Train a Long Short-Term Memory Network for Predicting Displacement of CNC Machine Centre using Temperature Sensor Data

Use a Long Short-Term Memory (LSTM) network to predict z-axis error due to the thermal deformation of CNC machine tools.

```
% load data files for use with several Deep Learning apps
clear; clc; close all;

% change the file path to load a different csv file
datapath = "C:\Users\Mareim\OneDrive - Coventry University\Data\Data on CNC temp\data\2017-03-0

dateOfDay = extractBetween(datapath,"data\",".csv");

[tsdata, ndata] = importfile1(datapath, [2, Inf]);
interval = 1:size(tsdata,1);
tsdata_interval = tsdata(interval,:);
tt1 =  table2timetable(tsdata_interval);
% fix date and time
t = tsdata.Time;
d = datetime(dateOfDay);
t.Year = d.Year;
t.Month = d.Month;
t.Day = d.Day;


% Response and predictor (temperature sensor feeds) variables (pre-normalised)


X_pre = tsdata.X(interval);
Y_pre = tsdata.Y(interval);
Z_pre = tsdata.Z(interval);


predictors_pre = tsdata{interval,5:end};%
ts_1 = tsdata(interval,1);
% Response and predictor variables (post-normalised)

Xval = ndata.X;
Yval = ndata.Y;
Zval = ndata.Z;

%predictors = [interval, ndata{:,4:end}];
predictors = ndata{:,4:end};
% take a sample sensor as input (column 1:48 specifying sensor location in
% 6x8 grid; column 49:52 specifying sensor 1:4 around the spindle)
pred1 = predictors(:,6);

% Plot figures
ts_interval = tsdata.Time(interval);
d = datetime(dateOfDay);
```

```
ts_interval.Year = d.Year;
ts_interval.Month = d.Month;
ts_interval.Day = d.Day;

%plotdata();
```

Split the dataset into training and testing data

Create three two-hour-long sampling intervals, with random starting points

```
numSamplingIntervals = 3;
% calculate the length of each sample such that 25% of the timeseries is
% used for testing
numHours = t.Hour(end)-t.Hour(1)+1;
samplingMinutes = 0.25*numHours*120;
sampleDuration = samplingMinutes/numSamplingIntervals;
startTimes = sort(randsample(numHours-3,numSamplingIntervals)*sampleDuration);
% assign random start points for sampling
startTimes = startTimes - 2*randsample(60,numSamplingIntervals)
```

```
startTimes = 3×1
   102
   358
   926
```

```
for i = 1:size(startTimes,1)
    if startTimes(i) < 0
        startTimes(i) = startTimes(i)*-1
    end
end
endTimes = startTimes(:) + sampleDuration
```

```
endTimes = 3×1
   242
   498
  1066
```

Generate a subset of testing samples

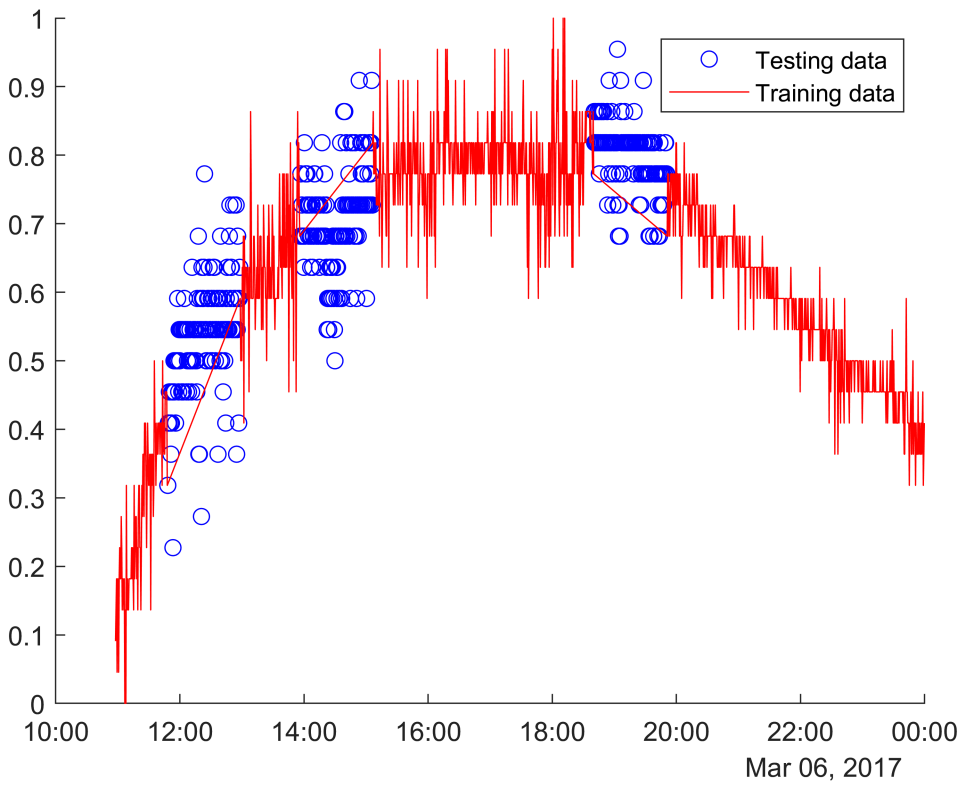
Add an index to predictors so the desired data points can be selected by their relative index in the input data.

```
predictors(:,1);

for i = 1:numSamplingIntervals
    % calculate sampling interval duration
    intervalPoints = startTimes(i):endTimes(i);
    ezs = [intervalPoints', Zval(intervalPoints')];
    newStartPoint = startTimes(i);
    % starting at the first point in the interval + the interval
    newEndPoint = newStartPoint + intervalPoints(end);
    eps = [predictors(startTimes(i):endTimes(i),:)];
    % set new start index
    startIndex = 1+(i-1)*sampleDuration
    extZSamples(startIndex:startIndex+sampleDuration,:) = ezs;
    extPredSamples(startIndex:startIndex+sampleDuration,:) = eps;
```

```
    %extPredSample() = eps;
    %Ztest((i-1)*240+1:i*240+i,1) = extSample;
end
```

```
startIndex = 1
startIndex = 141
startIndex = 281
```

```
predictors_test =  extPredSamples;
predictors_train = predictors;
indexesToRemove = extZSamples(:,1);
% remove indexes in indexesToRemove
predictors_train(indexesToRemove,:) =  [];
% similarly, remove indexes in Zvals_train
Zvals_test = extZSamples;
Zvals_train = [interval', Zval];
Zvals_train(indexesToRemove,:) = [];

figure(3); clf reset;
testIndeces = Zvals_test(:,1);
trainIndeces = Zvals_train(:,1);
scatter(t(testIndeces),Zvals_test(:,2),'b');
hold on;
plot(t(trainIndeces),Zvals_train(:,2),'r');
legend('Testing data','Training data');
```



Step 1: Feature selection for regression using Neighborhood Component Analysis

Create a Feature Selection for Regression Neighborhood Component Analysis (FSRNCA) model to weight each feature based on a predefined threshold \Lambda, equal to 0.5/numObservations

```
lambda = 0.5/size(interval,2);
mdl = fsrnca(predictors, Zval,'Verbose',1,'FitMethod','Exact','Solver','lbfgs');
```

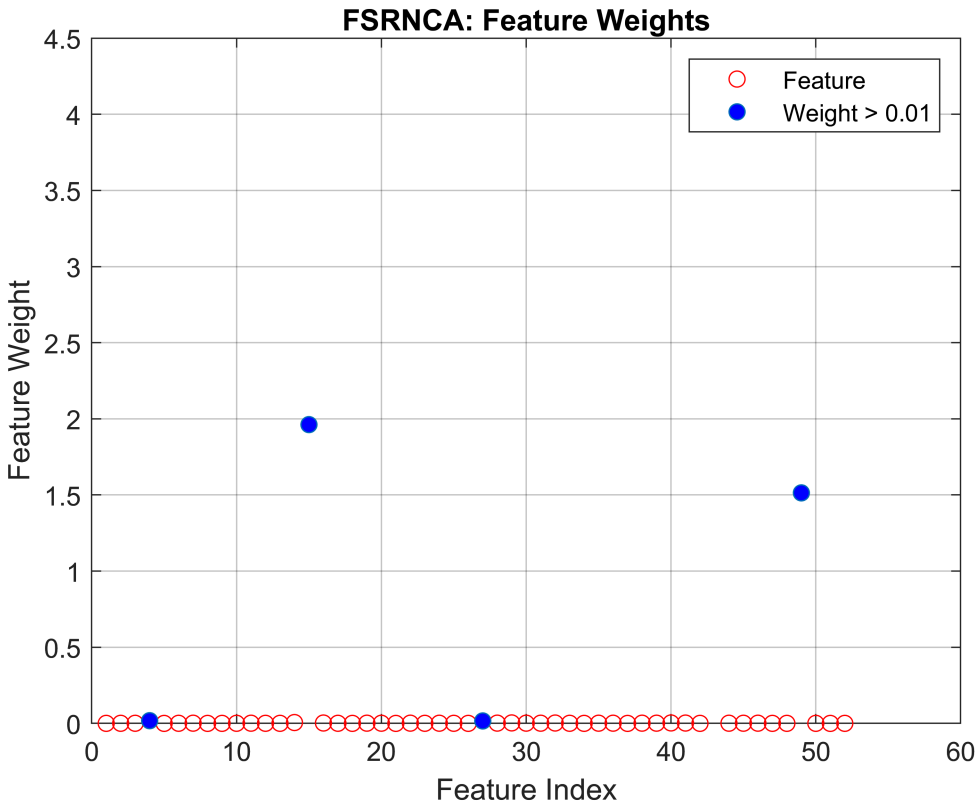
```
 o Solver = LBFGS, HessianHistorySize = 15, LineSearchMethod = weakwolfe

|=====================================================================================================|
|   ITER   |   FUN VALUE    |  NORM GRAD   |  NORM STEP   |  CURV  |    GAMMA     |    ALPHA     | ACCEPT |
|=====================================================================================================|
|        0 |  9.603917e-02  |   1.052e-03  |   0.000e+00  |        |   5.703e+02  |   0.000e+00  |   YES  |
|        1 |  9.102973e-02  |   2.667e-03  |   1.625e+00  |   OK   |   1.781e+02  |   5.000e-01  |   YES  |
|        2 |  8.774561e-02  |   2.103e-03  |   1.051e+00  |   OK   |   2.951e+02  |   1.000e+00  |   YES  |
|        3 |  8.414418e-02  |   5.378e-04  |   2.497e+00  |   OK   |   4.224e+02  |   5.000e-01  |   YES  |
|        4 |  8.383733e-02  |   4.789e-04  |   3.816e-01  |   OK   |   1.085e+03  |   1.000e+00  |   YES  |
|        5 |  8.321317e-02  |   7.151e-04  |   1.530e+00  |   OK   |   1.092e+03  |   1.000e+00  |   YES  |
|        6 |  8.290021e-02  |   2.114e-04  |   6.471e-01  |   OK   |   3.475e+02  |   1.000e+00  |   YES  |
|        7 |  8.278288e-02  |   3.115e-04  |   7.406e-01  |   OK   |   1.315e+03  |   1.000e+00  |   YES  |
|        8 |  8.271017e-02  |   3.167e-04  |   6.743e-01  |   OK   |   3.151e+02  |   1.000e+00  |   YES  |
|        9 |  8.269814e-02  |   3.439e-04  |   6.241e-01  |   OK   |   2.743e+02  |   1.000e+00  |   YES  |
|       10 |  8.267506e-02  |   9.129e-05  |   2.724e-01  |   OK   |   2.161e+02  |   1.000e+00  |   YES  |
|       11 |  8.266842e-02  |   4.655e-05  |   1.147e-01  |   OK   |   6.469e+02  |   1.000e+00  |   YES  |
|       12 |  8.265333e-02  |   7.826e-05  |   4.573e-01  |   OK   |   8.649e+02  |   1.000e+00  |   YES  |
|       13 |  8.264446e-02  |   4.231e-05  |   1.501e-01  |   OK   |   5.269e+02  |   1.000e+00  |   YES  |
|       14 |  8.263843e-02  |   1.445e-05  |   2.195e-01  |   OK   |   2.583e+03  |   1.000e+00  |   YES  |
|       15 |  8.263470e-02  |   1.947e-05  |   1.396e-01  |   OK   |   2.379e+03  |   1.000e+00  |   YES  |
|       16 |  8.263012e-02  |   5.545e-05  |   2.880e-01  |   OK   |   6.904e+02  |   1.000e+00  |   YES  |
|       17 |  8.262761e-02  |   5.318e-05  |   1.892e-01  |   OK   |   3.715e+03  |   1.000e+00  |   YES  |
|       18 |  8.262170e-02  |   1.440e-04  |   3.603e-01  |   OK   |   3.750e+02  |   1.000e+00  |   YES  |
|       19 |  8.261862e-02  |   2.722e-04  |   5.823e-01  |   OK   |   7.548e+02  |   1.000e+00  |   YES  |

|=====================================================================================================|
|   ITER   |   FUN VALUE    |  NORM GRAD   |  NORM STEP   |  CURV  |    GAMMA     |    ALPHA     | ACCEPT |
|=====================================================================================================|
|       20 |  8.260315e-02  |   3.912e-05  |   3.256e-01  |   OK   |   2.761e+02  |   1.000e+00  |   YES  |
|       21 |  8.259643e-02  |   8.932e-05  |   2.052e-01  |   OK   |   5.125e+02  |   1.000e+00  |   YES  |
|       22 |  8.259060e-02  |   1.935e-04  |   3.333e-01  |   OK   |   6.178e+02  |   1.000e+00  |   YES  |
|       23 |  8.258346e-02  |   1.499e-04  |   1.006e-01  |   OK   |   4.314e+02  |   1.000e+00  |   YES  |
|       24 |  8.257346e-02  |   5.661e-05  |   3.700e-01  |   OK   |   1.560e+03  |   1.000e+00  |   YES  |
|       25 |  8.257126e-02  |   6.093e-05  |   1.180e-01  |   OK   |   2.581e+02  |   1.000e+00  |   YES  |
|       26 |  8.256994e-02  |   1.497e-05  |   1.397e-01  |   OK   |   3.786e+02  |   1.000e+00  |   YES  |
|       27 |  8.256948e-02  |   1.287e-05  |   3.131e-02  |   OK   |   1.948e+03  |   1.000e+00  |   YES  |
|       28 |  8.256710e-02  |   1.934e-05  |   1.950e-01  |   OK   |   1.856e+03  |   1.000e+00  |   YES  |
|       29 |  8.256658e-02  |   4.113e-05  |   1.918e-01  |   OK   |   9.819e+02  |   1.000e+00  |   YES  |
|       30 |  8.256518e-02  |   1.013e-05  |   6.371e-02  |   OK   |   6.513e+02  |   1.000e+00  |   YES  |
|       31 |  8.256463e-02  |   2.289e-05  |   6.595e-02  |   OK   |   6.812e+02  |   1.000e+00  |   YES  |
|       32 |  8.256407e-02  |   2.015e-05  |   8.268e-02  |   OK   |   2.882e+03  |   1.000e+00  |   YES  |
|       33 |  8.256353e-02  |   3.350e-05  |   1.226e-01  |   OK   |   1.021e+03  |   1.000e+00  |   YES  |
|       34 |  8.256319e-02  |   5.645e-06  |   5.969e-02  |   OK   |   4.341e+02  |   1.000e+00  |   YES  |
|       35 |  8.256309e-02  |   8.063e-06  |   3.346e-02  |   OK   |   2.259e+03  |   1.000e+00  |   YES  |
|       36 |  8.256294e-02  |   6.870e-06  |   4.605e-02  |   OK   |   2.222e+03  |   1.000e+00  |   YES  |
|       37 |  8.256285e-02  |   3.831e-06  |   5.365e-02  |   OK   |   2.283e+03  |   1.000e+00  |   YES  |
|       38 |  8.256280e-02  |   2.927e-06  |   3.052e-02  |   OK   |   9.133e+02  |   1.000e+00  |   YES  |
|       39 |  8.256277e-02  |   2.252e-06  |   2.822e-02  |   OK   |   7.989e+03  |   1.000e+00  |   YES  |

|=====================================================================================================|
|   ITER   |   FUN VALUE    |  NORM GRAD   |  NORM STEP   |  CURV  |    GAMMA     |    ALPHA     | ACCEPT |
|=====================================================================================================|
|       40 |  8.256274e-02  |   3.940e-06  |   5.565e-02  |   OK   |   6.619e+03  |   1.000e+00  |   YES  |
```

```
|       41 |  8.256272e-02 |   6.417e-06 |   2.812e-02 |    OK   |   2.538e+03 |   5.000e-01 |   YES   |
|       42 |  8.256270e-02 |   5.058e-06 |   1.876e-02 |    OK   |   4.330e+02 |   1.000e+00 |   YES   |
|       43 |  8.256269e-02 |   1.947e-06 |   1.900e-02 |    OK   |   4.774e+02 |   1.000e+00 |   YES   |
|       44 |  8.256268e-02 |   7.175e-07 |   7.499e-03 |    OK   |   1.343e+03 |   1.000e+00 |   YES   |

         Infinity norm of the final gradient = 7.175e-07
             Two norm of the final step     = 7.499e-03, TolX   = 1.000e-06
Relative infinity norm of the final gradient = 7.175e-07, TolFun = 1.000e-06
EXIT: Local minimum found.
```

```
[fi,fj,fs]  = find(mdl.FeatureWeights > 0.01);
```

Plot the results

```
figure(2); clf reset;
plot(mdl.FeatureWeights,'ro');
hold on;
scatter(fi,mdl.FeatureWeights(fi),'MarkerFaceColor','b');
legend('Feature','Weight > 0.01');
xlabel('Feature Index');
ylabel('Feature Weight');
grid on;
title('FSRNCA: Feature Weights');
```



Approach 2: Use the Diagnostic Feature Designer App to generate condition indicators

TODO

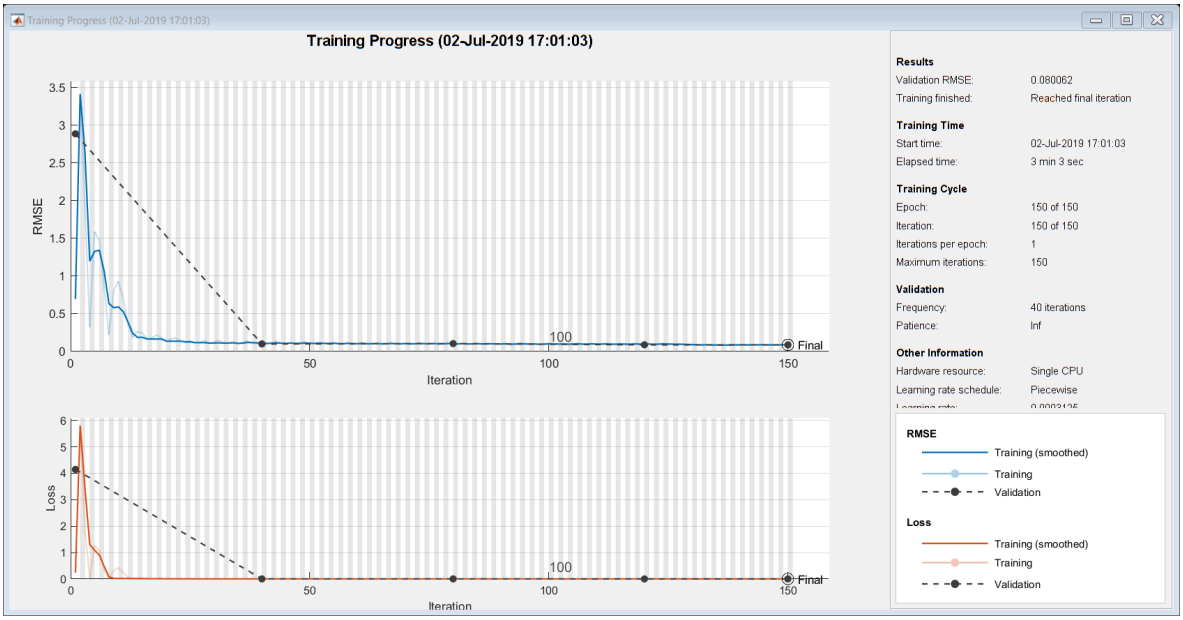
Approach 3: Use a simple LSTM layer to predict the output of the z-axis displacement taking in the file ensemble datastore as input

```
% Training Data
XTrain = predictors_train';
YTrain = Zvals_train(:,2:end)';
% Testing Data
XTest = predictors_test';
YTest = Zvals_test(:,2:end)';

numFeatures = 52;
numHiddenUnits = 180;
numResponses = 1;


layers = [...
    sequenceInputLayer(numFeatures),...
    lstmLayer(numHiddenUnits,'OutputMode','sequence'),...
    fullyConnectedLayer(100),...
    dropoutLayer(0.4),...
    fullyConnectedLayer(numResponses),...
    regressionLayer;
    ];
options = trainingOptions('adam', ...
    'MaxEpochs',150, ...
    'GradientThreshold',1, ...
    'InitialLearnRate',0.0025, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',125, ...
    'LearnRateDropFactor',0.125, ...
    'Verbose',0, ...
    "ValidationData",[{XTest},{YTest}],...
    "ValidationFrequency",40,...
    'Plots','training-progress');
```

Designate the training and testing data and pre-train the model

Vary the training options to improve the performance, or try training the model for longer.

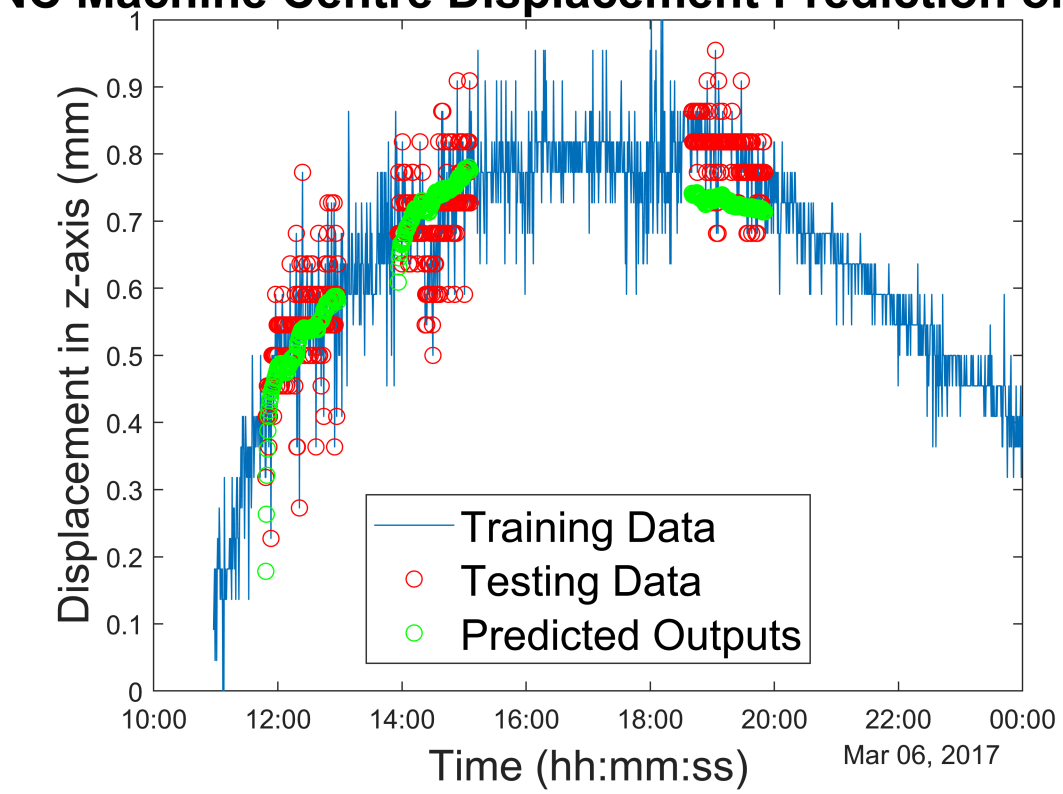
```
net = trainNetwork(XTrain, YTrain, layers, options);
```



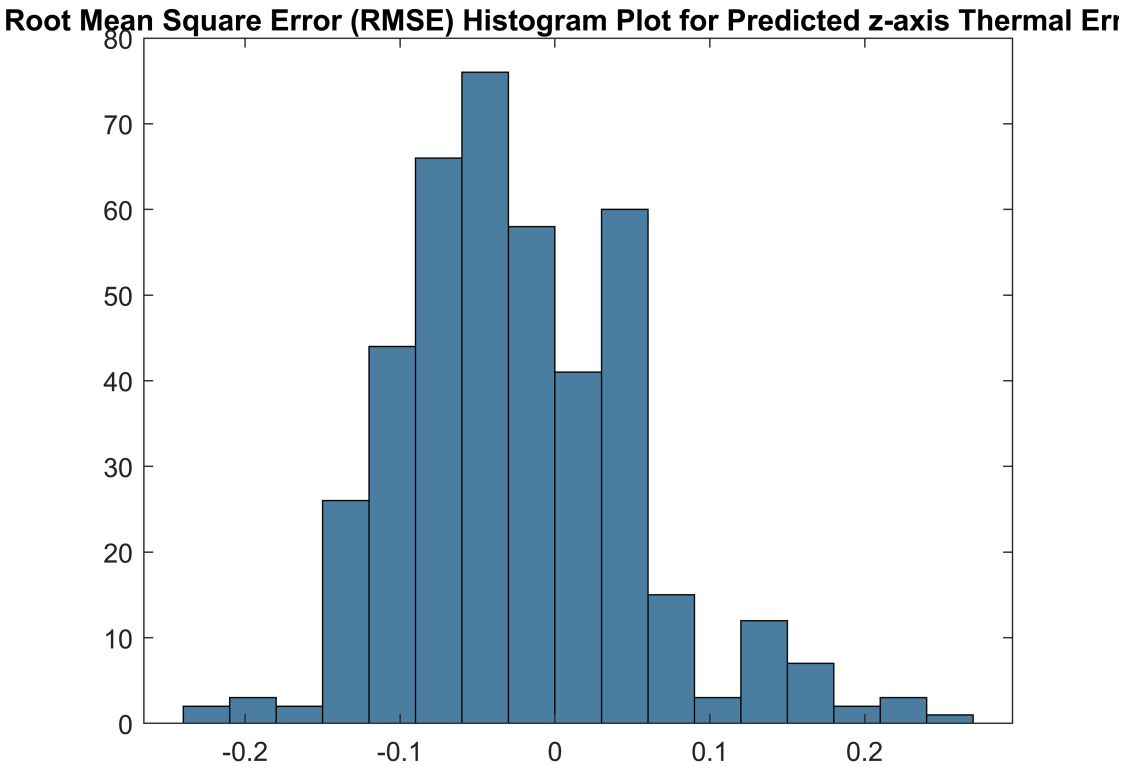

```
YPred = predict(net,XTest,'MiniBatchSize',1);
```

Plot the results

```
testData = [{t(testIndeces)},{YTest}];
predData = [{t(testIndeces)},{YPred}];

% Create a figure to show the prediction outputs
figure(4); clf reset;
plot(t,Zval(:,1));
hold on
scatter(testData{:,1},testData{:,2},'r');
scatter(predData{:,1},predData{:,2},'g');
hold off
legend('Training Data','Testing Data','Predicted Outputs','Location','Best','FontSize',16)
ylabel('Displacement in z-axis (mm)','FontSize',16);
xlabel('Time (hh:mm:ss)','FontSize',16);
title('LSTM z-axis CNC Machine Centre Displacement Prediction on 24-hour Sequence','FontSize',1
```




```
% calculate RMSE and display it in the histogram for all
% predictions
RMSE = sqrt(mean(YPred - YTest.^2));
figure(4); clf reset;
h1 = histogram(YPred - YTest);
title('Root Mean Square Error (RMSE) Histogram Plot for Predicted z-axis Thermal Error');
```




```
acc_t = [0.10 0.20];
errors = abs(YPred - YTest);
% accuracy of estimation error magnitude
numCorrect_1 = errors(errors < acc_t(1));
numCorrect_2 = errors(errors < acc_t(2));
acc_10 = size(numCorrect_1,2)/size(YPred,2)
```

```
acc_10 = 0.8385
```

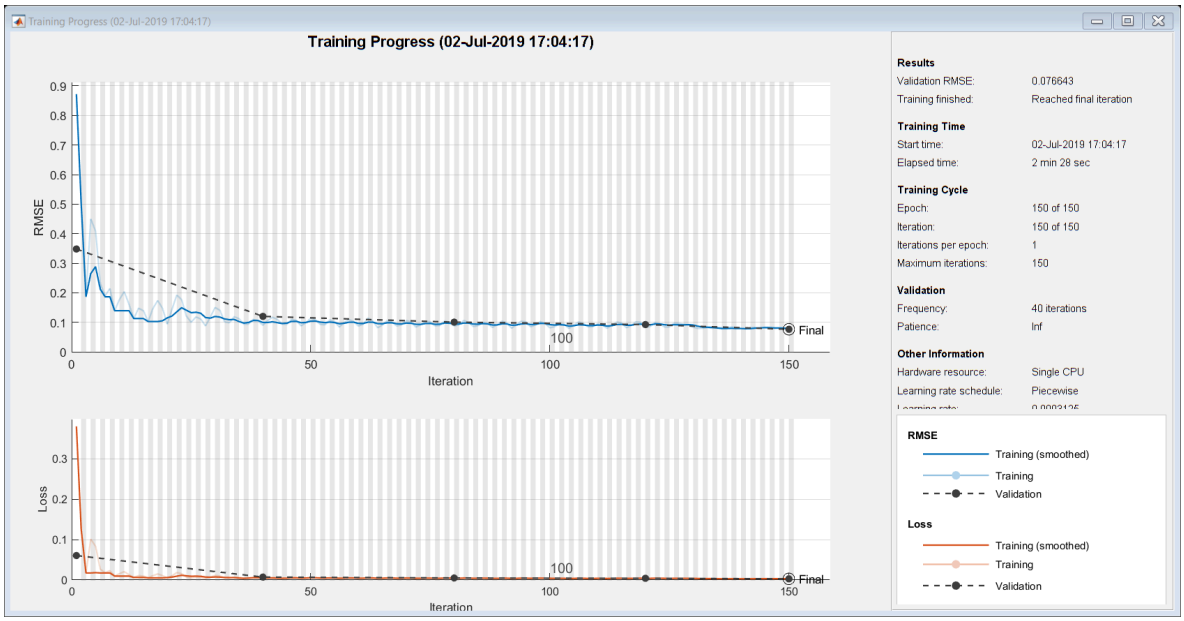
```
acc_20 = size(numCorrect_2,2)/size(YPred,2)
```

```
acc_20 = 0.9857
```

Approach 3b: Use only the features identified from FSRNCA implementation

```
% Training Data
fsrnca_features = fi;
XTrain_fsrnca = predictors_train(:,fsrnca_features)';
YTrain = Zvals_train(:,2:end)';
% Testing Data
XTest_fsrnca = predictors_test(:,fsrnca_features)';
YTest_fsrnca = Zvals_test(:,2:end)';

numFeatures_fsrnca = size(fi,1);
numHiddenUnits = 180;
numResponses = 1;
```

```
layers = [...
    sequenceInputLayer(numFeatures_fsrnca),...
    lstmLayer(numHiddenUnits,'OutputMode','sequence'),...
    fullyConnectedLayer(100),...
    dropoutLayer(0.4),...
    fullyConnectedLayer(numResponses),...
    regressionLayer;
    ];
options = trainingOptions('adam', ...
    'MaxEpochs',150, ...
    'GradientThreshold',1, ...
    'InitialLearnRate',0.0025, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',125, ...
    'LearnRateDropFactor',0.125, ...
    'Verbose',0, ...
    "ValidationData",[{XTest_fsrnca},{YTest}],...
    "ValidationFrequency",40,...
    'Plots','training-progress');

net2 = trainNetwork(XTrain_fsrnca, YTrain, layers, options);
```

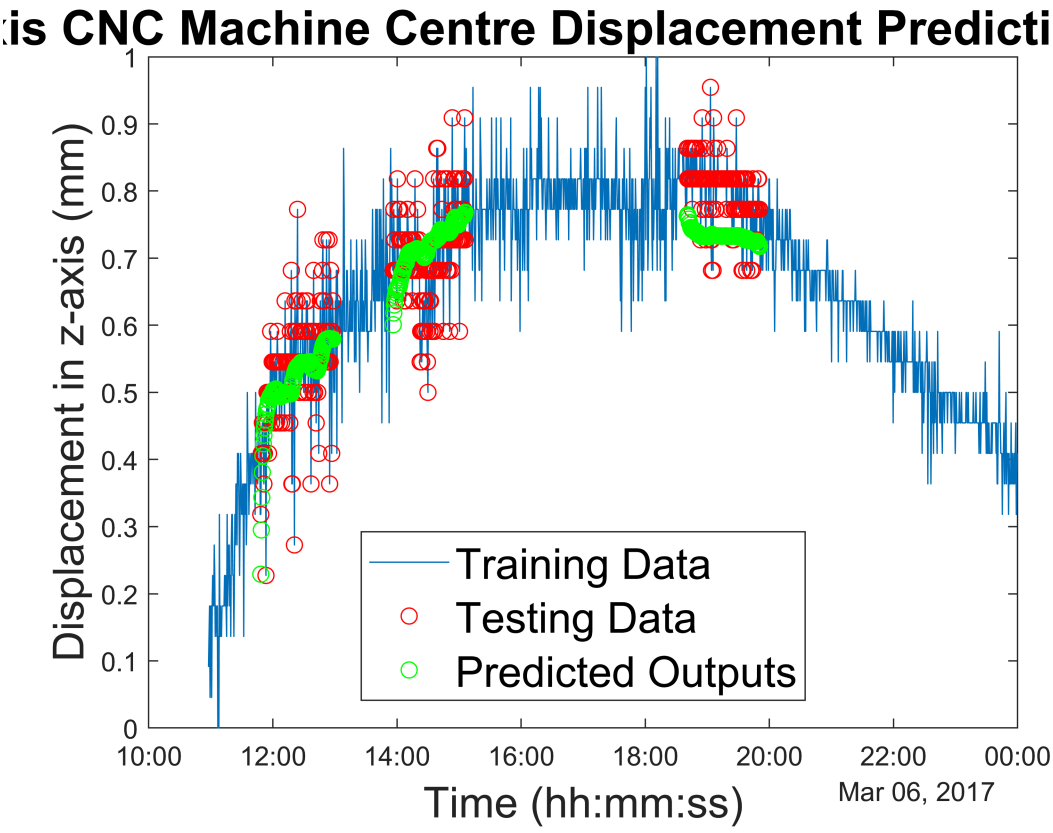



```
YPred_fsrnca = predict(net2,XTest_fsrnca,'MiniBatchSize',1);
```

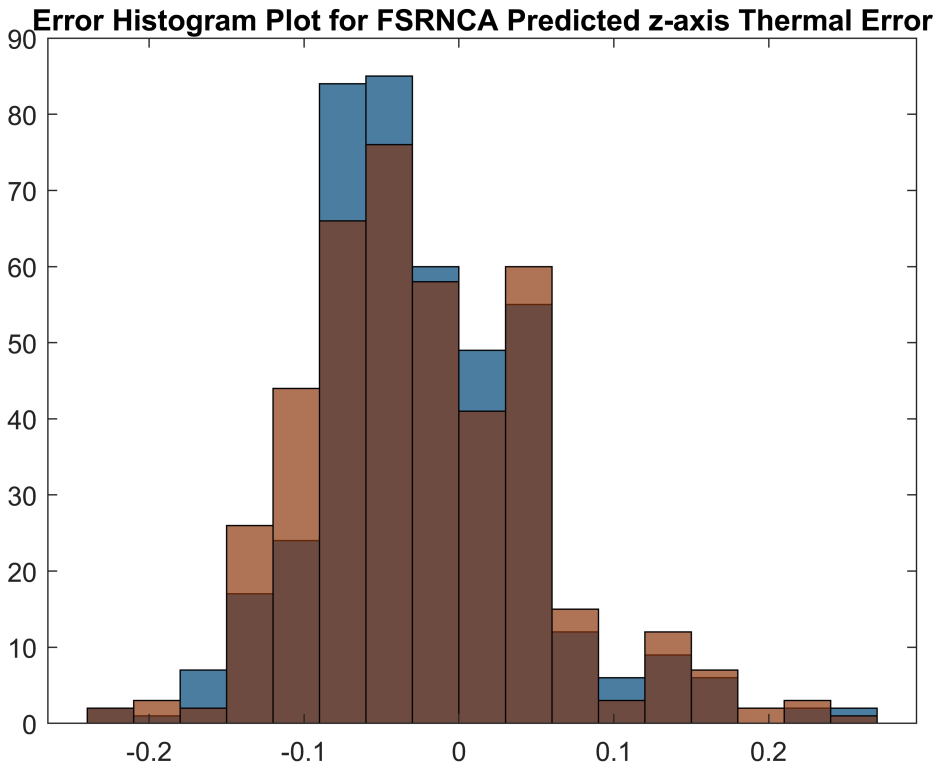
Plot the results

```
testData_fsrnca = [{t(testIndeces)},{YTest}];
predData_fsrnca = [{t(testIndeces)},{YPred_fsrnca}];

% Create a figure to show the prediction outputs
figure(5); clf reset;
plot(t,Zval(:,1));
```

```
hold on
scatter(testData_fsrnca{:,1},testData_fsrnca{:,2},'r');
scatter(predData_fsrnca{:,1},predData_fsrnca{:,2},'g');
hold off
legend('Training Data','Testing Data','Predicted Outputs','Location','Best','FontSize',16)
ylabel('Displacement in z-axis (mm)','FontSize',16);
xlabel('Time (hh:mm:ss)','FontSize',16);
title('FSRNCA-LSTM z-axis CNC Machine Centre Displacement Prediction on 24-hour Sequence','Font
```




```
% calculate RMSE and display it in the histogram for all
% predictions
RMSE = sqrt(mean(YPred_fsrnca - YTest_fsrnca.^2));
figure(6); clf reset;
h1 = histogram(YPred_fsrnca - YTest_fsrnca);
hold on;
h2 = histogram(YPred - YTest);
title('Error Histogram Plot for FSRNCA Predicted z-axis Thermal Error');
```




```
acc_t = [0.10 0.20];
errors_fsrnca = abs(YPred_fsrnca - YTest_fsrnca);
% accuracy of estimation error magnitude
numCorrect_1_fsrnca = errors_fsrnca(errors_fsrnca < acc_t(1));
numCorrect_2_fsrnca = errors_fsrnca(errors_fsrnca < acc_t(2));
acc_10_fsrnca = size(numCorrect_1_fsrnca,2)/size(YPred_fsrnca,2)
```

```
acc_10_fsrnca = 0.8480
```

```
acc_20_fsrnca = size(numCorrect_2_fsrnca,2)/size(YPred_fsrnca,2)
```

```
acc_20_fsrnca = 0.9857
```

As observed, we can increase the validation accuracy by 0.95% when using FSRNCA. Tuning this quantity further could help us further improve the prediction performance.