MATLAB EXPO

건전성 예측관리 시스템의 개발/운용 효율을 위한 DevOps 구축

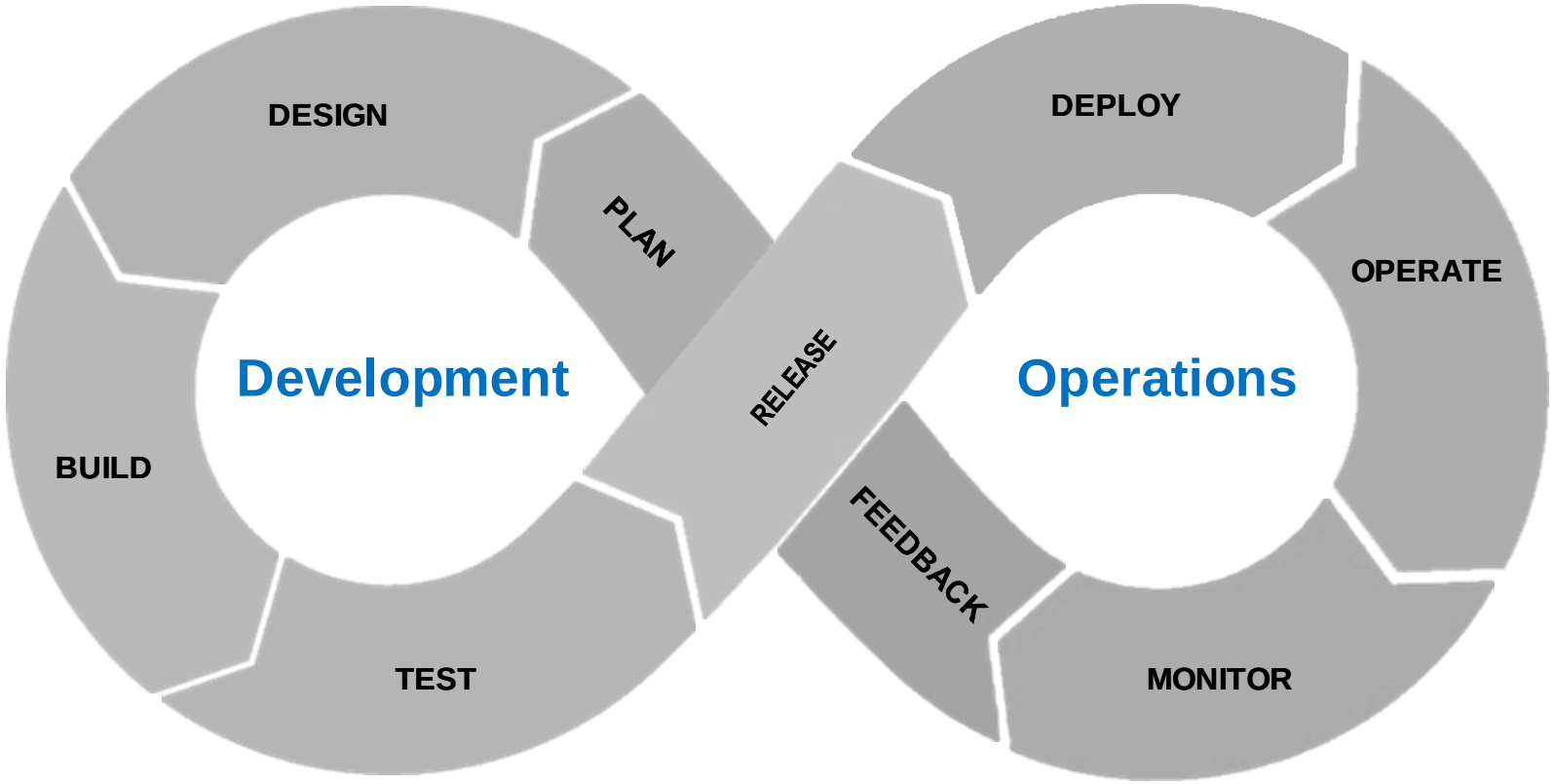
엄준상 부장, 매스웍스코리아

MathWorks

Key Takeaways

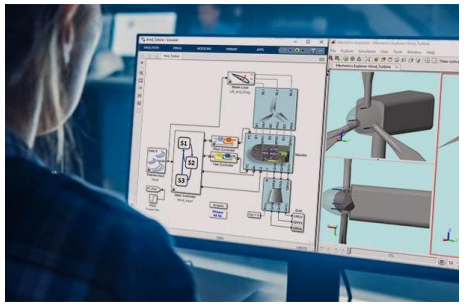
- **Incorporate familiar MATLAB capabilities**, including Predictive Maintenance and Drift Detection, in operations
- **Integrate with production systems** like data sources and dashboards, and translate those integrations from desktop to cloud servers
- **Automatically build, test, package, and deploy MATLAB code and Simulink models** with CI/CD

DevOps: Develop and Operate Production Software



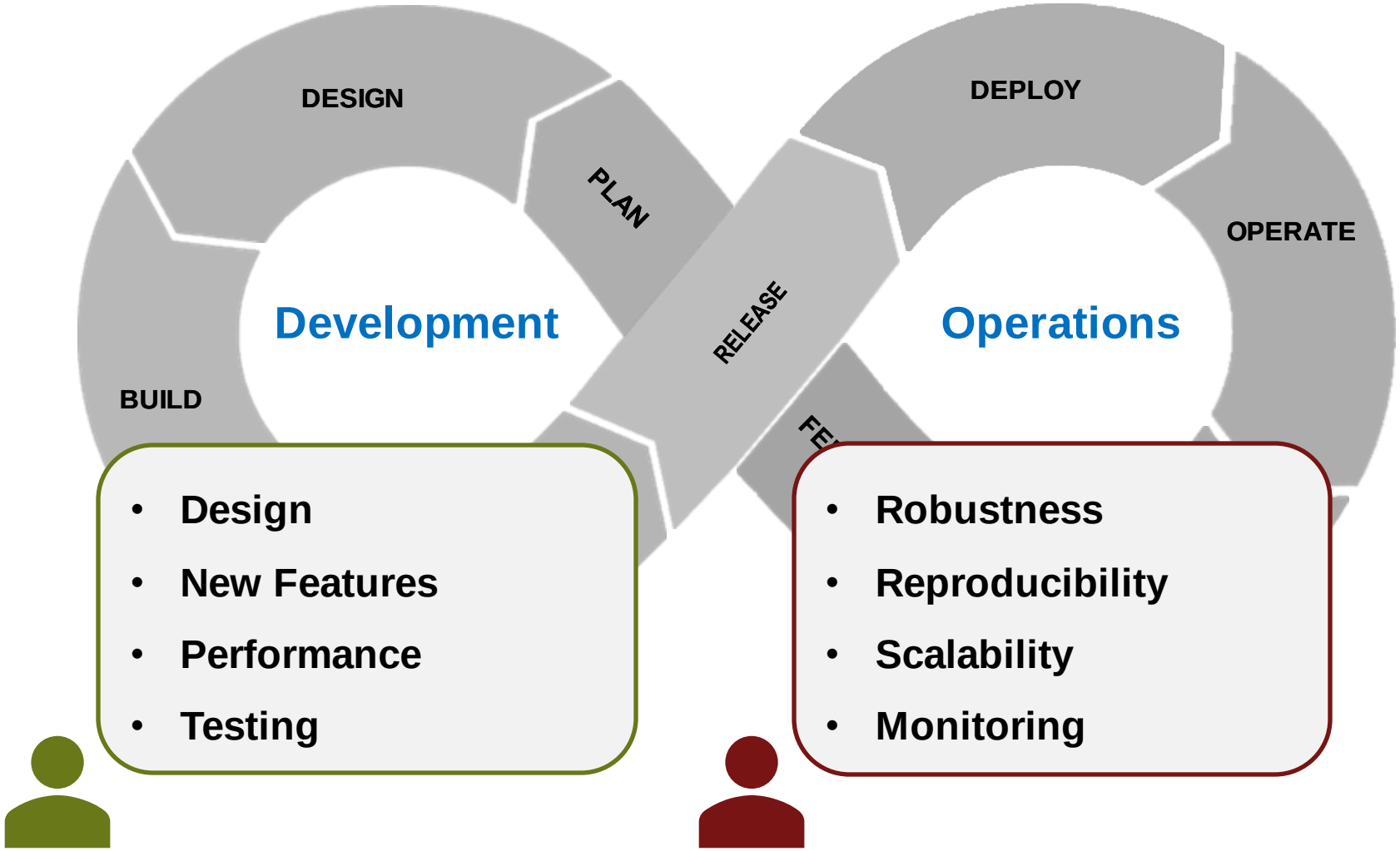
Predictive Maintenance

Financial Modeling

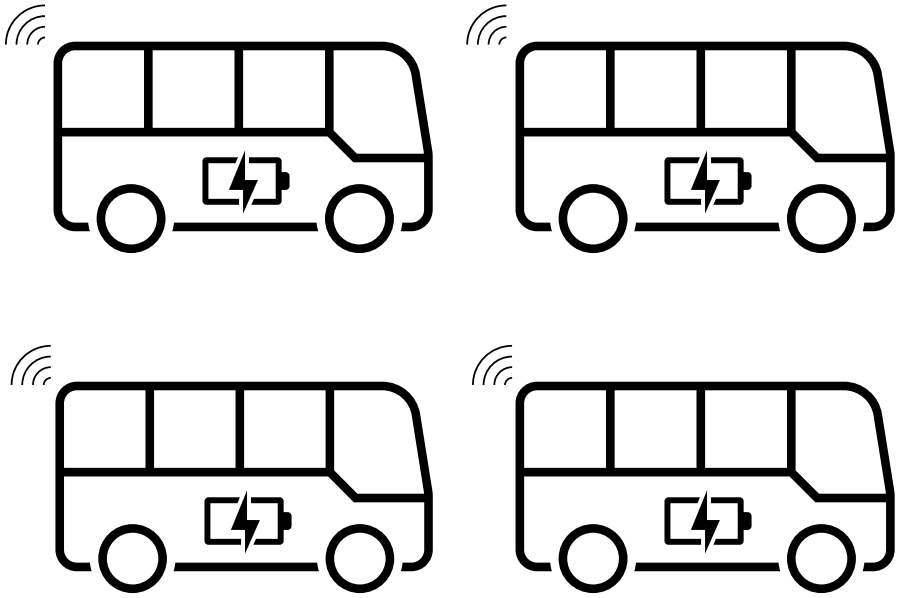


Embedded Controls

Not necessarily a conflict of interests, but certainly different interests



Example:
Predicting Battery State-of-Health



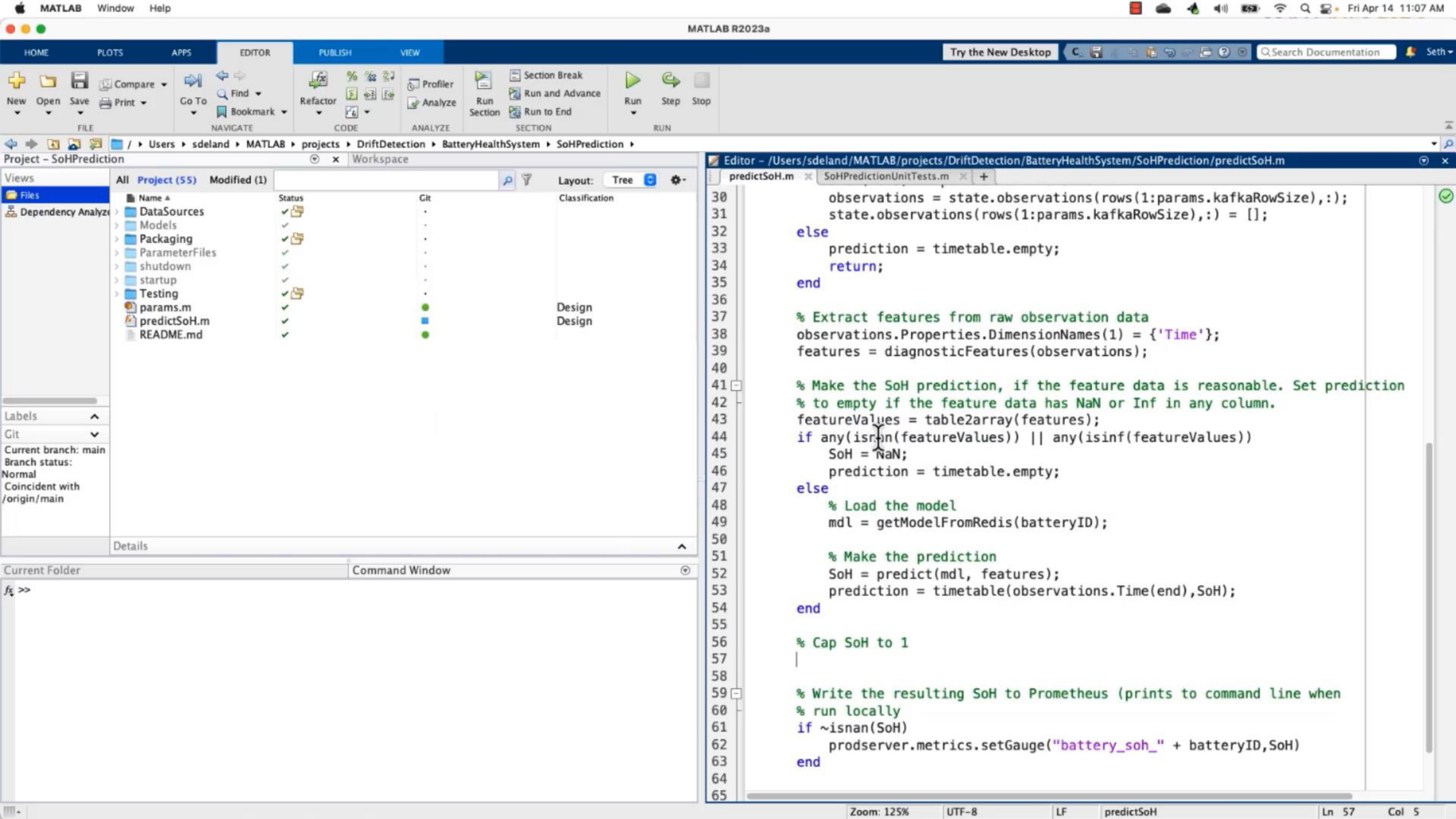
- Fleet of electric buses
- Maintenance is expensive. **Could we do a better job predicting when batteries need replacing?**
- Started gathering telemetry data on batteries

Variables – observations

observations

1800x7 timetable

	timestamp	1 Current	2 Voltage	3 Temperature1	4 Temperature2	5 SoC_B1	6 SoC_B2	7 BatteryID
1	01-Nov-2021 00:...	2.6869	7.4436	333.1463	332.7619	0.4995	0.4995	1
2	01-Nov-2021 00:...	2.6872	7.4426	333.1317	332.3924	0.4990	0.4990	1
3	01-Nov-2021 00:...	2.6876	7.4417	333.1073	332.0405	0.4985	0.4985	1
4	01-Nov-2021 00:...	2.6879	7.4408	333.0740	331.7048	0.4980	0.4980	1
5	01-Nov-2021 00:...	2.6882	7.4399	333.0327	331.3844	0.4975	0.4975	1
6	01-Nov-2021 00:...	2.6885	7.4390	332.9843	331.0783	0.4970	0.4970	1
7	01-Nov-2021 00:...	2.6889	7.4381	332.9295	330.7857	0.4965	0.4965	1


MATLAB Window Help
Fri Apr 14 11:07 AM
MATLAB R2023a
HOME PLOTS APPS EDITOR PUBLISH VIEW
Try the New Desktop
Search Documentation
Seth
New Open Save Compare Print FILE
Go To Find Bookmark NAVIGATE
Refactor CODE
Profiler Analyze ANALYZE
Run Section Section Break Run and Advance Run to End SECTION
Run Step Stop RUN
/ Users sdeland MATLAB projects DriftDetection BatteryHealthSystem SoHPrediction
Project – SoHPrediction
Workspace
Views
Files
Dependency Analyzer
All Project (55) Modified (1)
Layout: Tree
Name Status Git Classification
DataSources
Models
Packaging
ParameterFiles
shutdown
startup
Testing
params.m Design
predictSoH.m Design
README.md
Labels
Git
Current branch: main
Branch status:
Normal
Coincident with
/origin/main
Details
Current Folder
Command Window
fx >>
Editor – /Users/sdeland/MATLAB/projects/DriftDetection/BatteryHealthSystem/SoHPrediction/predictSoH.m
predictSoH.m
SoHPredictionUnitTests.m
30 observations = state.observations(rows(1:params.kafkaRowSize),:);
31 state.observations(rows(1:params.kafkaRowSize),:) = [];
32 else
33 prediction = timetable.empty;
34 return;
35 end
36
37 % Extract features from raw observation data
38 observations.Properties.DimensionNames(1) = {'Time'};
39 features = diagnosticFeatures(observations);
40
41 % Make the SoH prediction, if the feature data is reasonable. Set prediction
42 % to empty if the feature data has NaN or Inf in any column.
43 featureValues = table2array(features);
44 if any(isnan(featureValues)) || any(isinf(featureValues))
45 SoH = NaN;
46 prediction = timetable.empty;
47 else
48 % Load the model
49 mdl = getModelFromRedis(batteryID);
50
51 % Make the prediction
52 SoH = predict(mdl, features);
53 prediction = timetable(observations.Time(end),SoH);
54 end
55
56 % Cap SoH to 1
57
58
59 % Write the resulting SoH to Prometheus (prints to command line when
60 % run locally
61 if ~isnan(SoH)
62 prodserver.metrics.setGauge("battery_soh_" + batteryID,SoH)
63 end
64
65
Zoom: 125% UTF-8 LF predictSoH Ln 57 Col 5

Batteries - Dashboards - Grafa

Not Secure | batteryhealth.mwlab.io/d/H0NdyV5Vk/batteries?orgId=1&refresh=5s&from=now-30m&to=now

General / Batteries

Last 30 minutes

Battery All Feature All

Notes (1 panel)

Battery and Model Health

Battery Health

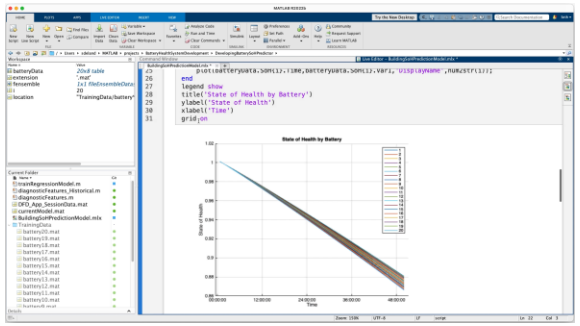
1	2	3	4	5	6	7	8	9	10
81.0%	82.9%	83.7%	87.2%	81.4%	82.3%	88.6%	96.7%	85.6%	89.7%
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
84.4%	81.7%	85.5%	84.6%	79.5%	83.7%	84.0%	82.3%	81.0%	93.5%

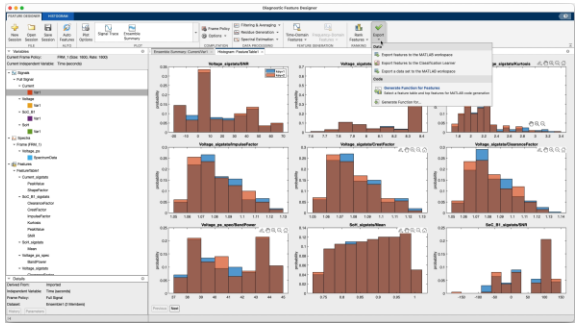
Battery 1 Health and Drift Status

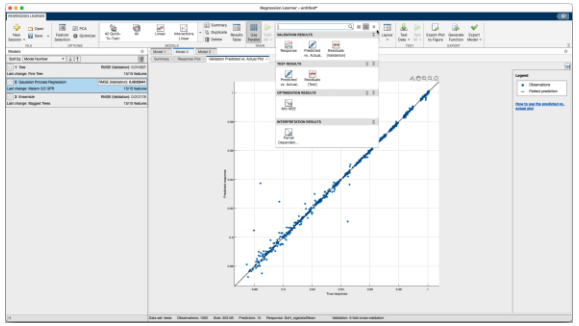
Battery 1 SoH

Battery 1 All Drift Status

Create a SoH prediction function using domain-specific tools for engineering data and predictive maintenance







Data Exploration → **Feature Extraction** → **Machine Learning**

MATLAB R2022b

HOME PLOTS APPS LIVE EDITOR INSERT VIEW

Try the New Desktop | Search Documentation | Seth

New Script | New Live Script | New | Open | Find Files | Compare — FILE
Import Data | Clean Data | Variable | Save Workspace | Clear Workspace — VARIABLE
Favorites | Analyze Code | Run and Time | Clear Commands — CODE
Simulink — SIMULINK
Layout | Preferences | Set Path | Parallel | Add-Ons — ENVIRONMENT
Help | Community | Request Support | Learn MATLAB — RESOURCES

/ ▸ Users ▸ sdeland ▸ MATLAB ▸ projects ▸ BatteryHealthSystemDevelopment ▸ DevelopingBatterySoHPredictor ▸

Workspace

Name ▲	Value

Current Folder

Name	Git
trainRegressionModel.m	
diagnosticFeatures_Historical.m	
diagnosticFeatures.m	
DFD_App_SessionData.mat	
currentModel.mat	
BuildingSoHPredictionModel.mlx	
TrainingData	
Models	
log	

Details

Command Window

Live Editor - BuildingSoHPredictionModel.mlx

BuildingSoHPredictionModel.mlx

Battery State-of-Health Prediction

Battery State-of-Health (SoH) prediction is important for making sure batteries have not degraded beyond their useful range, and for estimating the capacity of the battery. Making accurate SoH predictions is challenging due to the complex , and difficult-to-anticipate duty cycles that the battery will be subject to in normal operation.

Load historical data

We use historical data from several batteries to build our model. We format the data as an "ensemble", which helps us keep the data organized by the battery that it came from.

```
location = fullfile("TrainingData","battery*.mat");
extension = '.mat';
fensemble = fileEnsembleDatastore(location,extension);
fensemble.ReadFcn = @readBatteryData;
fensemble.DataVariables = ["Current","Voltage","T1","T2","SoC_B1","SoC_B2","SoH"];
fensemble.SelectedVariables = ["Current","Voltage","T1","T2","SoC_B1","SoC_B2","SoH"];
batteryData = readall(fensemble)
```

Look at SoH by battery

```
figure
hold on
for i = 1:height(batteryData)
    plot(batteryData.SoH{i}.Time, batteryData.SoH{i}.Var1, DisplayName=num2str(i));
end
legend show
title('State of Health by Battery')
```

Zoom: 150% | UTF-8 | LF | script | Ln 1 | Col 45

Diagnostic Feature Designer

FEATURE DESIGNER | ENSEMBLE SUMMARY

New Session | Open Session | Save Session — FILE
Auto Features — AUTO
Plot Options | Signal Trace | Ensemble Summary — PLOT
Frame Policy | Options — COMPUTATION
Filtering & Averaging | Residue Generation | Spectral Estimation — DATA PROCESSING
Time-Domain Features | Frequency-Domain Features — FEATURE GENERATION
Rank Features — RANKING
Export — EXPORT

Variables

Current Frame Policy: FRM_1 (Size: 1800, Rate: 1800)
Current Independent Variable: Time (seconds)

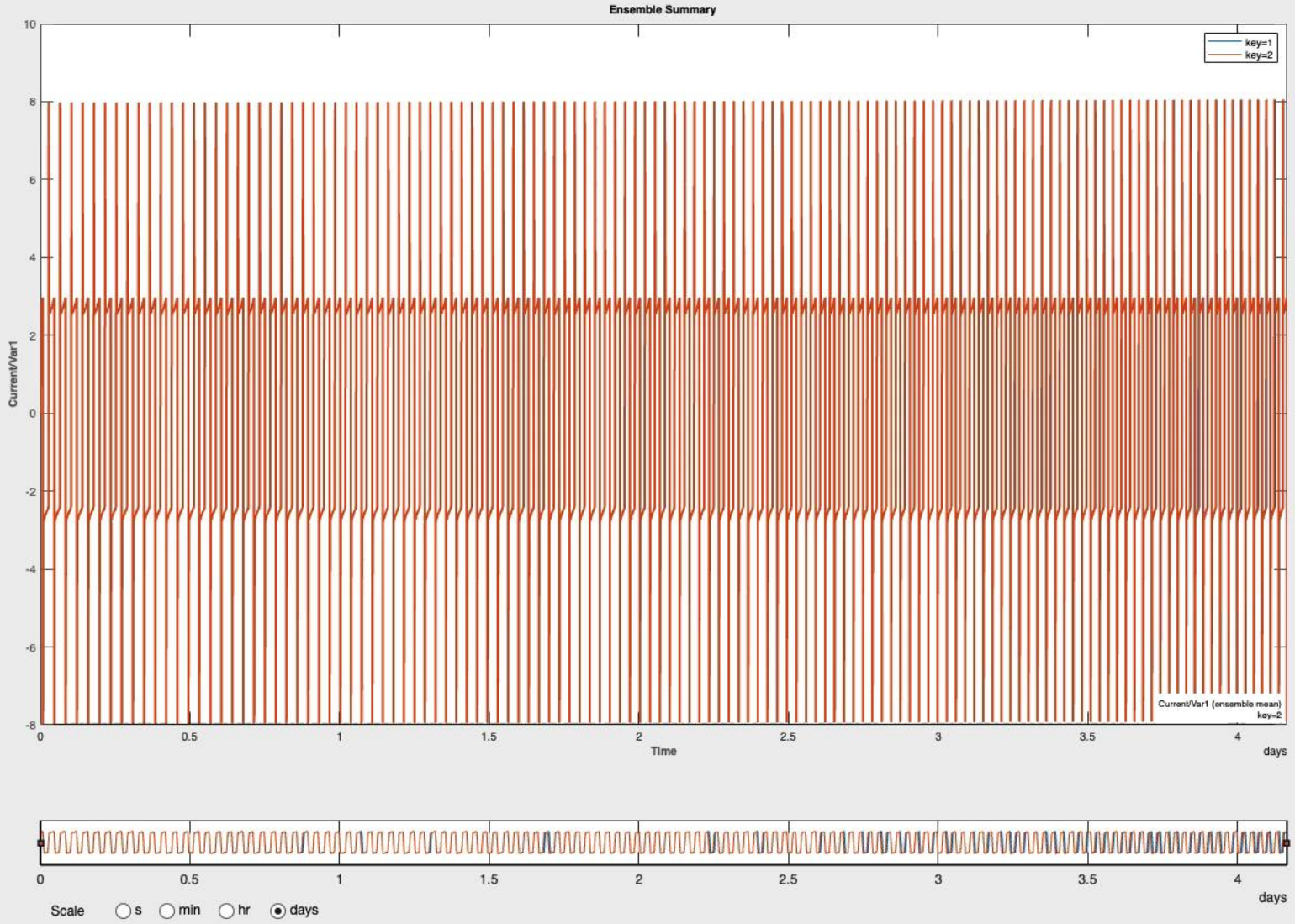
- Signals
  - Full Signal
    - Current
      - Var1
    - Voltage
      - Var1
    - SoC_B1
      - Var1
    - SoH
      - Var1
- Spectra
  - Frame (FRM_1)
    - Voltage_ps
      - SpectrumData
- Features
  - FeatureTable1
    - Current_sigstats
      - PeakValue
      - ShapeFactor
    - SoC_B1_sigstats
      - ClearanceFactor
      - CrestFactor
      - ImpulseFactor
      - Kurtosis
      - PeakValue
      - SNR
    - SoH_sigstats
      - Mean
    - Voltage_ps_spec
      - BandPower
    - Voltage_sigstats

Details

Derived From: Imported
Independent Variable: Time (seconds)
Frame Policy: Full Signal
Dataset: Ensemble1 (2 Members)

History | Parameters

Ensemble Summary: Current/Var1 | Histogram: FeatureTable1



Regression Learner - untitled*

REGRESSION LEARNER

New Session | Open | Save — FILE
Feature Selection | PCA | Optimizer — OPTIONS
All Quick-To-Train | All | Linear | Interactions Linear | Summary | Duplicate | Delete | Results Table — MODELS
Use Parallel | Train All — TRAIN
Response | Predicted vs. Actual.. | Residuals (Validation) | Predicted vs. Actual.. | Layout — PLOT AND INTERPRET
Test Data | Test All — TEST
Export Plot to Figure | Generate Function | Export Model — EXPORT

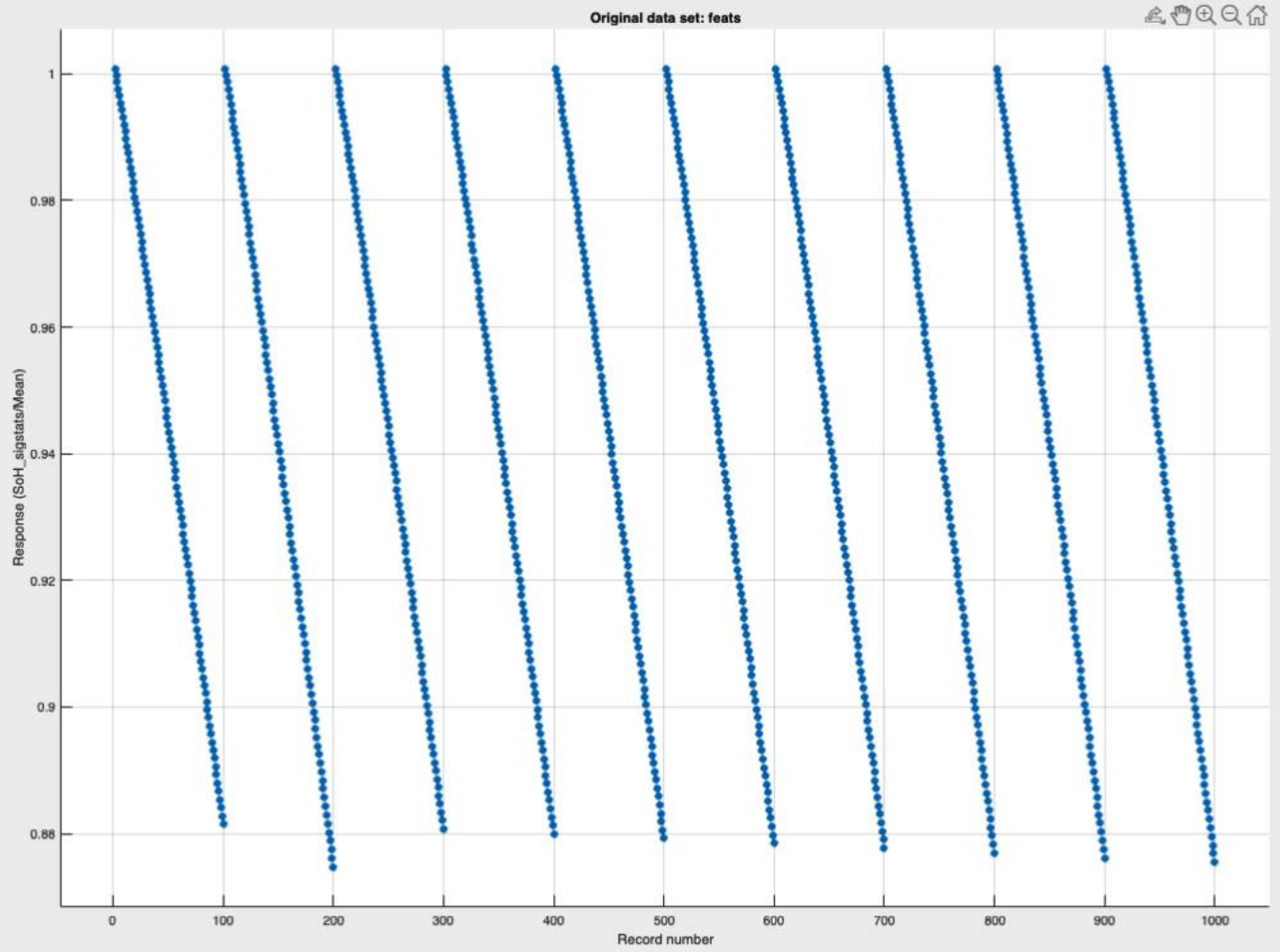
Models

Sort by Model Number

1 Tree — Draft
Last change: Fine Tree — 15/15 features

Model 1

Summary | Response Plot



Plot
True

Style
Markers
Box plot

X-axis
X Record number

How to use the response plot

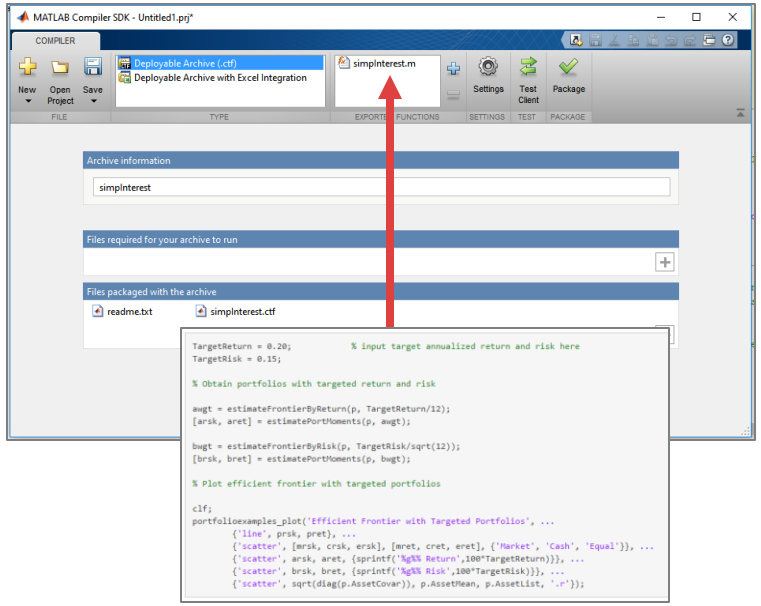
Data set: feats | Observations: 1000 | Size: 205 kB | Predictors: 15 | Response: SoH_sigstats/Mean | Validation: 5-fold cross-validation

Typical authoring workflow

Click the ‘Package’ button and wait for the compiler to generate the deployable archive

.ctf file

MATLAB Compiler SDK

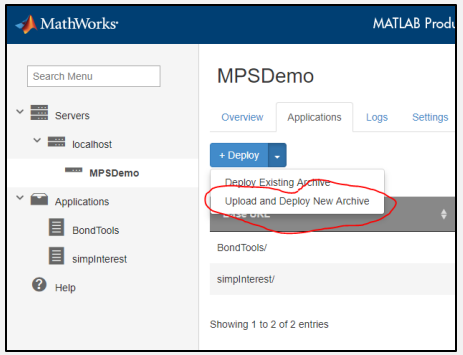


MATLAB code

MATLAB

Code / test

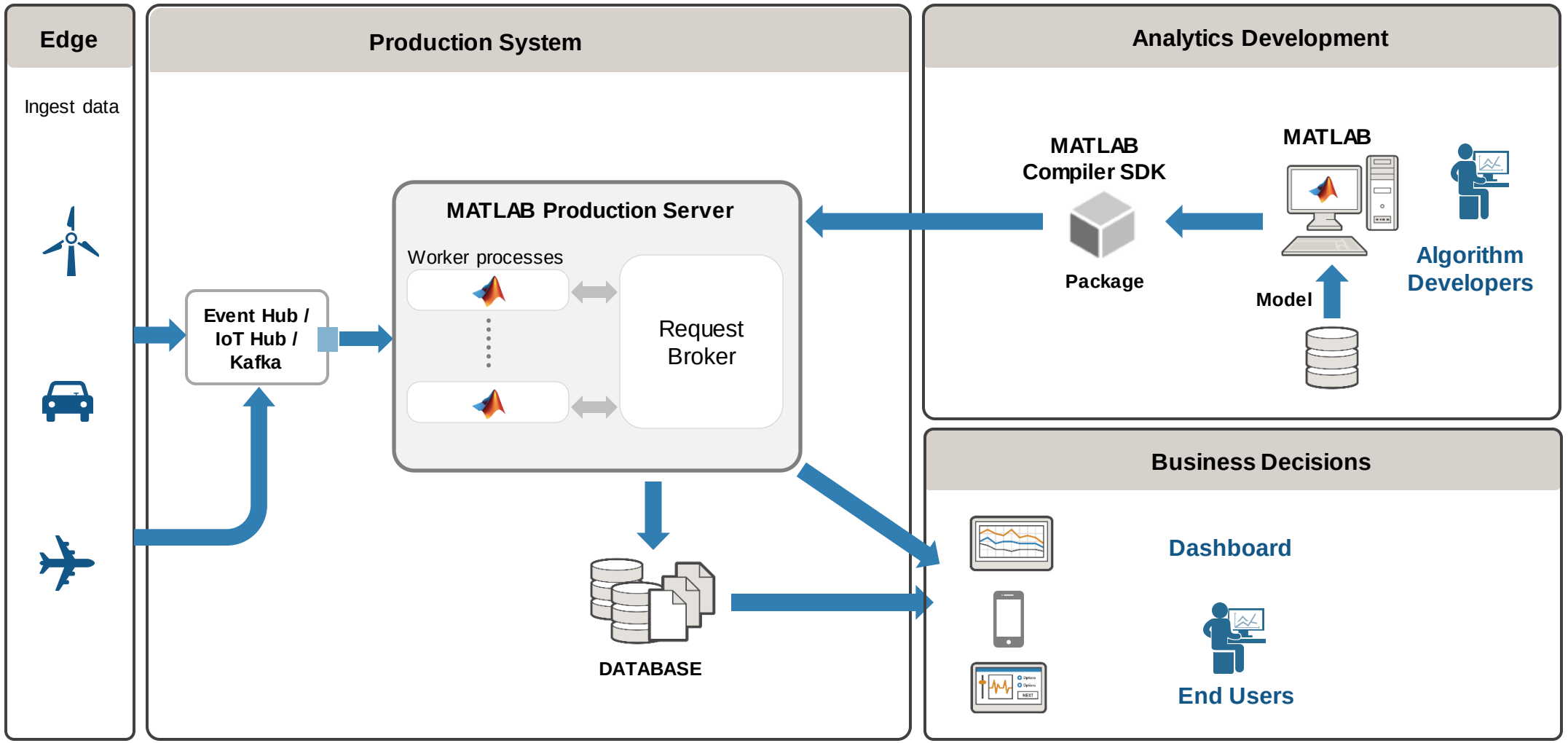
MATLAB Production Server



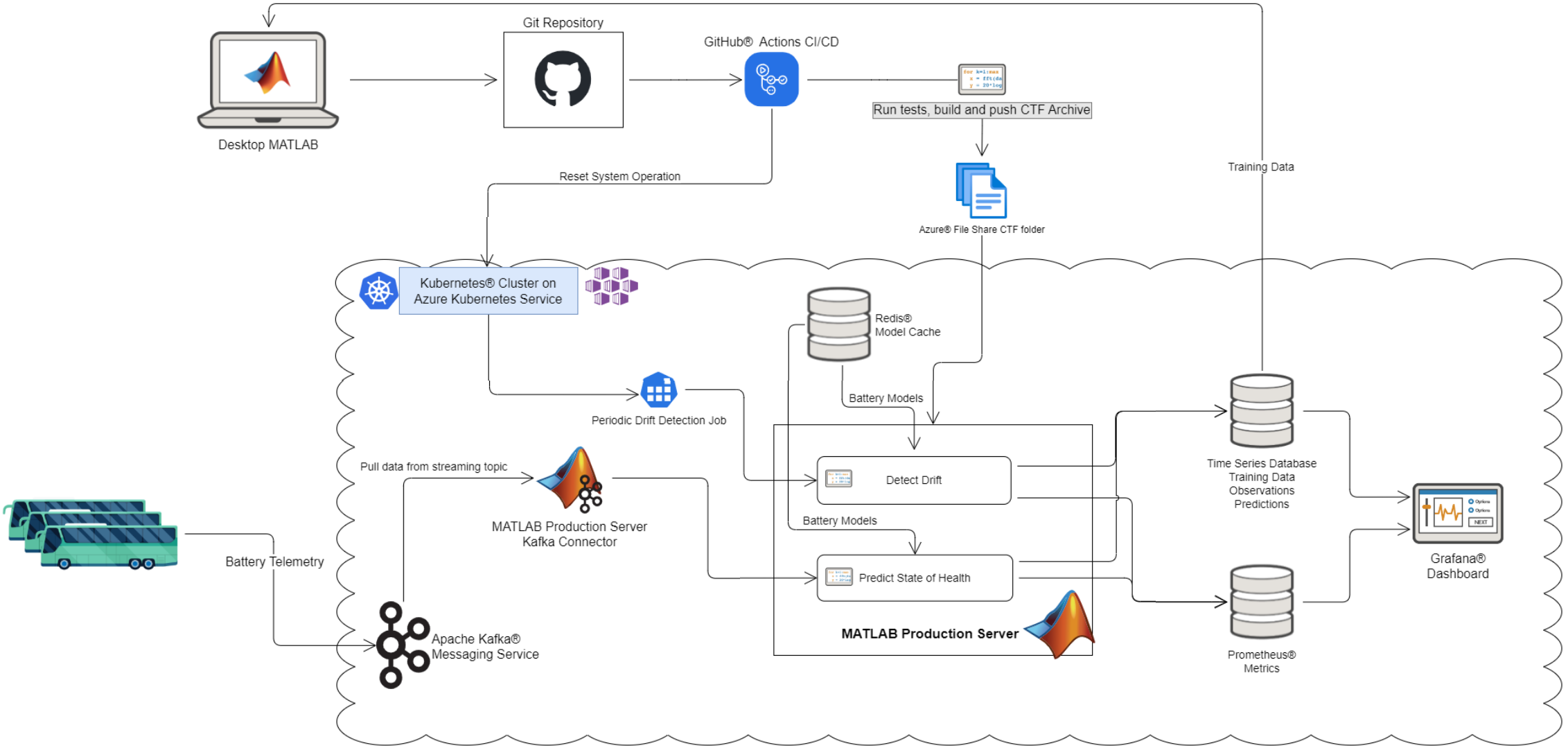
Copy .ctf file into the auto_deploy folder or use web dashboard

Server file system

Example reference architecture

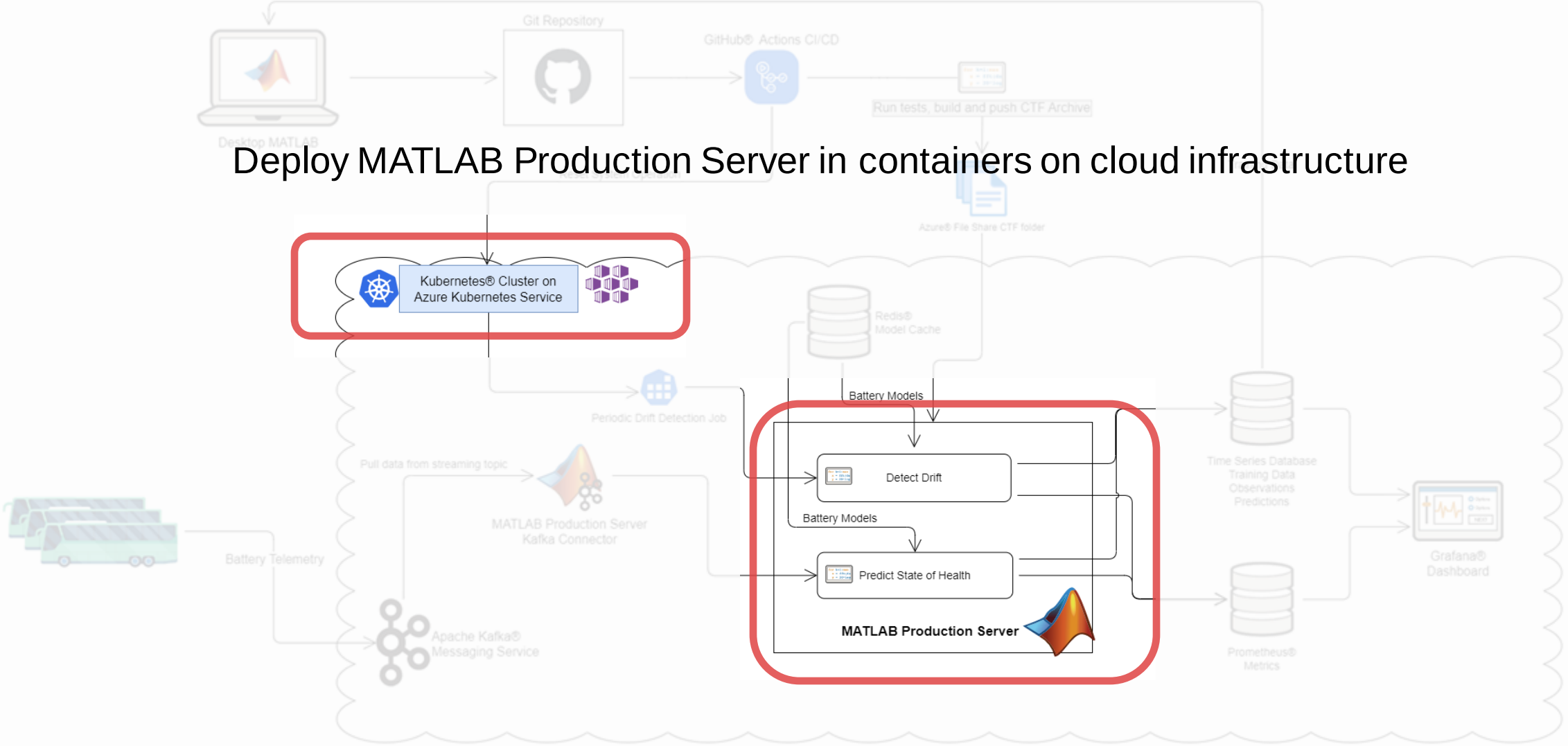


A scalable production system running on the cloud, using industry standard tools

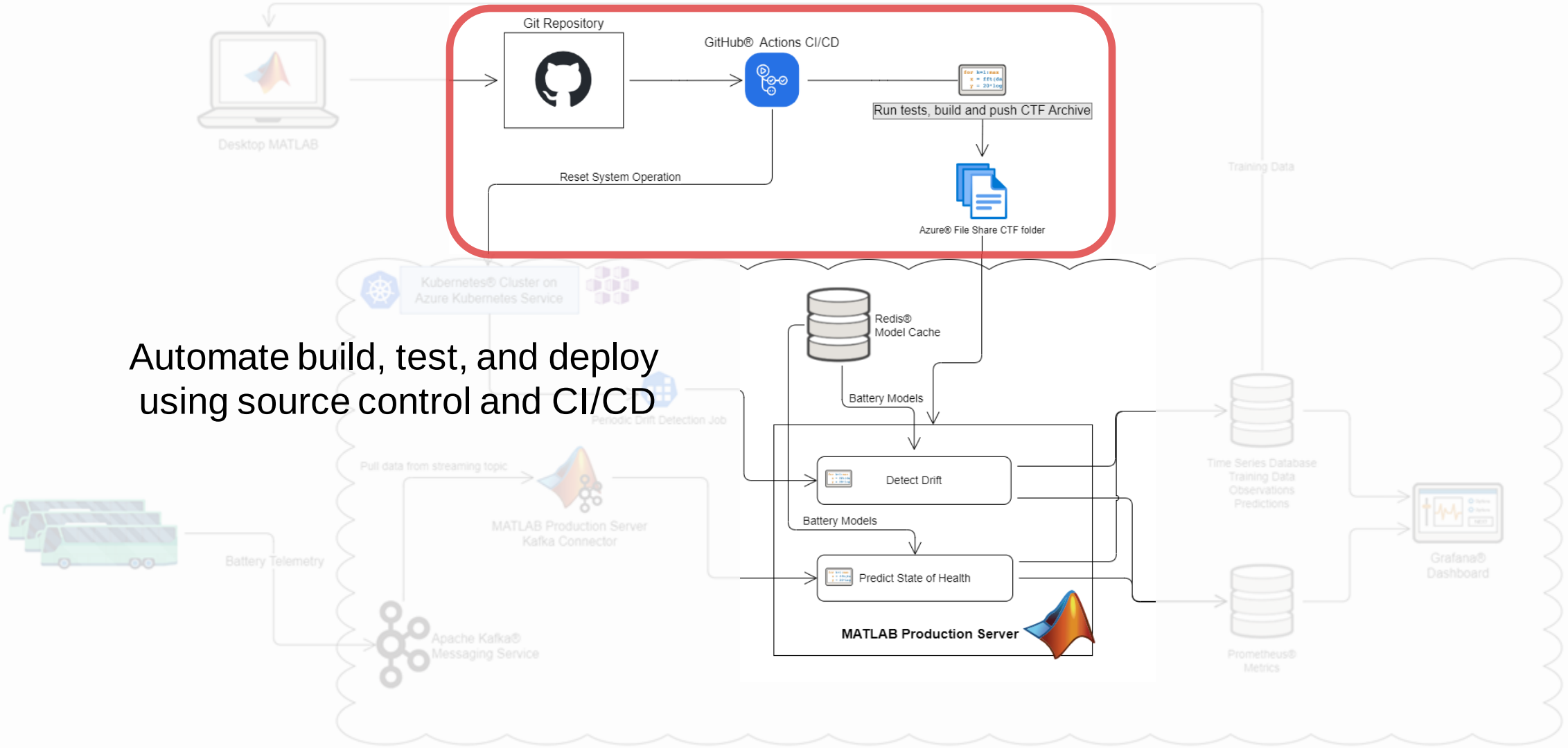


A scalable production system running on the cloud, using industry standard tools

Deploy MATLAB Production Server in containers on cloud infrastructure

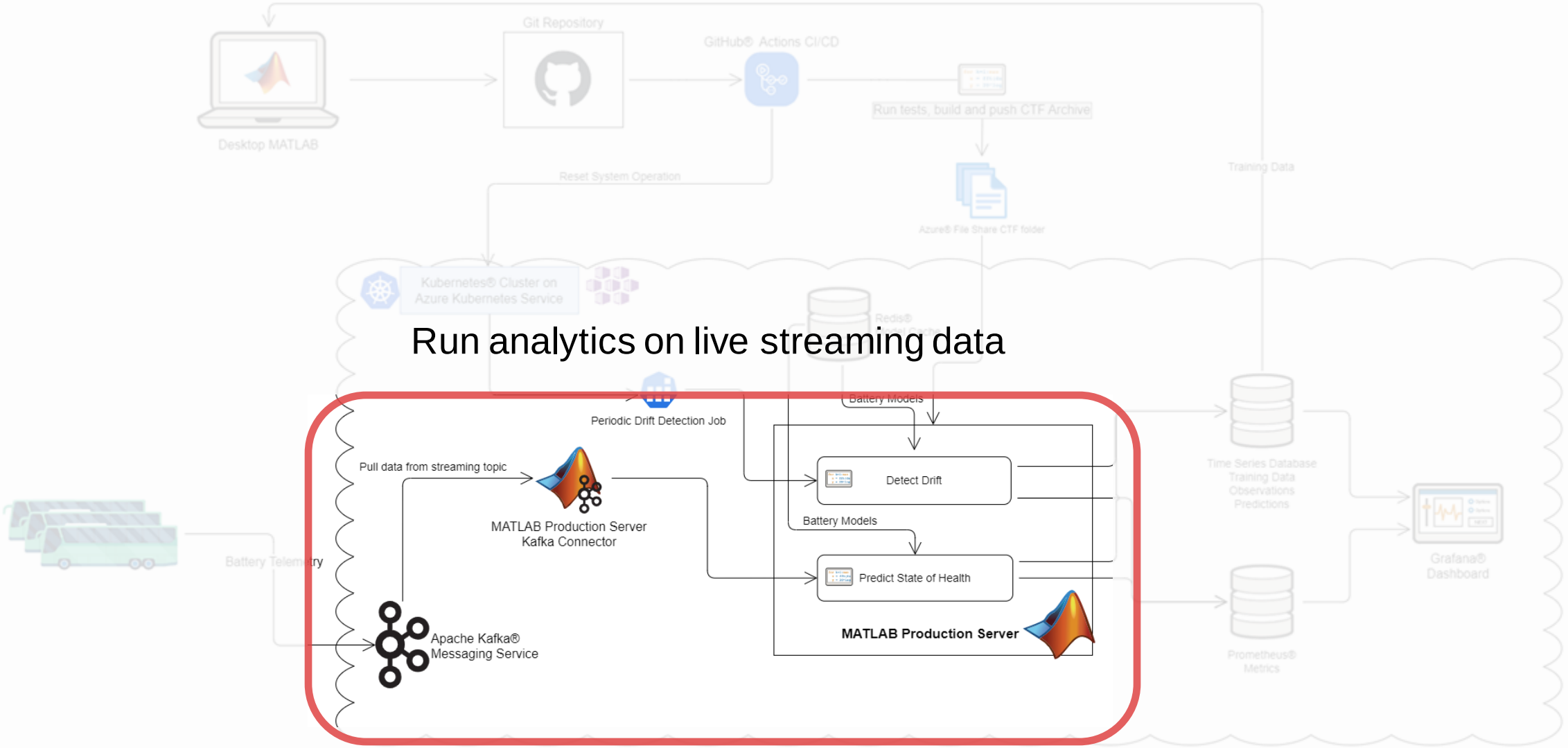


A scalable production system running on the cloud, using industry standard tools



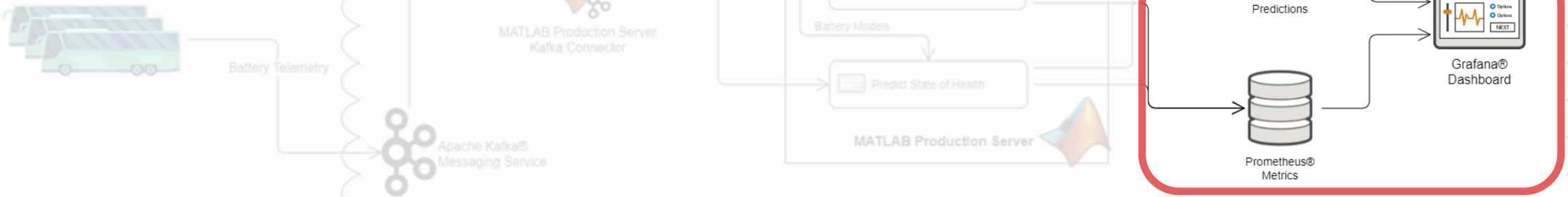
Automate build, test, and deploy using source control and CI/CD

A scalable production system running on the cloud, using industry standard tools

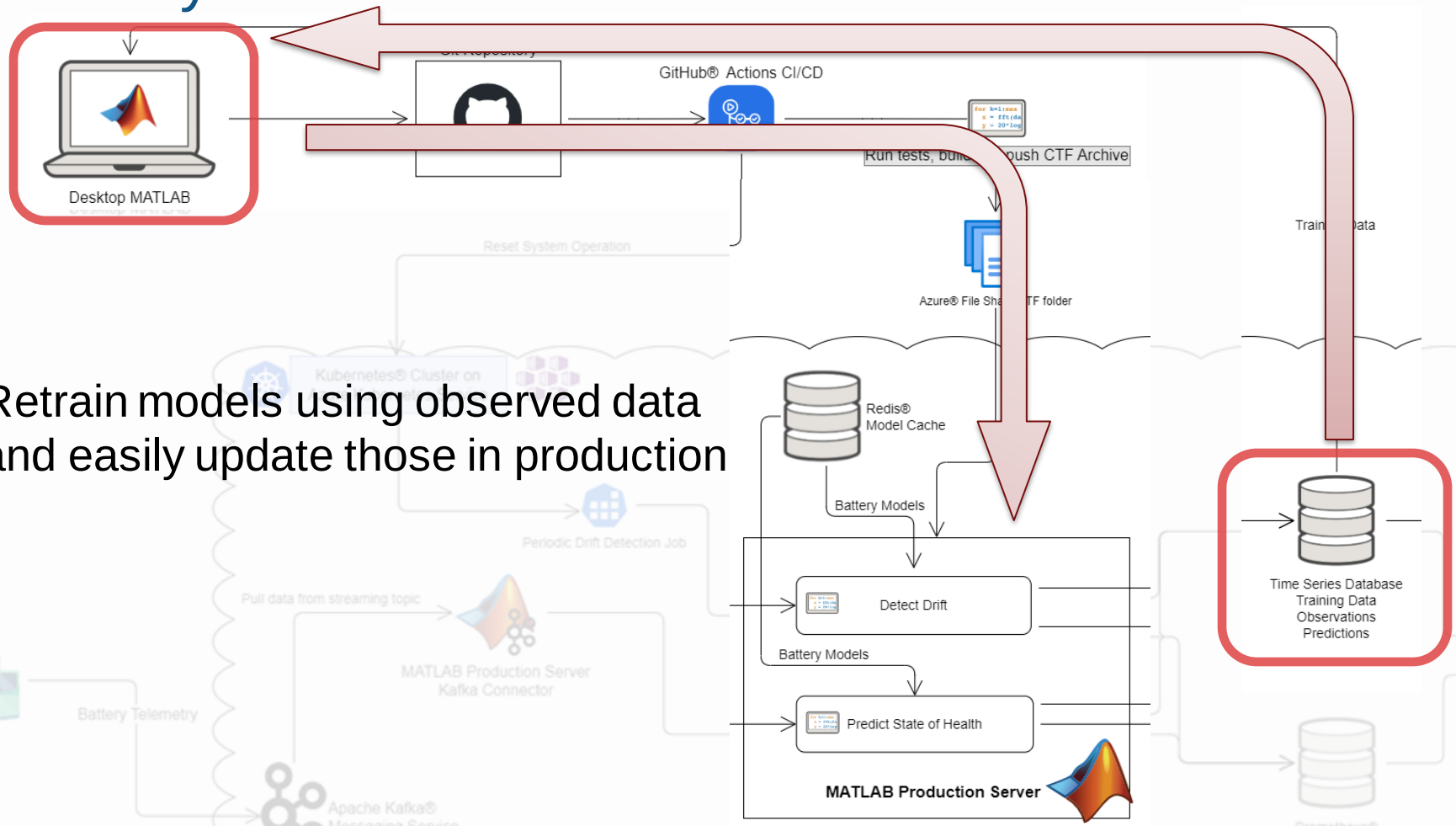


A scalable production system running on the cloud, using industry standard tools

Expose valuable metrics to business stakeholders using industry-standard dashboard integrations



A scalable production system running on the cloud, using industry standard tools



Retrain models using observed data and easily update those in production

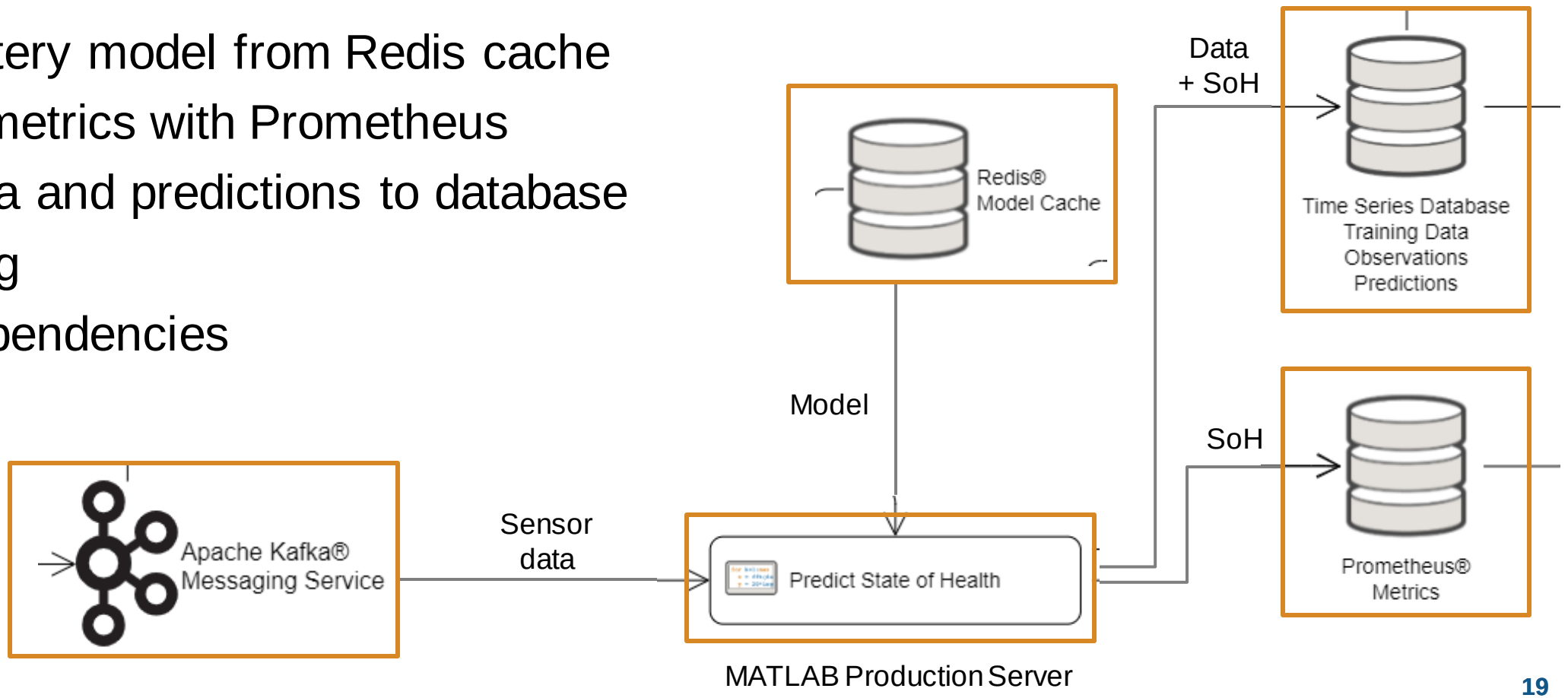
State of health algorithm in production

Production System

- Receive sensor data as kafka stream
- Load battery model from Redis cache
- Expose metrics with Prometheus
- Save data and predictions to database

Local testing

- Mock dependencies



MATLAB R2023a

HOME | PLOTS | APPS | EDITOR | PUBLISH | VIEW

Search Documentation

New | Open | Save | Compare | Print — FILE

Go To | Find | Bookmark — NAVIGATE

Refactor — CODE

Profiler | Analyze — ANALYZE

Insert Method | Insert Parameters — TEST

Run Section | Section Break | Run and Advance | Run to End — SECTION

Run Tests | Run Current Test | Step | Stop — RUN

C: ▸ Development ▸ Github ▸ esteiner ▸ Battery-Health-Estimation-Streaming-Demo ▸ BatteryHealthSystem ▸ SoHPrediction ▸

Current Folder

Name | Git

Folder

Data... · log · Mod... · Pack... · Para... · Reca... · reso... · shut... · startup · Testi...

Details

Command History

ctrl = mps...
stop(ctrl);
mps.cache....
params.kaf...

Workspace

Name | Va

Editor - SoHPredictionUnitTests.m

Project - SoHPrediction

SoHPredictionUnitTests.m | predictSoH.m

```
10      end
11
12      methods (Test)
13          function smokeTest(testCase)
14              % Test Kafka streams
15              inKS = testStream(Rows=params.kafkaRowSize);
16              s = load(fullfile("Testing","sampleData.mat"));
17              data = s.observations;
18              writetimetable(inKS,data);
19
20              esp = eventStreamProcessor(inKS, @predictSoH, @initRedis, ...
```

Test Browser

tests.SoHPredic
tests.SoHPre

Command Window

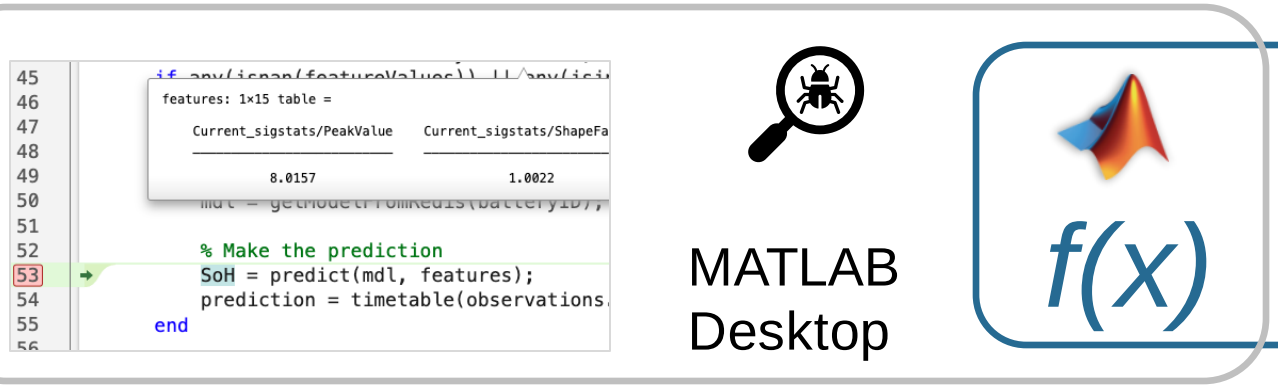
```
Using mock function for writeToInfluxDB.
.
Done tests.SoHPredictionUnitTests
```

>>

Write SoH prediction function to use kafka streams

Simulate streams with
- `inMemoryStream`
- `testStream`

Use production Kafka streams with `kafkaStream`



Debug locally, then deploy the same MATLAB code to production.

Automatically build, test, package, and deploy MATLAB code

esteinerMW / Battery-Health-Estimation-Streaming-Demo Private

forked from mathworks/Battery-Health-Estimation-Streaming-Demo

Watch 0 Fork 2 Star 1

Code Pull requests Actions Projects Wiki

Build and upload deployable archive (CTF) to MATLAB Production Server

update battery dashboard #19

Summary

Jobs

build

Reset Demo Operations

Run details

Usage

Workflow file

build

succeeded 12 hours ago in 3m

Set up job

Run actions/checkout@v3 3s

Setup MATLAB Support Packages 0s

Setup MATLAB

Run MATLAB buildtool

1 ▶ Run matlab-actions/run-build@v1

8 ▶ Generate script

10 ▶ Run command

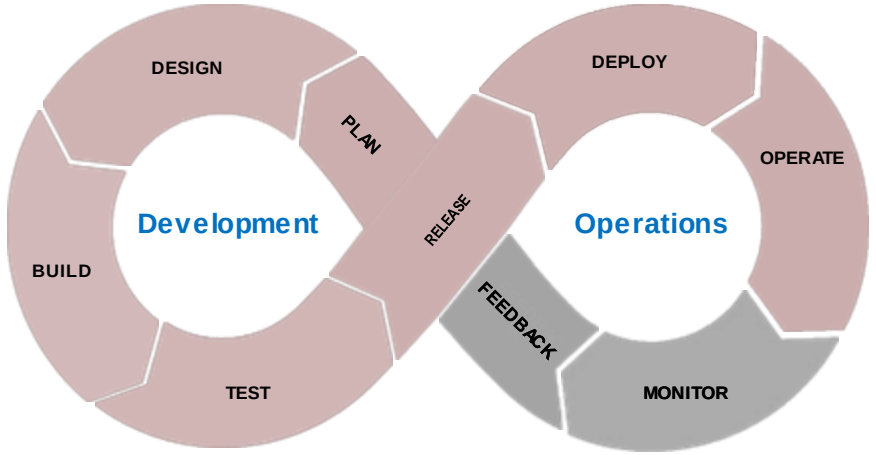
Azure login

Azure CLI script - upload CTF to az file-share

Post Run actions/checkout@v3

Complete job

```
- name: Run MATLAB buildtool
uses: matlab-actions/run-build@v1
with:
tasks: packageDriftDetection
packageSoHPrediction
```




```
function plan = buildfile
plan = buildplan(localfunctions);
plan("packageDriftDetection").Dependencies = "test";
plan("packageSoHPrediction").Dependencies = "test";
plan("test").Dependencies = "validate";
end
```

Batteries - Dashboards - Grafa

Not Secure | batteryhealth.mwlab.io/d/H0NdyV5Vk/batteries?orgId=1&refresh=5s&from=now-30m&to=now

General / Batteries

Last 30 minutes

Battery All Feature All

Notes (1 panel)

Battery and Model Health

Battery Health

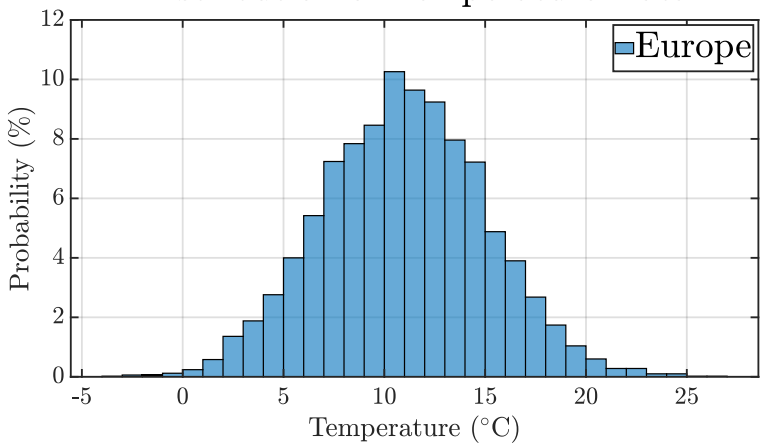
1	2	3	4	5	6	7	8	9	10
81.0%	82.9%	83.7%	87.2%	81.4%	82.3%	88.6%	96.7%	85.6%	89.7%
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
84.4%	81.7%	85.5%	84.6%	79.5%	83.7%	84.0%	82.3%	81.0%	93.5%

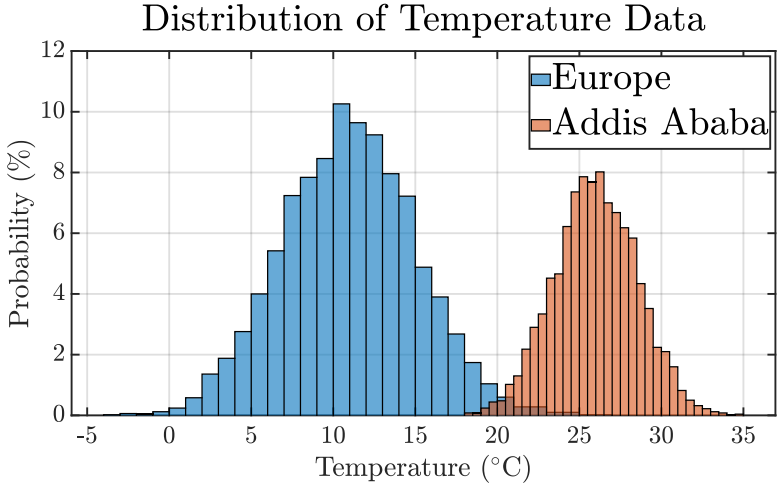
Battery 1 Health and Drift Status

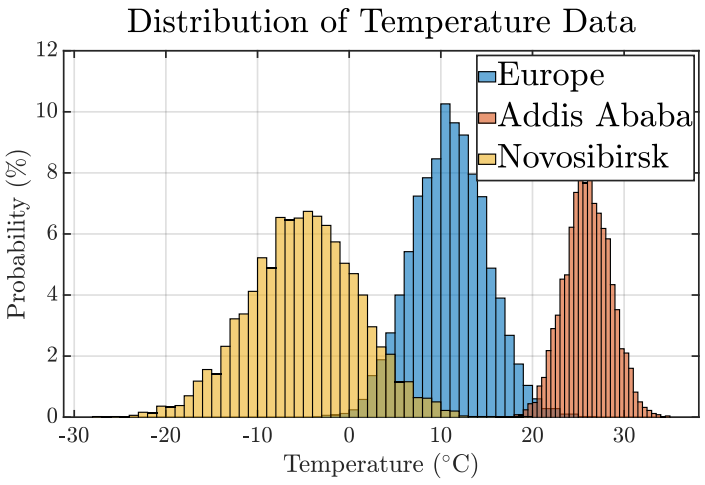
Battery 1 SoH

Battery 1 All Drift Status

The static data assumption rarely holds in the real world

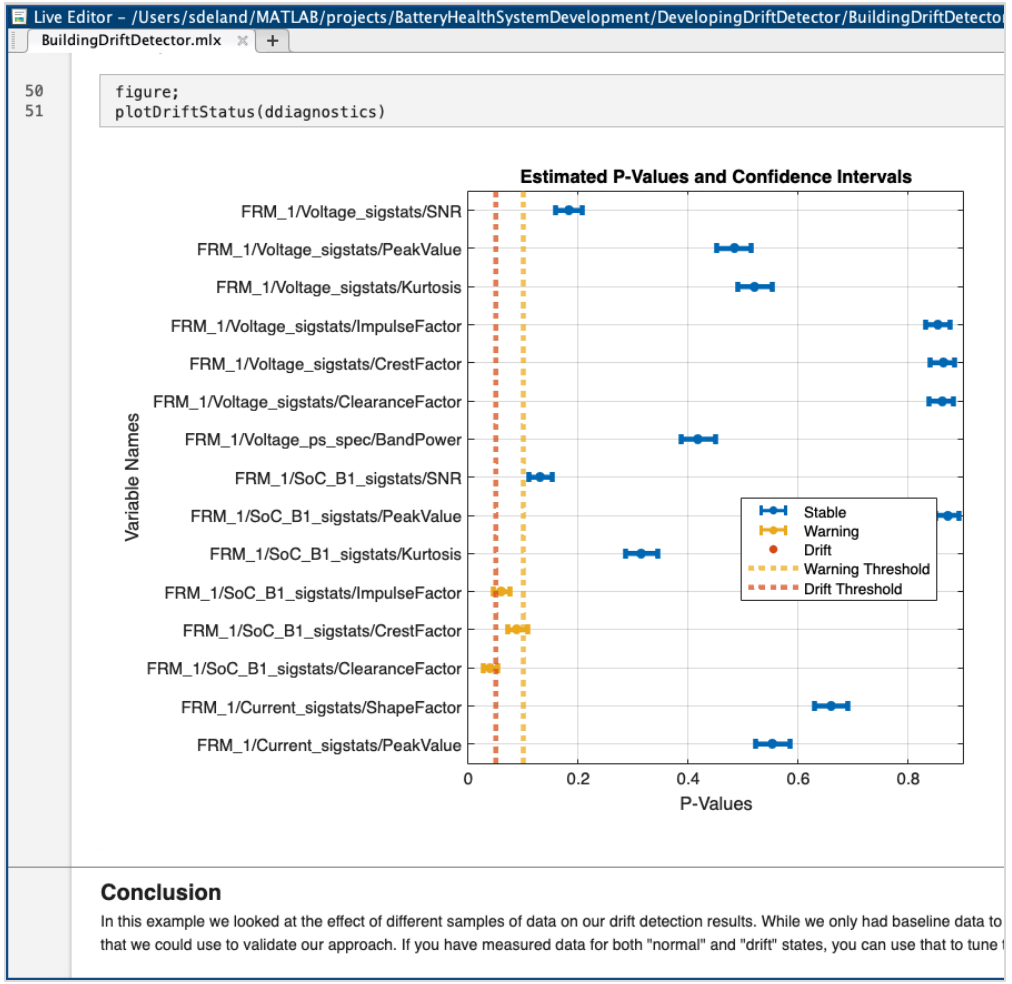






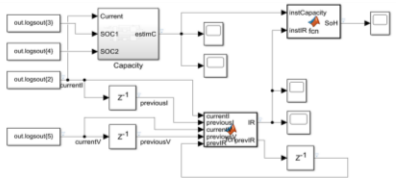
Developing drift detection with `detectdrift`

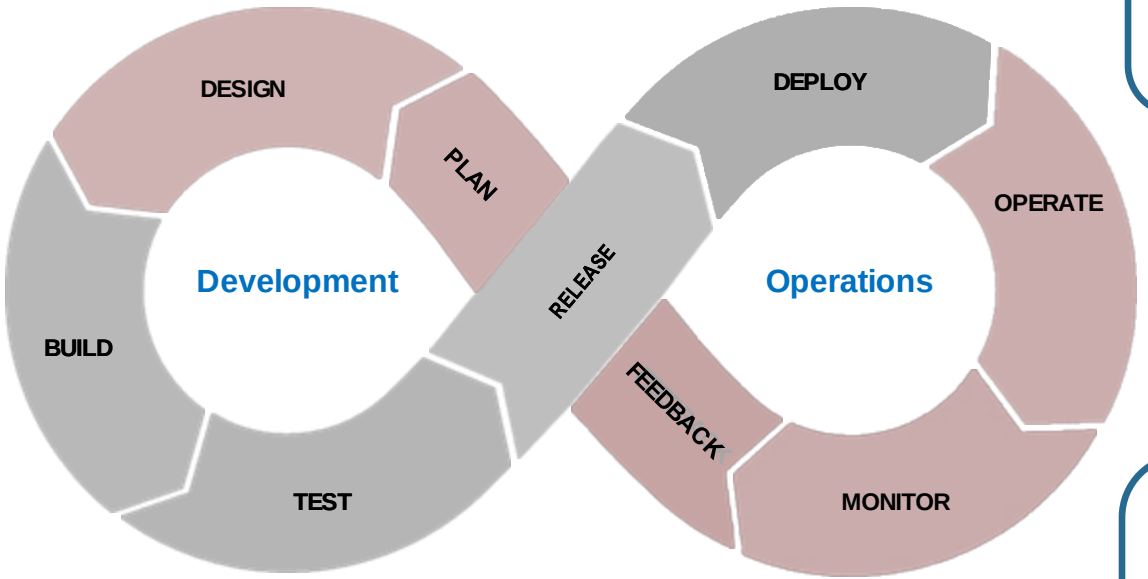
- Use historical data (training data) to create a baseline distribution
- Generate synthetic data to test for drift
  - This will be replaced by streaming data in the production system
- Out-of-Distribution Detection for Deep Neural Networks



Update model when drift is detected

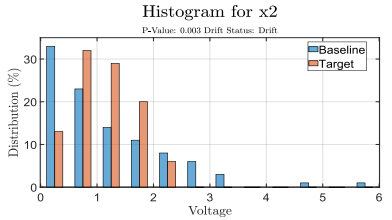
Data labeling





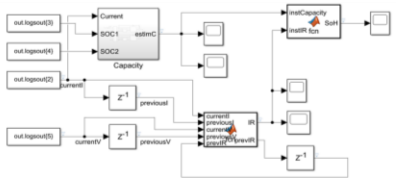
$\hat{f}_1(x)$

Drift Detection

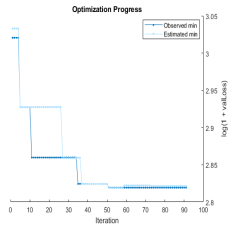


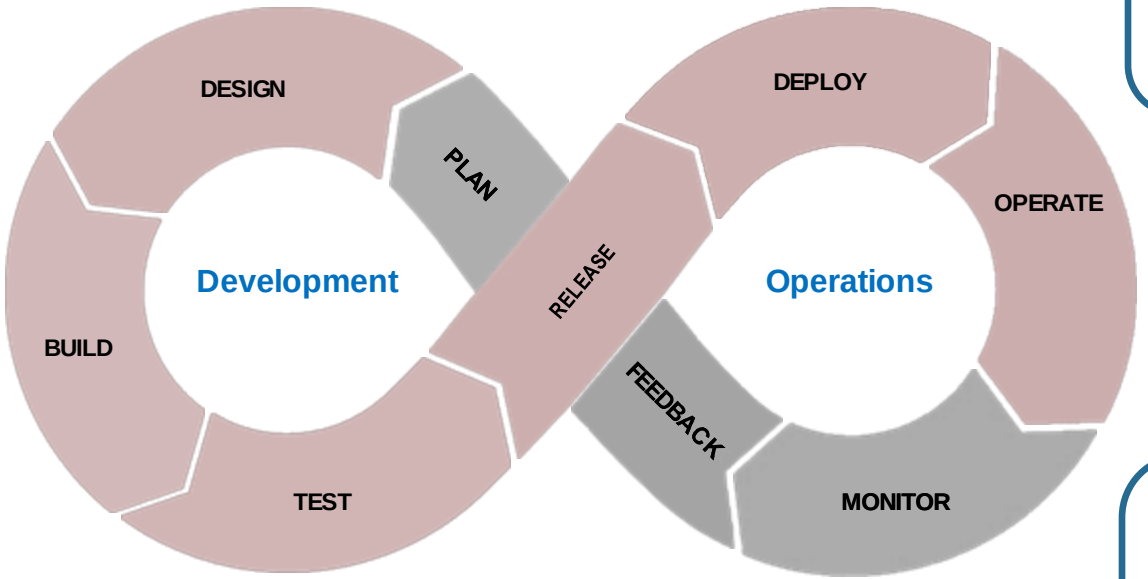
Update model when drift is detected

Data labeling



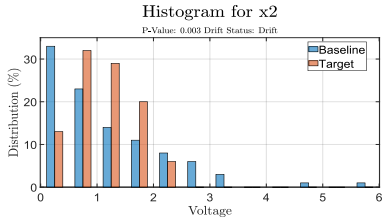
Retrain



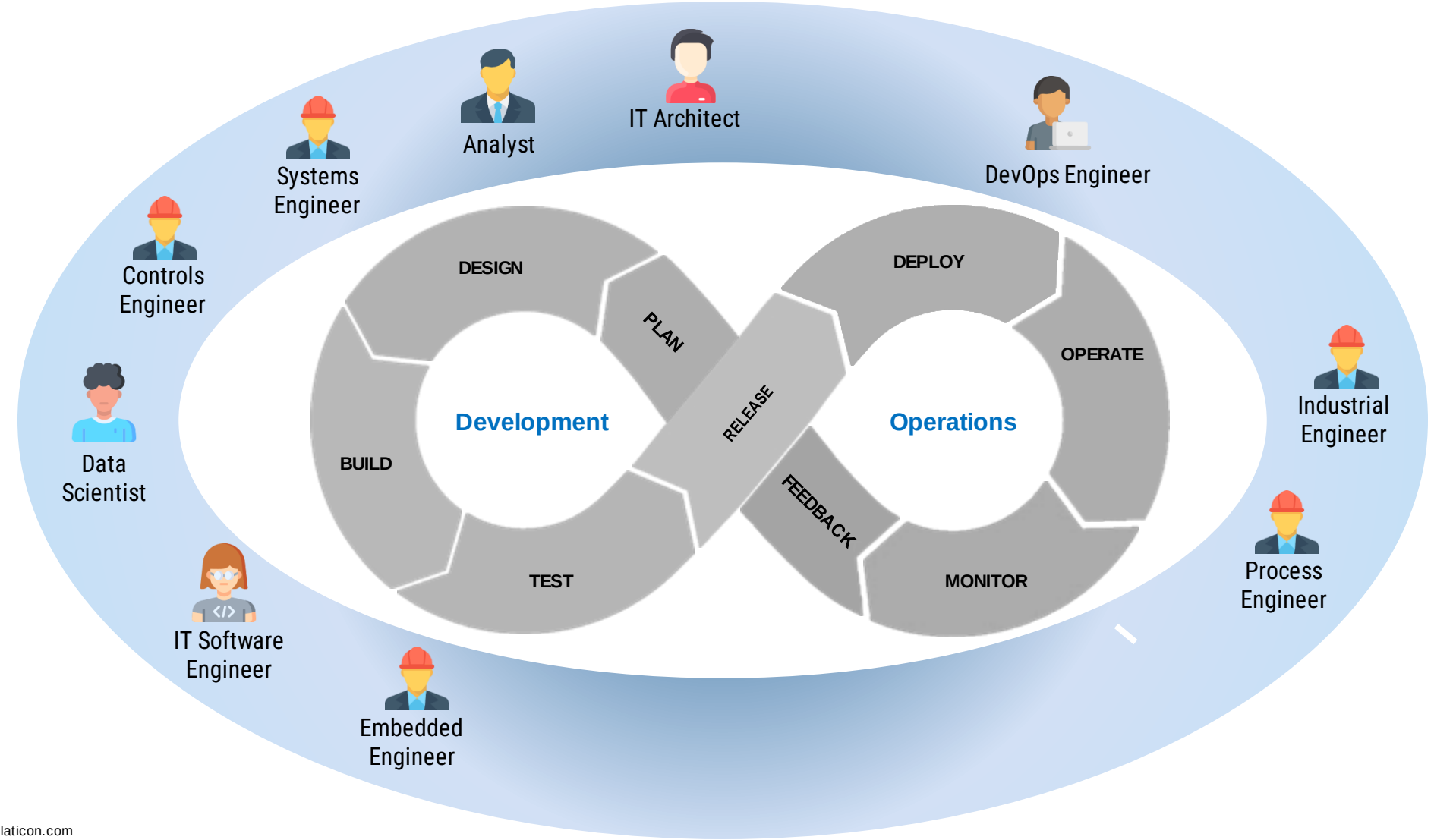


$$\hat{f}_2(x)$$

Drift Detection



Diverse teams across Engineering, IT, Line of Business, and Operations must collaborate to achieve success in DevOps



Icons made by Freepik from www.flaticon.com

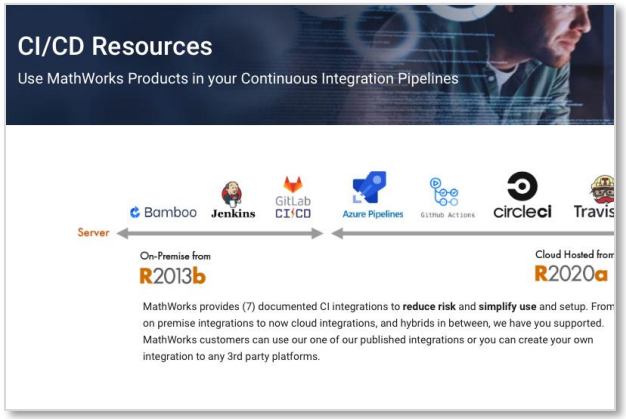
Key Takeaways

- **Incorporate familiar MATLAB capabilities**, including Predictive Maintenance and Drift Detection, in operations
- **Integrate with production systems** like data sources and dashboards, and translate those integrations from desktop to cloud servers
- **Automatically build, test, package, and deploy MATLAB code and Simulink models** with CI/CD

Learn More

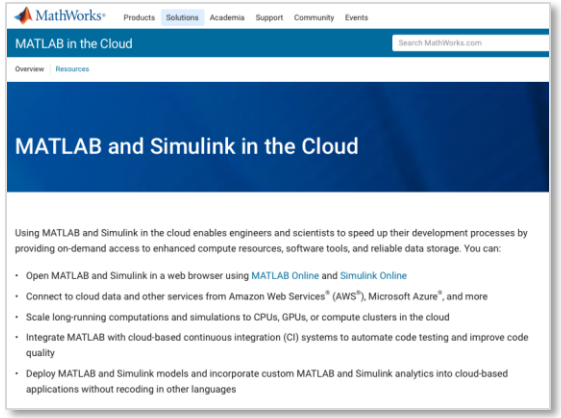
How MATLAB and Simulink are used with Enterprise IT

https://www.mathworks.com/solutions/enterprise-it-systems.html



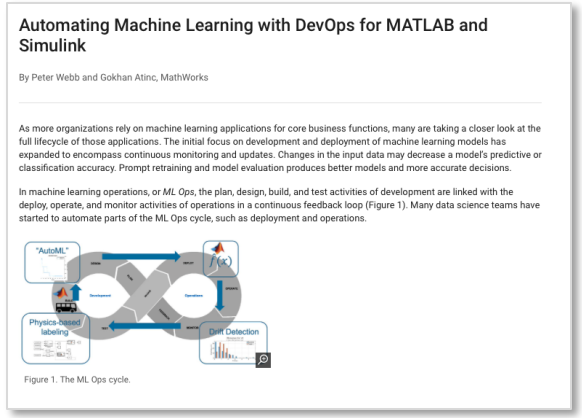
CI/CD Resources

https://www.mathworks.com/solutions/enterprise-it-systems/ci-cd.html



MATLAB and Simulink in the Cloud

https://www.mathworks.com/solutions/cloud.html



Automating Machine Learning with DevOps for MATLAB and Simulink

https://www.mathworks.com/company/newsletters/articles/automating-machine-learning-with-devops-for-matlab-and-simulink.html

MATLAB EXPO

Thank you

Questions?

Attributions

- Apache, Apache Kafka, Kafka and the Kafka logo are trademarks of the Apache Software Foundation. The Apache Software Foundation has no affiliation with and does not endorse the materials provided at this event.
- The Grafana Labs Marks are trademarks of Grafana Labs, and are used with Grafana Labs' permission. We are not affiliated with, endorsed or sponsored by Grafana Labs or its affiliates.
- Microsoft, Azure, Azure Kubernetes Service, GitHub, GitHub Actions, and their associated logos are trademarks of the Microsoft group of companies.
- Prometheus, Kubernetes, and their associated logos are registered trademarks of The Linux Foundation.
- Redis is a registered trademark of Redis Ltd. Any rights therein are reserved to Redis Ltd. Any use by MathWorks is for referential purposes only and does not indicate any sponsorship, endorsement or affiliation between Redis and MathWorks