
125 % Construct app
126 function app = app1 ...
138
139 % Code that executes before app deletion
140 function delete(app) ...
145 end
146 end
Search
View:
app1
app.UIFigure
app.GridLayout
app.Panel3
app.Button3
app.Slider
app.Panel2
app.Button2
app.Label
app.DropDown
app.LoadDataPanel
app.Button
app.EditField
app.UIAxes
COMPONENT BROWSER
App
Callbacks
Search
SHARING DETAILS

Step 1: Load Feature Data -LoadTest Button

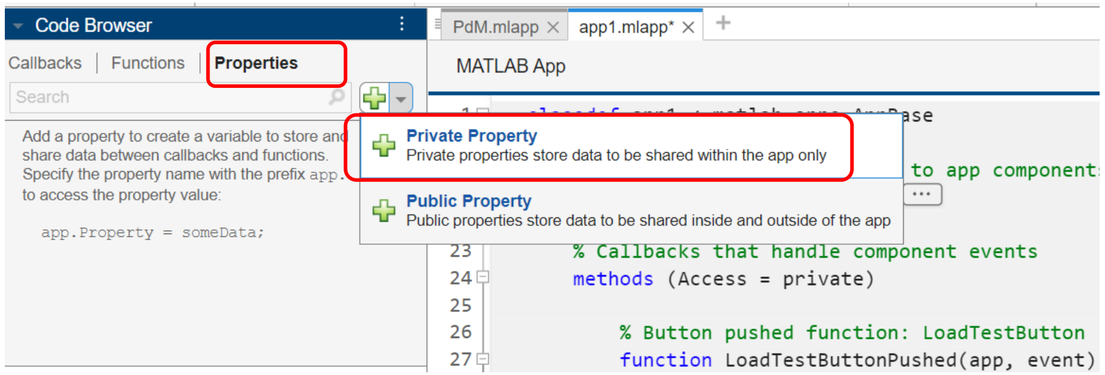


MATLAB App

```
classdef app1 < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public) ...

    % Callbacks that handle component events
    methods (Access = private)

        % Button pushed function: LoadTestButton
        function LoadTestButtonPushed(app, event)

        end
    end
```

```
try
    [file, path] = uigetfile({'*.mat;*.xlsx', 'MAT or Excel Files (*.mat, *.xlsx)'}, ...
                             'Select Test Data File');
    if isequal(file, 0)
        return;
    end

    fullpath = fullfile(path, file);
    [~, ~, ext] = fileparts(file);

    switch lower(ext)
        case '.mat'
            data = load(fullpath);
            if isfield(data, 'featureTest')
                app.featureTest = data.featureTest;
            else
                error('The MAT file must contain "featureTest" or "featureAll" + "idx".');
            end
        case '.xlsx'
            app.featureTest = readtable(fullpath);
        otherwise
            error('Unsupported file type.');
    end
    % Extract input features (your function to process table)
    app.XTestAll = extractLabeledData(app.featureTest, 'Label', 'All');
    % Show file name in EditField
    app.EditField.Value = file;
    % Success alert
    uialert(app.UIFigure, ['Test data loaded: ' file], 'Success');
catch ME
    uialert(app.UIFigure, ['Error loading test data: ' ME.message], 'Error');
end
```

Add Properties (Private)

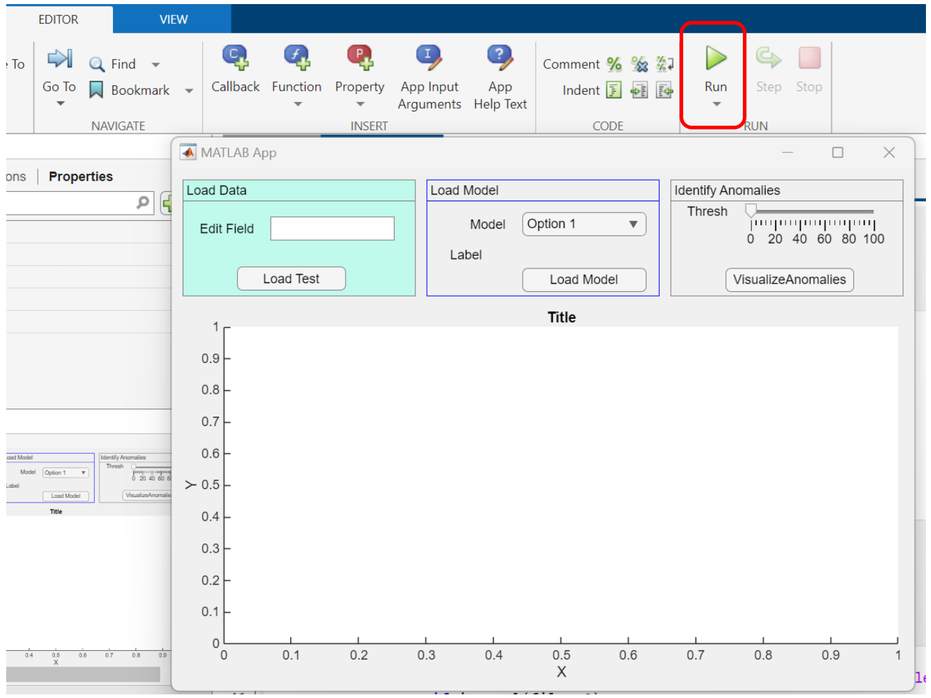


This will:

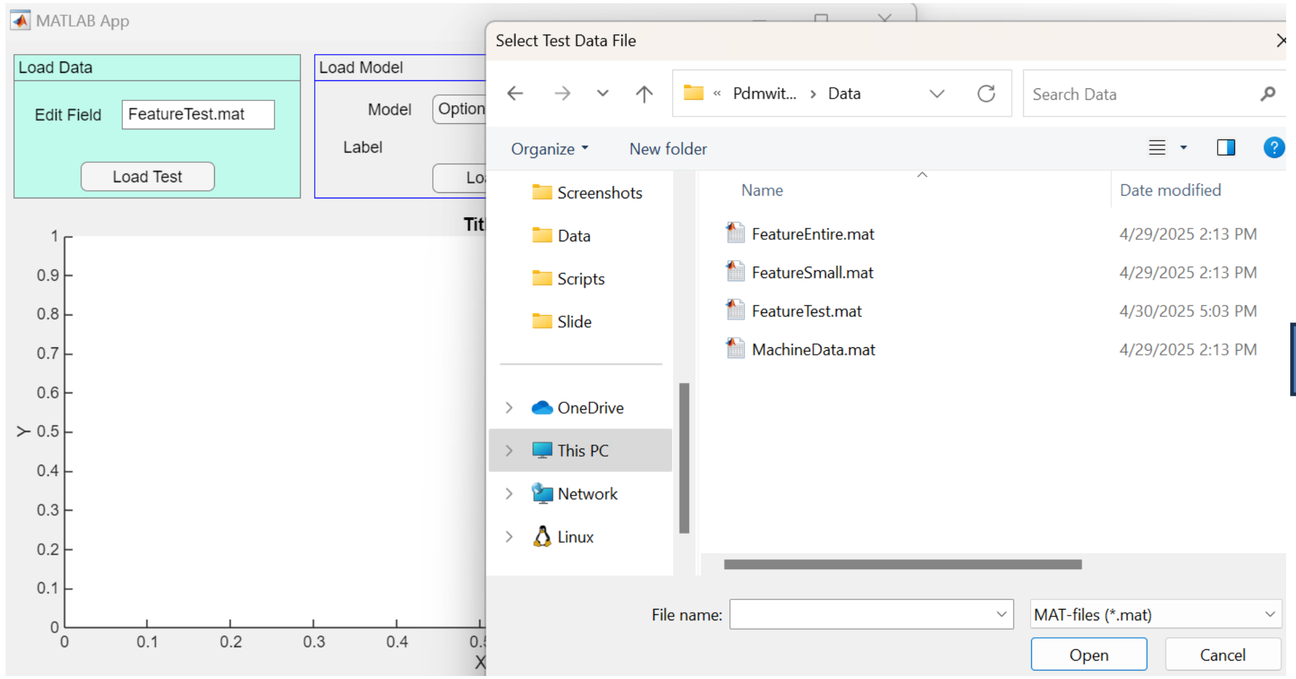
- Load the selected file
- Extract the test data
- Alert success or failure

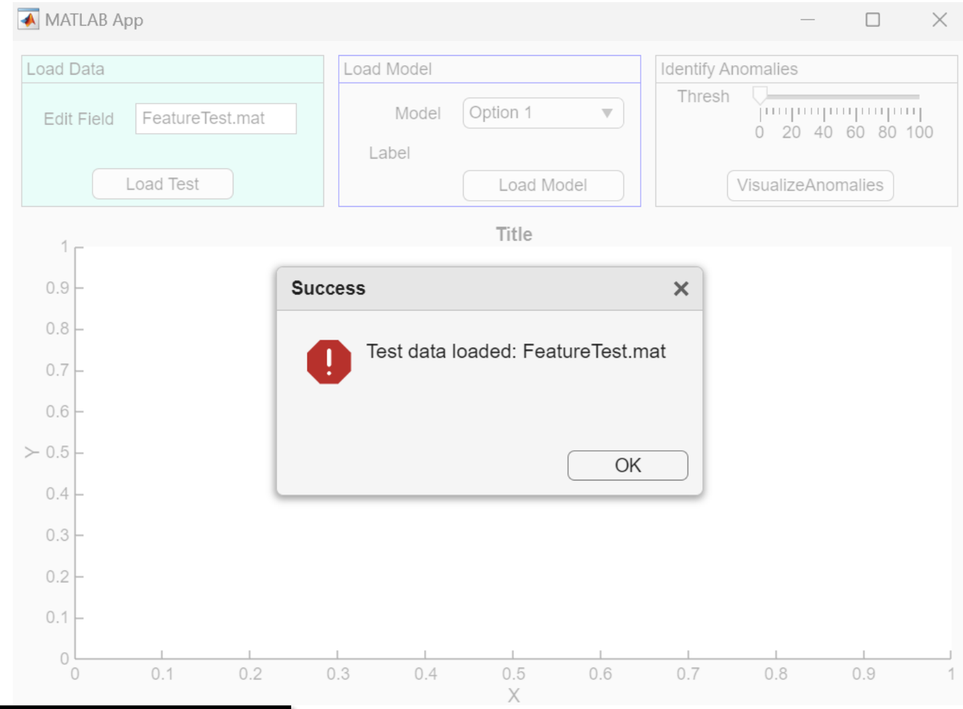
Define App Properties

```
properties (Access = private)
    Property % Description
    featureTest           % Table for test data
    XTestAll              % Cell array extracted from featureTest
    net          % The trained LSTM network
    yHatAll      % Predicted output from the network
    errorAll     % Reconstruction error between predicted and input
end
```



Output **-Load Feature Data -**LoadTest Button





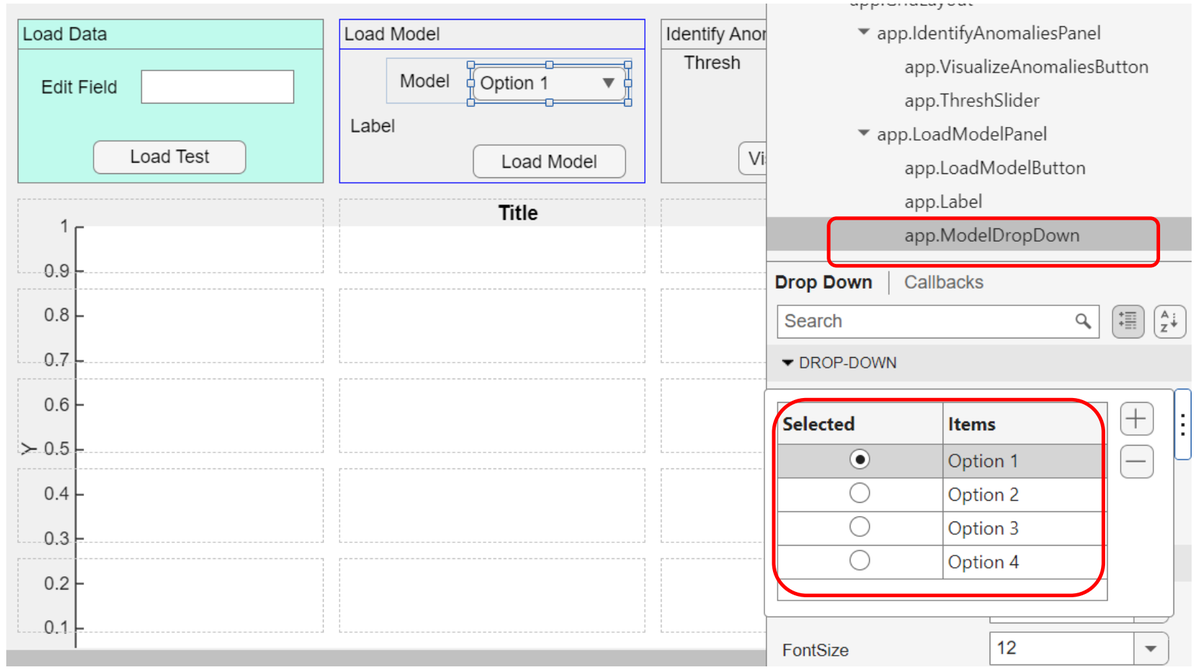
Tip : How to change Icon in uialert

Predefined Icon

This table lists the values for the predefined icons. For example, to show the check mark icon, specify the name-value pair `'Icon','success'`.

Value	Icon
`'error'` (default)	
`'warning'`	
`'info'`	
`'question'`	
`'success'`	
`''`	No icon displays.

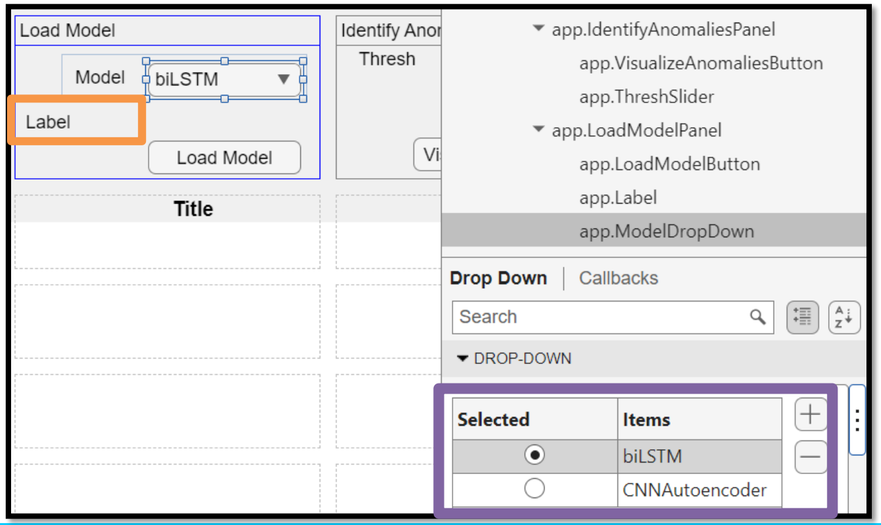
Step 2: Load Model- Select model option

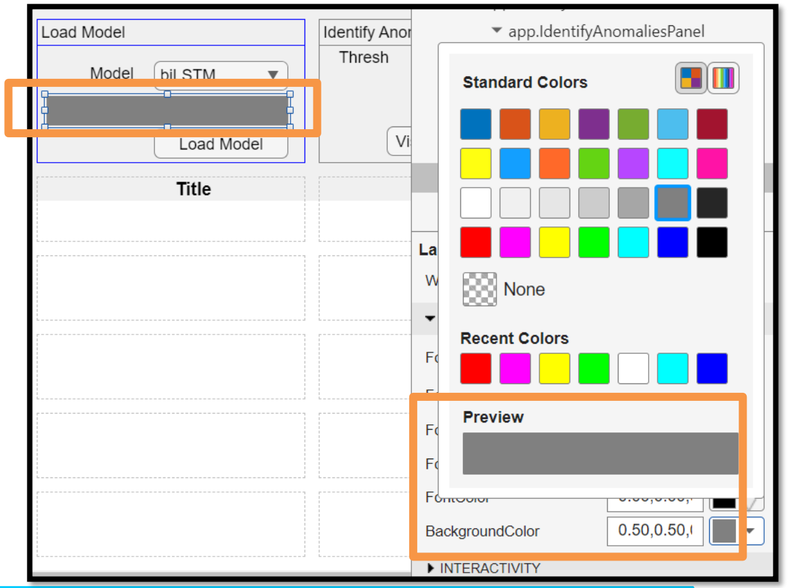


This will:

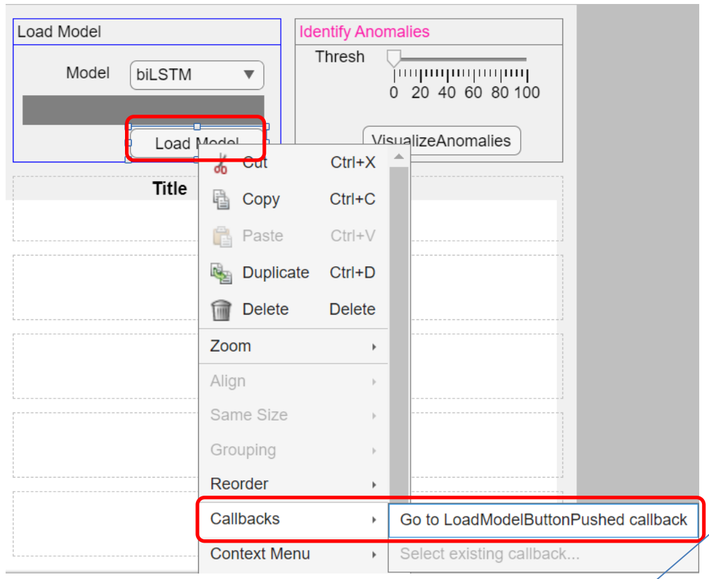
- Delete option
- Change name items
- Delete Text Label
- Change BackgroundColor

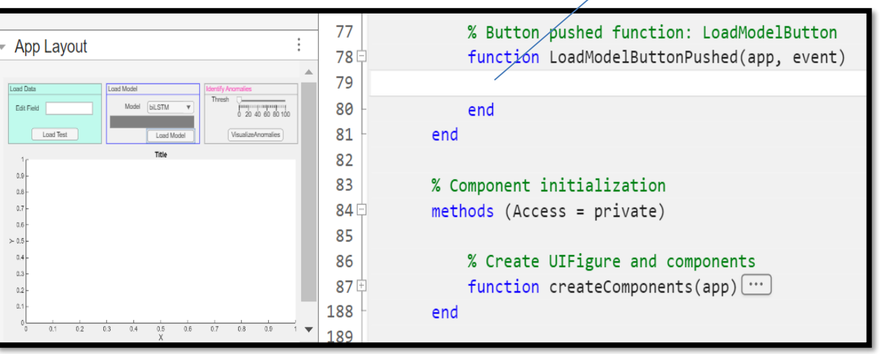
Model Selected : 1.biLSTM
2.CNNAutoencoder





Load Model Button

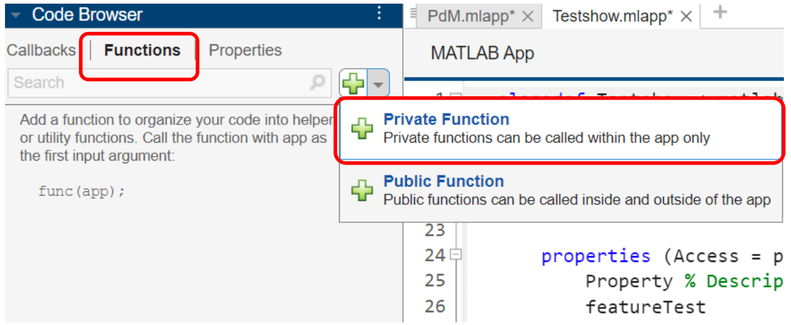





```
try
  % Select dropdown
  selectedModel = app.ModelDropDown.Value;

  [file, path] = uigetfile('*.mat', ['Select Trained Network File for ' selectedModel]);
  if isequal(file, 0)
       return; % User cancelled
  end

  modelPath = fullfile(path, file);
  modelData = load(modelPath);

  % Show Label
  app.Label.Text = file;

  % Check ‘net’ in file
  if isfield(modelData, 'net’)
      app.net = modelData.net;
  else
      error('Model file does not contain variable ''net''.’);
  end

  % Predict and Cal reconstruction error
  app.yHatAll = predict(app.net, app.XTestAll);
  app.errorAll = app.calculateError(app.XTestAll, app.yHatAll);

  % alert
  uialert(app.UIFigure, ['Model loaded and detection complete: ' file], 'Success');
catch ME
  uialert(app.UIFigure, ['Error loading model or detecting: ' ME.message], 'Error');
end
```

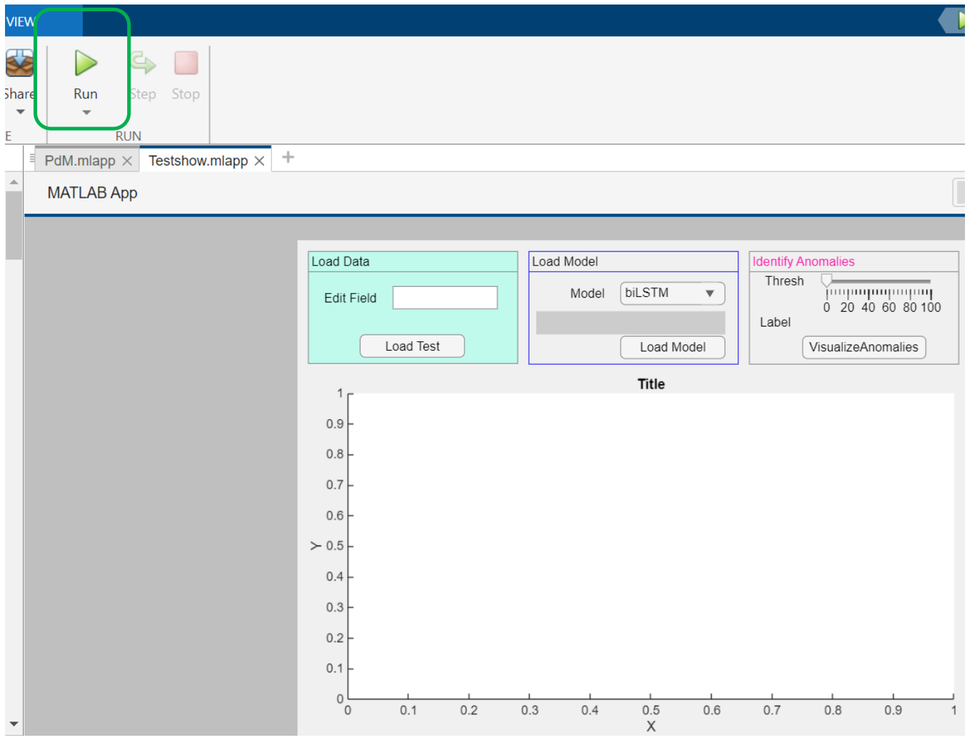
Add Functions (Private)

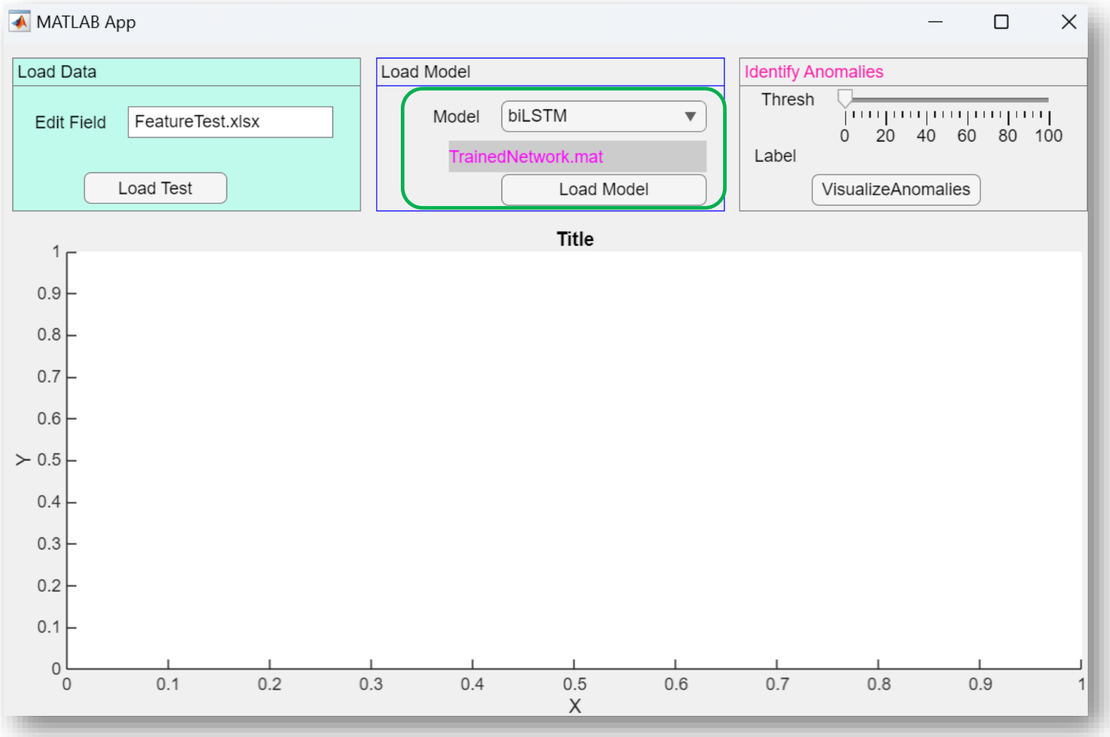
app.calculateError



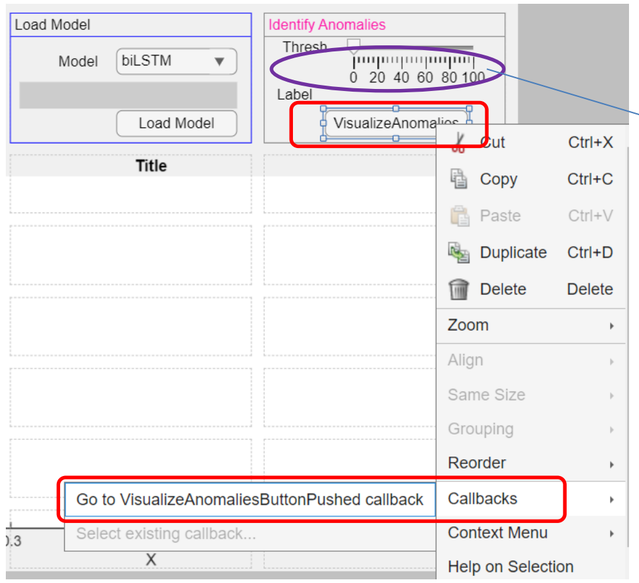

```
33      methods (Access = private)
34
35          function results = func(app)
36
37          end
38      end
```

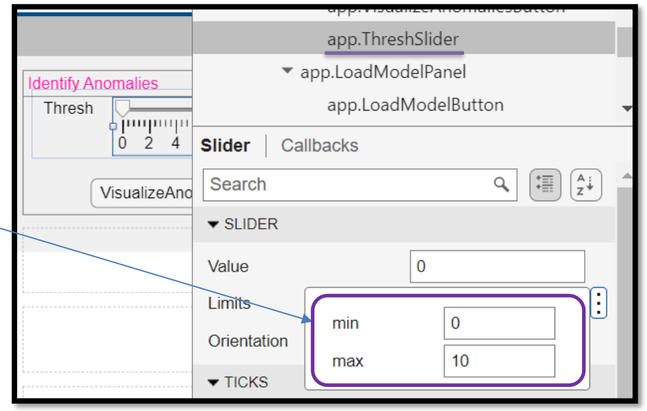
```
33      methods (Access = private)
34
35          function E = calculateError(app, X, Y)
36              E = zeros(length(X), 1);
37              for i = 1:length(X)
38                  E(i) = sqrt(sum((Y{i} - X{i}).^2));
39              end
40          end
41
```





Step 3: Visualize Anomalies Botton





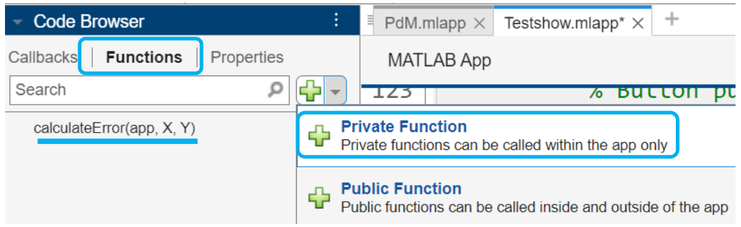
This will:

- Adjust slider limits min – max
- To 0 - 10

```
124        function VisualizeAnomaliesButtonPushed(app, event)
125
126        end
```

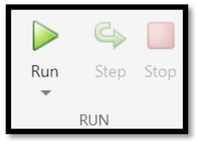
```
try
    % Get slider value
    thresh = app.ThreshSlider.Value;

    % Compute anomaly flags
    anomalies = app.errorAll > thresh * mean(app.errorAll);

    % Call visualization helper
    app.visualizeAnomaliesInApp(anomalies, app.errorAll, app.featureTest);

catch ME
    uialert(app.UIFigure, ['Error visualizing anomalies: ' ME.message], 'Error');
end
```

Add Functions (Private)




```
function visualizeAnomaliesInApp(app, anomalies, errorAll, featureTest)
        % Prepare anomaly index and error
        anomalyIdx = find(anomalies);
        anomalyErr = errorAll(anomalies);

        % Predicted and true class labels
        predAE = categorical(anomalies, [1 0], ["Anomaly", "Healthy"]);
        if ~iscategorical(featureTest.label)
            featureTest.label = categorical(featureTest.label);
        end
        trueAE = renamecats(featureTest.label, ["Before", "After"], ["Anomaly", "Healthy"]);

        % Calculate test accuracy
        acc = numel(find(trueAE == predAE)) / numel(predAE) * 100;

        % === Plot to UIAxes ===
        cla(app.UIAxes);
        plot(app.UIAxes, errorAll, 'b'); hold(app.UIAxes, 'on');
        plot(app.UIAxes, anomalyIdx, anomalyErr, 'rx');
        hold(app.UIAxes, 'off');
        title(app.UIAxes, sprintf("Reconstruction Error (Thresh: %.2f)",
        app.ThreshSlider.Value));
        xlabel(app.UIAxes, 'Observation');
        ylabel(app.UIAxes, 'Error');
        legend(app.UIAxes, 'Error', 'Candidate Anomaly', 'Location', 'northeast');
        % Update Label with accuracy
        app.Label_2.Text = sprintf('Test Accuracy: %.2f%%', acc);
        % Show confusion chart
        figure('Name', 'Confusion Matrix');
        confusionchart(trueAE, predAE);
        title('Anomaly Detection Confusion Matrix');
end
```

Run Test your APP



TECHSOURCE Dynamic Solutions Precise Results

ASCENDAS SYSTEMS

MATLAB App

Load Data

FeatureTest.mat

Load Test

Load Model

Model CNNAutoEncoder

TrainedNet1DConAutoen.mat

Load Model

Identify Anomalies

Thresh 0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5 5.5 6 6.5 7 7.5 8 8.5 9 9.5 10

Test Accuracy: 75.57%

VisualizeAnomalies

Reconstruction Error (Thresh: 2.63)

Error
Candidate Anomaly

Error

Observation

Figure 2: Confusion Matrix

File Edit View Insert Tools Desktop Window Help

Anomaly Detection Confusion Matrix

True Class \ Predicted Class	Anomaly	Healthy
Anomaly	192	430
Healthy	1	1141

Overview | Monitoring Parameter Display | Detailed Analysis using Statistical approach | Detailed Analysis using AI-based approach

KIER Wind Turbine Health Monitoring System

Time: 2019-02-02 01-26-10

Turbine Status: Power Production

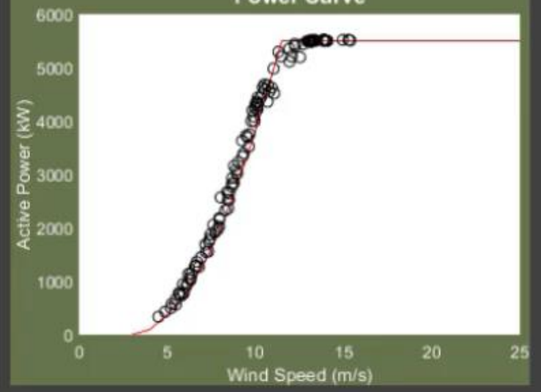
Active Power (kW) 2551.8

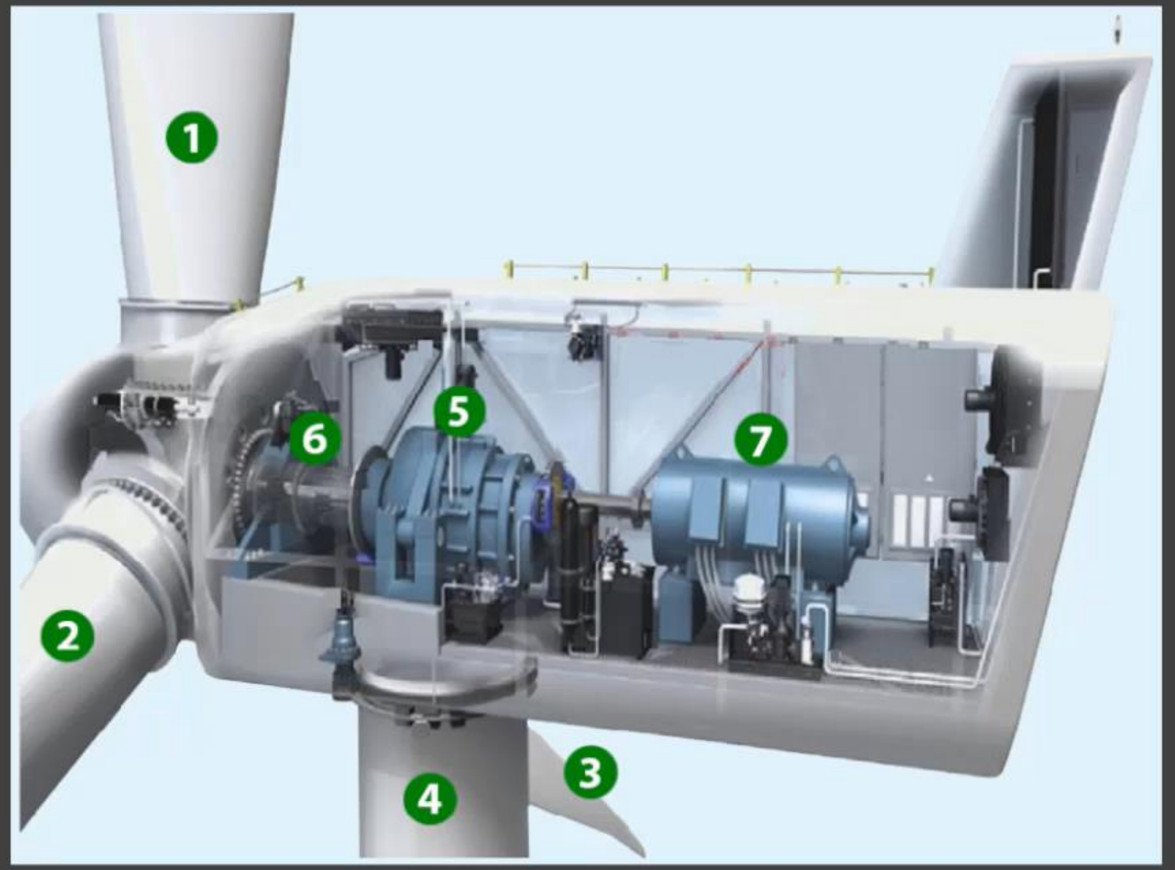
Wind Speed (m/s) 8.2

Wind Direction (deg) 331.5

Rotor RPM (rpm)	10.5
Pitch Angle (deg)	0.0
Yaw Angle (deg)	-35.5
Outside Temp. (℃)	3.9

Power Curve





① No. 1 Blade ② No. 2 Blade ③ No. 3 Blade ④ Tower

⑤ Gearbox ⑥ Shaft ⑦ Generator

Remain Useful Life

No. 1 Blade RUL 180 months

No. 2 Blade RUL 180 months

No. 3 Blade RUL 180 months

Shaft RUL 178 months

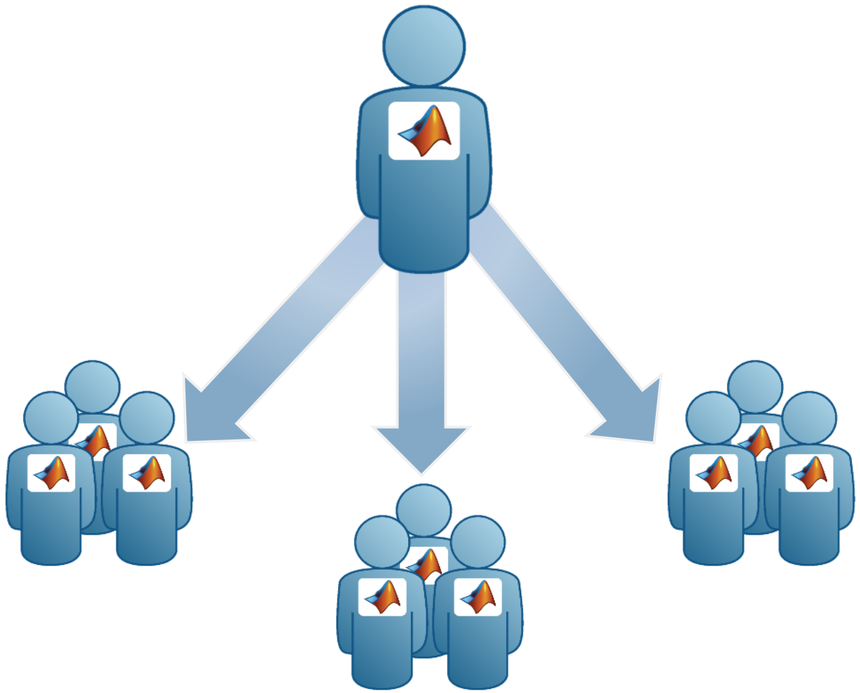
Tower top RUL 180 months

Tower mid RUL 180 months

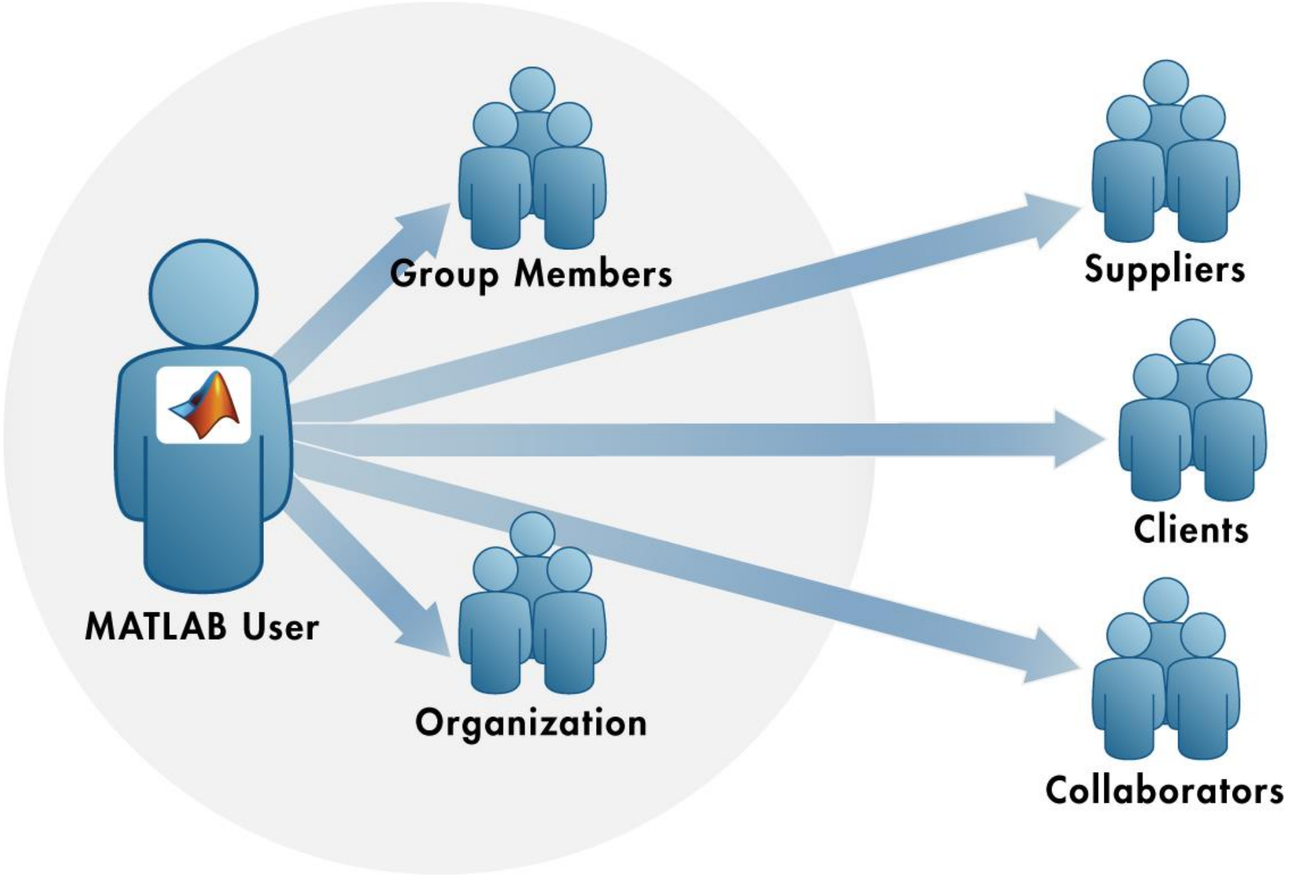
Tower base RUL 180 months

MATLAB Programs Can be Shared With Anyone

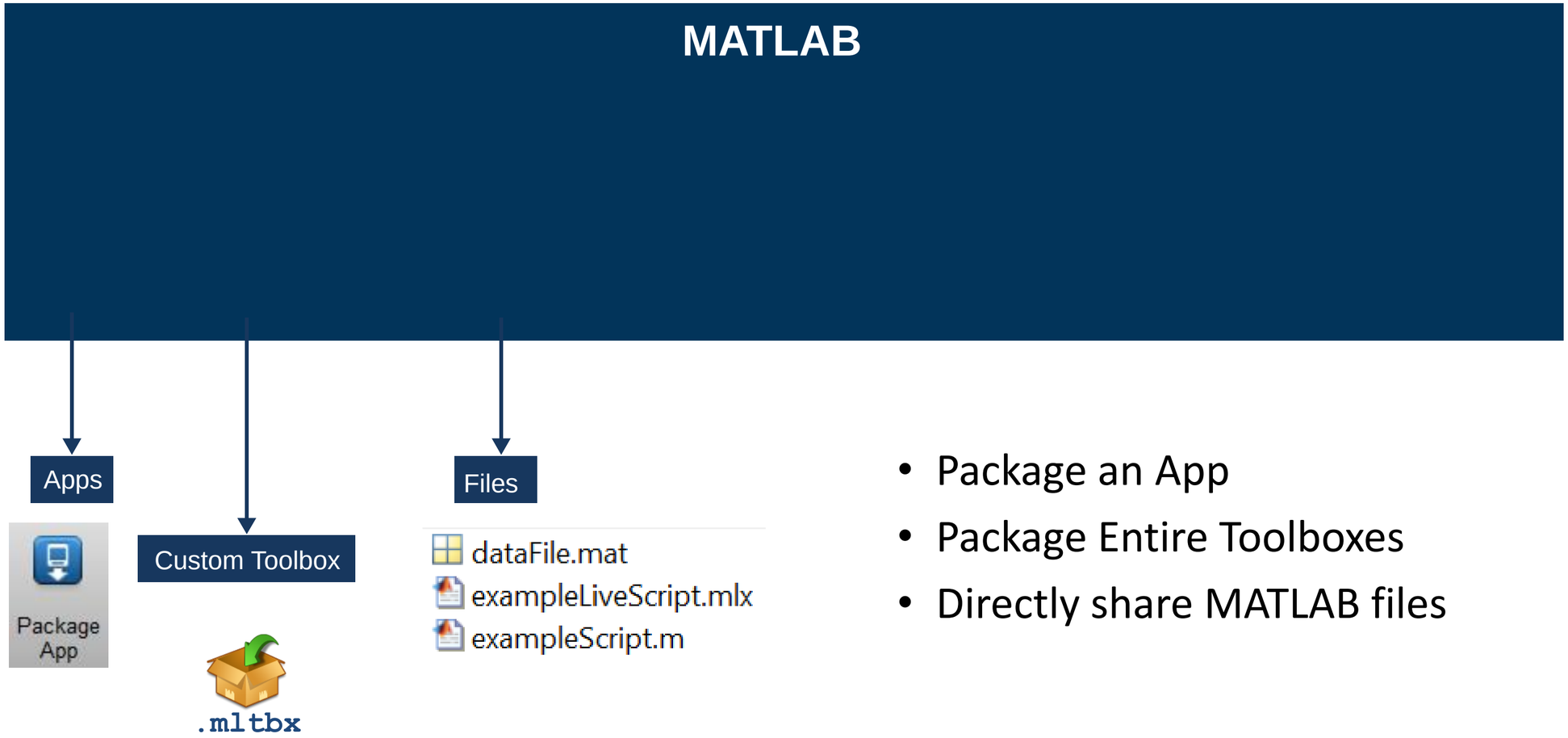
Share With Other MATLAB Users



Share With People Who do Not Have MATLAB

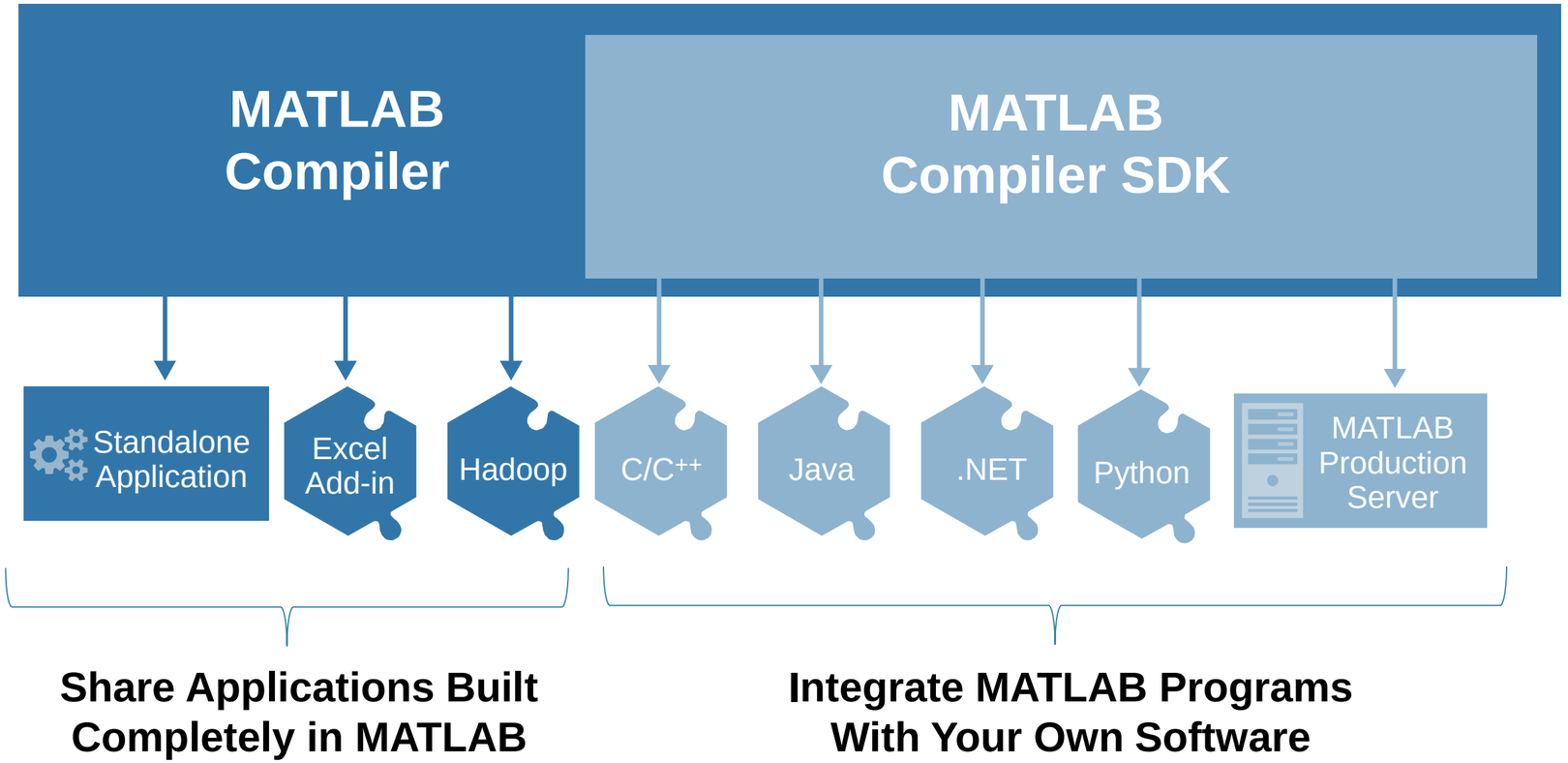


Share with MATLAB Users



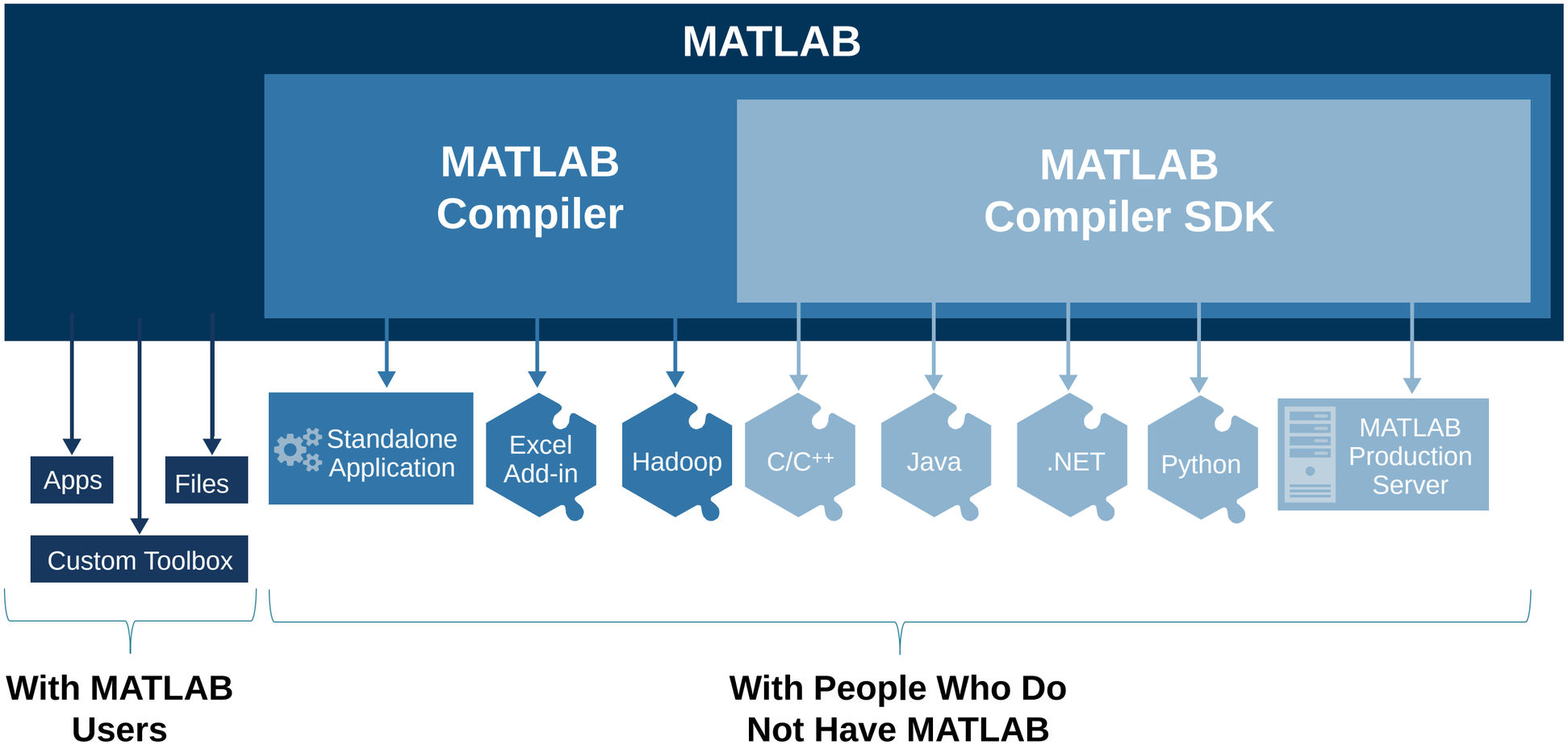
- Package an App
- Package Entire Toolboxes
- Directly share MATLAB files

Share with People Who Do Not Have MATLAB



- Royalty-free Sharing
- IP Protection via Encryption

Write Your Programs Once
Then Share To Different Targets

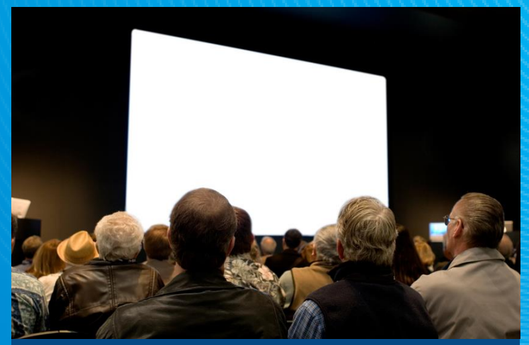


Scan Feedback From

TECHSOURCE ASCENDAS SYSTEMS

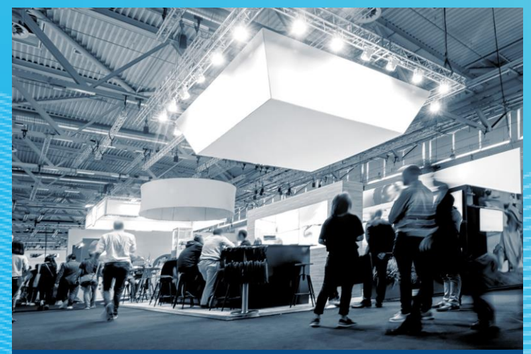
UPCOMING EVENTS....

Webinars



Seminars

Hands-on Workshops



Tradeshows

www.techsource-asia.com/events

techsource-systems

techsourcesystem

TECHSOURCE ASCENDAS SYSTEMS

CONSULTANCY SERVICES

Work with MathWorks and TechSource Systems to speed up your MATLAB and Simulink projects.

Leverage the Expertise of the MathWorks and TechSource Systems Organization

- Transparent Approach
- Customized Engagement
- Return on Investment

Click Here For More Info

Why Invest in Training?

- ✓ MORE THAN 54 COURSES
- ✓ BASIC, ADVANCED & CUSTOMIZED COURSES
- ✓ TAUGHT BY EXPERTS
- ✓ CERTIFICATE OF COMPLETION

TECHSOURCE ACADEMY

AMD Authorized Training Provider

For more info, visit **https://www.techsource-asia.com/training-calendar/**

TECHSOURCE ASCENDAS SYSTEMS

TECHSOURCE ASCENDAS SYSTEMS

TECHSOURCE ASCENDAS SYSTEMS