Integrating Digital Twin & AI for Smarter Engineering Decisions

Amit Doshi
Sr. Application Engineer- Data Science, MathWorks
adoshi@mathworks.com

ODSC-India, 8 Aug 2019

Key Takeaways

- Understanding concepts of Digital Twin
- Key techniques to build a Digital Twin – use case
- Application of Digital twin in AI

Why do sometimes well-designed **engineering** systems fail or underperform?

- Example: faulty braking system leads to windmill disaster
  - https://youtu.be/-YJuFvjtMos?t=39s

- Systems like these cost millions of dollars; also, frequent maintenance and unexpected failures are a large cost in many industries

- Many failures are hazardous

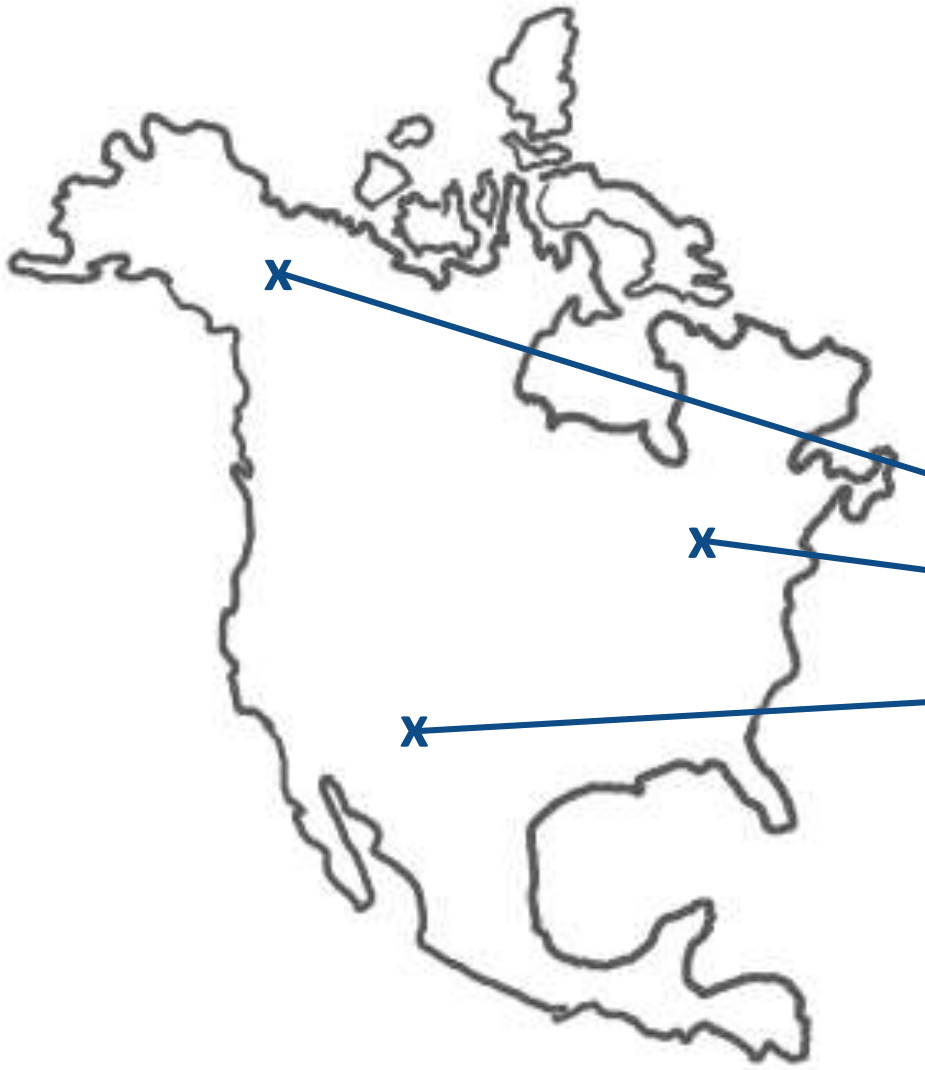
Solution: Digital Twin?

Digital Twin: What?

A Digital Twin is an **up-to-date representation** of a real asset in operation

Cyber objects Digital-Avatars Cyber Physical System

A Digital Twin is an up-to-date representation of a real asset in operation.



Cost of rig: >$1M
Repair cost: $100,000
Cost of valve: $200

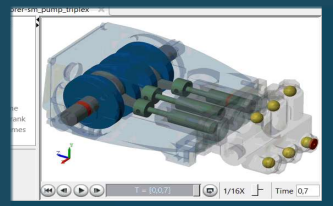
A Digital Twin is an up-to-date representation of a real **asset** in operation.

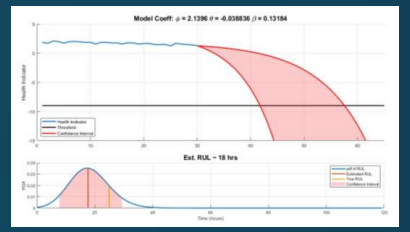
Component

System

System of systems

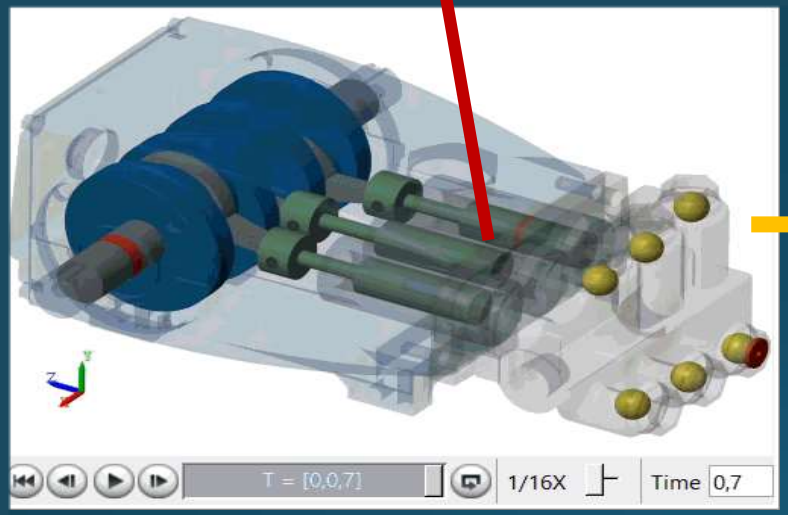
Asset

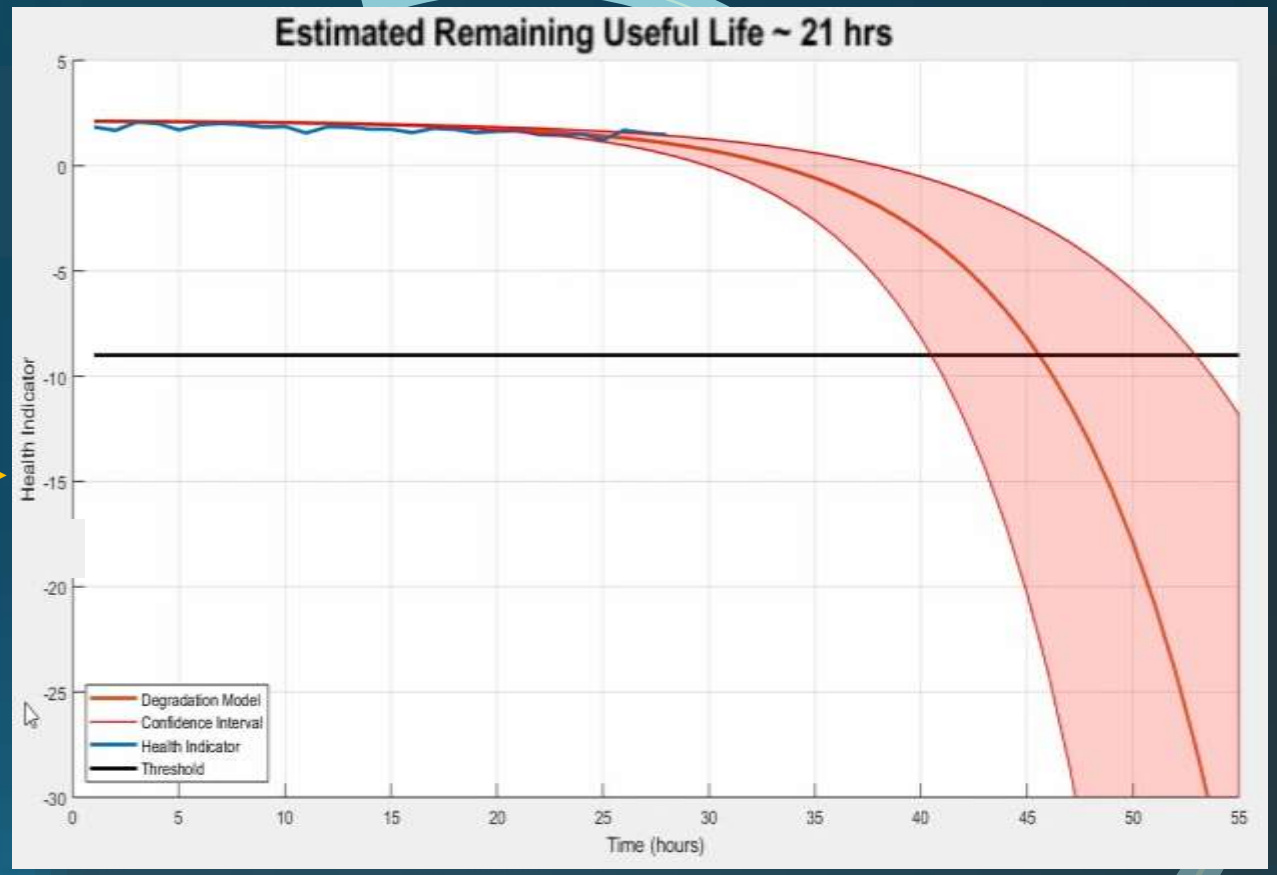




Digital Twin

Visual Indication of Fault in Model

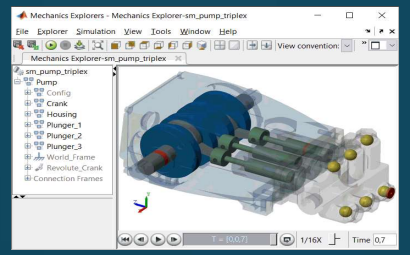


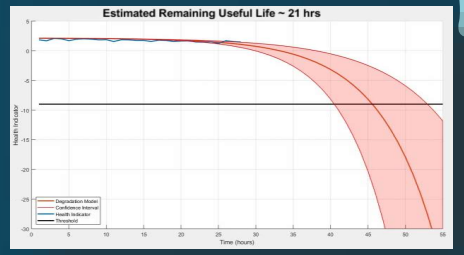


Condition Monitoring

Predictive maintenance

Asset





Digital Twin

ALARM:
Pump-15, location- Rocky Mountain Site, needs Urgent maintenance.
One of it‘s cylinders is blocked. It will shut down your line in 15 hours.

Inventory management
Run what-if simulations
Operational planning

Digital Twin: How?

Use Case: Fault Classification of a Pump

Objective:
- Using machine data to determine what needs to be fixed

Solution:
- Develop Predictive Maintenance Algorithm

Raw Data →

Predictive maintenance algorithm

Identify fault type →

- Healthy
- Blocked Inlet
- Seal Leakage
- Worn Bearing
- Blocked Inlet & Worn Bearing
- Seal Leakage & Worn Bearing
- Seal Leakage & Blocked Inlet

Use Case: Fault Classification of a Pump

Objective:
- Using machine data to determine what needs to be fixed

Solution:
- Develop Predictive Maintenance Algorithm

Challenge:
- Failure sensor data is not available

Raw Data

Flow

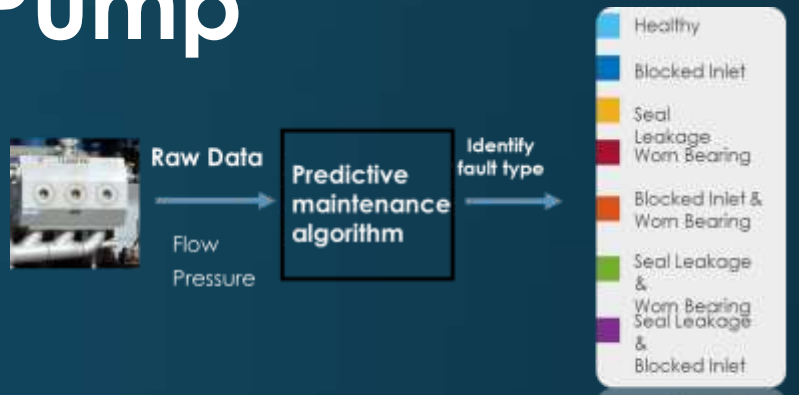
Pressure

Predictive maintenance algorithm

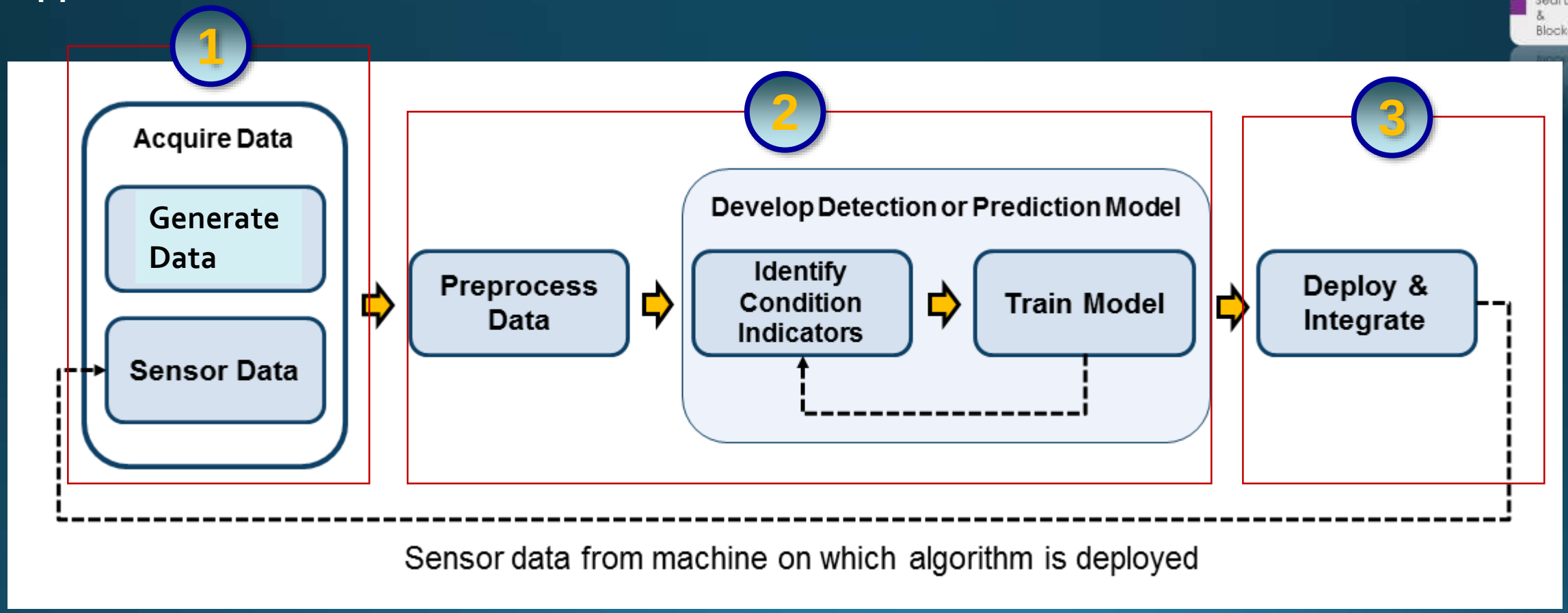
Identify fault type

- Healthy
- Blocked Inlet
- Seal Leakage
- Worn Bearing
- Blocked Inlet & Worn Bearing
- Seal Leakage & Worn Bearing
- Seal Leakage & Blocked Inlet

Use Case: Fault Classification of a Pump

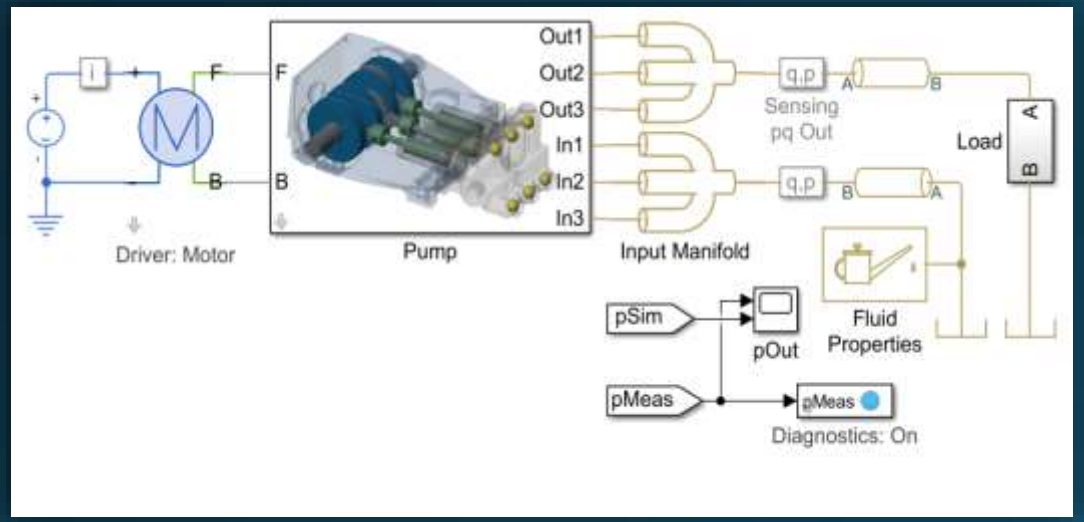


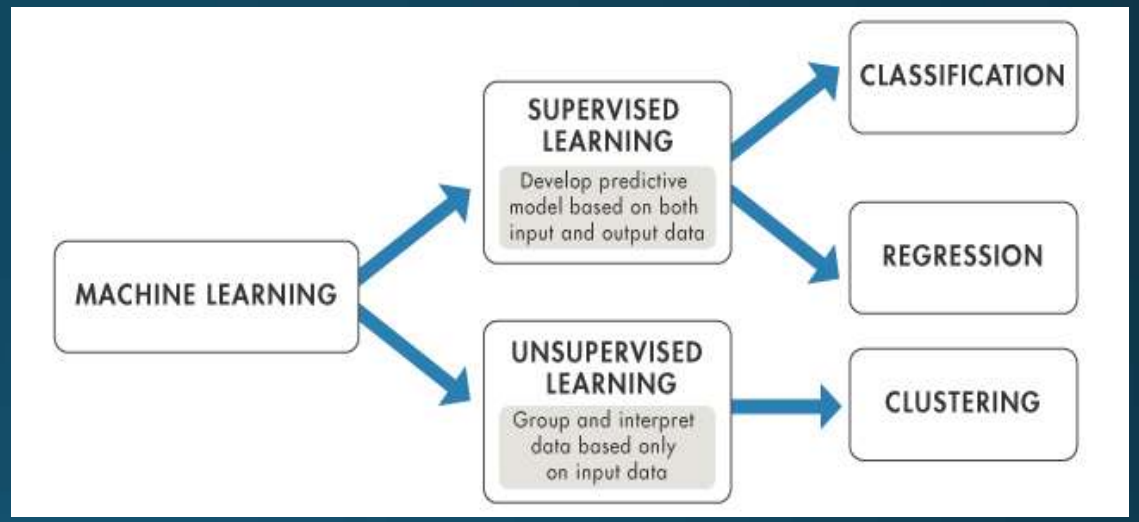
Approach



Digital Twin: How?

Physics-based modeling ⟷ **Data-driven modeling**





Captures physics

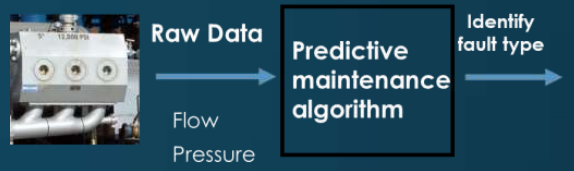
- Transparent
- Sometimes it is not possible to capture all the dynamics, model failures

Captures behavior using sensor data,

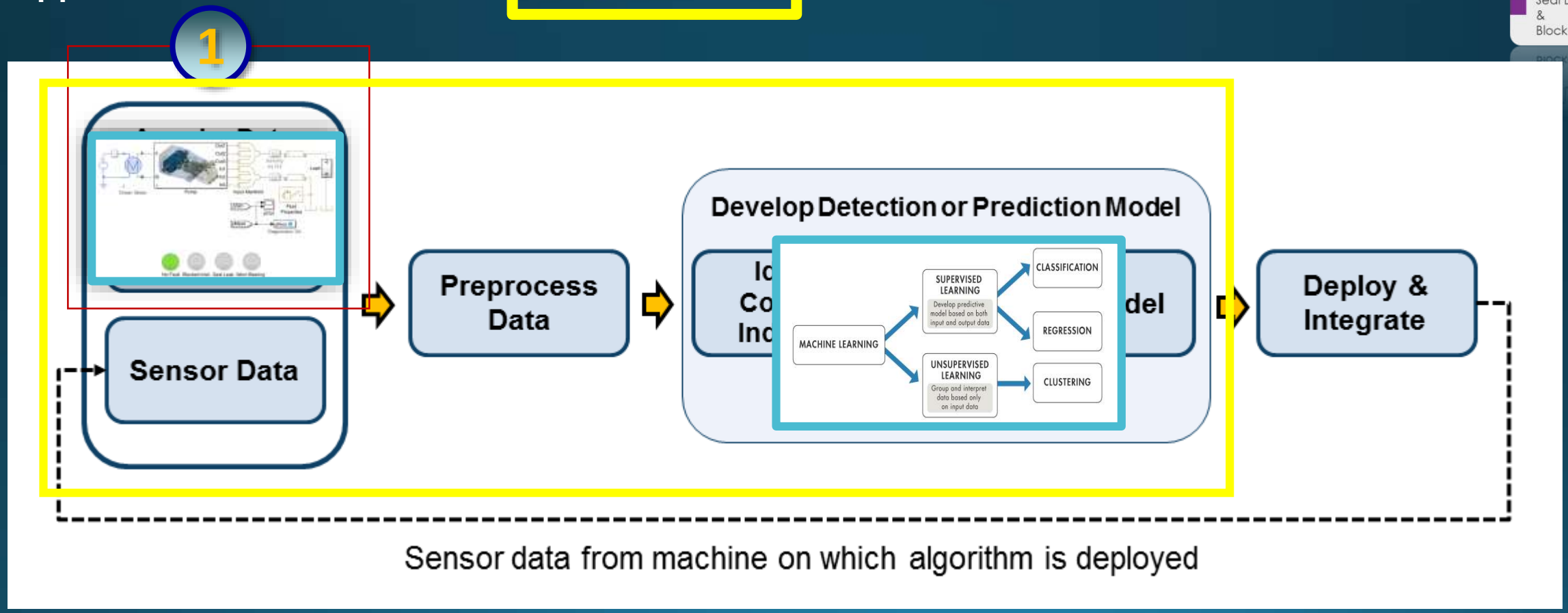
- Could be black box
- Not possible to have sensors for every parameter

Use Case: Fault Classification of a Pump

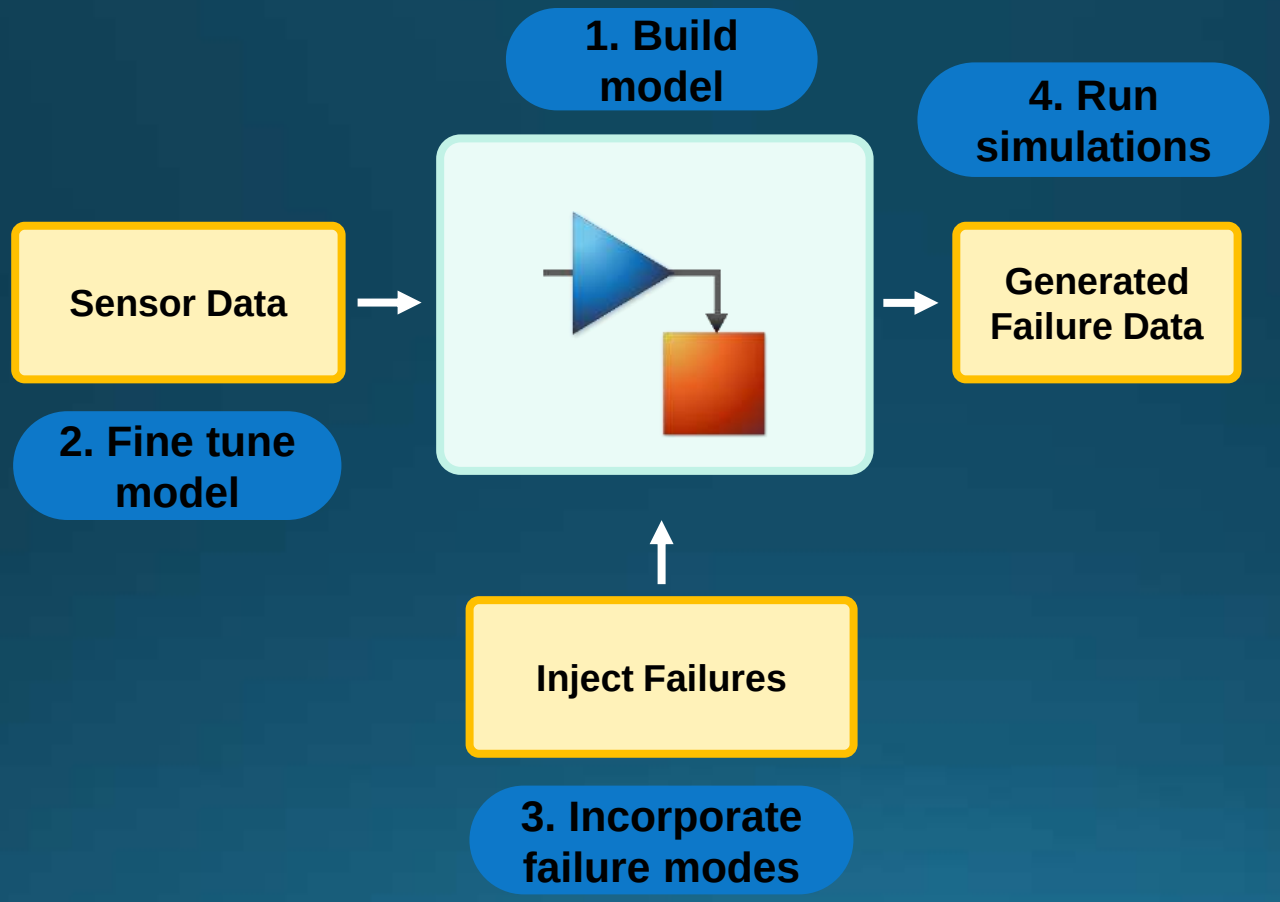
Digital Twin



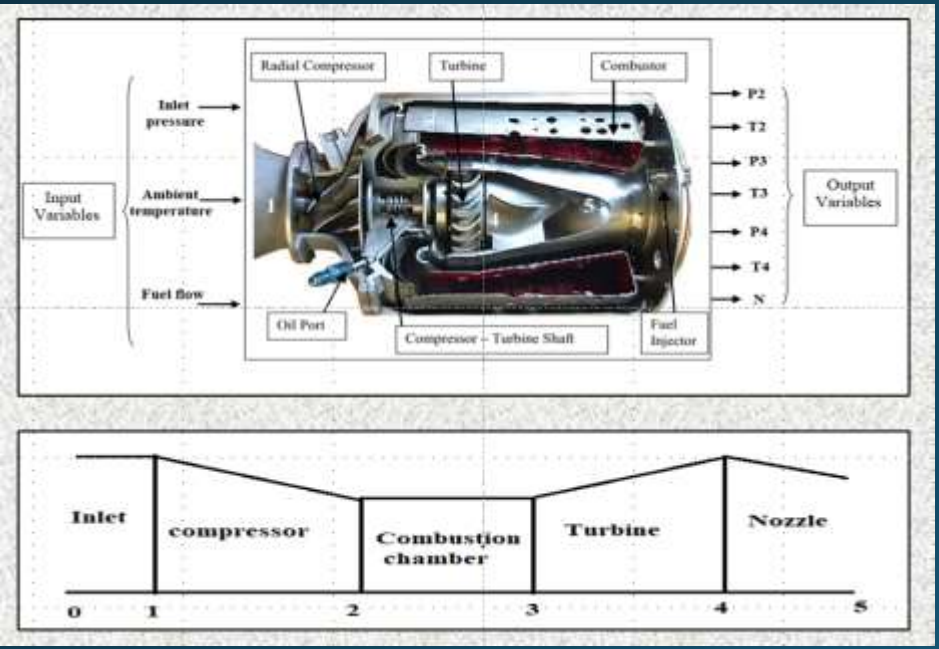
Approach



Building Digital Twin using Physics based approach



Domain Experts



First Principle Modeling Approach

State variable method:

1 - Selection of state Variable

$$x = [P_2 \;\; P_4 \;\; N]$$

2 - Compressor calculation

$$\left(\frac{P_2}{P_1}, \frac{N}{\sqrt{T_1}}\right) \xrightarrow{Compressor\ map} (\dot{m}_c, \eta_c)$$

$$(T_1, \eta_C, PR_{21}) \xrightarrow{isentropic\ equation} (T_2)$$

$$W_c = \dot{m}_c c_p (T_2 - T_1)$$

3 - Combustion chamber

$$(P_2, \sigma_{cc}) = P_3$$

$$(LHV, \dot{m}_c, \dot{m}_f) \xrightarrow{Energy\ balance} T_3$$

4- Turbine calculation

$$\left(\frac{P_3}{P_4}, \frac{N}{\sqrt{T_3}}\right) \xrightarrow{turbine\ map} (\dot{m}_t, \eta_t)$$

$$(T_3, \eta_t, PR_{34}) \xrightarrow{isentropic\ equation} (T_4)$$

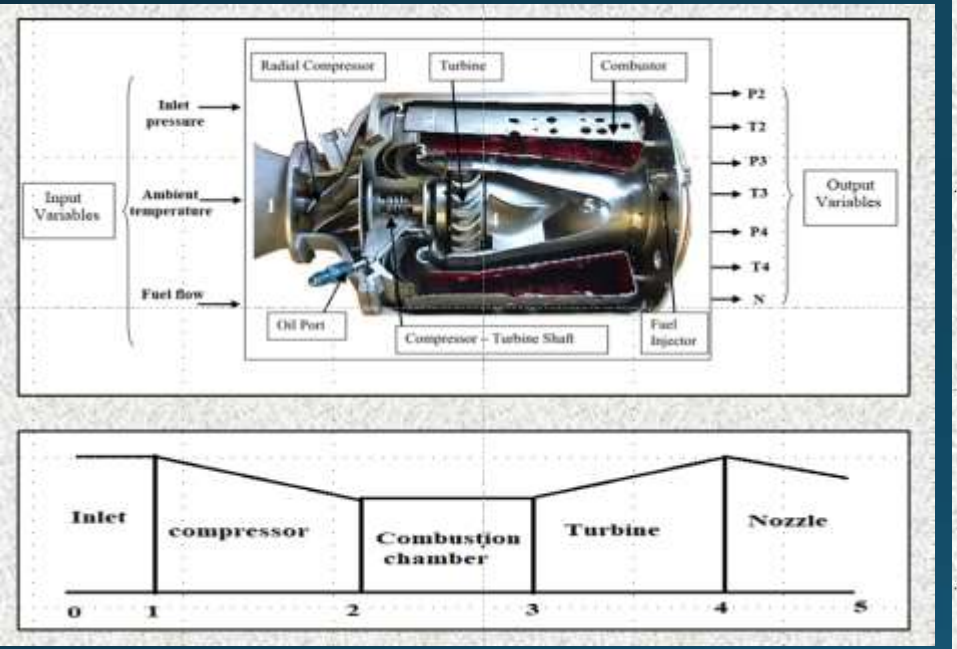
$$W_t = \dot{m}_t c_p (T_3 - T_4)$$

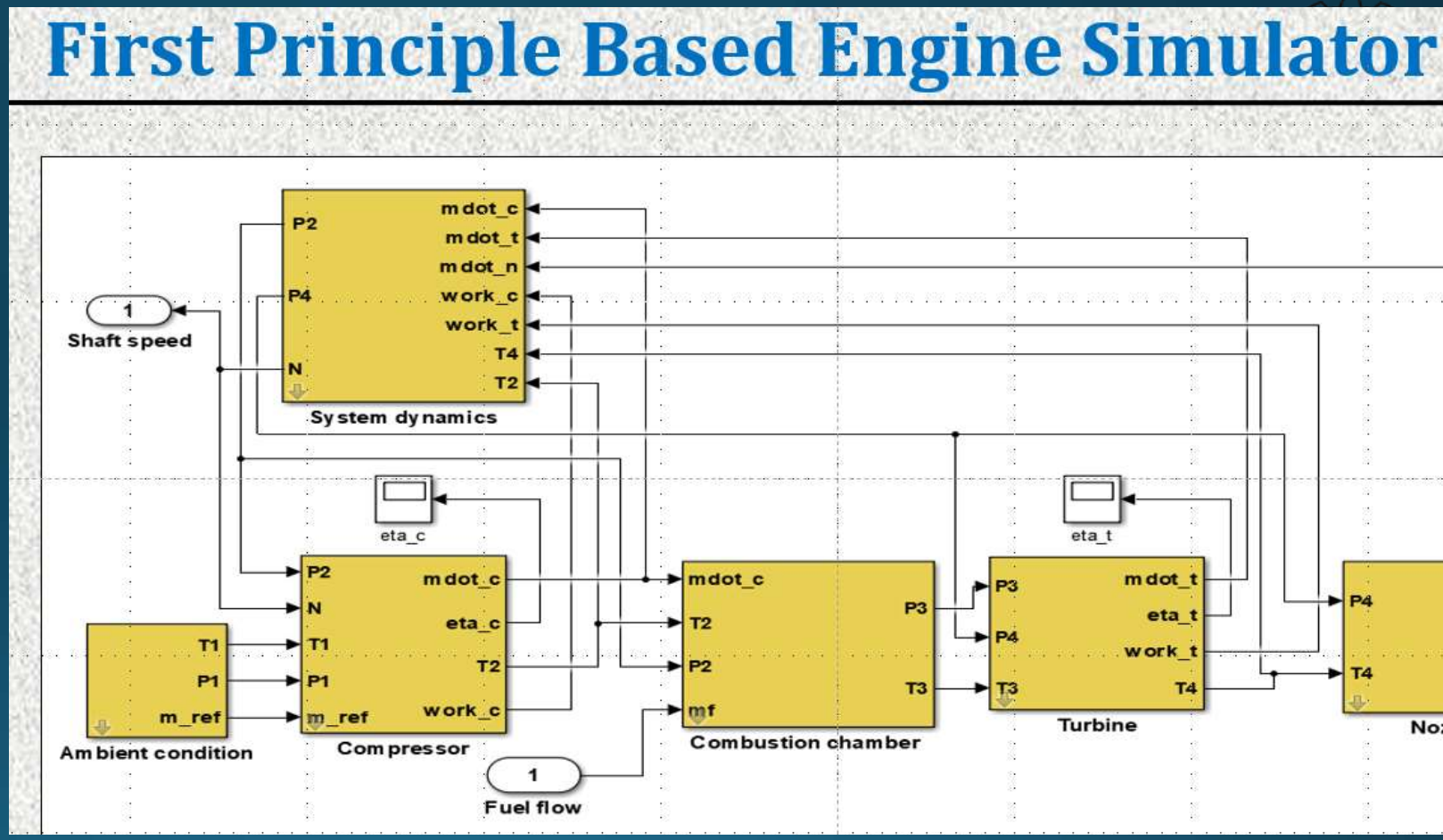
5- Nozzle equation

$$\left(\frac{P_4}{P_5}\right) \xrightarrow{nozzle\ map} (\dot{m}_t, \eta_t)$$

Source: IIT-Bombay, MATLAB-Expo-2018

Domain Experts + Simulation Tool

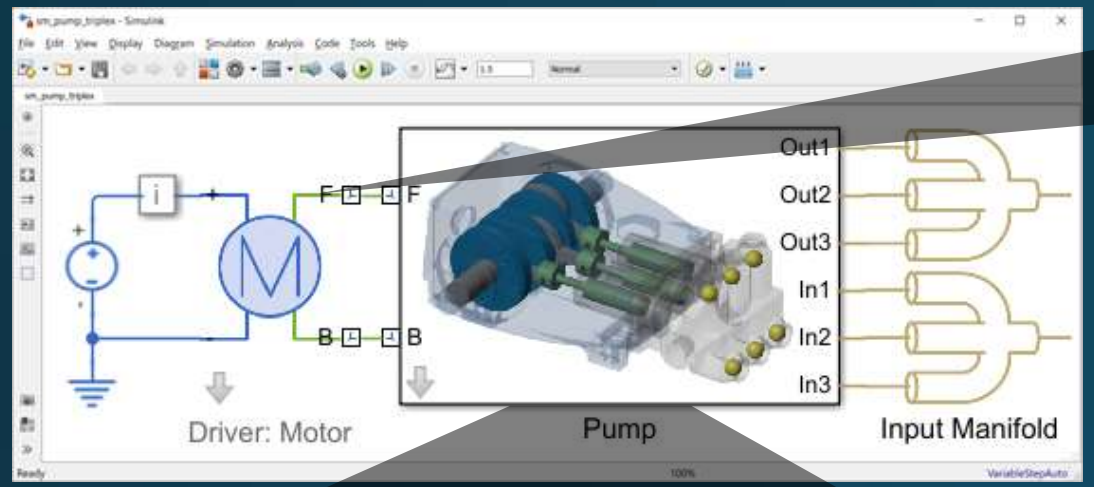


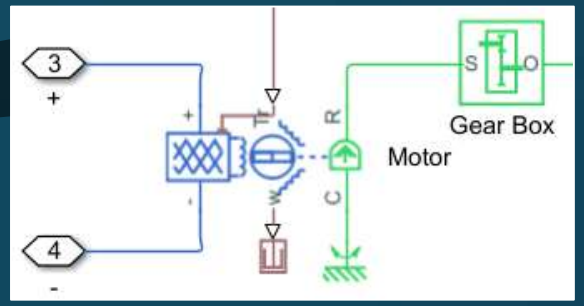


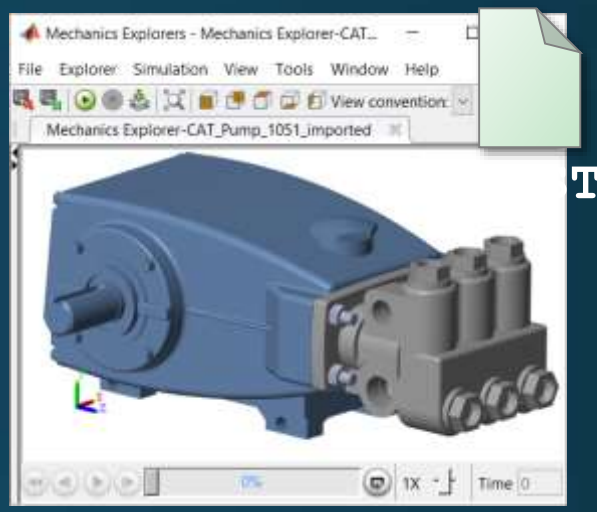
Source: IIT-Bombay, MATLAB-Expo-2018

1. Build model

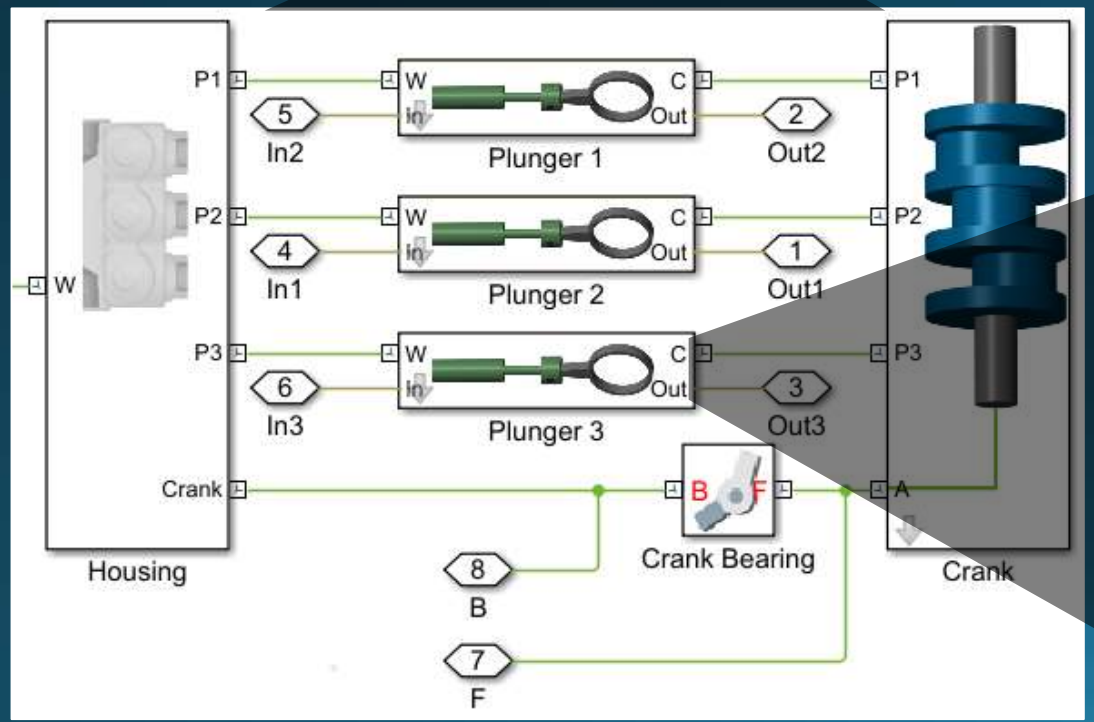
Dynamic Physics Based Model

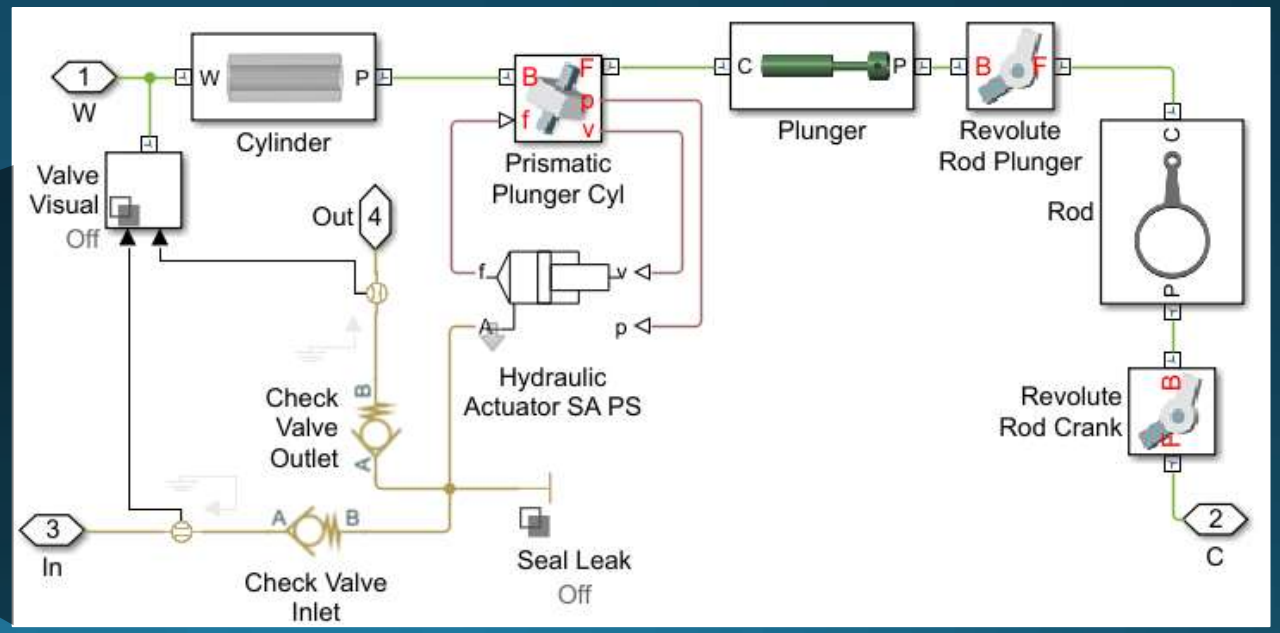






TEP

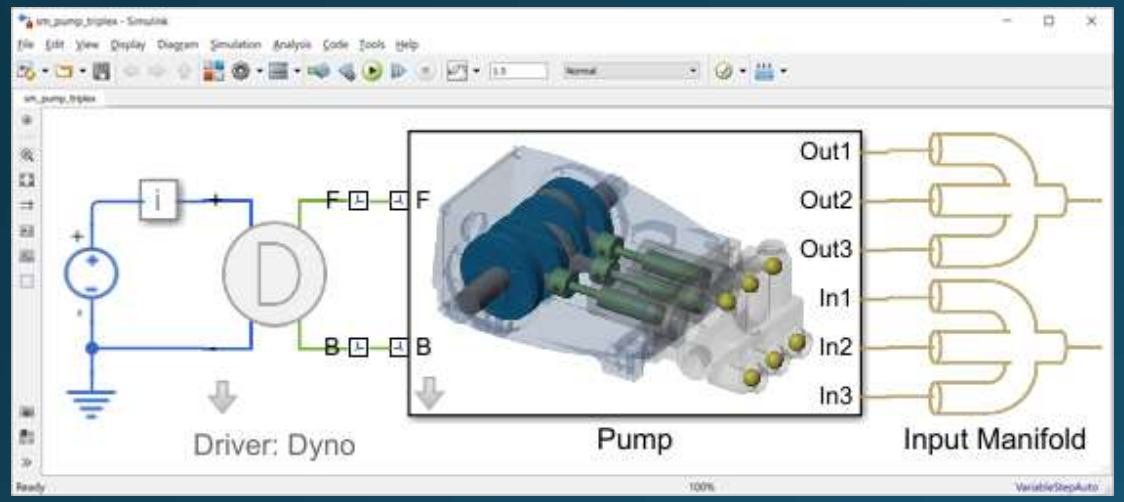


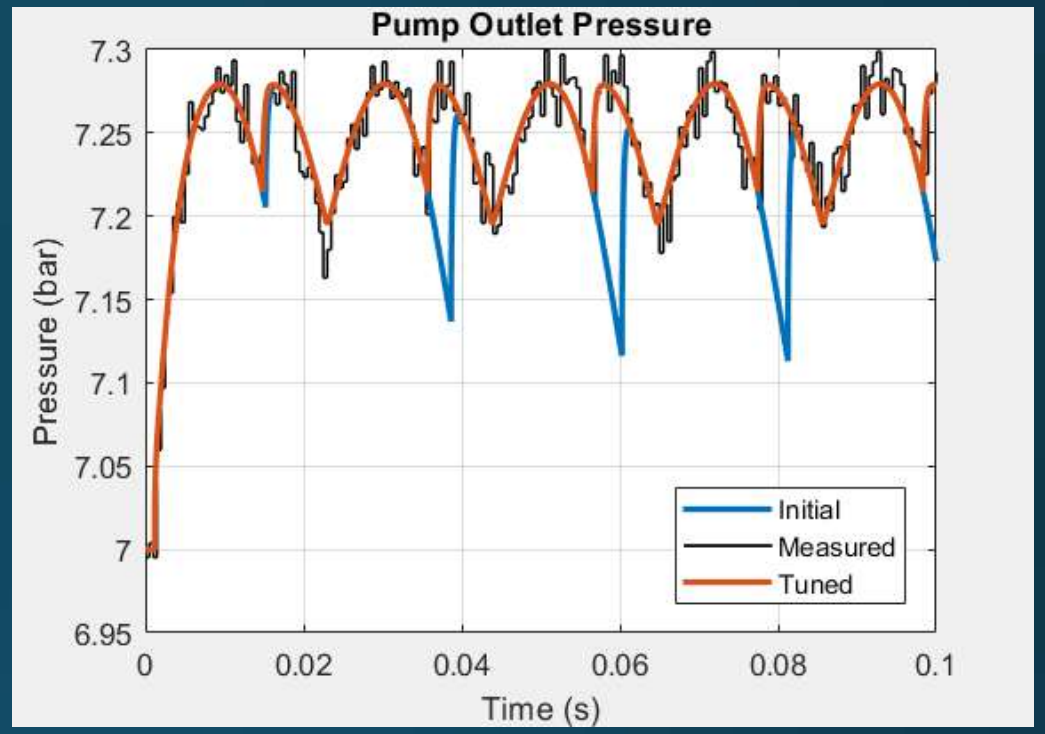


2. Fine tune model

Estimate Parameters Using Measured Data

Model:





Challenge: Simulation results do not match behavior of real system

Solution: Use Optimization to tune model parameters

```
% Group the model parameters and initial states to be estimated together.
%
v = [p;s];

% Estimation Function
estFcn = @(v) sdoPumpEstimation_Objective(v, Simulator, Exp);


% Optimization options
opt = sdo.OptimizeOptions;
opt.Method = 'lsqnonlin';

% Estimate the Parameters
vOpt = sdo.optimize(estFcn, v, opt)
```

Initialize

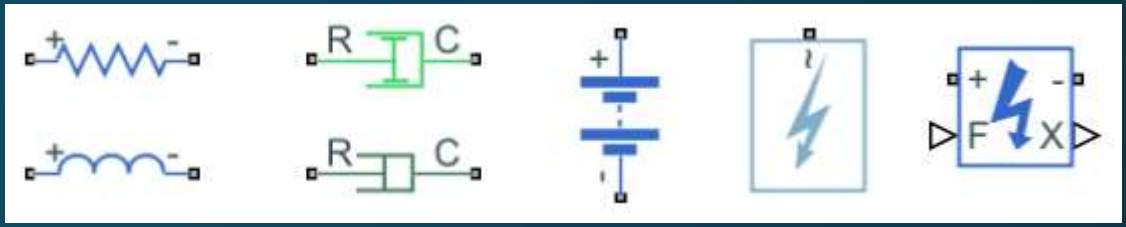
Set Objective

Select solver

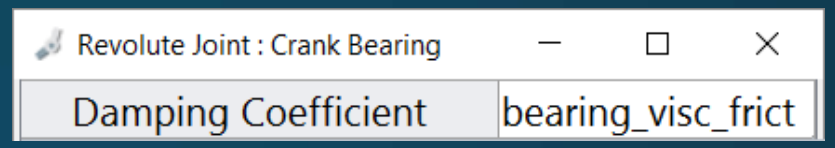
Estimate

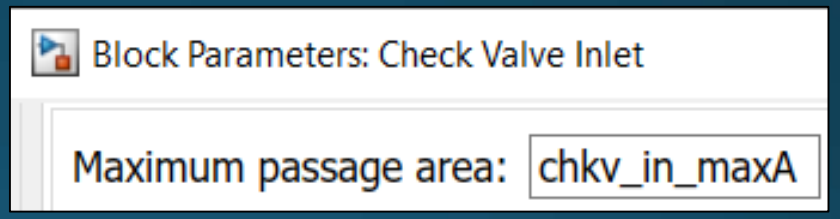
3. Incorporate failure modes

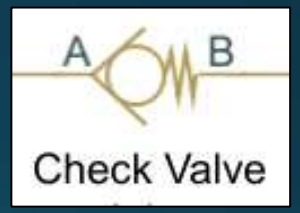
Model Component Failure

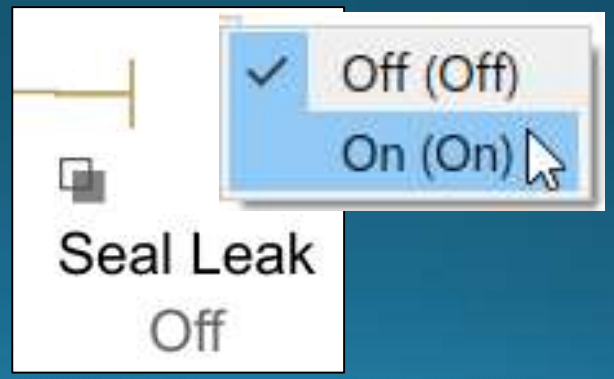


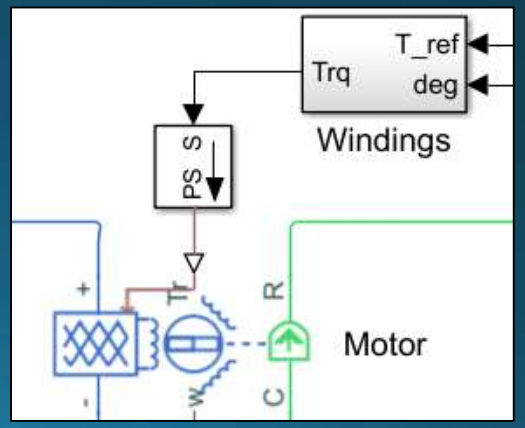
- Generic faults in many components
  - Short circuit, open circuit, friction, fade, etc.
  - Trigger based on time or conditions
- Adjust parameter values
  - Worn bearing adds friction
  - Blocked inlet has reduced passage area
- Adjust network
  - Seal leakage adds flow path
- Custom effects
  - Broken winding applies no torque for 1/3 of every revolution

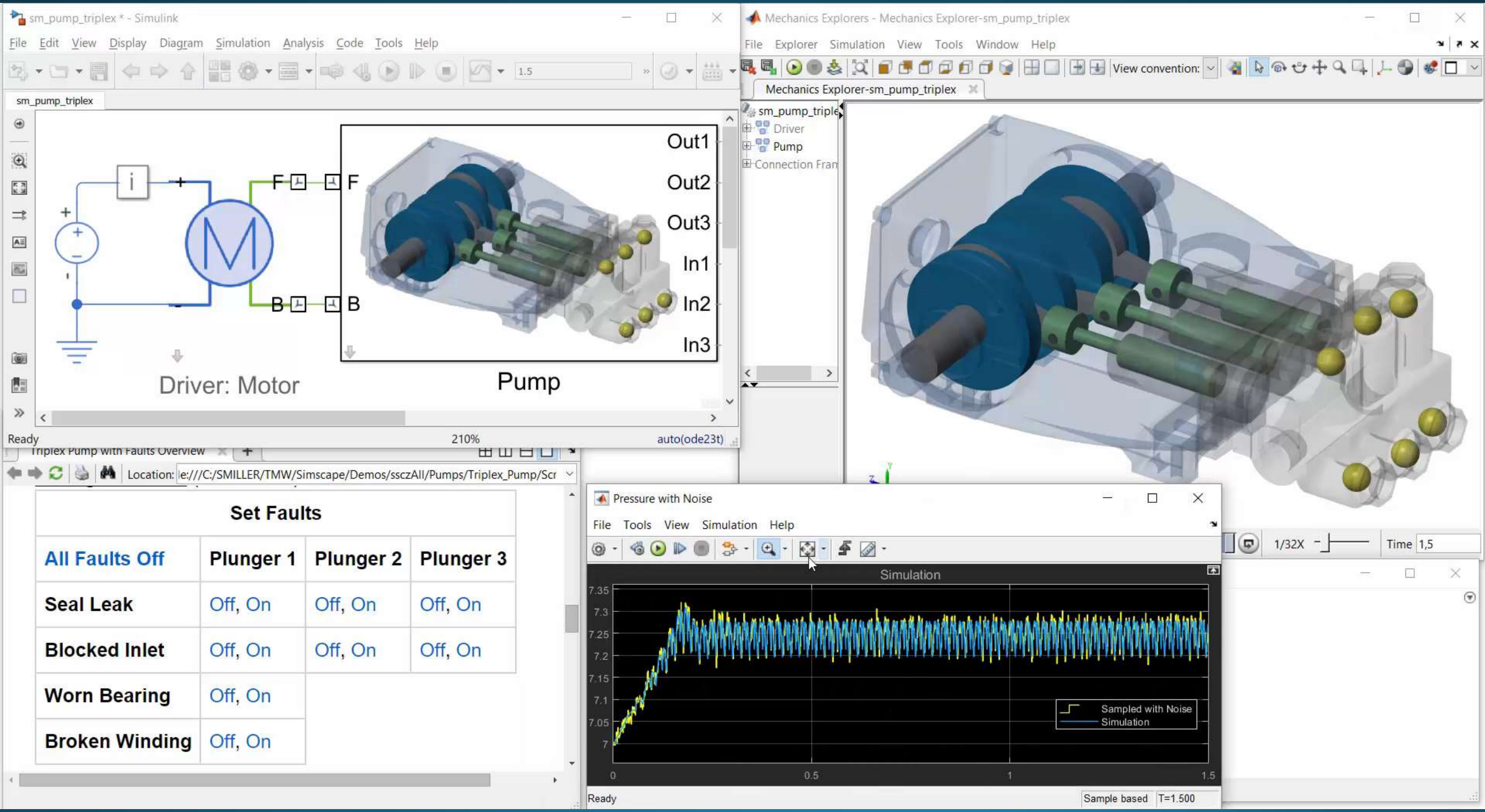










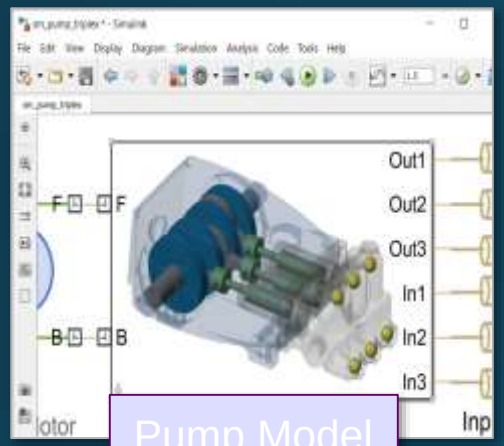

sm_pump_triplex * - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
sm_pump_triplex
F
B
Out1
Out2
Out3
In1
In2
In3
Driver: Motor
Pump
Ready
210%
auto(ode23t)
Mechanics Explorers - Mechanics Explorer-sm_pump_triplex
File Explorer Simulation View Tools Window Help
View convention:
Mechanics Explorer-sm_pump_triplex
sm_pump_triple
Driver
Pump
Connection Fran
1/32X
Time 1,5
Triplex Pump with Faults Overview
Location: le:///C:/SMILLER/TMW/Simscape/Demos/ssczAll/Pumps/Triplex_Pump/Scr
Set Faults
All Faults Off
Plunger 1
Plunger 2
Plunger 3
Seal Leak
Off, On
Off, On
Off, On
Blocked Inlet
Off, On
Off, On
Off, On
Worn Bearing
Off, On
Broken Winding
Off, On
Pressure with Noise
File Tools View Simulation Help
Simulation
7.35
7.3
7.25
7.2
7.15
7.1
7.05
7
0
0.5
1
1.5
Sampled with Noise
Simulation
Ready
Sample based
T=1.500

4. Run simulations

Generate Synthetic Data

Running simulations in parallel speeds up your testing process.

```
for ct = 1:length(final_block)
    time_WKSP = ct;
    simInput(ct) = Simulink.SimulationInput(mdl);
    simInput(ct) = setVariable(simInput(ct),'leak_cyl_area_WKSP',final_leak(1));
    simInput(ct) = setVariable(simInput(ct),'block_in_factor_WKSP',final_block(ct));
    simInput(ct) = setVariable(simInput(ct),'bearing_fault_frict_WKSP',final_bearing(1));
    simInput(ct) = setVariable(simInput(ct),'noise_seed_offset_WKSP',ct-1);
    simInput(ct) = setVariable(simInput(ct),'time_WKSP',time_WKSP);
end
```



Pump Model

Computer Cluster

Workers

Simulation 1

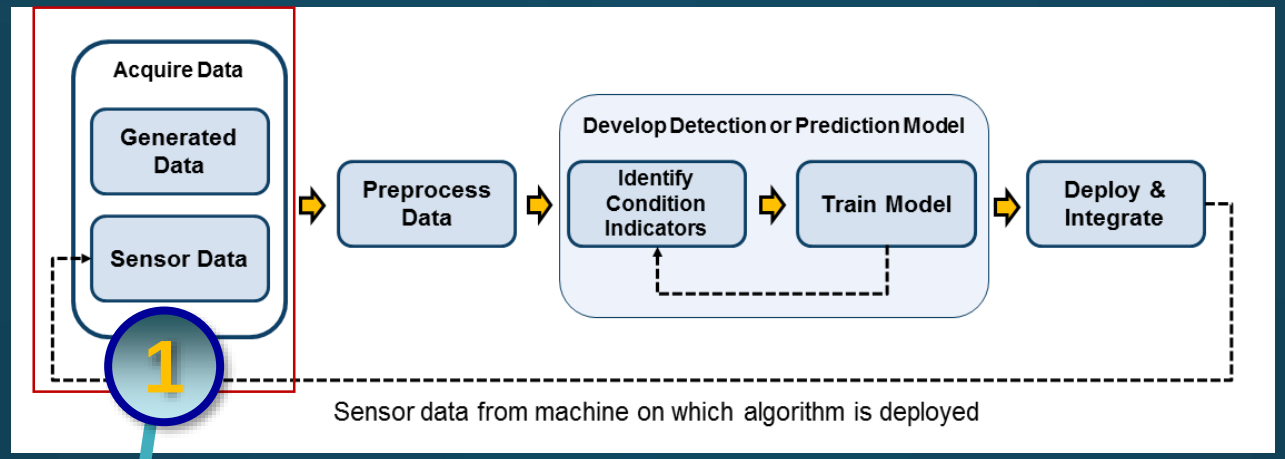
Simulation 2

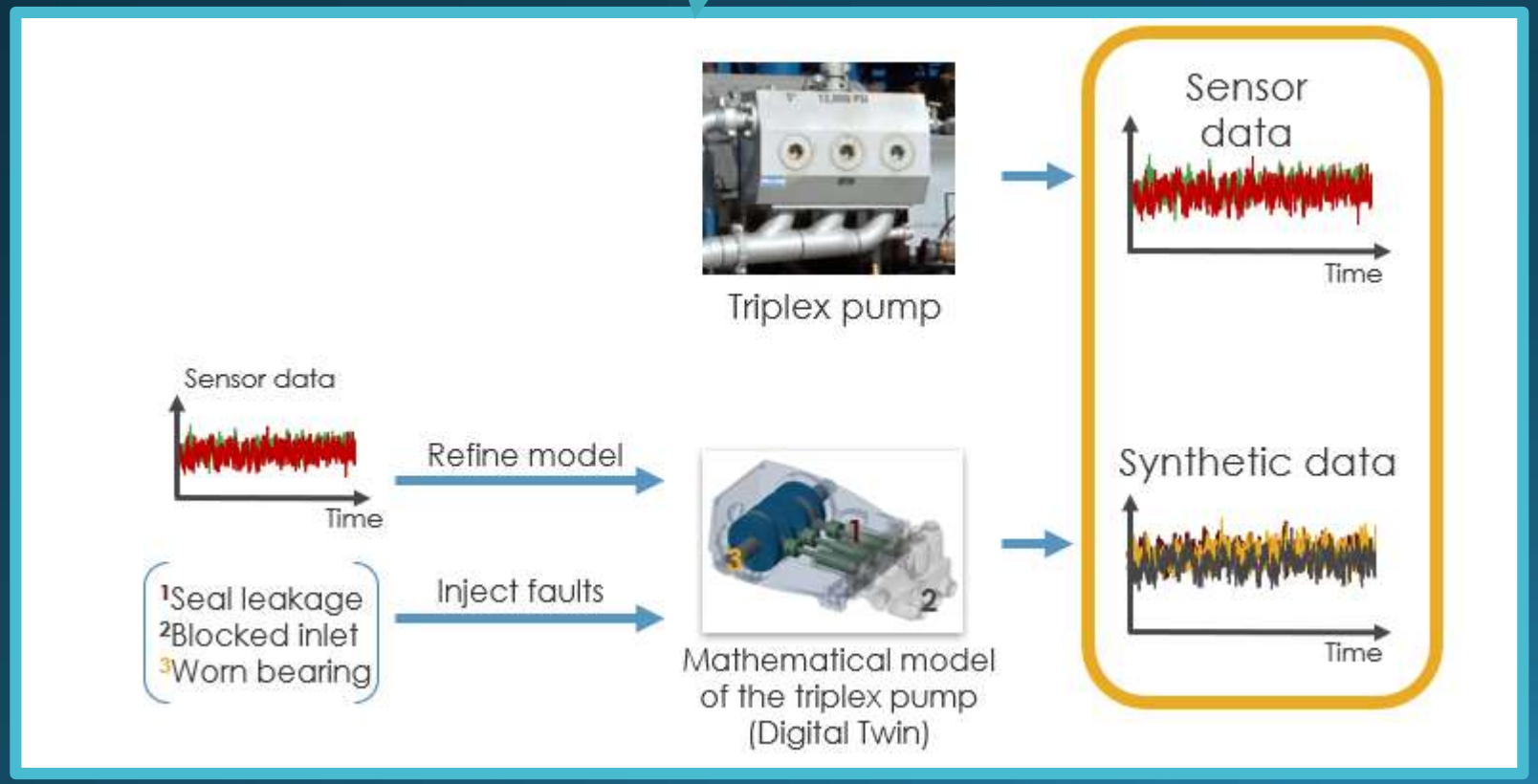
⋮

Desktop System

Workers

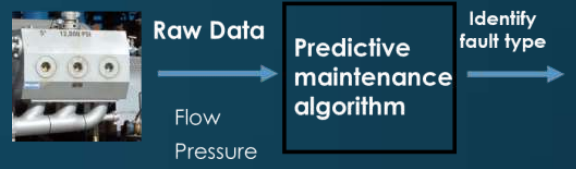
PumpData1.csv
PumpData2.csv
PumpData3.csv
PumpData4.csv
PumpData5.csv
PumpData6.csv
PumpData7.csv
PumpData8.csv
PumpData9.csv
PumpData10.csv
PumpData11.csv
PumpData12.csv
PumpData13.csv
PumpData14.csv
PumpData15.csv
PumpData16.csv
PumpData17.csv





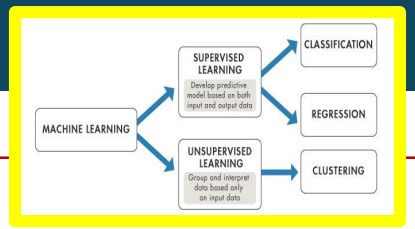
240 measurements of flow and pressure with a duration of 1.2 seconds

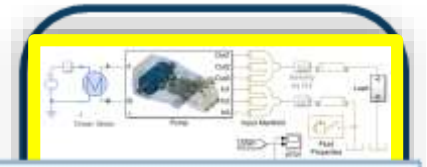
Use Case: Fault Classification of a Pump



Approach

Digital Twin





240 measurements of flow and pressure with a duration of 1.2 seconds

2

Preprocess Data

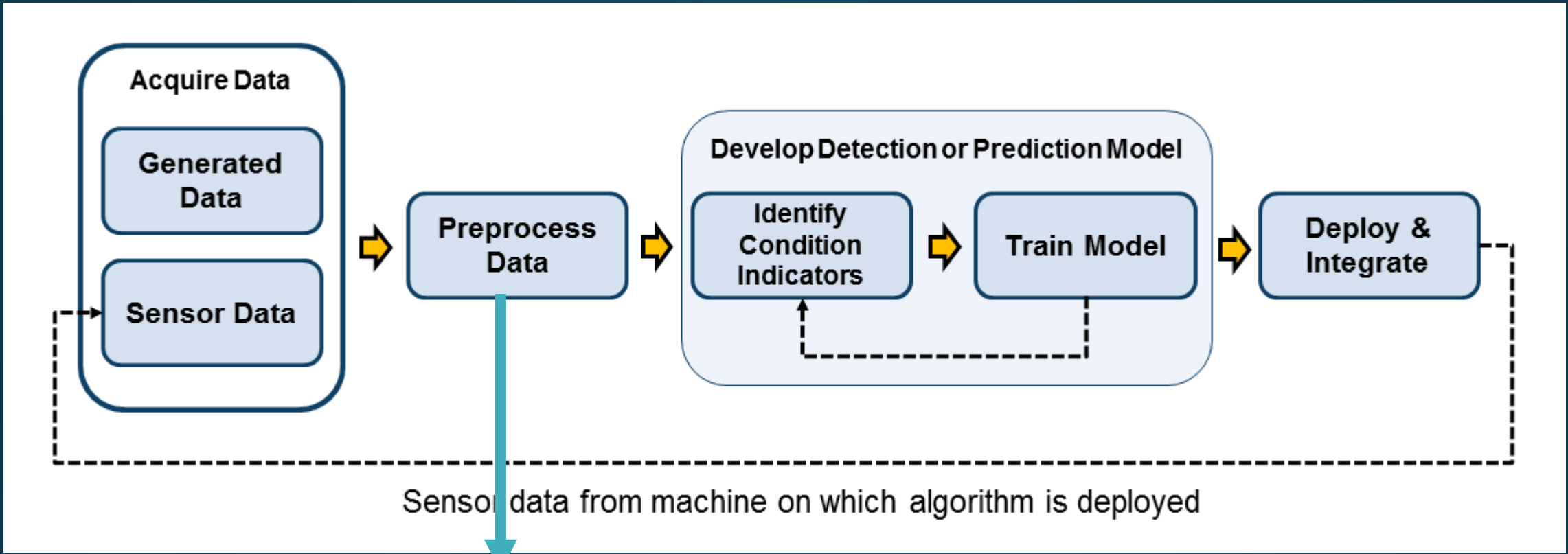
Develop Detection or Prediction Model

Identify Condition Indicators

Train Model

Deploy & Integrate

Sensor data from machine on which algorithm is deployed



Challenges:

- Data clean up
  - Poorly formatted files
  - Irregularly sampled data
  - Redundant data, outliers, missing data etc.
- Data specific processing
  - Signals: Smoothing, resampling, denoising, Wavelet transforms, etc.
  - Images: Image registration, morphological filtering, deblurring, etc.

MATLAB R2019a

« adoshi ▸ Documents ▸ MATLAB ▸ Examples ▸ R2019a ▸ predmaint ▸ MultiClassFaultDetec

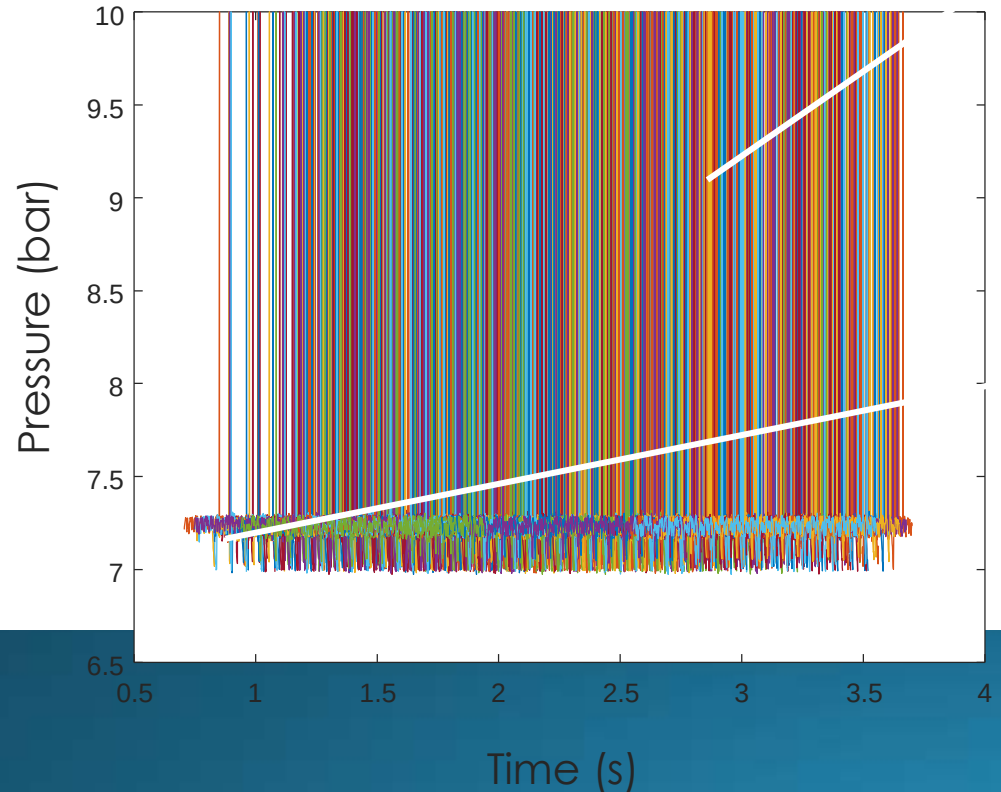
Live Editor - plotRawData.mlx | Command Window | Variables - pumpData

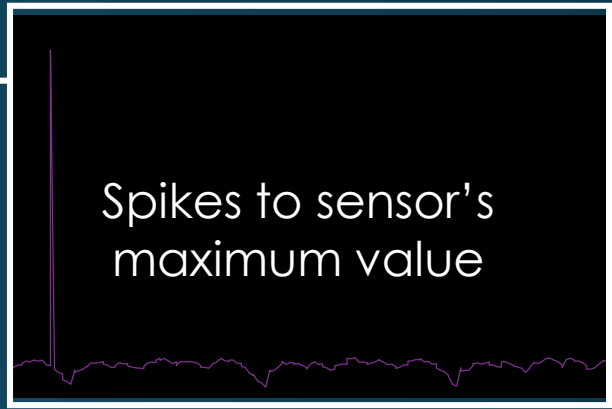
plotRawData.mlx

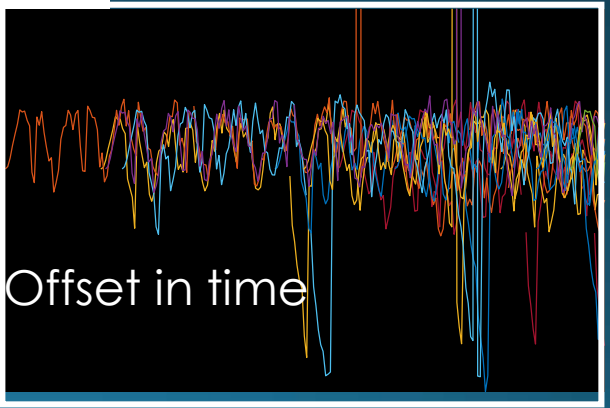
Plot time vs presssure data for all the 240 runs

```
f= @(x) plot(pumpData.pressure{x,1}.Time,pumpData.pressure{x,1}.Data);
for i=1:240, f(i),end
```

Raw data

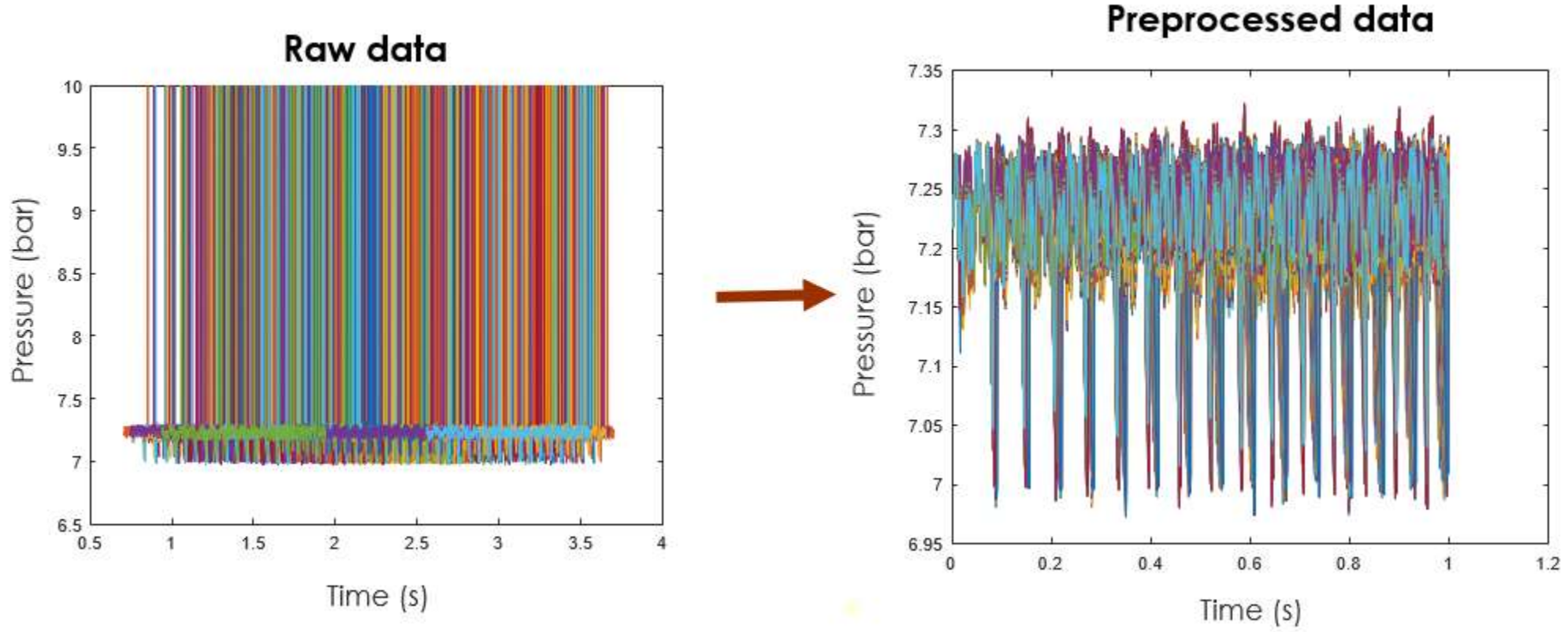


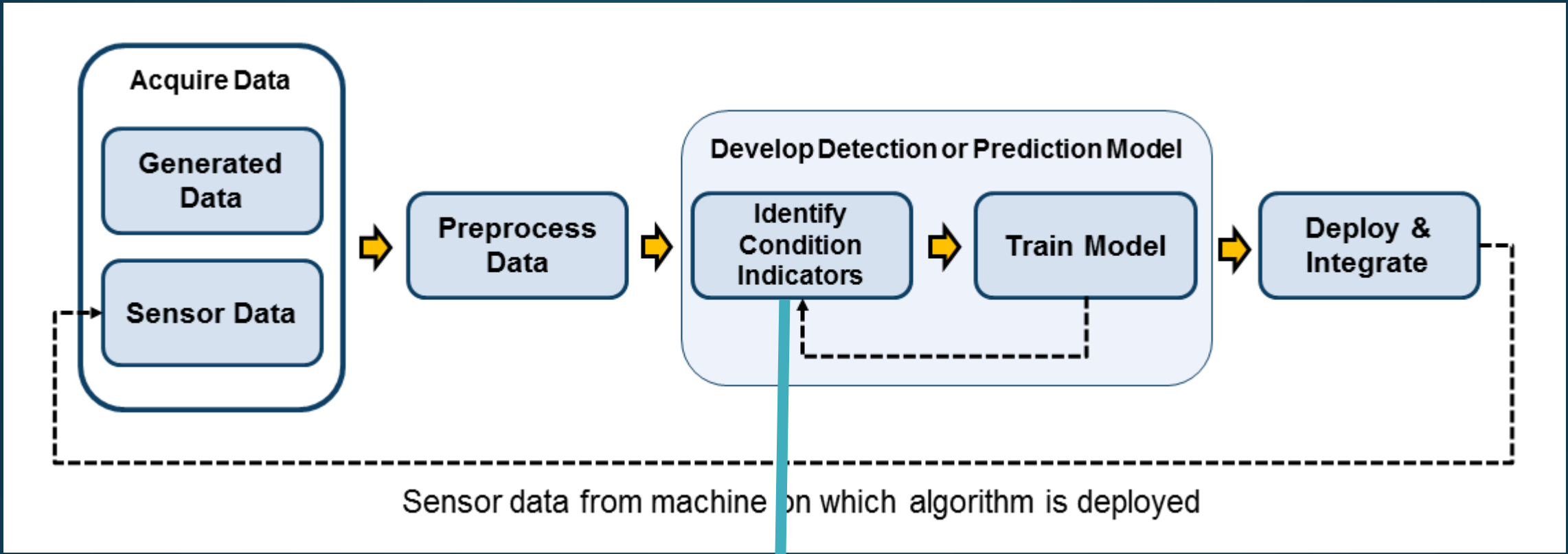




```
%fileLoc = 'hdfs://hadoop01glnxa64:54310/datasets/PumpData*.csv'
fileLoc = fullfile('PumpData*.csv')
ds = datastore(fileLoc)

while hasdata(ds)
    T=read(ds);
    idx_Noise = T.pOut>9;
    T.pOut(idx_Noise) = NaN;
    idx_Missing = isnan(T.pOut);
    T.pOut = fillmissing(T.pOut,'linear');
    T.time = T.time-T.time(1);
    plot(T.time,T.pOut);
    hold on
end
```

```
%fileLoc = 'hdfs://hadoop01glnxa64:54310/datasets/PumpData*.csv'
fileLoc = fullfile('PumpData*.csv')
ds = datastore(fileLoc)

while hasdata(ds)
    T=read(ds);
    idx_Noise = T.pOut>9;
    T.pOut(idx_Noise) = NaN;
    idx_Missing = isnan(T.pOut);
    T.pOut = fillmissing(T.pOut,'linear');
    T.time = T.time-T.time(1);
    plot(T.time,T.pOut);
    hold on
end
```





Challenges

- It is important to have various features to help us develop a robust condition monitoring (ML)algorithm;
- We might not have all the knowledge of various features as well as we might not know which ones are more significant

Signal-Based Condition Indicators (features)

Time-domain features

- Mean
- Standard deviation
- Skewness
- Root-mean square
- Kurtosis

⋮

Frequency-domain features

- Power bandwidth
- Mean frequency
- Peak values
- Peak frequencies
- Harmonics

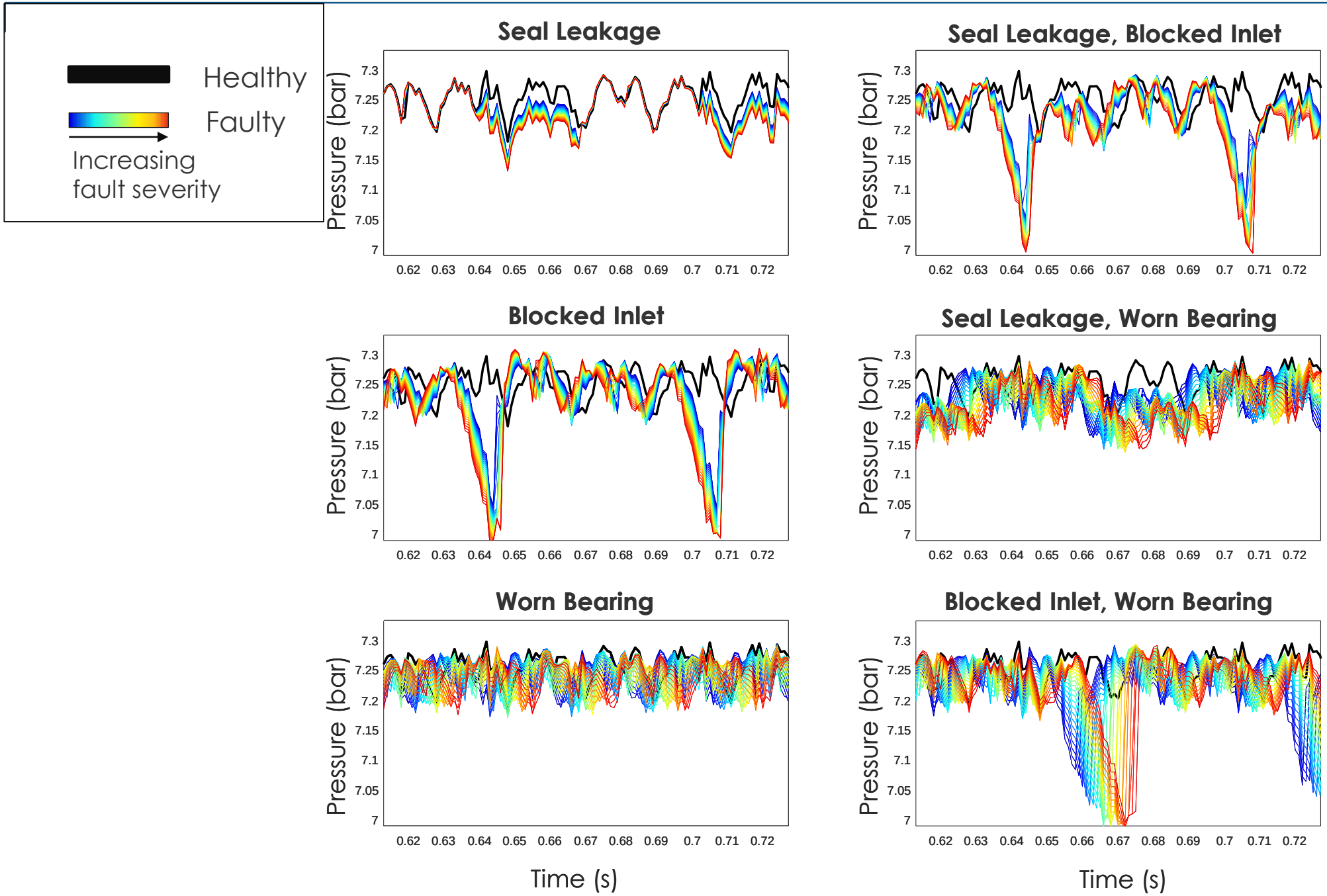
⋮

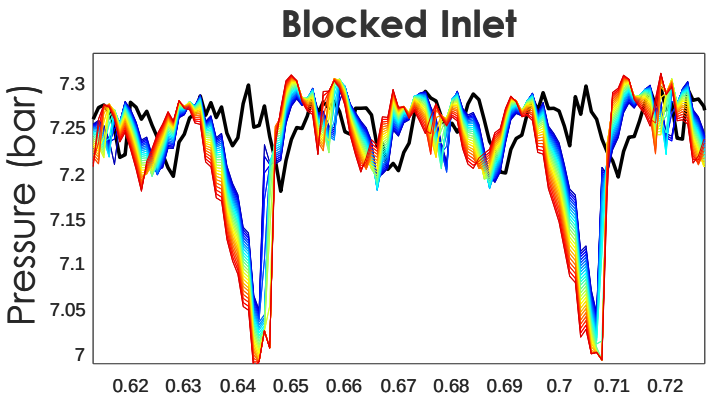
Time-frequency domain features

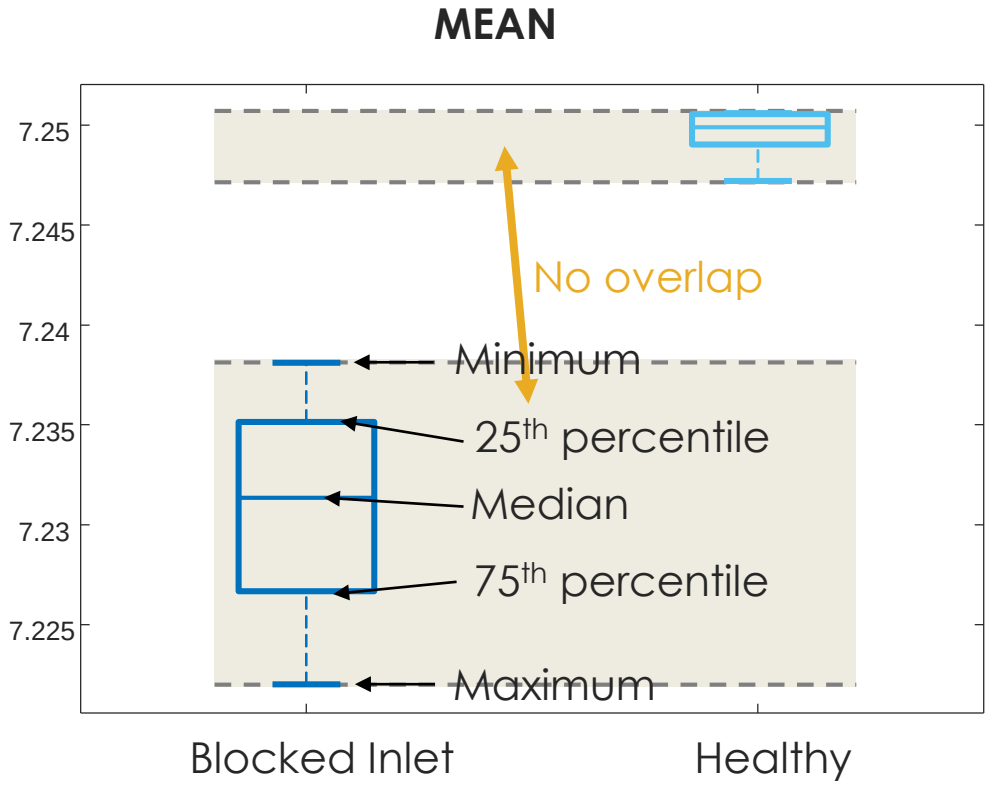
- Spectral entropy
- Spectral kurtosis

⋮

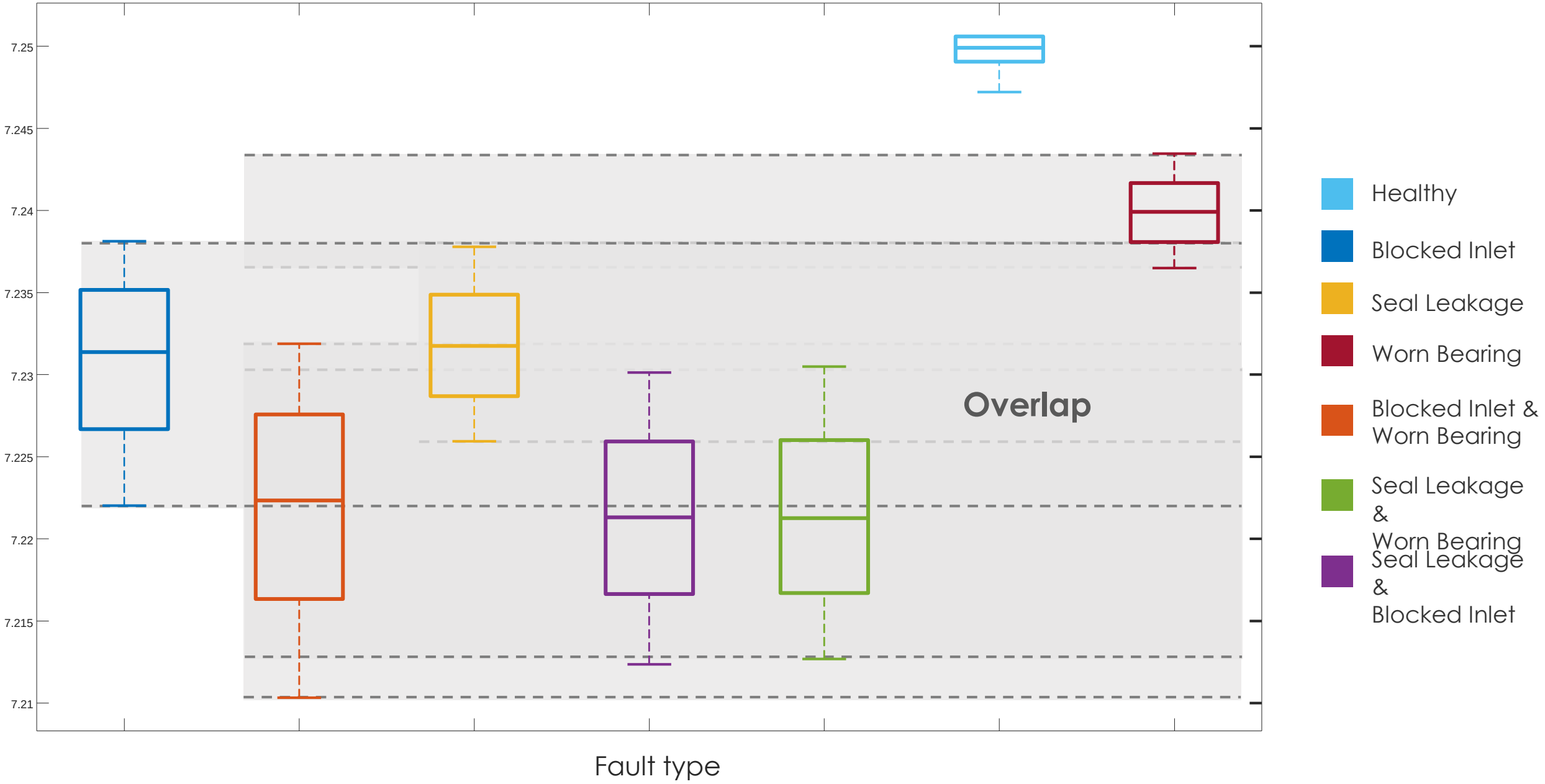
Learn more about Condition Indicators

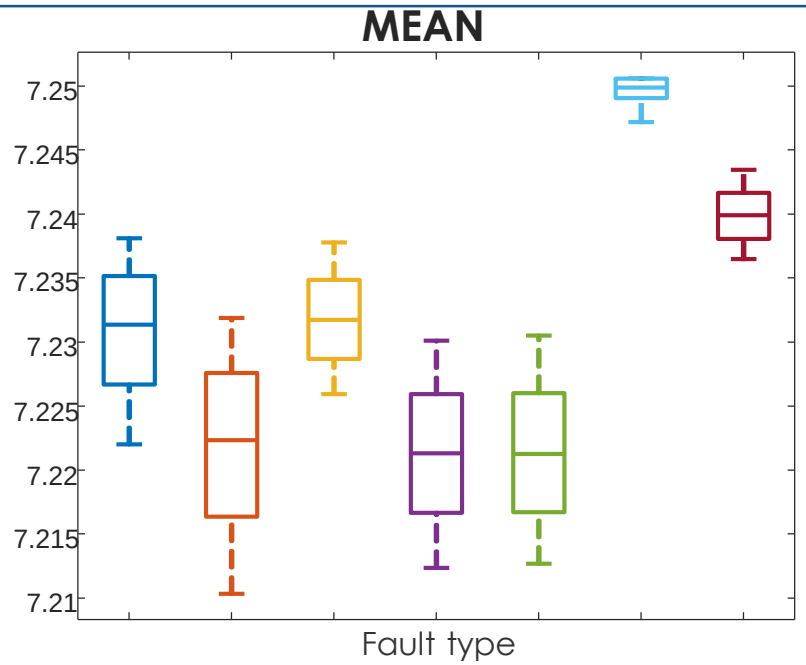

Healthy
Faulty
Increasing fault severity
Seal Leakage
Seal Leakage, Blocked Inlet
Blocked Inlet
Seal Leakage, Worn Bearing
Worn Bearing
Blocked Inlet, Worn Bearing
Pressure (bar)
7.3
7.25
7.2
7.15
7.1
7.05
7
0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.7 0.71 0.72
Time (s)

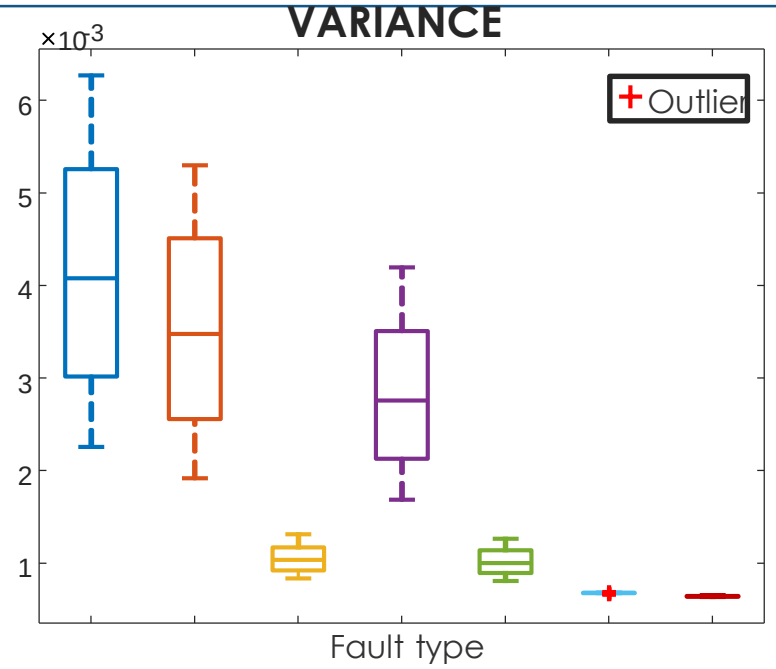

Blocked Inlet
Pressure (bar)
7.3
7.25
7.2
7.15
7.1
7.05
7
0.62
0.63
0.64
0.65
0.66
0.67
0.68
0.69
0.7
0.71
0.72

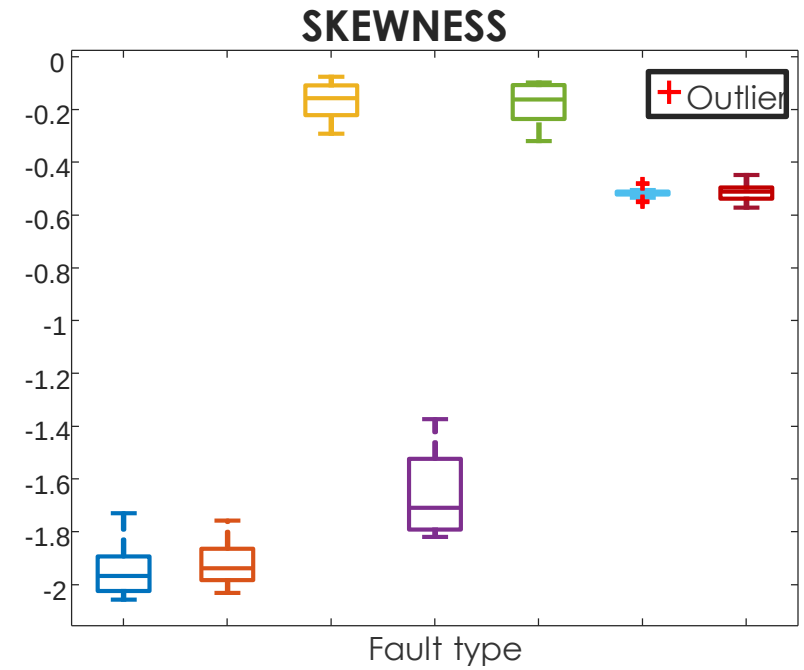

MEAN
7.25
7.245
7.24
7.235
7.23
7.225
No overlap
Minimum
25th percentile
Median
75th percentile
Maximum
Blocked Inlet
Healthy

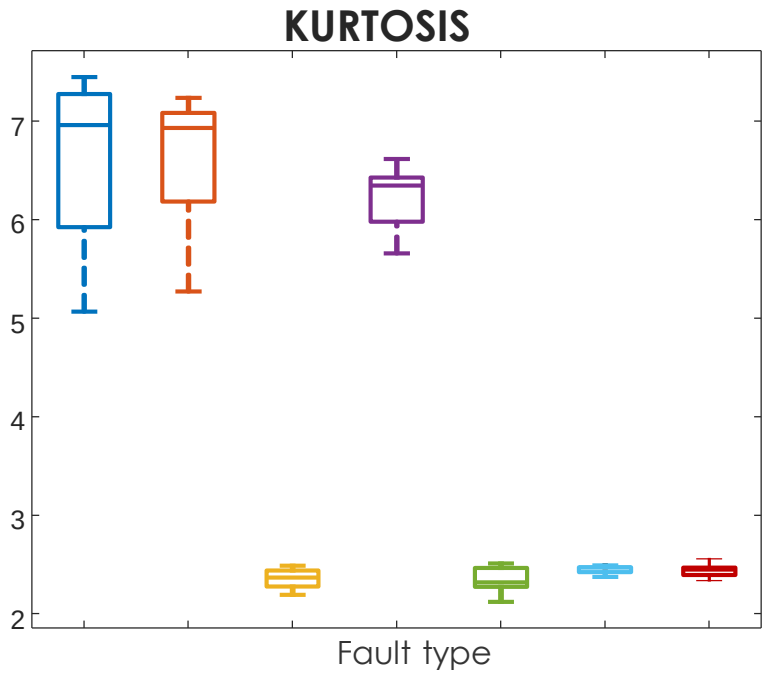
MEAN


7.25
7.245
7.24
7.235
7.23
7.225
7.22
7.215
7.21
Overlap
Fault type
Healthy
Blocked Inlet
Seal Leakage
Worn Bearing
Blocked Inlet & Worn Bearing
Seal Leakage & Worn Bearing
Seal Leakage & Blocked Inlet

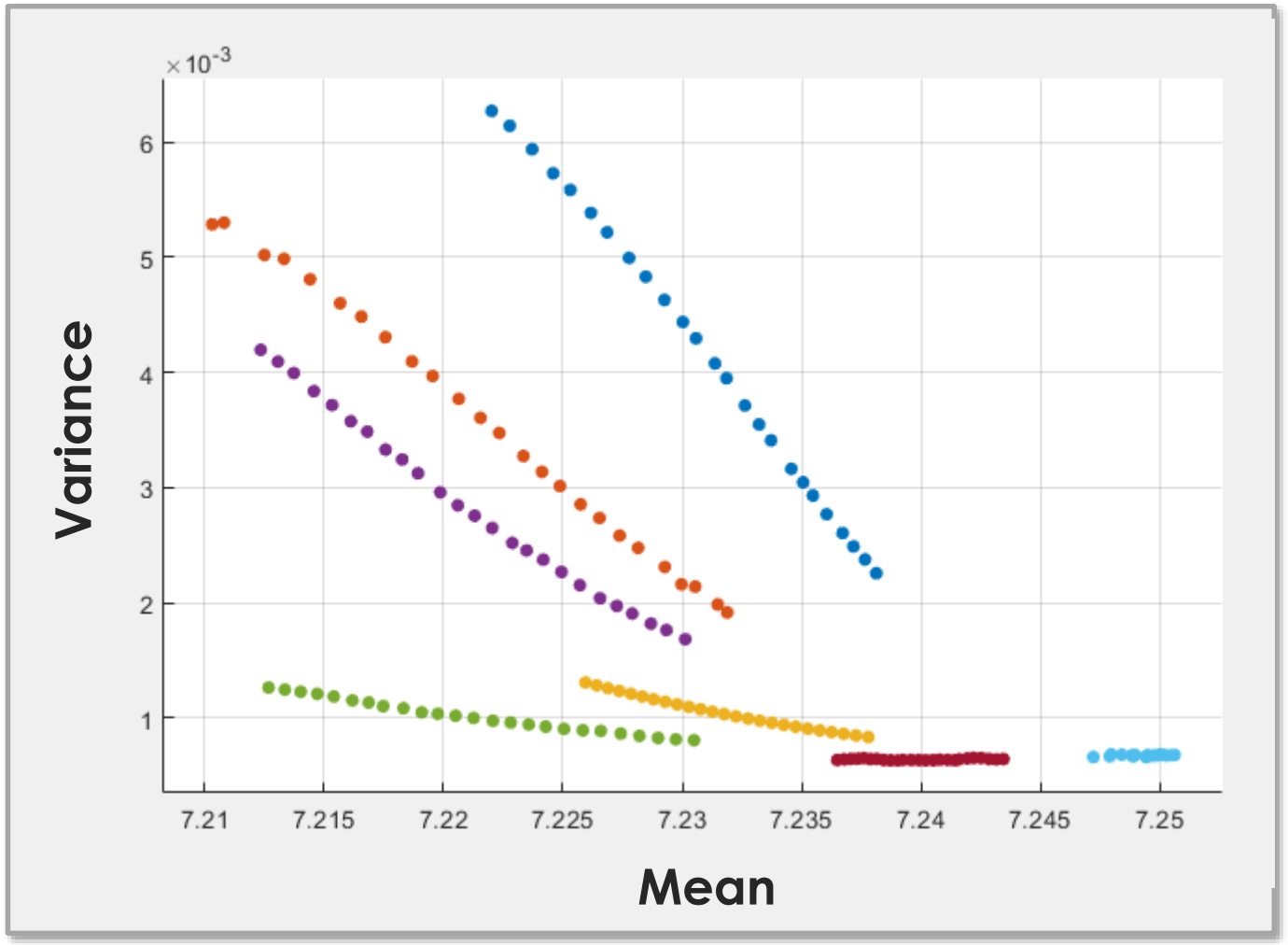

MEAN
7.25
7.245
7.24
7.235
7.23
7.225
7.22
7.215
7.21
Fault type


VARIANCE
×10-3
6
5
4
3
2
1
Outlier
Fault type


SKEWNESS
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4
-1.6
-1.8
-2
Outlier
Fault type


KURTOSIS
7
6
5
4
3
2
Fault type

Healthy
Blocked Inlet
Seal Leakage
Worn Bearing
Blocked Inlet & Worn Bearing
Seal Leakage & Worn Bearing
Seal Leakage & Blocked Inlet



- Healthy
- Blocked Inlet
- Seal Leakage
- Worn Bearing
- Blocked Inlet, Worn Bearing
- Seal Leakage, Worn Bearing
- Seal Leakage, Blocked Inlet

Signal-Based Condition Indicators (features)

Time-domain features

Mean
Standard deviation
Skewness
Root-mean square
Kurtosis
⋮

Frequency-domain features

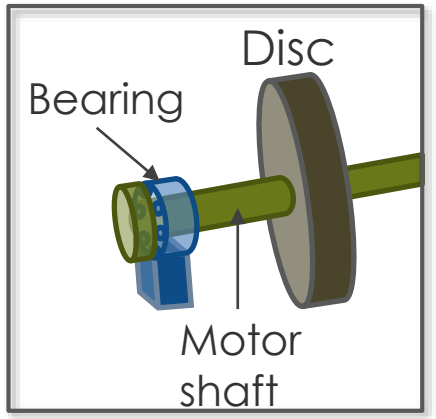
Power bandwidth
Mean frequency
Peak values
Peak frequencies
Harmonics
⋮

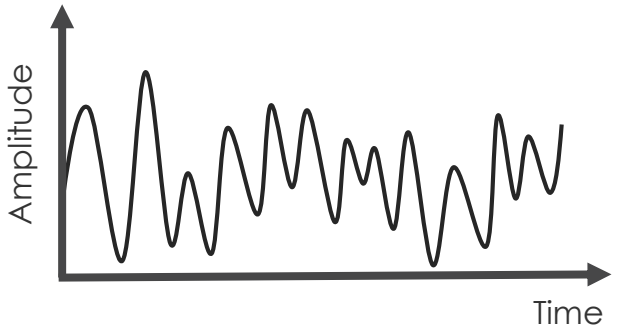
Time-frequency domain features

Spectral entropy
Spectral kurtosis
⋮

Learn more about Condition Indicators

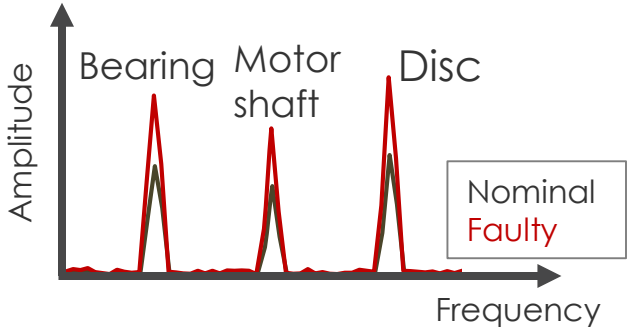
Machine with rotating components



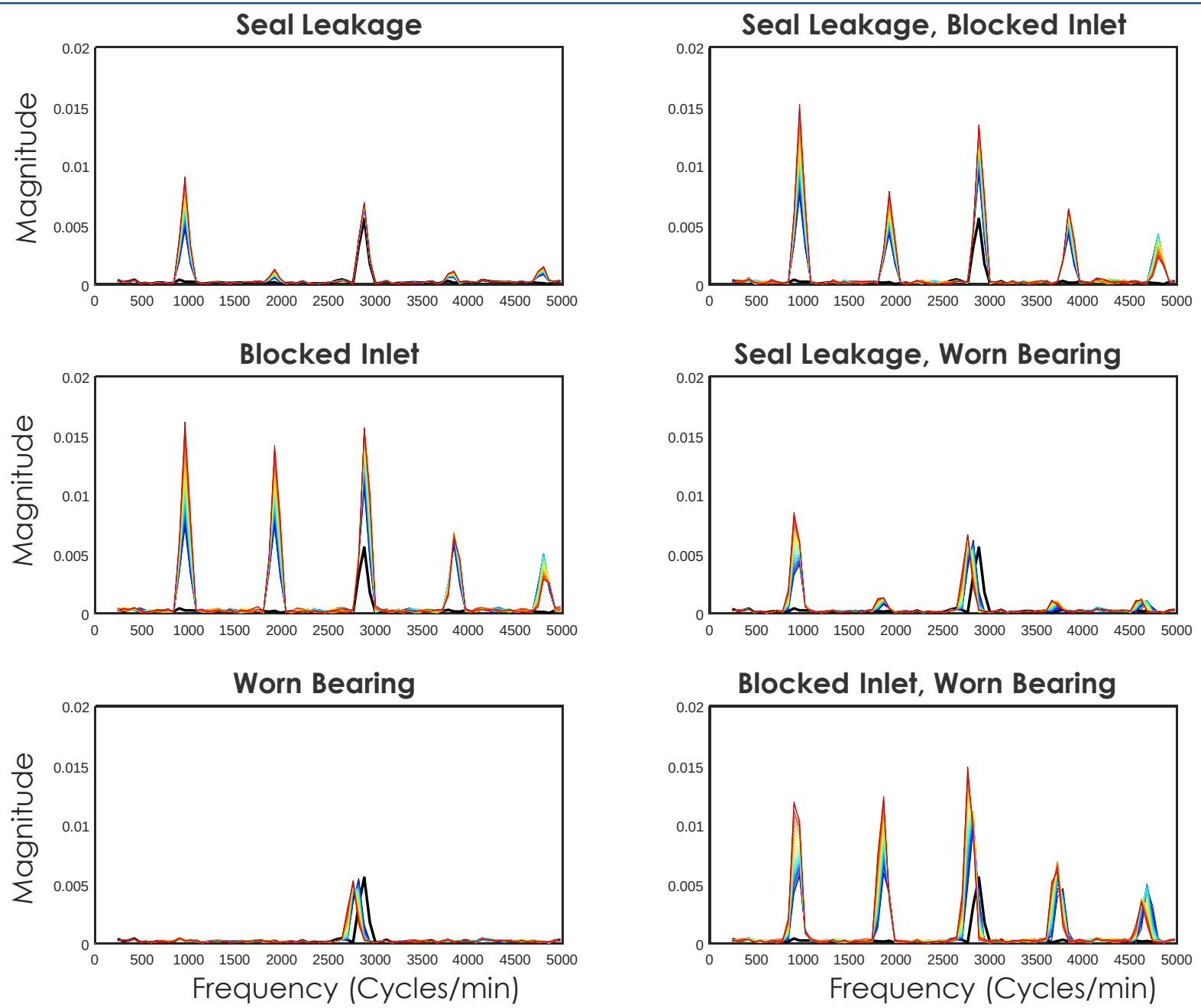


- In time-domain, we observe the combined effect of different sources of vibration.

Three different vibration sources:
- Bearing
- Motor shaft
- Disc



- Using frequency-domain analysis, we can distinguish different sources of vibration.

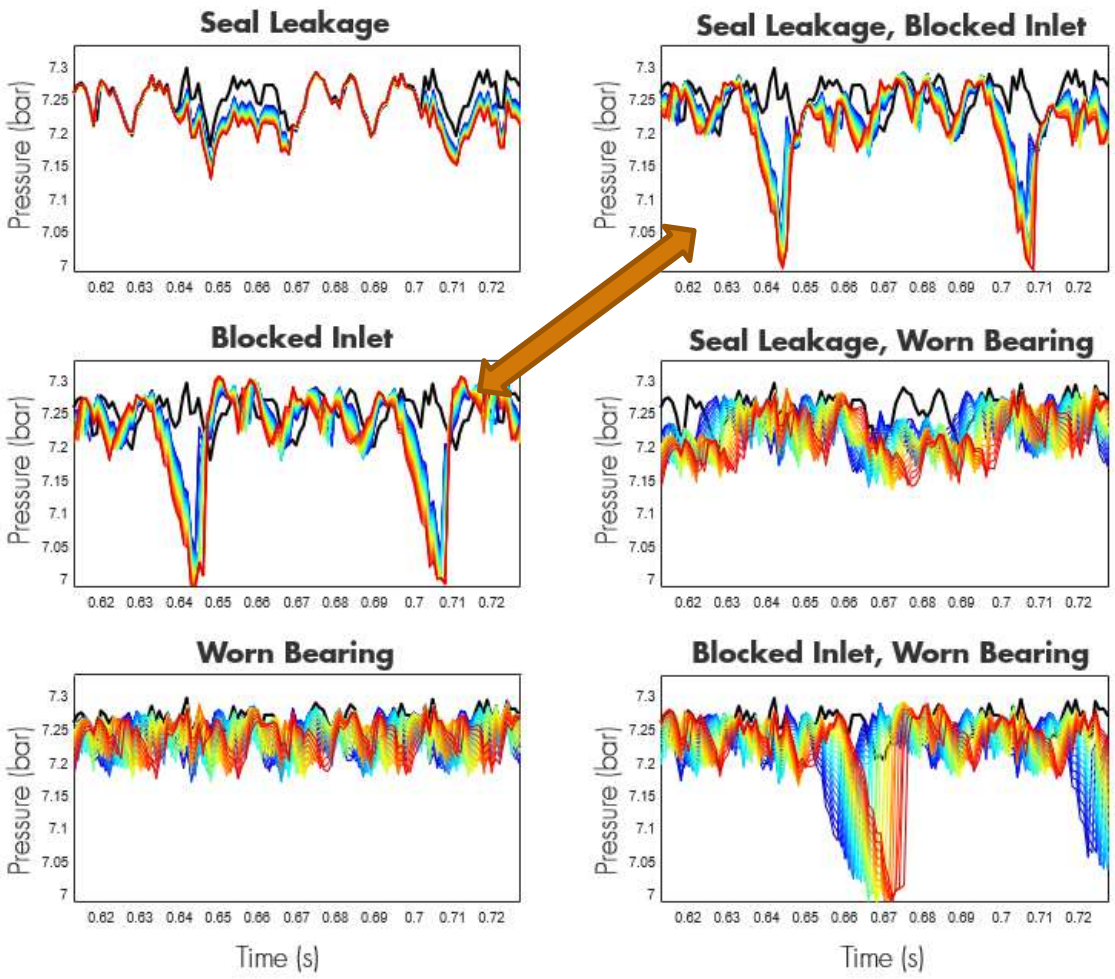


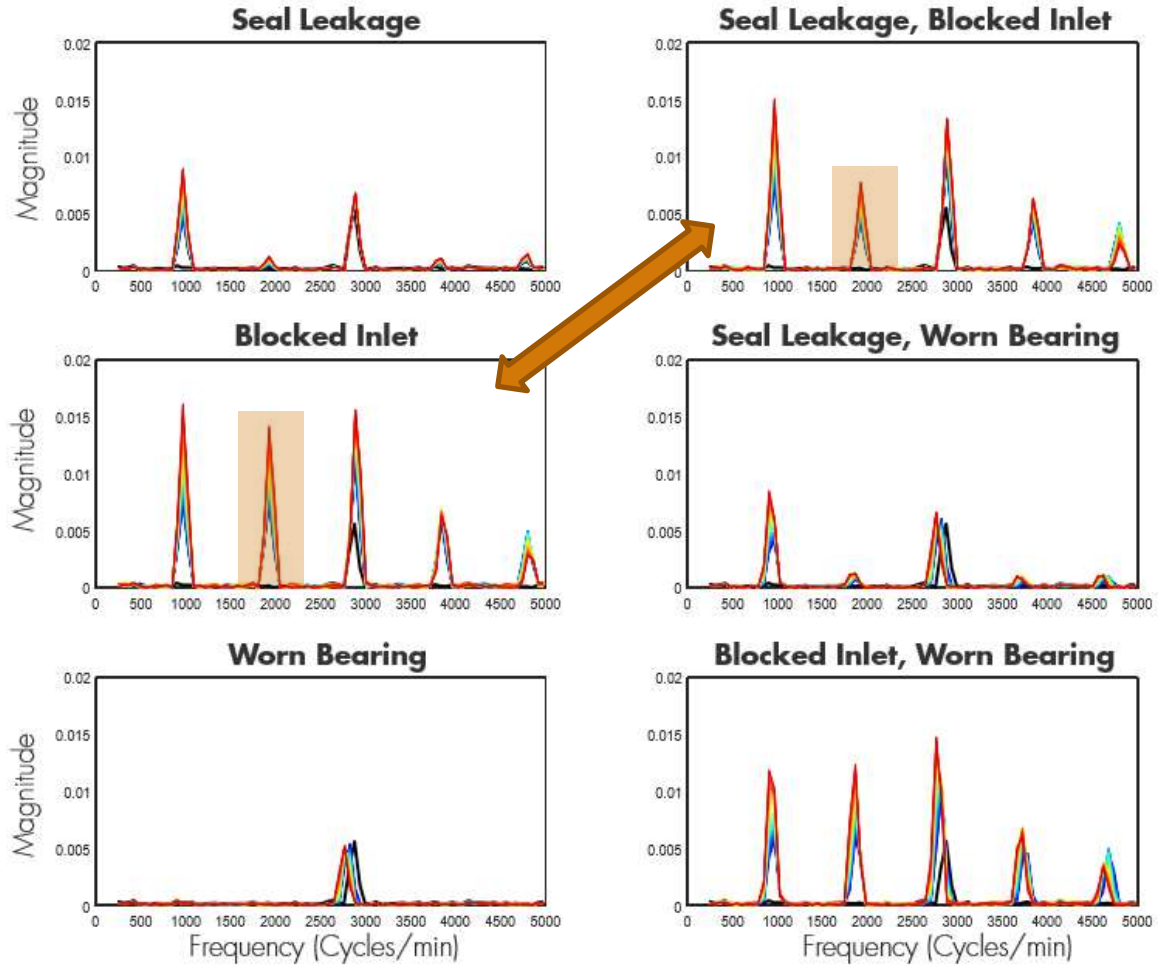
Frequency-domain features:

- Peaks
- Peak frequencies

Time-domain

Frequency-domain





Analyze and Select Features for

https://www.mathworks.com/help/predmaint/ug/analyze-and-select-features-for-pump-diagnostics.html

MathWorks Products Solutions Academia Support Community Events Get MATLAB

Documentation All Examples Functions Apps Search R2019a Documentation Documentation

CONTENTS Trial Software Product Updates Translate This Page

Analyze and Select Features for Pump Diagnostics

R2019a

This example shows how to use the **Diagnostic Feature Designer** app to analyze and select features to diagnose faults in a triplex reciprocating pump.

The example uses simulated pump fault data generated by the Multi-Class Fault Detection Using Simulated Data example. The data has been preprocessed to remove the pump startup transients.

Try This Example

Open Diagnostic Feature Designer

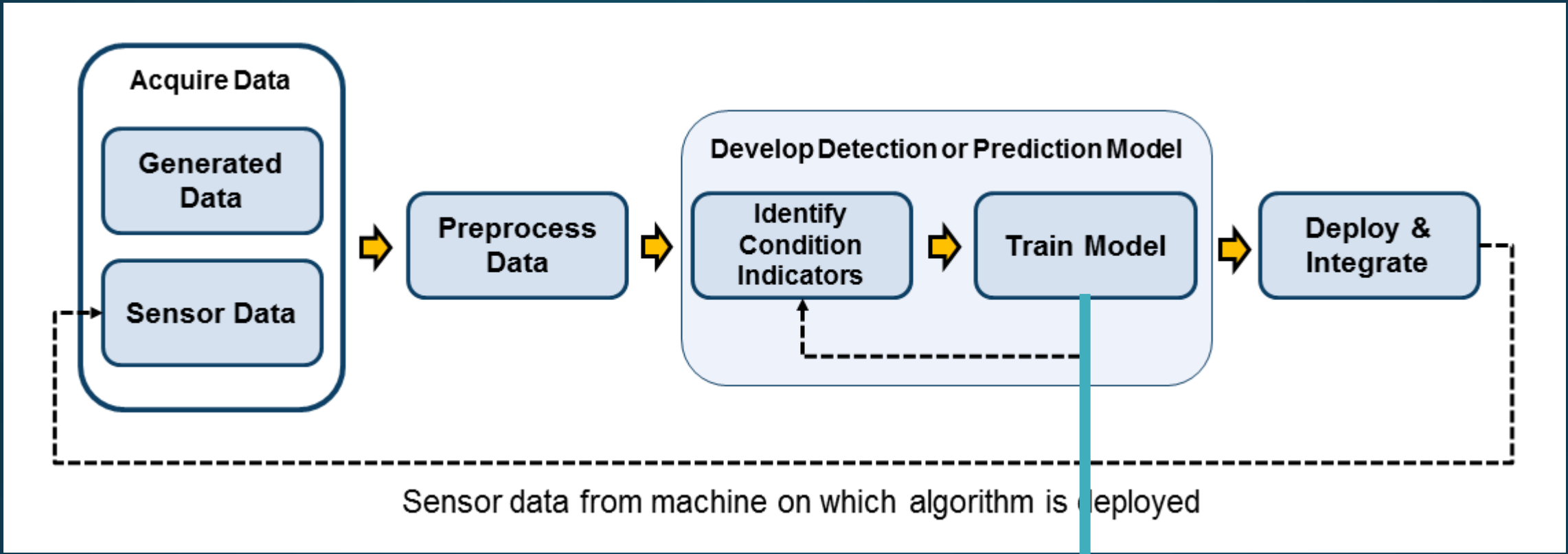
Load the triplex pump fault data. The pump data contains 240 flow and pressure measurements for different fault conditions. There are three fault types (leaking pump cylinder, blocked pump inlet, increased pump bearing friction). The measurements cover conditions where none, one, or multiple faults are present. The data is collected in a table where each row is a different measurement.

```
load('savedPumpData')
pumpData
```

```
pumpData=240×3 table
          flow                pressure          faultCode
    ________________    ________________    _________

    [1201x1 timetable]  [1201x1 timetable]      0
    [1201x1 timetable]  [1201x1 timetable]      0
    [1201x1 timetable]  [1201x1 timetable]      100
    [1201x1 timetable]  [1201x1 timetable]      100
    [1201x1 timetable]  [1201x1 timetable]      100
    [1201x1 timetable]  [1201x1 timetable]      100
    [1201x1 timetable]  [1201x1 timetable]      100
    [1201x1 timetable]  [1201x1 timetable]      100
    [1201x1 timetable]  [1201x1 timetable]      100
    [1201x1 timetable]  [1201x1 timetable]      100
    [1201x1 timetable]  [1201x1 timetable]      100
    [1201x1 timetable]  [1201x1 timetable]      100
    [1201x1 timetable]  [1201x1 timetable]      0
    [1201x1 timetable]  [1201x1 timetable]      100
    [1201x1 timetable]  [1201x1 timetable]      100
    [1201x1 timetable]  [1201x1 timetable]      100
    |
```

Open **Diagnostic Feature Designer** by using the `diagnosticFeatureDesigner` command. Import the pump data into the app. The data is organized as a multi-member ensemble, so use that option for import.

Link



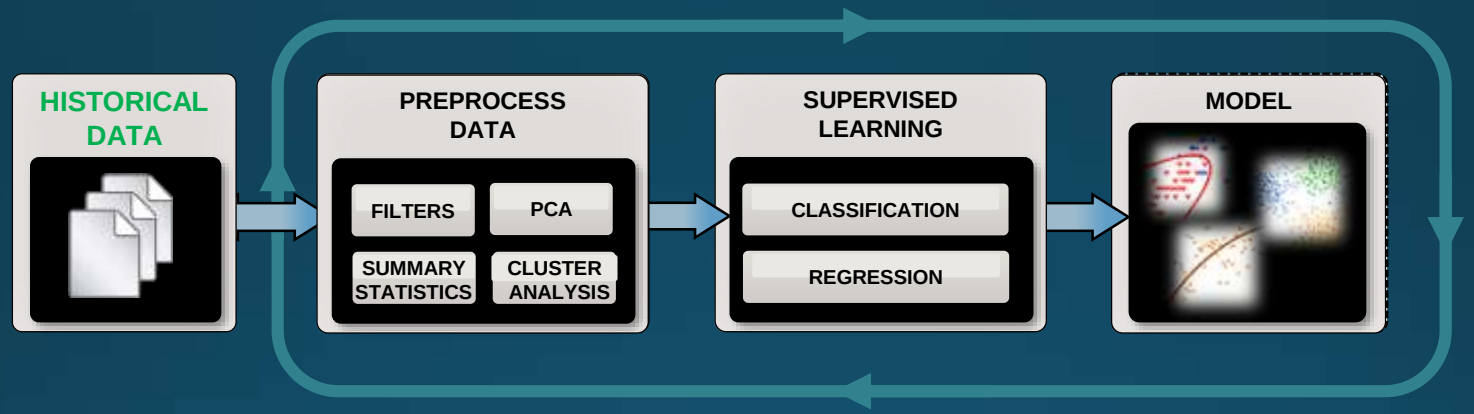
Challenges

- One solution doesn't fit all problems; compare various models;
- Need to also keep in mind usability of the model in production

Machine Learning Workflow

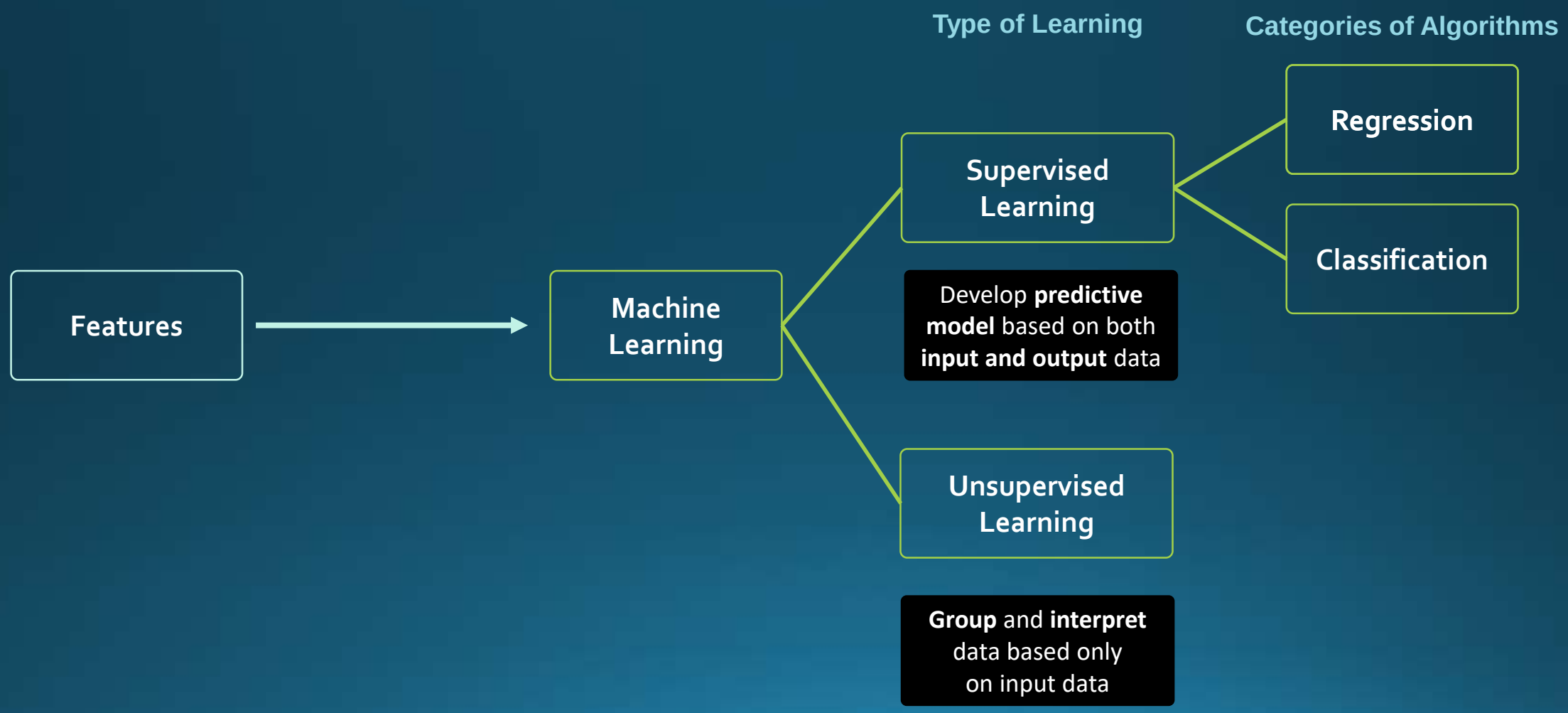
Machine learning uses data and produces a program to perform a task

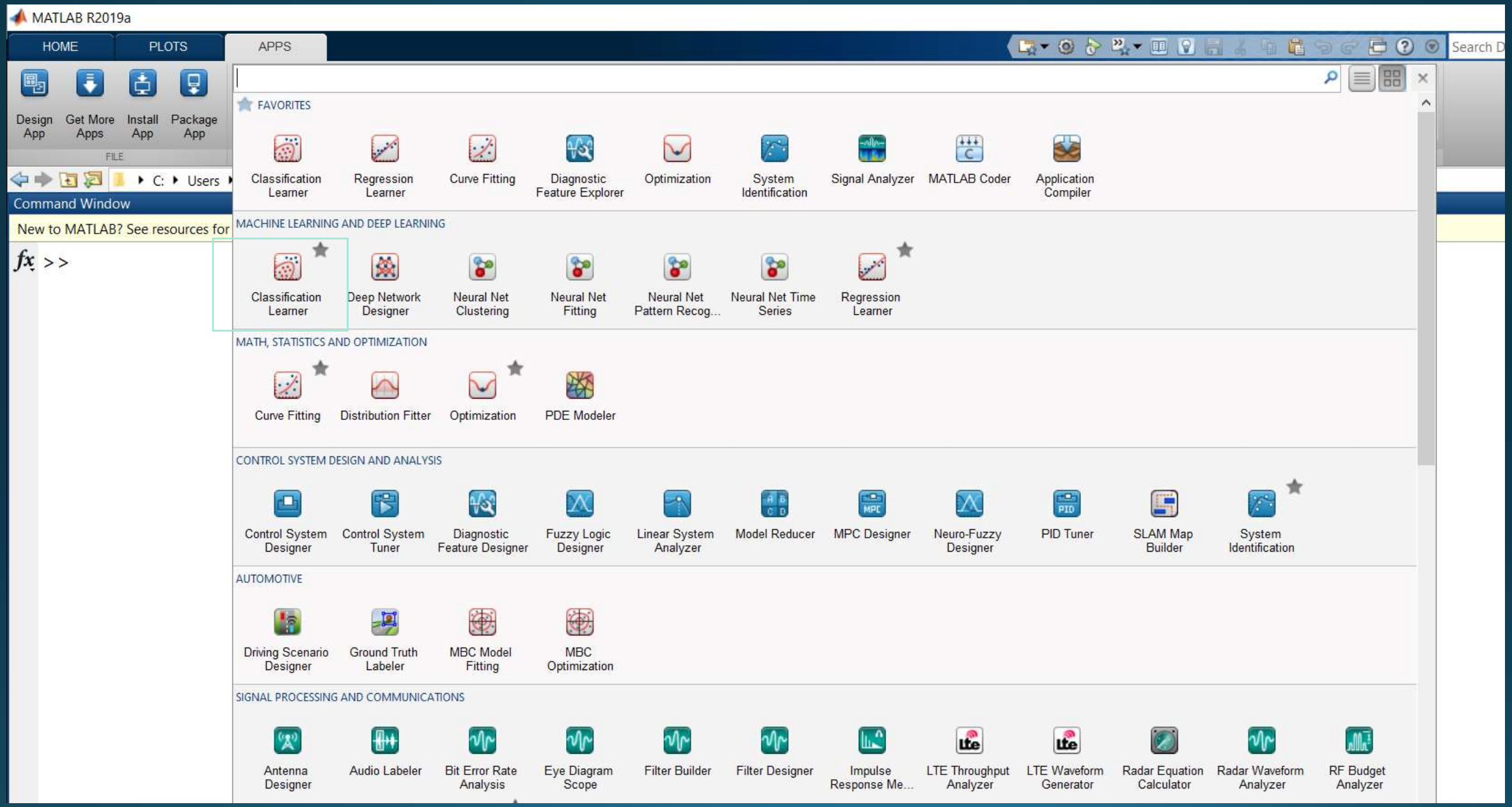
Train: Iterate till you find the best model using historical data



Predict: Integrate trained models into applications

Machine Learning: Types




MATLAB R2019a
HOME
PLOTS
APPS
Search D
Design App
Get More Apps
Install App
Package App
FILE
C: ▸ Users ▸
Command Window
New to MATLAB? See resources for
fx >>
FAVORITES
Classification Learner
Regression Learner
Curve Fitting
Diagnostic Feature Explorer
Optimization
System Identification
Signal Analyzer
MATLAB Coder
Application Compiler
MACHINE LEARNING AND DEEP LEARNING
Classification Learner
Deep Network Designer
Neural Net Clustering
Neural Net Fitting
Neural Net Pattern Recog...
Neural Net Time Series
Regression Learner
MATH, STATISTICS AND OPTIMIZATION
Curve Fitting
Distribution Fitter
Optimization
PDE Modeler
CONTROL SYSTEM DESIGN AND ANALYSIS
Control System Designer
Control System Tuner
Diagnostic Feature Designer
Fuzzy Logic Designer
Linear System Analyzer
Model Reducer
MPC Designer
Neuro-Fuzzy Designer
PID Tuner
SLAM Map Builder
System Identification
AUTOMOTIVE
Driving Scenario Designer
Ground Truth Labeler
MBC Model Fitting
MBC Optimization
SIGNAL PROCESSING AND COMMUNICATIONS
Antenna Designer
Audio Labeler
Bit Error Rate Analysis
Eye Diagram Scope
Filter Builder
Filter Designer
Impulse Response Me...
LTE Throughput Analyzer
LTE Waveform Generator
Radar Equation Calculator
Radar Waveform Analyzer
RF Budget Analyzer

Diagnostic Feature Designer - Feature Ranking: FeatureTable1

FEATURE DESIGNER | FEATURE RANKING | VIEW

RL Func | RL Examples | CleanUp

Ranking Techniques (ANALYZE) | Rank By: faultCode (CONDITION) | Sort By: One-way ANOVA (SORT) | Delete Scores (SCORE) | Export (EXPORT)

Data Browser

Signals & Spectra

- flow/Data
- pressure/Data
- flow_ps/Data
- flow_ps_1/Data
- pressure_ps/Data

Feature Tables

FeatureTable1

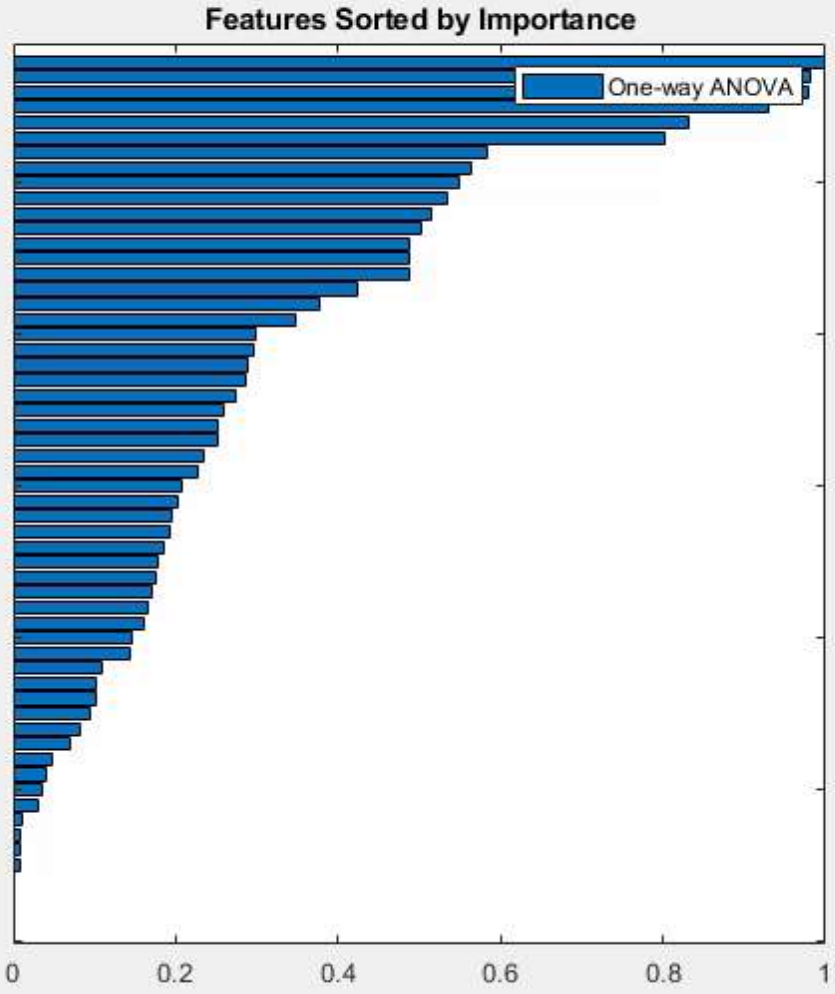
Features:

- flow_stats/Data_ClearanceFactor
- flow_stats/Data_CrestFactor
- flow_stats/Data_ImpulseFactor
- flow_stats/Data_Kurtosis
- flow_stats/Data_Mean
- flow_stats/Data_PeakValue

Datasets

Ensemble1

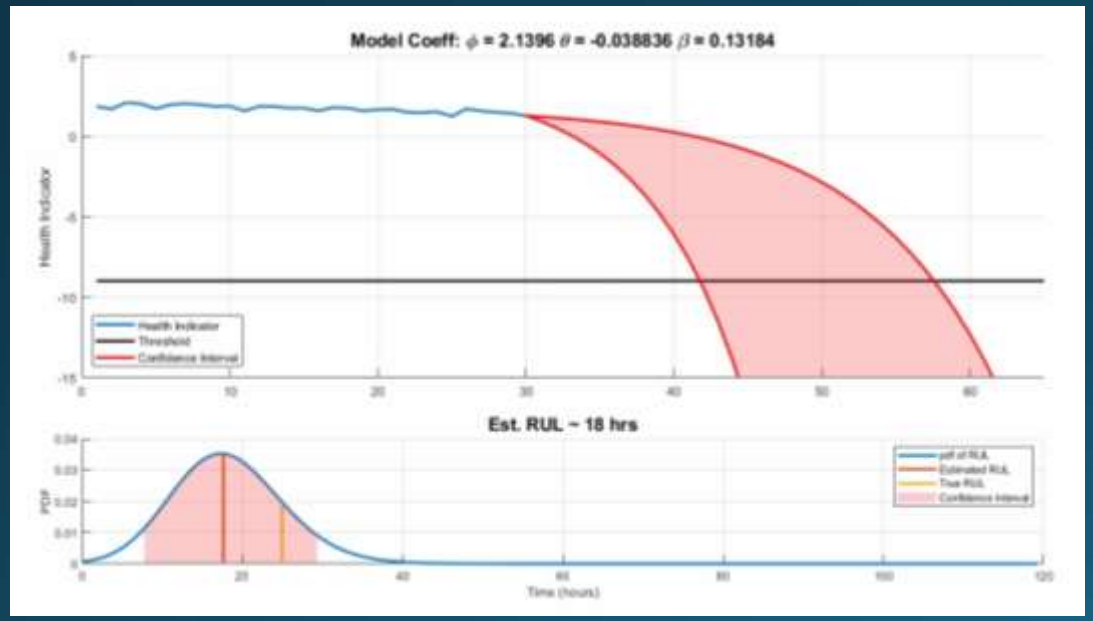
Histogram: FeatureTable1 | Feature Ranking: FeatureTable1

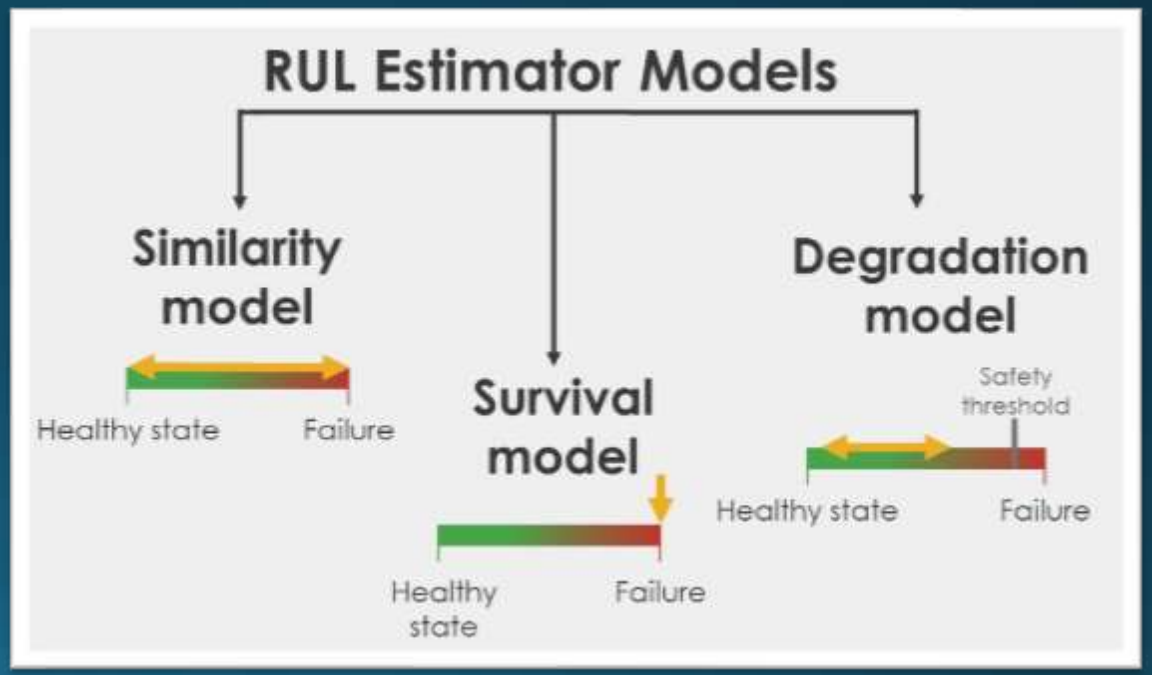


Feature	One-way ANOVA
flo...ts/Data_RMS	126.9504
pre...ts/Data_RMS	124.5145
pre...ts/Data_Mean	124.2274
flo...ts/Data_Mean	118.1523
pre...ec/Data_PeakAmp1	105.5479
flo...ec/Data_PeakAmp1	101.7428
flo...ts/Data_CrestFactor	74.1650
flo...ts/Data_ImpulseFactor	71.7221
flo...ts/Data_ClearanceFactor	69.6891
flo...ts/Data_THD	67.8899
flo...ec/Data_PeakAmp3	65.4610
pre...ts/Data_THD	63.6488
pre...ts/Data_CrestFactor	61.9233
pre...ts/Data_ImpulseFactor	61.8776
pre...ts/Data_ClearanceFactor	61.8542
pre...ec/Data_PeakAmp3	53.6584
pre...ec/Data_PeakAmp2	47.9822
flo...ec/Data_PeakAmp2	44.0777
flo...ts/Data_SNR	37.7644
pre...ts/Data_SNR	37.5568
flo...ec/Data_Wn1	36.7957
pre...ec/Data_Wn1	36.2216
pre...ec/Data_Zeta1	34.7913
flo...ec/Data_PeakAmp4	32.9371

Feature ranking plot for "FeatureTable1" is in focus.

Learn further: Estimate Remaining Useful Life

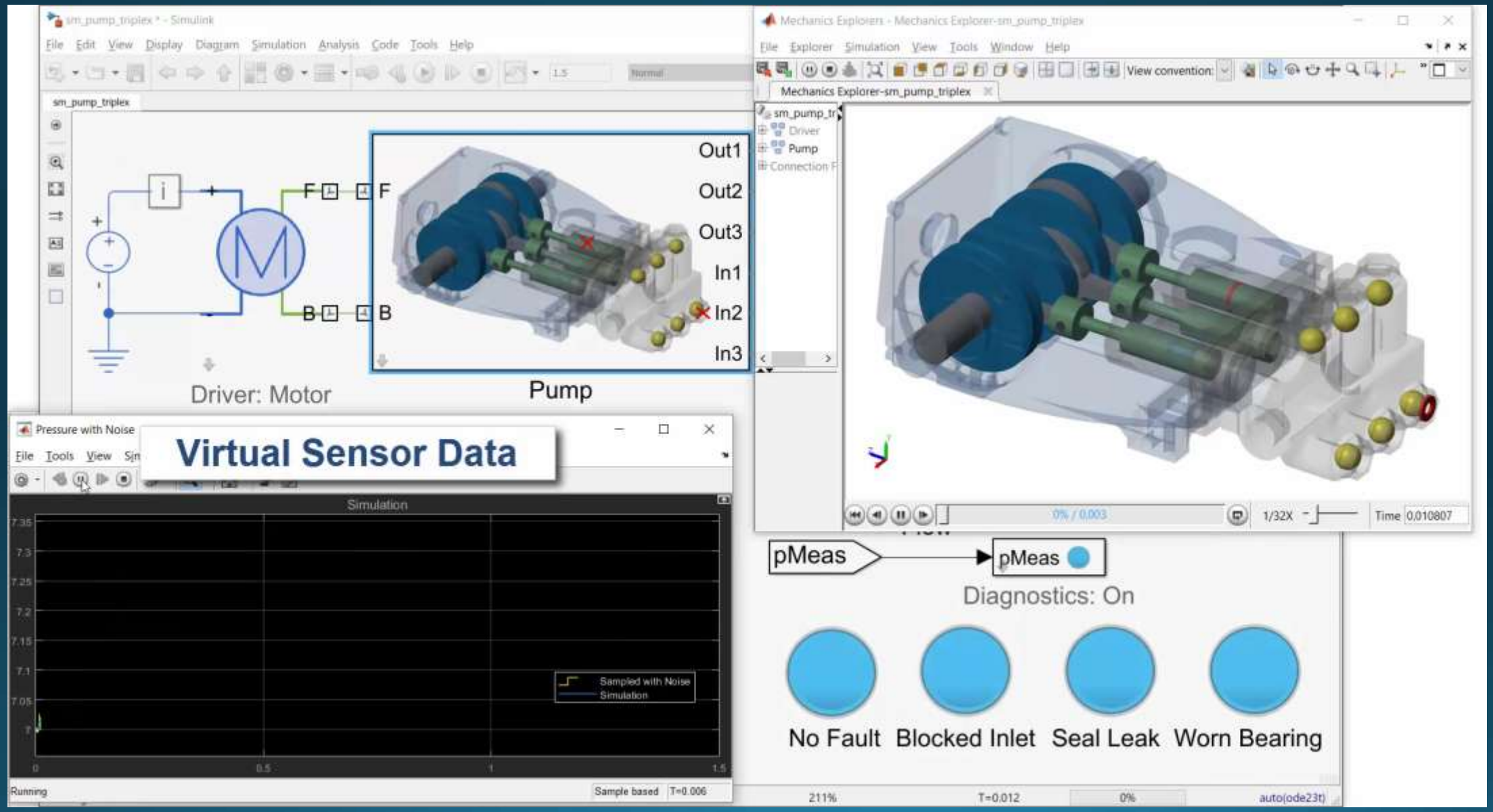


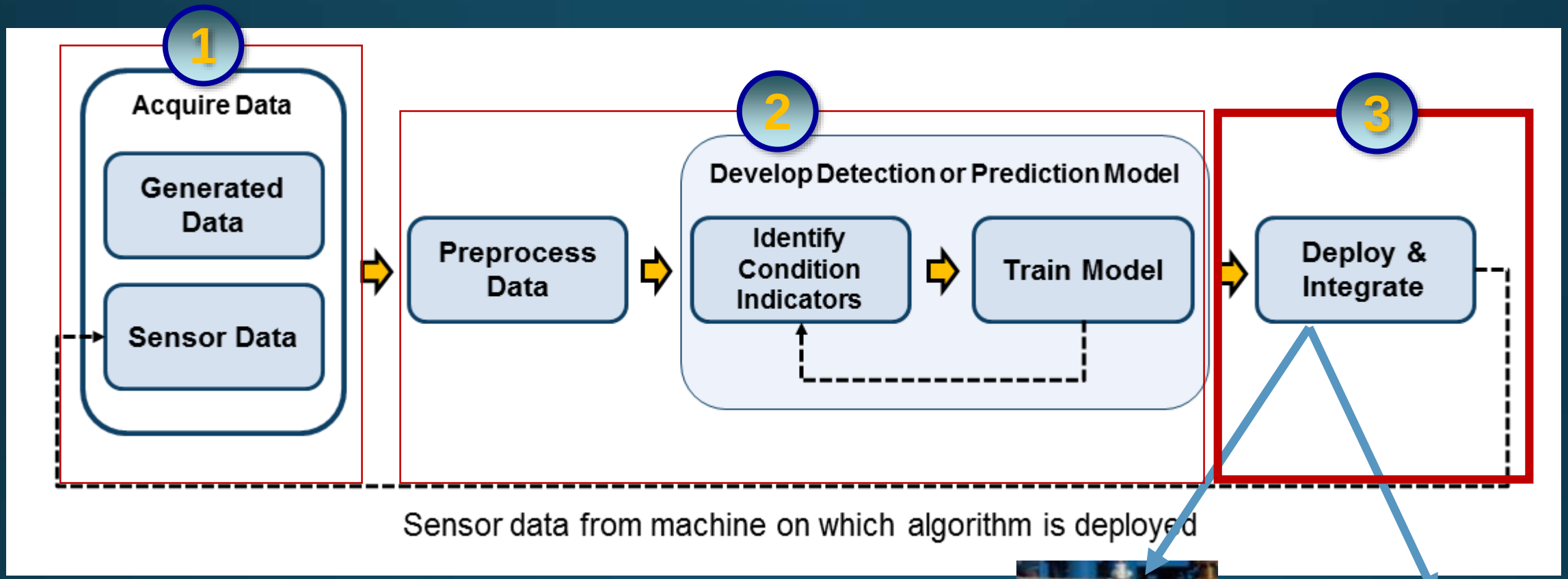


$$S(t) = \phi + \theta(t)\, e^{\left(\beta(t)t + \epsilon(t) - \frac{\sigma}{2}\right)}$$

Digital twin – Physics Based Model + Data Driven Model working together:

Once tuned acts as Golden Reference



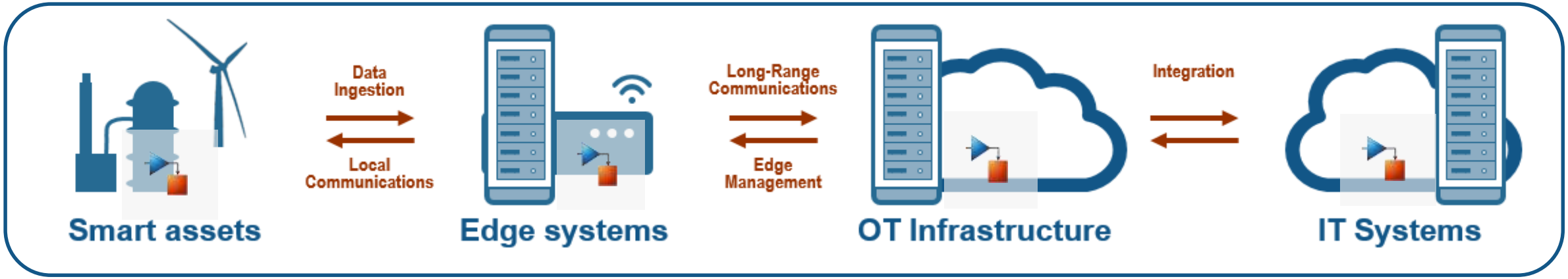


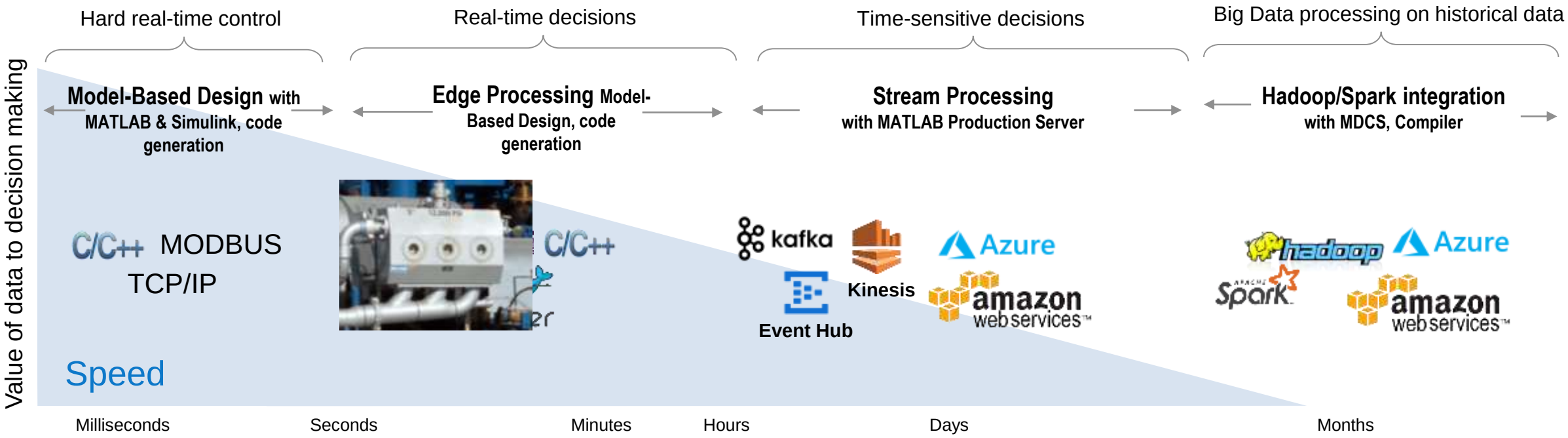
Edge device

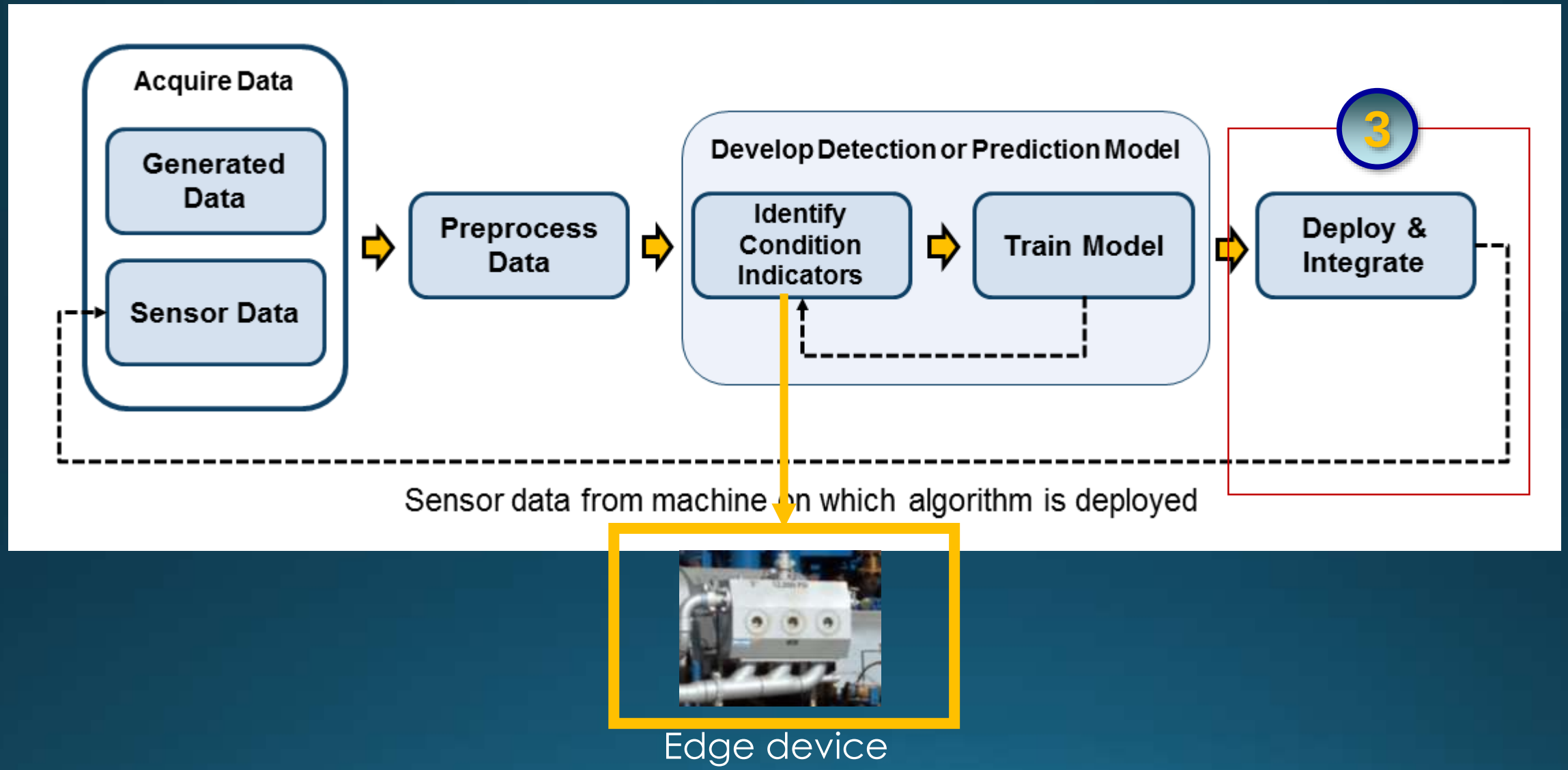
Challenge:

- Decide where to deploy these algorithms – edge device / cloud

How much time your AI model has to act?





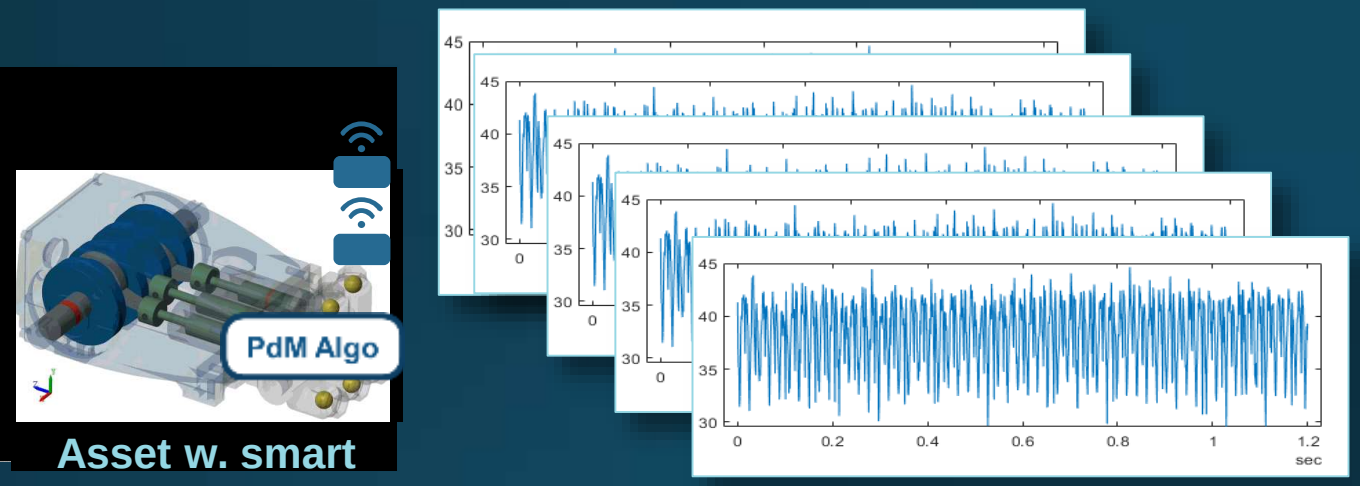


Challenges

- Decide where to deploy these algorithms – edge device / cloud

Feature Extraction Algorithm at the Edge

Pump flow sensor *1 sec ~ 1000 samples ~16kB*



- 1 day ~ 1.3 GB
- 20 sensors/pump ~26 GB/day
- 3 pumps ~ 78 GB/day

Challenge:
Data transmission cost is pretty high

Solution:
Extract only relevant information and send it to predictive model

MATLAB R2018a

HOME | PLOTS | APPS | EDITOR | PUBLISH | VIEW

New | Open | Save | Find Files | Compare | Print | Go To | Find | Insert | Comment | Indent | Breakpoints | Run | Run and Advance | Run Section | Advance | Run and Time

C: ▸ Users ▸ abaru ▸ Desktop ▸ Expo 2018 ▸ FinalDemo ▸ Demo_Files ▸ Data_Reduction ▸

Current Folder

Name

- Folder
  - codegen
  - Copy_of_Data
  - Data
- Function
  - featureExtraction.m
  - featureExtractionBuffer.m
  - helperSortedBarPlot.m
  - monotonicity.m
- MEX-file
  - featureExtraction_mex.mexw64
  - featureExtractionBuffer_mex.mexw64
- Live Script
  - Expo_Data_Preprocessing_CodeGe...
- MATLAB Coder Project
  - featureExtraction.prj
  - featureExtractionBuffer.prj

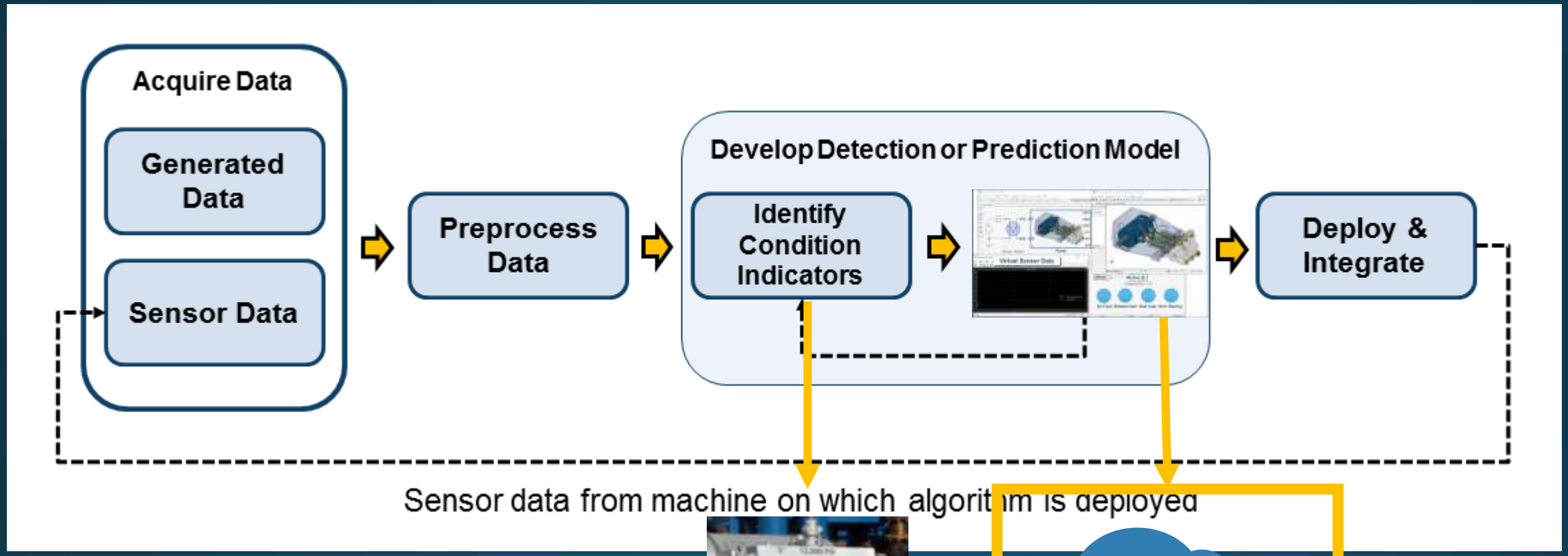
featureExtractionBuffer.m (Function)

Editor - C:\Users\abaru\Desktop\Expo 2018\FinalDemo\Demo_Files\Data_Reduction\featureExtractionBuffer.m

Expo_Data_Preprocessing_CodeGen.mlx | featureExtractionBuffer.m

```
function [feature_list] = featureExtractionBuffer(data,timestamp)

persistent flow_array
persistent time_array
Np = 1000;

if isempty(flow_array)
    flow_array = nan(Np,1);
end

if isempty(time_array)
    time_array = nan(Np,1);
end

flow_array = [data; flow_array(1:Np-1)];
data = flow_array;

time_array = [timestamp; time_array(1:Np-1)];
timestamp = time_array;


if isempty(find(isnan(data),1))

    flow = data;

    % Ensure the flow is sampled at a uniform sample rate
    t_flow = timestamp;
```

featureExtractionBuffer | Ln 17 | Col 1

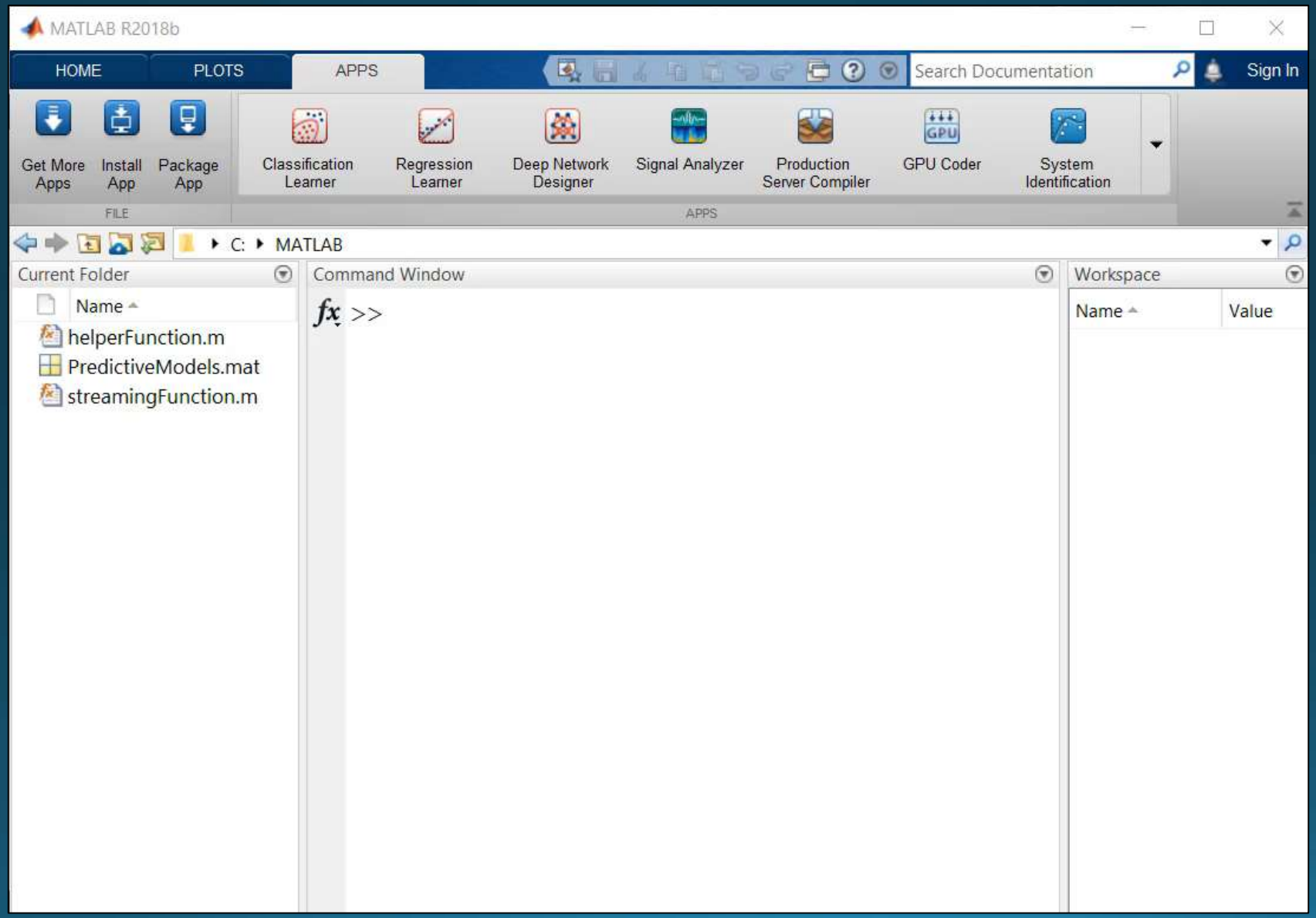


Edge device

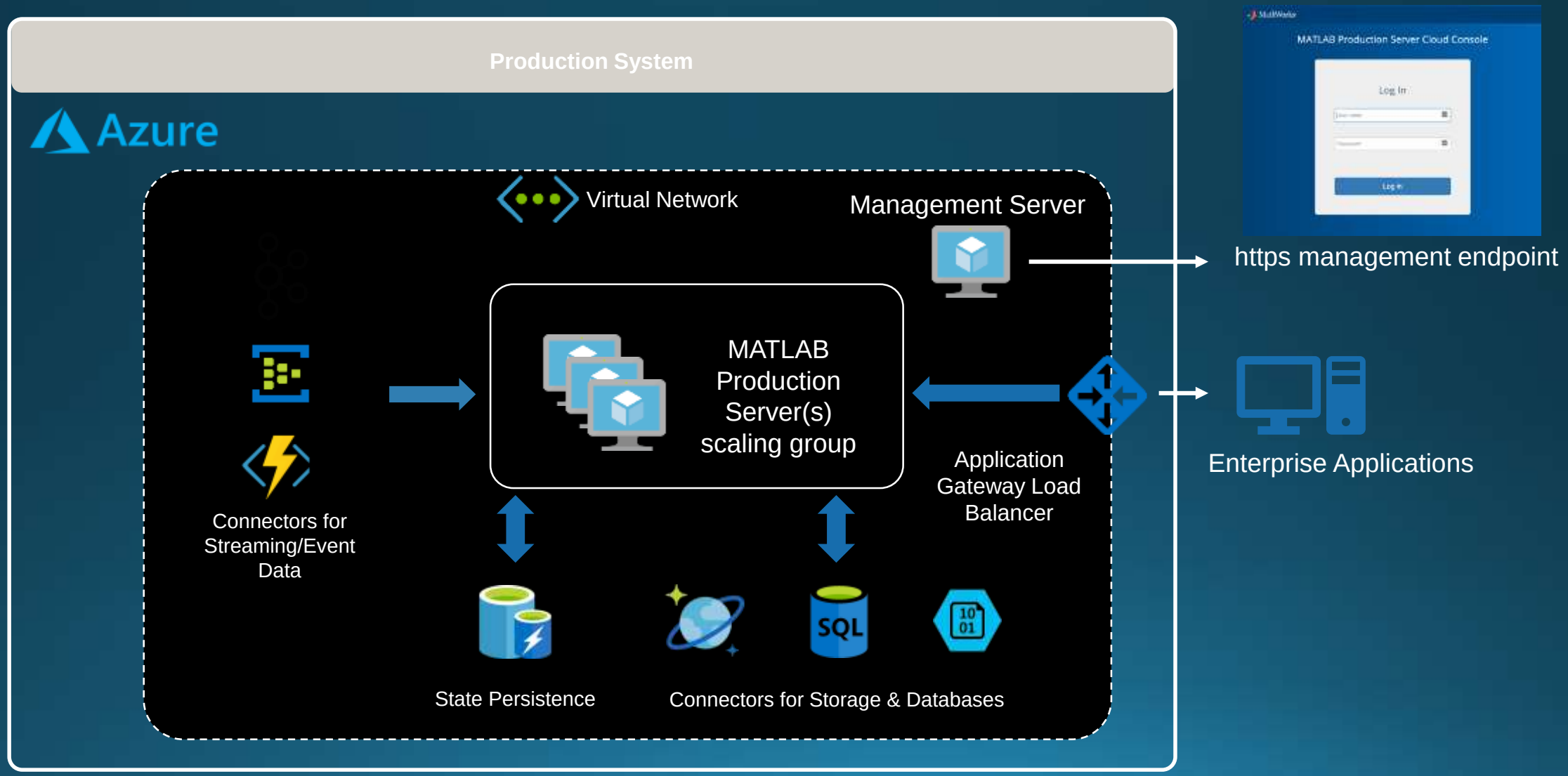
Challenges

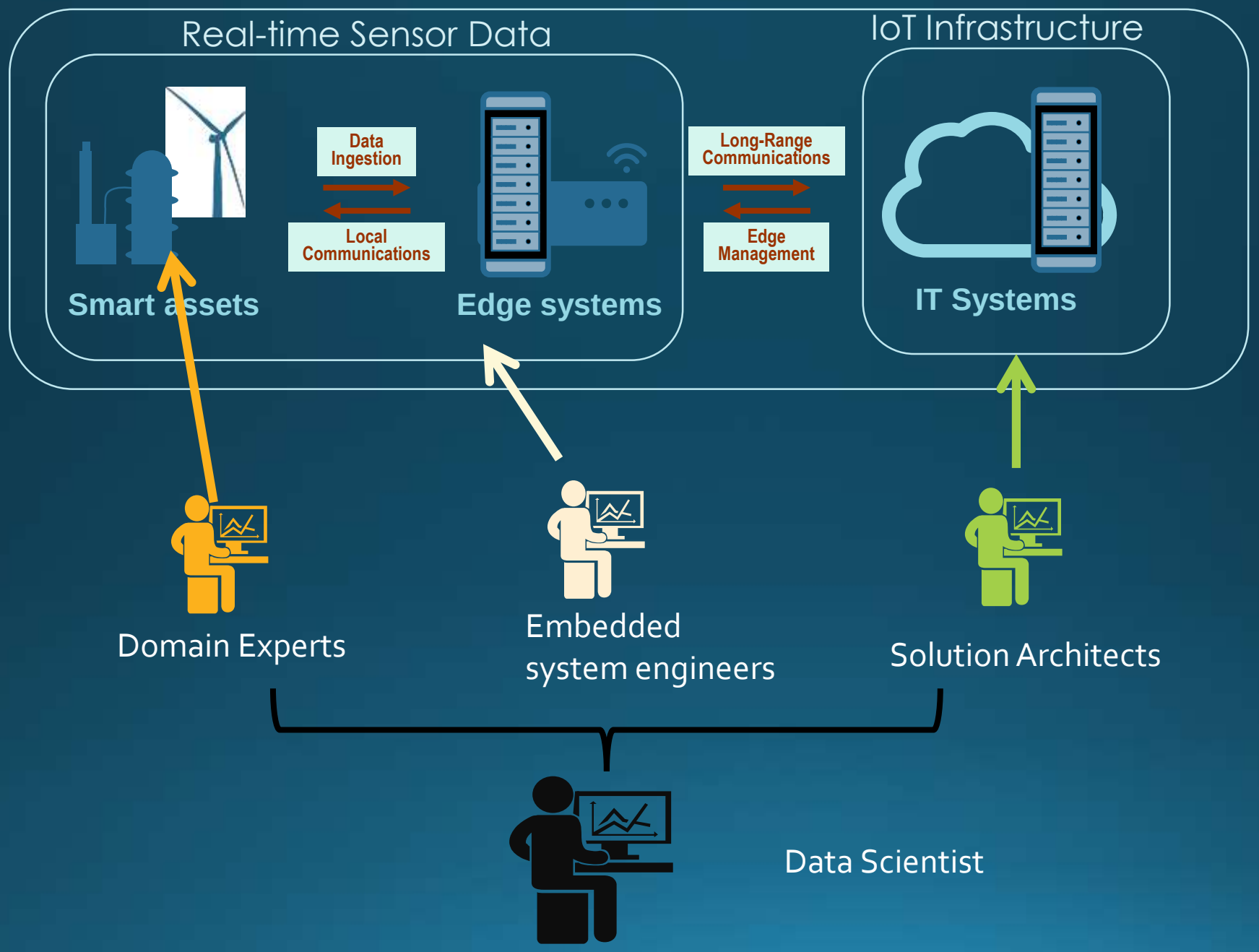
- Decide where to deploy these algorithms – edge device / cloud

Package Stream Processing Function



Receive rapid streams of data to maintain effective Digital Twins




Real-time Sensor Data
IoT Infrastructure
Data Ingestion
Local Communications
Long-Range Communications
Edge Management
Smart assets
Edge systems
IT Systems
Domain Experts
Embedded system engineers
Solution Architects
Data Scientist

Summary : Fault Classification of a Pump

Raw Data

Flow

Pressure

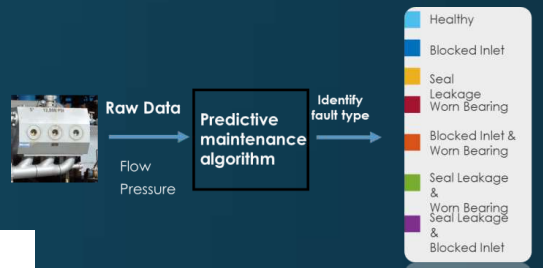
Predictive maintenance algorithm

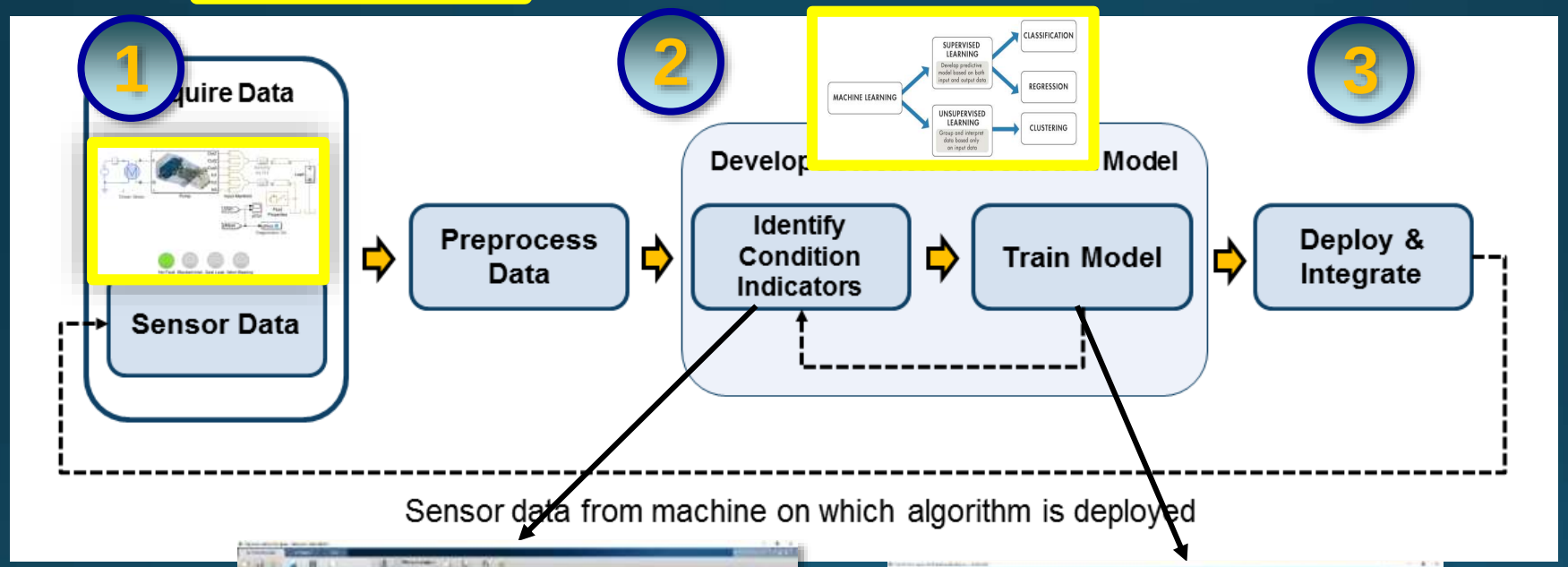
Identify fault type

- Healthy
- Blocked Inlet
- Seal Leakage
- Worn Bearing
- Blocked Inlet & Worn Bearing
- Seal Leakage & Worn Bearing
- Seal Leakage & Blocked Inlet

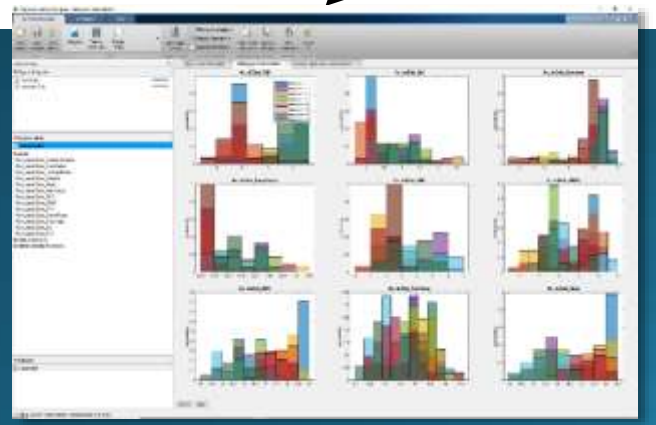
Summary : Fault Classification of a Pump

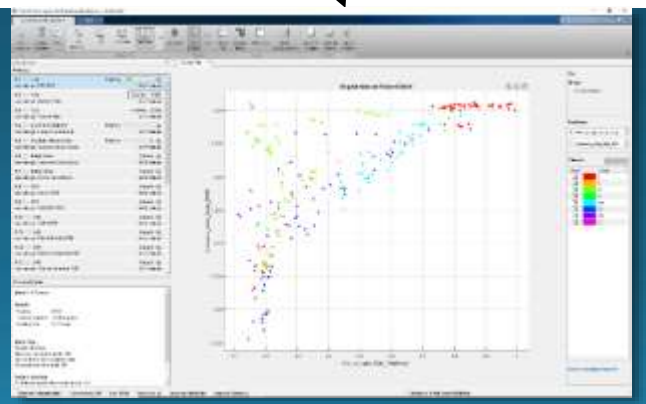
Approach **Digital Twin** = Physics Based + Data Driven





Edge device





Digital Twin: Other Applications

Example Applications in Operation
Asset Performance Management (APM)
Predictive Maintenance
Operations Optimization
Business Optimization
Advanced Controls/Edge Computing
Cyber (security)

Ensuring power grid reserves

Cloud-based HVAC optimization

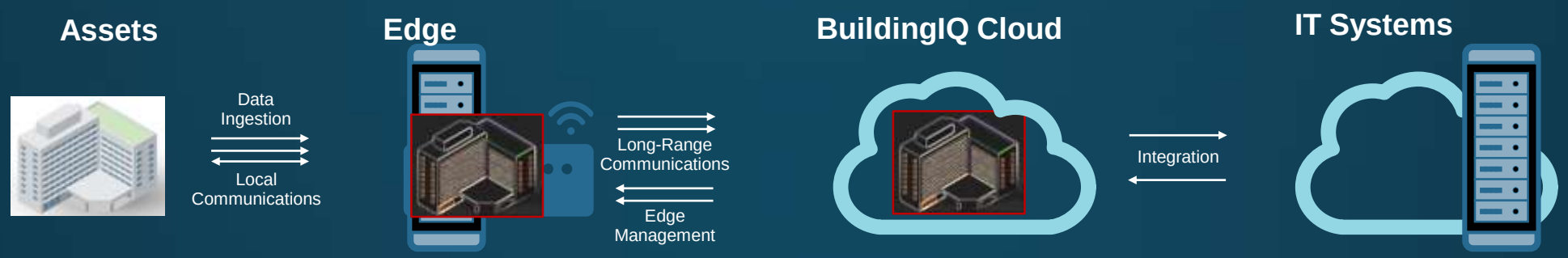
Efficiency improvement

BuildingIQ Develops Proactive Algorithms for HVAC Energy Optimization in Large-Scale Buildings

Challenge

Develop a real-time system to minimize HVAC energy costs in large-scale commercial buildings via proactive, predictive optimization

Example: BuildingIQ (Operations Optimization)



25% cost reduction

Supervisory Control

HVAC hard-real-time closed-loop control

Current building condition

Filtering, processing and visualizing

Optimal set points on each building

Supervisory control applied to HVAC

Operations Optimization

Multi-objective optimization to optimize periodically for energy efficiency

Time of Use Energy Price
Demand Forecast
Projected Weather

Robustness analysis done

Action applied across multiple buildings

IoT Applications Using Digital Twins

Transpower Ensures Reliability of New Zealand National Grid with Reserve Management Tool

Tata Steel Saved 40% on Cooling Towers Through Software Algorithms

Predictive Maintenance, Part 5: Digital Twin

new

Related Topics

IoT

Predictive Maintenance

Physical Modeling

"We predicted that by 2022, over two-thirds of companies that have implemented IoT will have deployed at least one digital twin in production. We might actually reach that number within a year." - *Gartner*

What’s your Digital Twin?

Thank You!!

Email: adoshi@mathworks.com
LinkedIn: https://www.linkedin.com/in/amit-doshi

@Mr_AmitD

- **Visit MathWorks Demo Booth** :
  - Develop and Deploy Deep Learning Algorithms
  - Predictive Maintenance and Digital Twin
- **Register for a workshop on**:
  - Addressing Deep Learning Challenges
  - 10th Aug@4:30 pm
- Are you Start-up?
  - Talk to us!