

# Advanced Capabilities for Embedding Machine Learning into ECUs

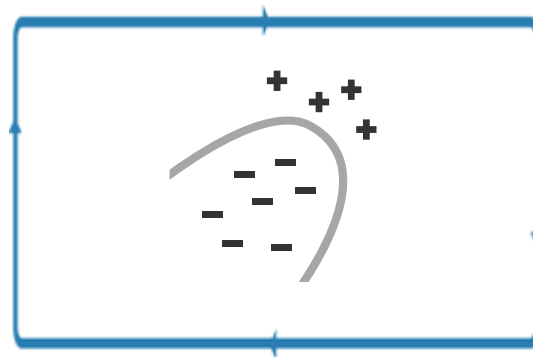
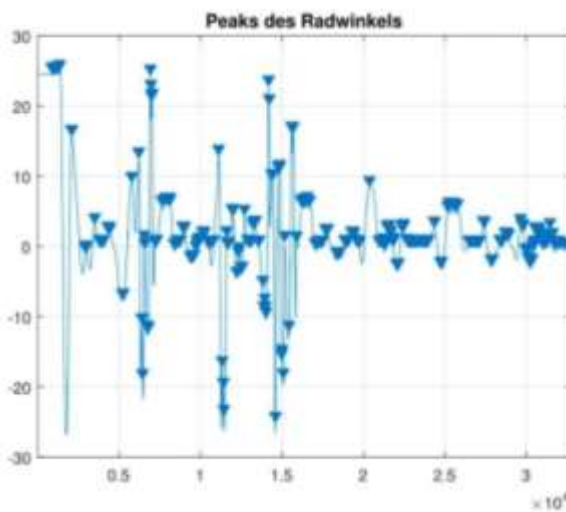
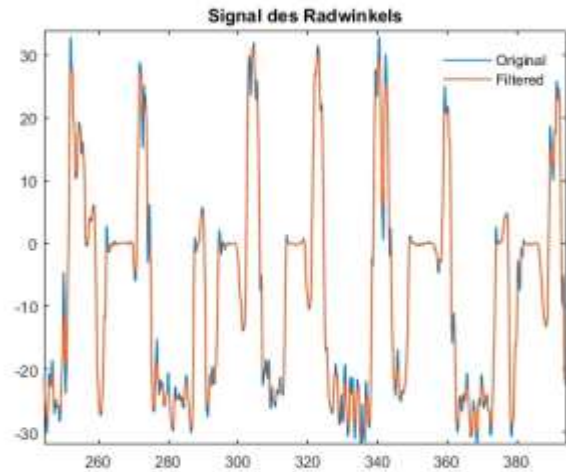
Christoph Stockhammer

July 02 | Europe

MathWorks  
AUTOMOTIVE  
CONFERENCE 2020

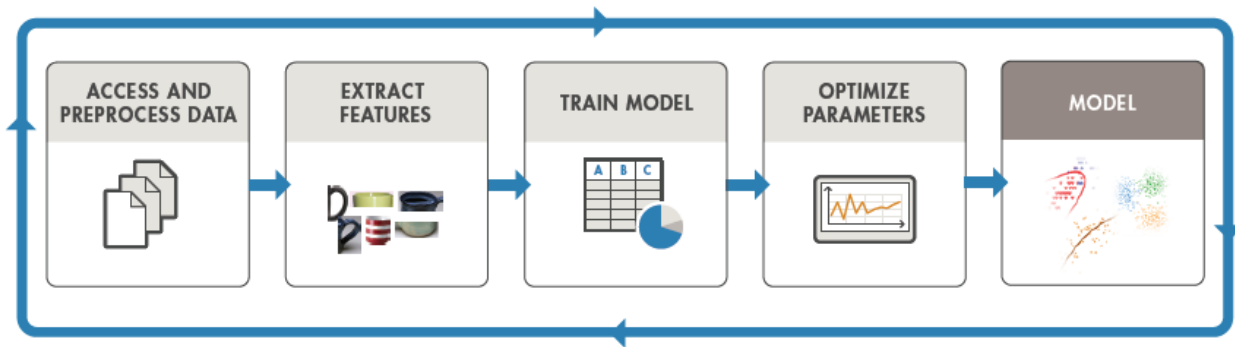


# BMW designs, tests and deploys data-driven systems that enhance vehicles' capabilities with MATLAB & Simulink



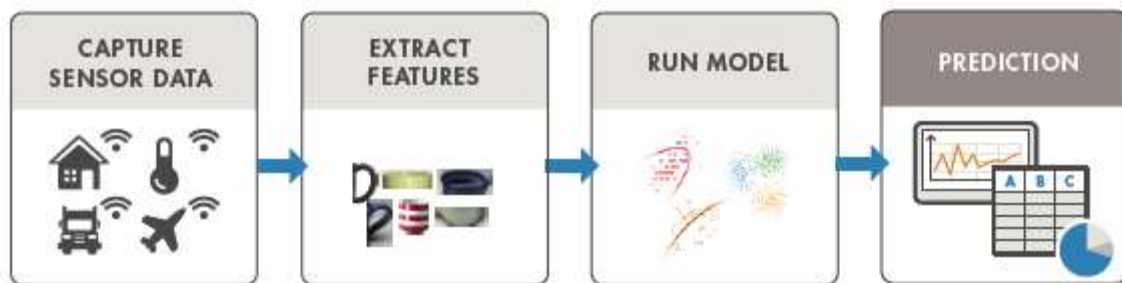
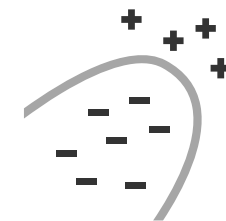
> 95% accuracy

# MathWorks provides tools to design and verify smart, data-driven machine learning systems



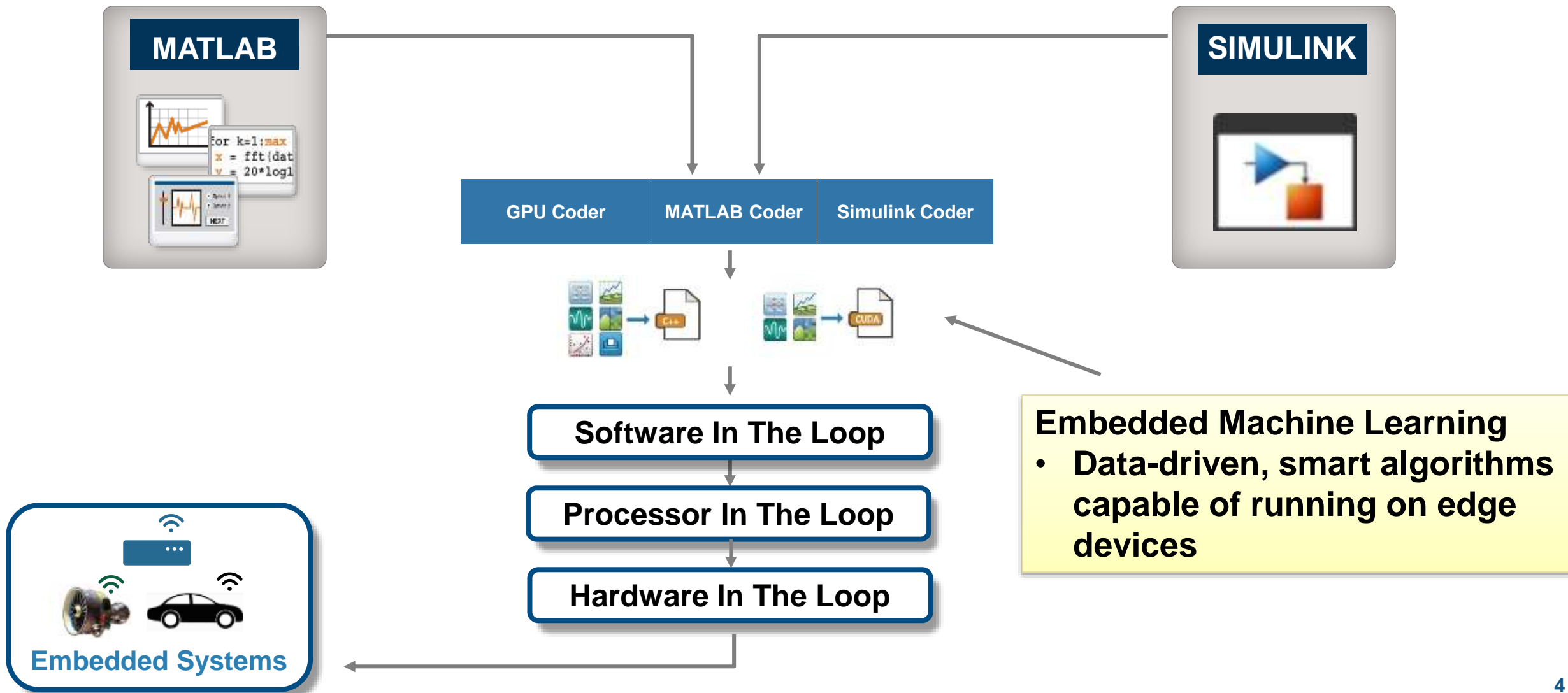
★  $f(x) = y$

$\hat{f}(x_{train}) = y_{train}$



$\hat{f}(x_{new}) = \hat{y}$  ✓

# MathWorks provides embedded machine learning workflows that integrate nicely with Model-Based Design



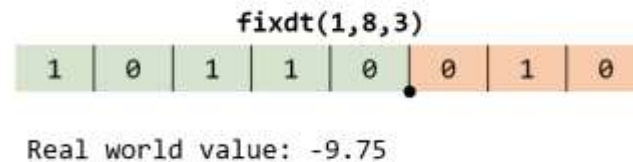


# Machine Learning algorithms are supported for a variety of embedded systems workflows

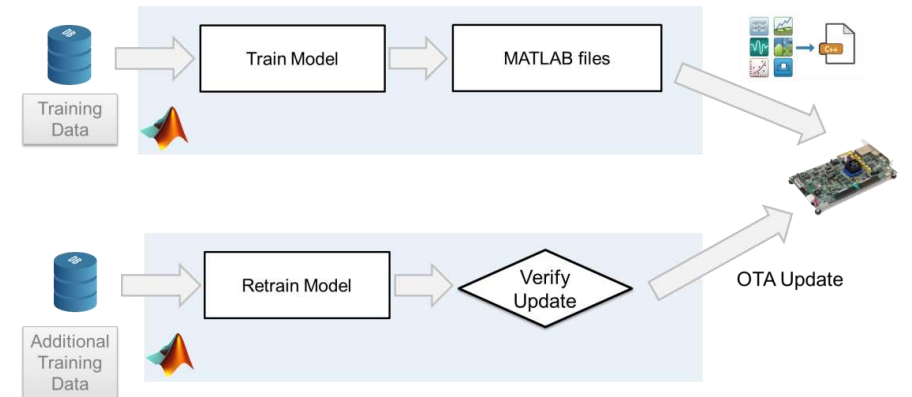
*Deploy machine learning models in MATLAB & Simulink*



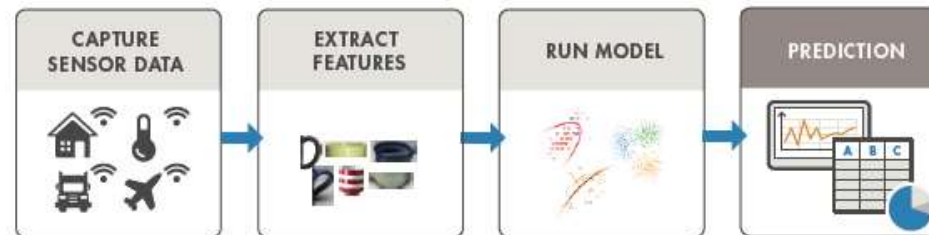
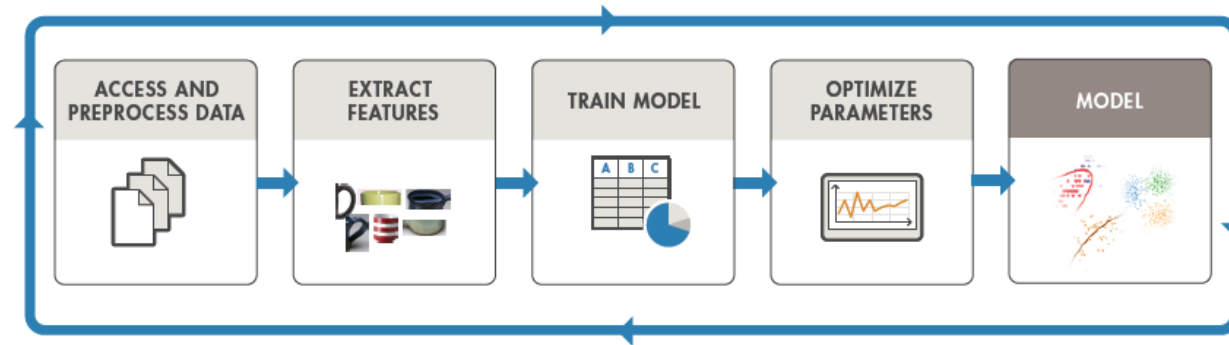
*Deploy reduced precision machine learning models*



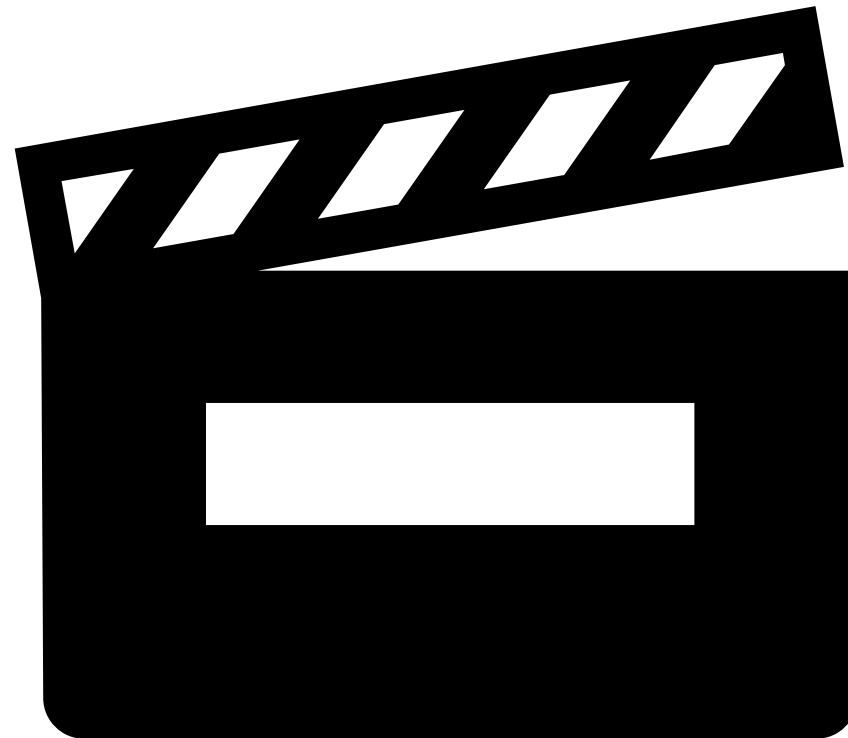
*In-place modification of deployed models*



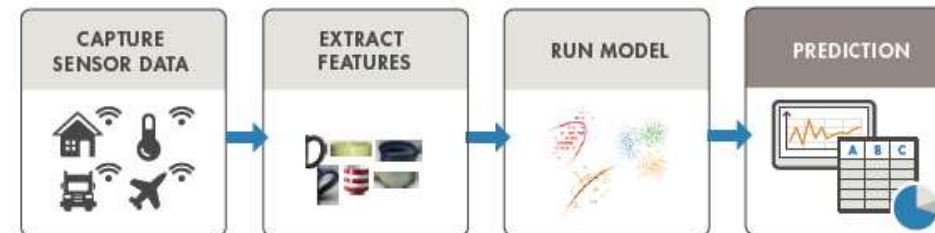
Learner Apps provide convenient ways to compare and iterate over different machine learning algorithms



# Classification learner App demonstration



# Model trained using Learner App can be saved for deployment



## Extract Trained Model

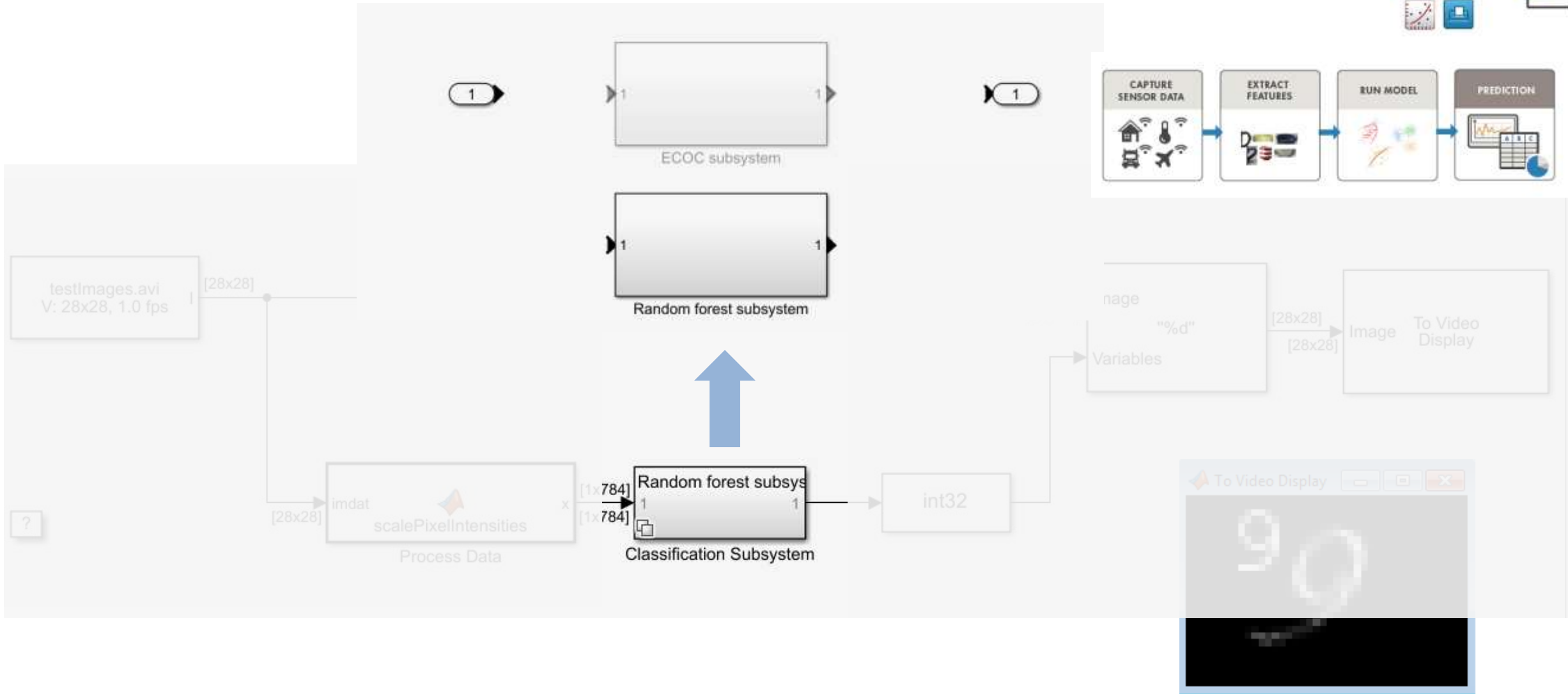
```
ensembleModel =  
  
struct with fields:  
  
    predictFcn: @(x)exportableModel.predictFcn(predictorExtractionFcn(x))  
    ClassificationEnsemble: [1x1 clasreg.learning.classif.CompactClassificationEnsemble]  
    HyperParameterOptimizationResult: [1x1 BayesianOptimization]  
    About: 'This struct is a trained model exported from Classification Learner R2020a.'  
    HowToPredict: 'To make predictions on a new predictor column matrix, X, use: ↵ yfit = c.predictFcn(X)'
```

## Save Trained Model for Deployment

```
saveLearnerForCoder(ensembleModel.ClassificationEnsemble, 'DigitImagesRF');
```

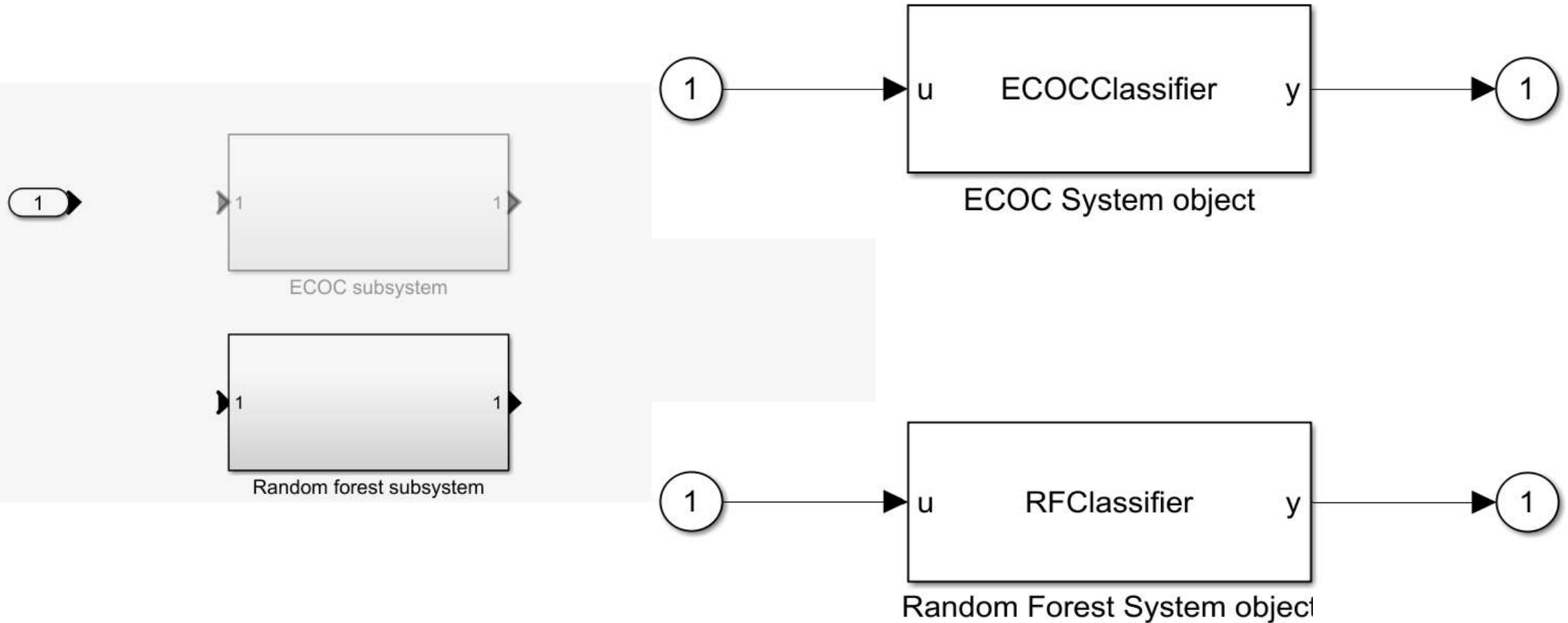


# Saved models can be used in Simulink models



```
openExample('stats/SystemObjectsForClassificationAndCodeGenerationExample')
```

# Saved models can be used in Simulink models



# Saved models can be used in Simulink models via System Blocks



```
classdef RFClassifier < matlab.System
% RFCLASSIFIER Predict image labels from trained random forest
%
% RFCLASSIFIER loads the trained random forest from
% |'DigitImagesRF.mat'|, and predicts labels for new observations based
% on the trained model. The random forest in |'DigitImagesRF.mat'|
% was cross-validated using the training data in the sample data
% |digitimages.mat|.

properties(Access = private)
    CompactMdl % The compacted, trained random forest
end

methods(Access = protected)

function setupImpl(obj)
% Load random forest from file
obj.CompactMdl = loadLearnerForCoder('DigitImagesRF');
end

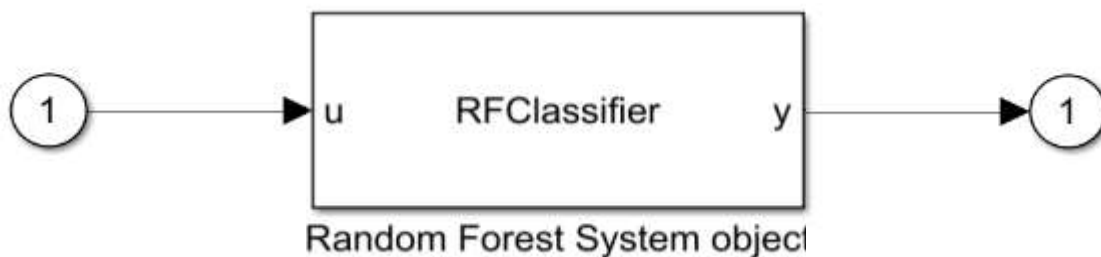
function y = stepImpl(obj,u)
y = predict(obj.CompactMdl,u);
end

function flag = isInputSizeMutableImpl(obj,index) %#ok<INUSD>
% Return false if input size is not allowed to change while
% system is running
flag = false;
end

function dataout = getOutputDataTypeImpl(~)
dataout = 'double';
end

function sizeout = getOutputSizeImpl(~)
sizeout = [1 1];
end

end
end
```



# Majority of machine Learning models are supported for Deployment



## Supported Models

- Linear Classification
- SVM
- Decision trees and Random Forests
- Linear Discriminant Analysis
- k-Nearest Neighbor models
- Ensemble models
- Naïve Bayes models
- Gaussian Process
- Linear/Generalized Linear Regression models
- Regression

## Simulink

- MATLAB Function Block
- MATLAB System Block
- Stateflow

*Deploy machine learning models  
in MATLAB & Simulink*



## Deploy machine learning models in MATLAB & Simulink



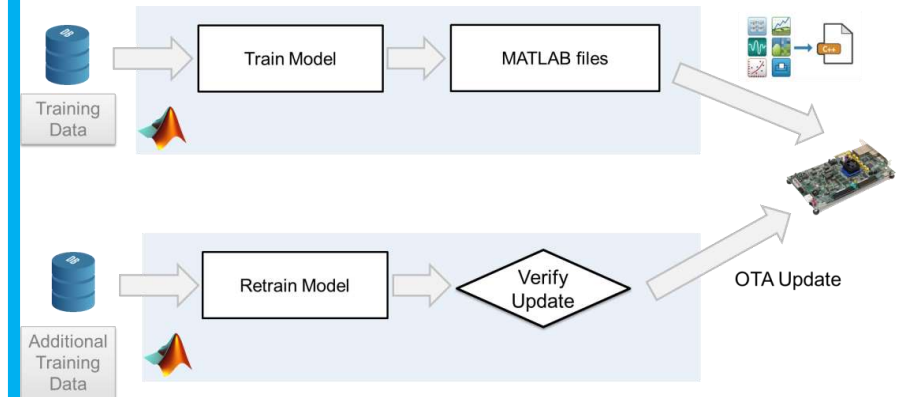
## Deploy reduced precision machine learning models

`fixdt(1,8,3)`

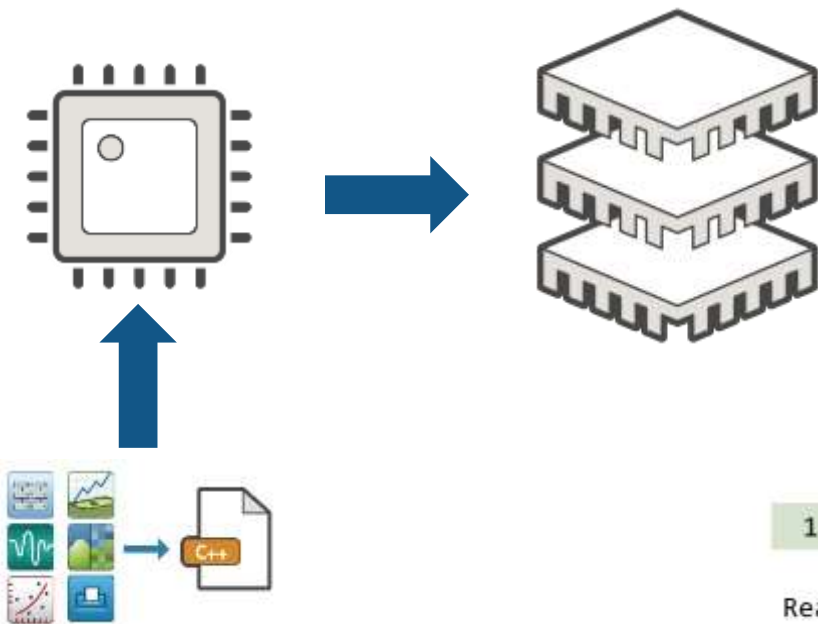
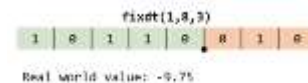


Real world value: -9.75

## In-place modification of deployed models

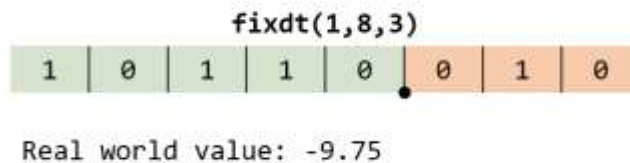


# Deploy reduced precision machine learning models



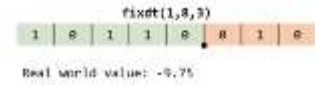
**Minimize energy consumption on EV's**

**Reduce cost**



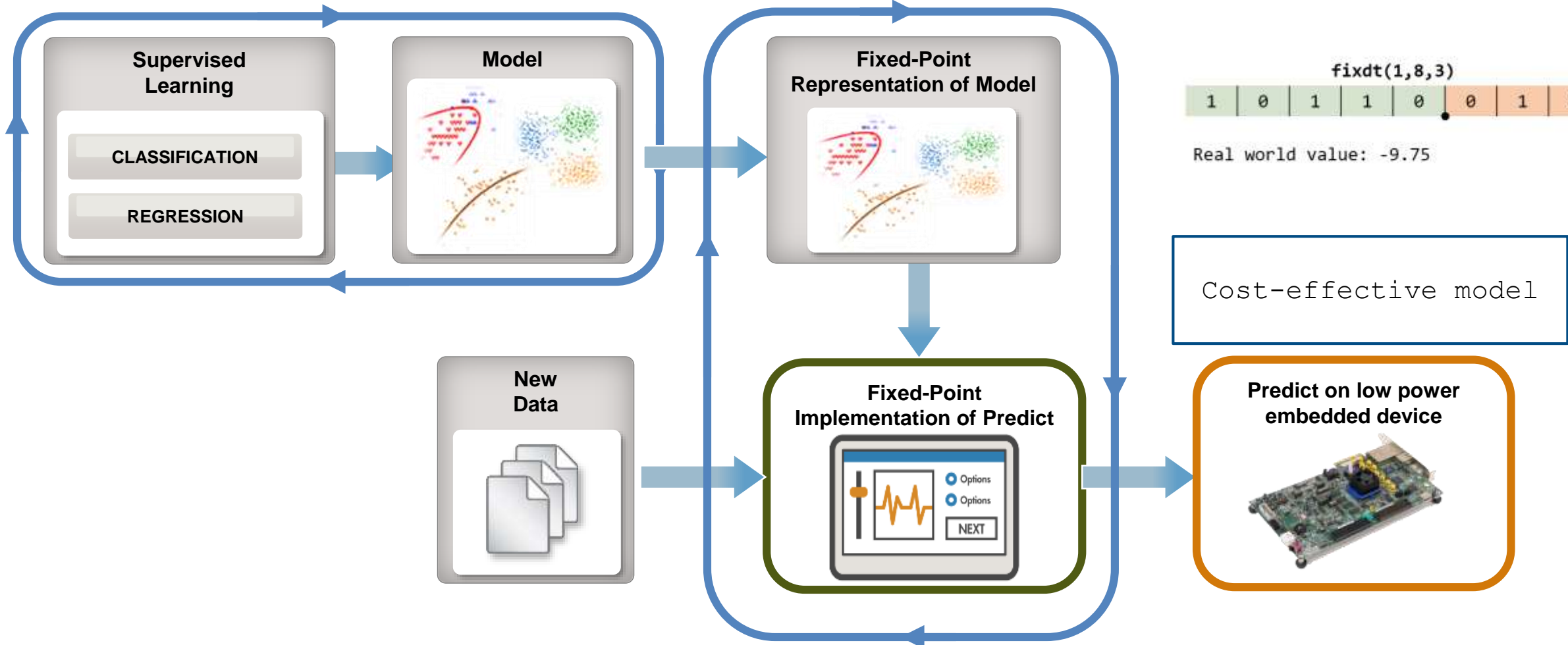


# Reduced precision workflows allow conversion to fixed-point and deployment of models with small memory footprint



## Train in MATLAB

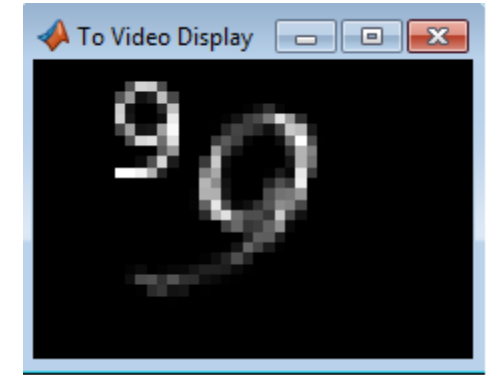
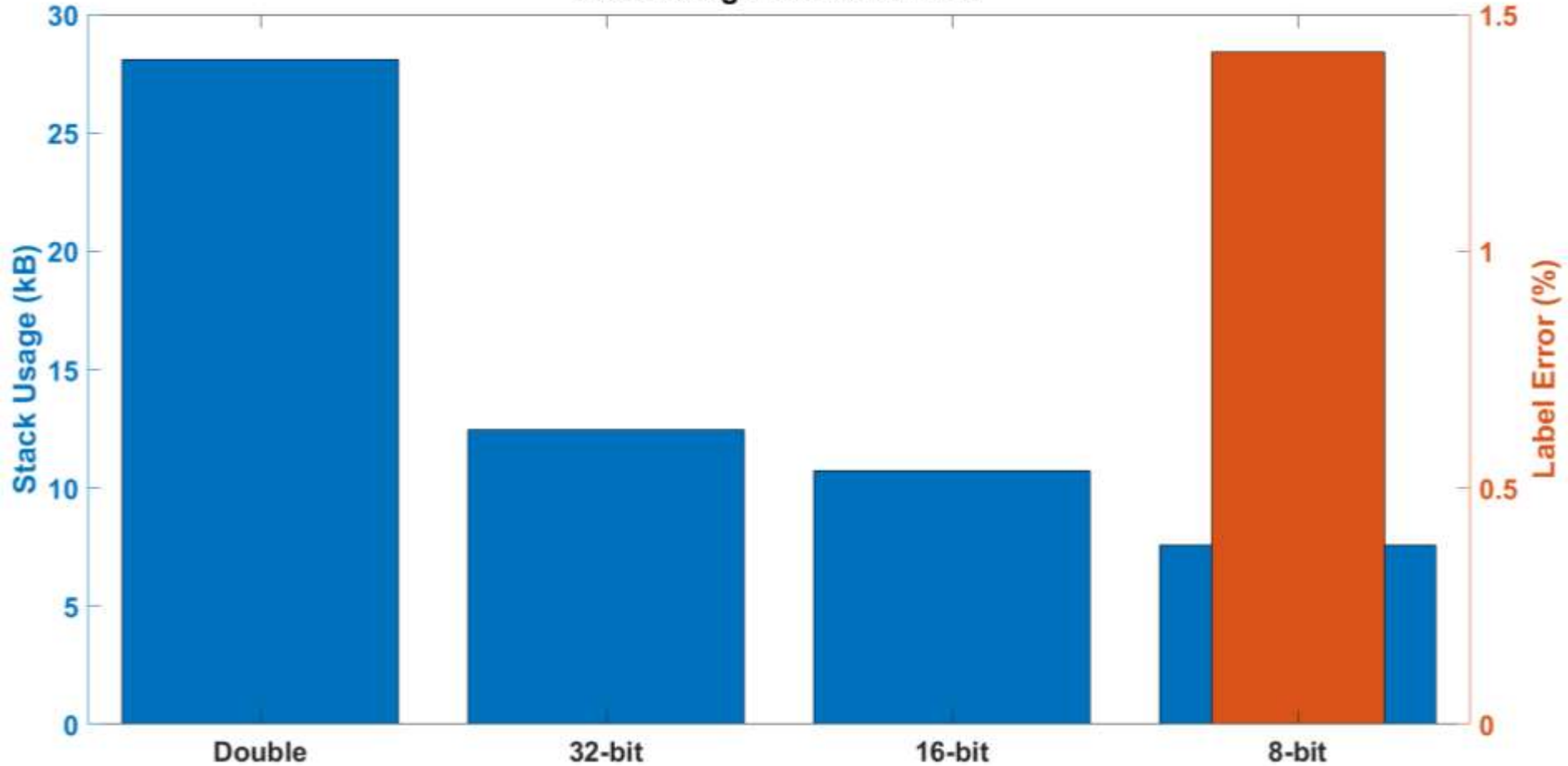
## Convert in Fixed-Point Designer



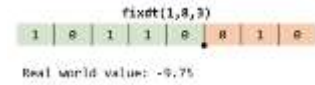
# Fixed-point conversion is a trade-off between resource usage optimization and accuracy

```
fixdt(1,8,3)
1 | 0 | 1 | 1 | 0 | 0 | 0 | 0
Real world value: -0.75
```

Stack usage vs. Label Error



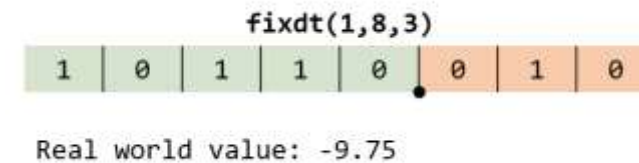
# Most Popular machine learning models are supported for fixed-point workflows



## Reduced Precision: Supported Models

- SVM
  - Multi-class not supported
- Decision Trees
- Ensembles of Decision Trees

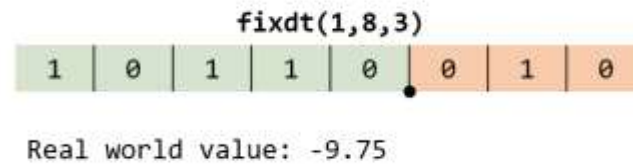
*Deploy reduced precision machine learning models*



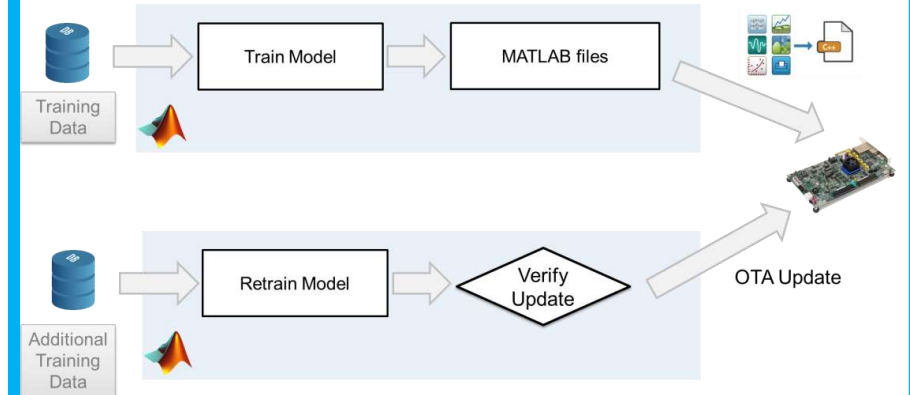
## Deploy machine learning models in MATLAB & Simulink



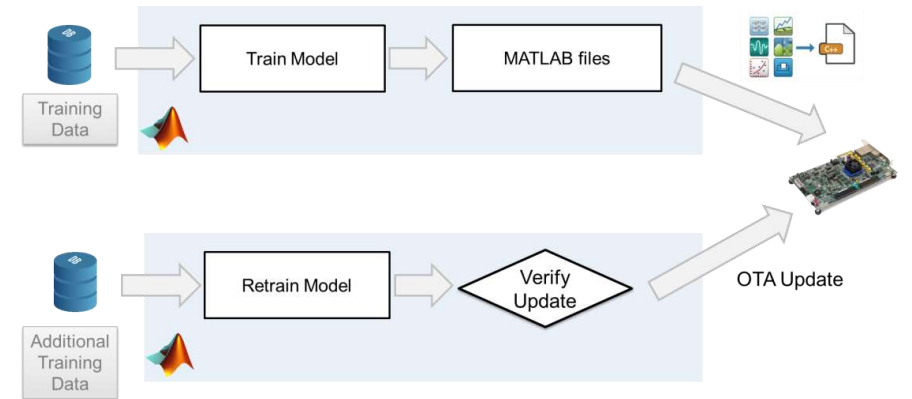
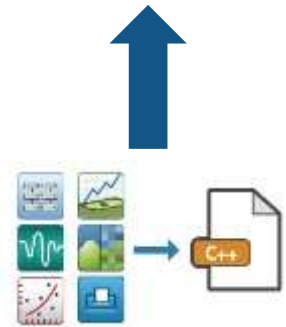
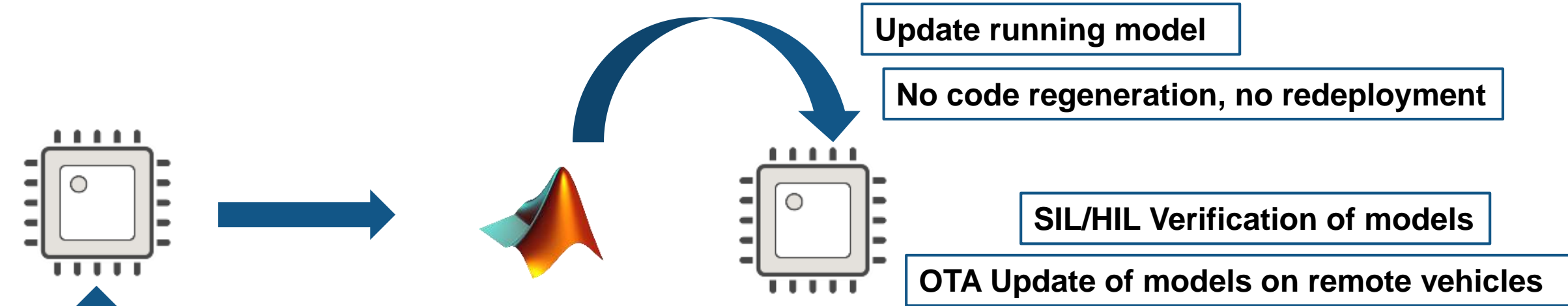
## Deploy reduced precision machine learning models



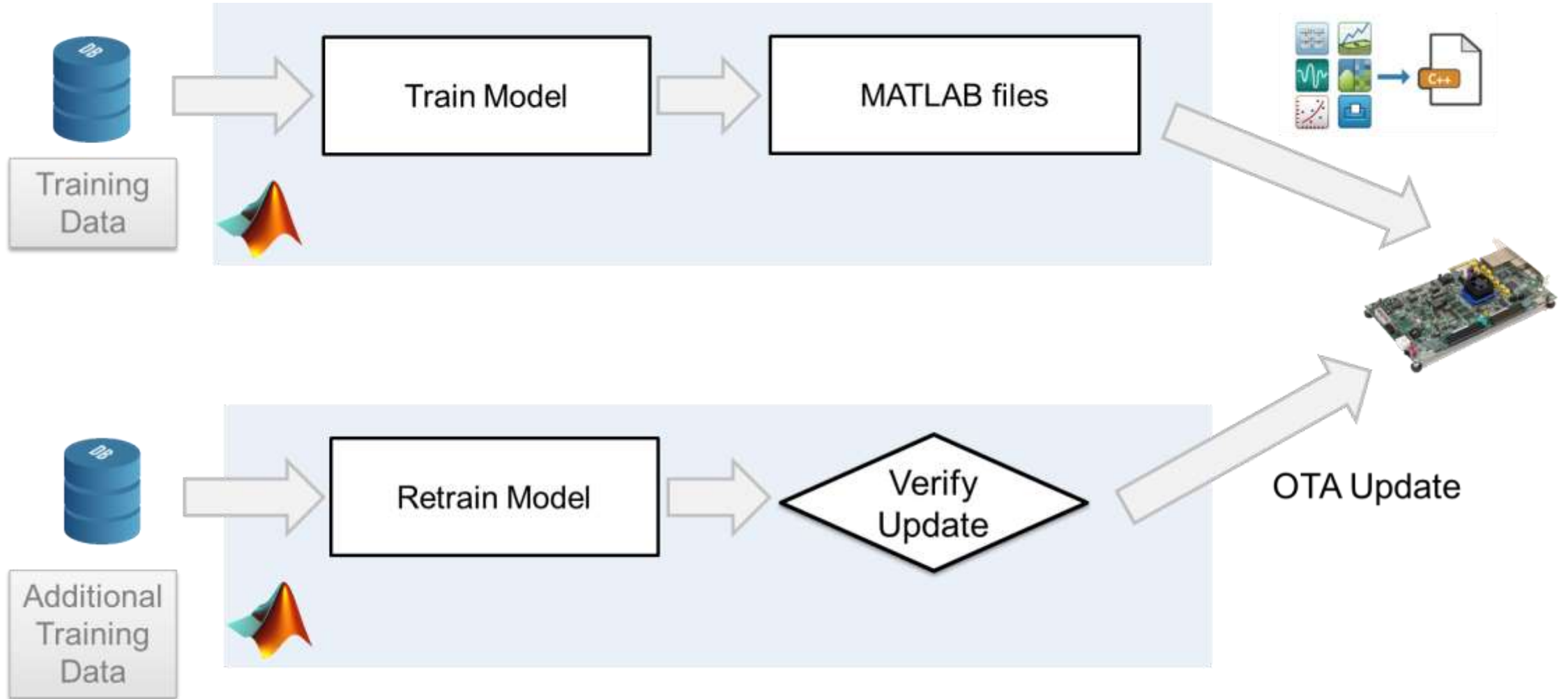
## In-place modification of deployed models



# In-place modification of deployed models



# In-place modification of deployed models allows model updates without code regeneration

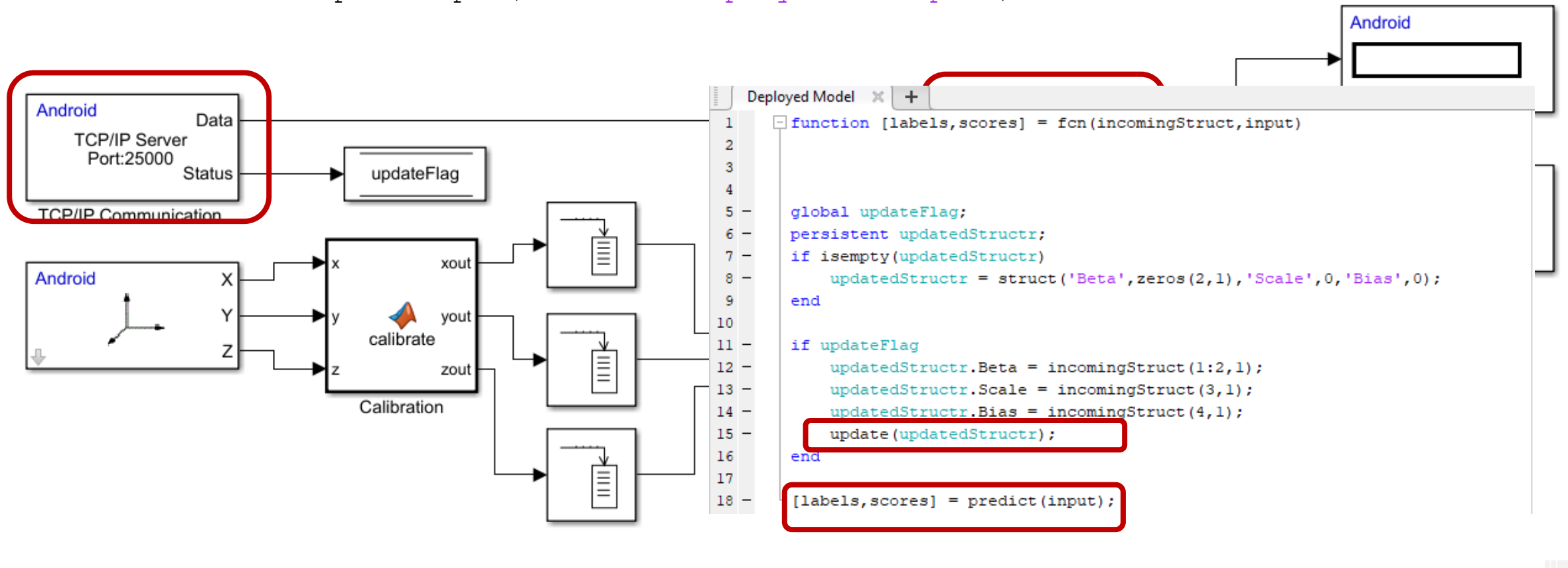




# In-place modification workflow is agnostic to communication method, works in Simulink

androidModel\_linear\_beta

Modified version of `openExample('stats/HARDeploymentExample')`

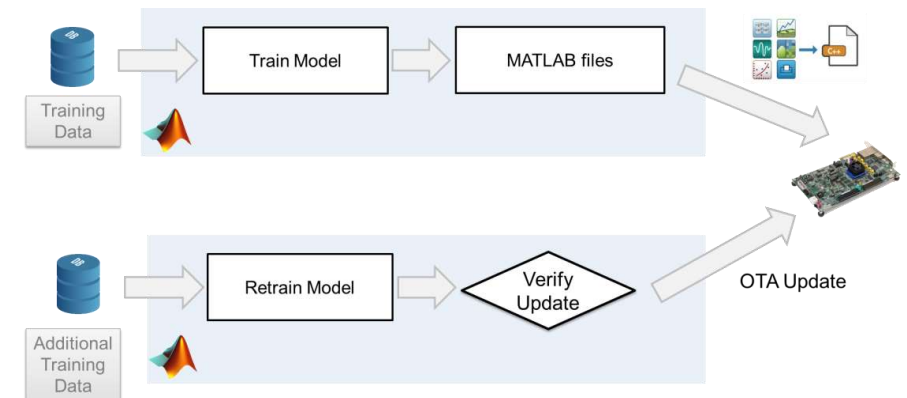


# Most Popular machine learning models are supported for in-place modification workflows

## In-place modification: Supported Models

- SVM
- Linear Models
- Decision Trees

## *In-place modification of deployed models*

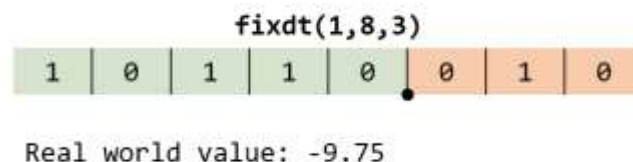


# Machine Learning algorithms are supported for a variety of embedded systems workflows

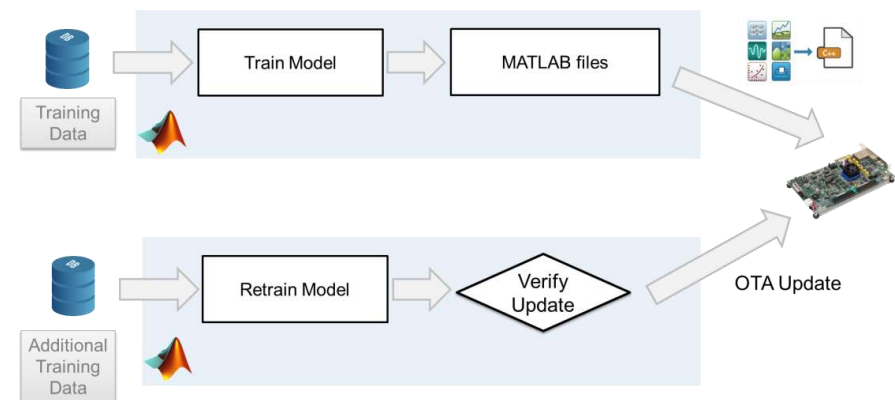
*Deploy machine learning models in MATLAB & Simulink*



*Deploy reduced precision machine learning models*



*In-place modification of deployed models*



# Q&A

Are you already working on a project that involves deploying a machine learning model to an edge device?

**A** YES

**B** NO

If you have questions, please reach out:



[cstockha@mathworks.com](mailto:cstockha@mathworks.com)