ansi_regression-final

April 14, 2018

1 ANSI Application analysis

```
In [1]: import numpy
        import pandas
        import matplotlib
        import matplotlib.pyplot as plotter
        from scipy.stats import pearsonr, probplot
        from sklearn.metrics import mean_squared_error, mean_absolute_error
        matplotlib.rcParams['agg.path.chunksize'] = 10000
```

```
In [2]: def view_boxplot(df):
            %matplotlib
            df.boxplot()
            plotter.show()
```

1.1 CPU data

```
In [3]: cpu_df = pandas.read_csv('data/ansi_final/ansi_final_cpu.csv', index_col='Time').drop(
```

```
In [4]: #cpu_df.columns
```

```
In [5]: cpu_df = cpu_df.clip(lower=0, upper=1000)
        #view_boxplot(cpu_df)
```

1.2 Network TX

```
In [6]: txnet_df = pandas.read_csv('data/ansi_final/ansi_final_network_tx.csv', index_col='Tim
```

```
In [7]: #txnet_df.columns
```

```
In [8]: txnet_df = txnet_df.clip(lower=0, upper=50000)
        #view_boxplot(txnet_df)
```

1.3 Network RX

```
In [9]: rxnet_df = pandas.read_csv('data/ansi_final/ansi_final_network_rx.csv', index_col='Tim
```

```
In [10]: #rxnet_df.columns
```

```
In [11]: rxnet_df = rxnet_df.clip(lower=0, upper=15000)
         #view_boxplot(rxnet_df)
```

1.4 Disk IO data

```
In [12]: disk_df = pandas.read_csv('data/ansi_final/ansi_final_disk_io.csv', index_col='Time')
```

```
In [13]: #disk_df.columns
```

```
In [14]: disk_df = disk_df.clip(lower=0, upper=4000)
         #view_boxplot(disk_df)
```

1.5 Context switching

```
In [15]: context_df = pandas.read_csv('data/ansi_final/ansi_final_context.csv', index_col='Tim
```

```
In [16]: #context_df.columns
```

```
In [17]: context_df = context_df.clip(lower=0, upper=5000)
         #view_boxplot(context_df)
```

1.6 Seperate into proper dataframes for each node

```
In [18]: dframes = [cpu_df, txnet_df, rxnet_df, context_df, disk_df]
         node = {}

         for i in range(1,13):
             frames = []

             for dframe in dframes:
                 columns = list(filter(lambda x: f'bb{i}l' in x, dframe.columns))
                 frames.append(dframe[columns])

             node[i] = pandas.concat(frames, join='inner', axis=1).fillna(0)[:68300]
```

```
In [19]: for i in range(1,13):
             print(node[i].shape)

         print(node[1].columns)
```

```
(68300, 29)
(68300, 29)
(68300, 29)
(68300, 29)
(68300, 29)
(68300, 29)
(68300, 29)
(68300, 29)
(68300, 29)
(68300, 29)
(68300, 29)
(68300, 29)
```

```
Index(['cpu_value host bb1localdomain type_instance idle',
       'cpu_value host bb1localdomain type_instance interrupt',
       'cpu_value host bb1localdomain type_instance nice',
       'cpu_value host bb1localdomain type_instance softirq',
       'cpu_value host bb1localdomain type_instance steal',
       'cpu_value host bb1localdomain type_instance system',
       'cpu_value host bb1localdomain type_instance user',
       'cpu_value host bb1localdomain type_instance wait',
       'interface_tx host bb1localdomain instance lo type if_dropped',
       'interface_tx host bb1localdomain instance lo type if_errors',
       'interface_tx host bb1localdomain instance lo type if_octets',
       'interface_tx host bb1localdomain instance lo type if_packets',
       'interface_tx host bb1localdomain instance wlan0 type if_dropped',
       'interface_tx host bb1localdomain instance wlan0 type if_errors',
       'interface_tx host bb1localdomain instance wlan0 type if_octets',
       'interface_tx host bb1localdomain instance wlan0 type if_packets',
       'interface_rx host bb1localdomain instance lo type if_dropped',
       'interface_rx host bb1localdomain instance lo type if_errors',
       'interface_rx host bb1localdomain instance lo type if_octets',
       'interface_rx host bb1localdomain instance lo type if_packets',
       'interface_rx host bb1localdomain instance wlan0 type if_dropped',
       'interface_rx host bb1localdomain instance wlan0 type if_errors',
       'interface_rx host bb1localdomain instance wlan0 type if_octets',
       'interface_rx host bb1localdomain instance wlan0 type if_packets',
       'contextswitch_value host bb1localdomain type contextswitch',
       'disk_io_time host bb1localdomain instance mmcblk1 type disk_io_time',
       'disk_io_time host bb1localdomain instance mmcblk1boot0 type disk_io_time',
       'disk_io_time host bb1localdomain instance mmcblk1boot1 type disk_io_time',
       'disk_io_time host bb1localdomain instance mmcblk1p1 type disk_io_time'],
      dtype='object')
```

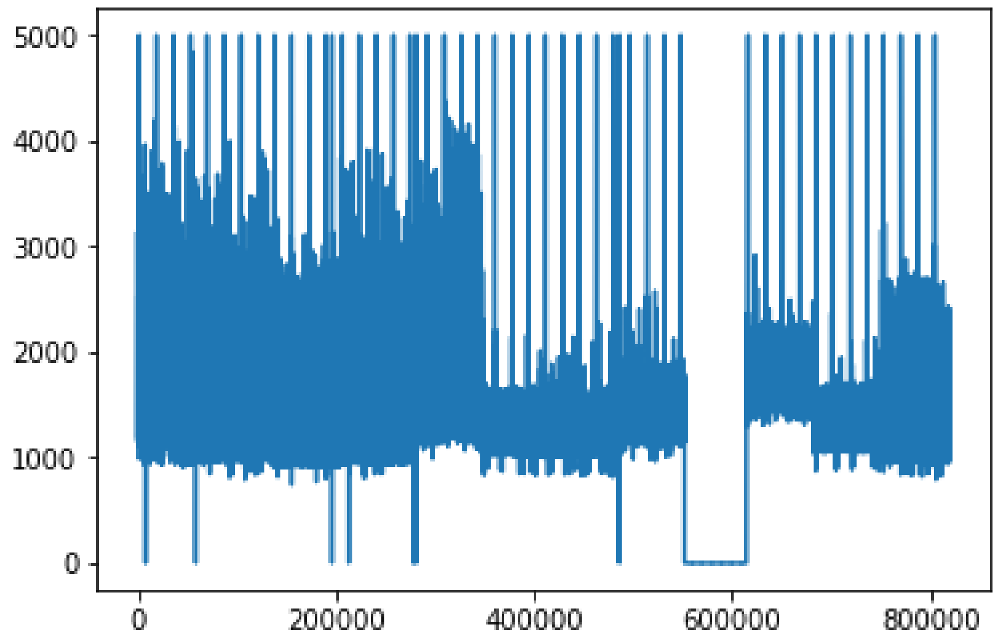
1.7 Get data

```
In [20]: data_matrices = []

         for i in range(1,13):
             data_matrices.append(node[i].as_matrix())

         data = numpy.vstack(data_matrices)
```

```
In [21]: data.shape
```

```
Out[21]: (819600, 29)
```

```
In [22]: tdata = data[:,24]
         plotter.plot(tdata.T)
         plotter.show()
         print(data.shape)
```




```
(819600, 29)
```

```
In [23]: #data = data[:,24]
```

1.8 Prepare scaler

```
In [24]: from sklearn.preprocessing import MinMaxScaler
         from sklearn.preprocessing import StandardScaler
         from sklearn.preprocessing import RobustScaler
         scaler = MinMaxScaler()

In [25]: scaler.fit(data)
         del data
```

1.9 Correrlation measurement

```
In [26]: for i in range(len(data_matrices)):

             transformed = scaler.transform(data_matrices[i])
             data_matrices[i] = transformed

         X = numpy.stack(data_matrices[:4], axis=1)
         test = numpy.stack(data_matrices[4:], axis=1)
         print(X.shape)
         print(test.shape)
```

```
(68300, 4, 29)
(68300, 8, 29)
```

```
In [27]: print(X.shape)
         LEN = X.shape[0]
         SPLIT = int(0.8*LEN)

         train_X = X[:SPLIT,:,:]
         val_X = X[SPLIT:,:,:]

         print(train_X.shape)
         print(val_X.shape)
```

```
(68300, 4, 29)
(54640, 4, 29)
(13660, 4, 29)
```

```
In [28]: test_X = numpy.transpose(test, (1,0,2))
         train_X = numpy.transpose(train_X, (1,0,2))
         val_X = numpy.transpose(val_X, (1,0,2))
         print(test_X.shape)
         print(train_X.shape)
         print(val_X.shape)
```

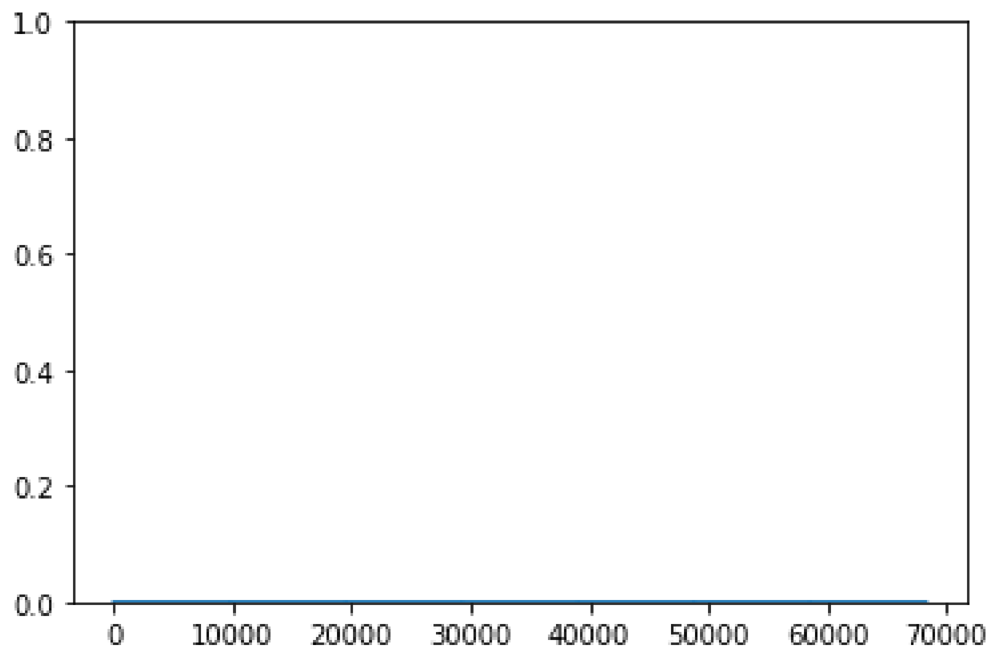
```
(8, 68300, 29)
(4, 54640, 29)
(4, 13660, 29)
```

```
In [29]: def flat_generator(X, tsteps = 5, ravel=1):
             i = 0

             while True:
                 batch_X = X[:,i:i+tsteps,:]
                 batch_y = X[:,i+tsteps,:]

                 if ravel:
```

```
                batch_X = batch_X.reshape((batch_X.shape[0], -1))
            #print(batch_X.shape)
            #print(batch_y.shape)

            yield batch_X, batch_y

            i += 1
            if i > (X.shape[1] - tsteps - 1):
                i = 0
                continue
```

In [30]:
```
series = test_X[7,:,21]
print(series.shape)
plotter.ylim(0,1)
plotter.plot(series)
plotter.show()
print(numpy.random.randint(29))
```

```
(68300,)
```




```
3
```

In [31]:
```
avg_load = test_X[3,:,:]
disk_io_start_late = test_X[0,:,:]
```

```
app_change_early = test_X[1,:,:]
#idle_early = test_X[4,:,:]
normal_test = test_X[7,:,:]
```

1.10 Training functions

```
In [32]: from keras.models import Model
         from keras.layers import Dense, Input, Dropout, GRU
         from keras.callbacks import EarlyStopping
```

```
Using TensorFlow backend.
```

```
In [33]: def train(model, tgen, vgen, name="none"):
             estopper = EarlyStopping(patience=15, min_delta=0.0001)
             history = model.fit_generator(tgen, steps_per_epoch=1000, epochs=10000, callbacks=
             plotter.plot(history.history['loss'],label='train')
             plotter.plot(history.history['val_loss'],label='validation')
             plotter.legend()
             plotter.xlim(0,150)
             plotter.xlabel("Epochs")
             plotter.ylabel("Error")
             plotter.savefig(f"{name}_train.png", dpi=500)
             plotter.show()
             print(f"Training loss for final epoch is {history.history['loss'][-1]}")
             print(f"Validation loss for final epoch is {history.history['val_loss'][-1]}")
```

```
In [34]: def plot_running_stats(error, name="none", window_size=5, bounds=None, qq=0):
             error = numpy.array(error)
             numpy.save(f"results/{name}_error.npy", error)
             window = numpy.ones(window_size)/window_size
             running_mean = numpy.convolve(error, window, mode="same")
             running_sigma = pandas.Series(error).rolling(window=window_size, center=True).std
             difference = 3.0 * running_sigma

             upper = running_mean + difference
             lower = running_mean - difference

             if bounds == None:
                 global_mean = numpy.mean(error) * numpy.ones(error.shape[0])
                 global_sigma = numpy.std(error) * numpy.ones(error.shape[0])
                 bound = (5.0 * global_sigma) + global_mean

             else:

                 global_mean = bounds[0]
                 bound = bounds[1]

             anomaly = ((error > bound) * error)
```

```
anomaly = numpy.array([float('nan') if x == 0.0 else x for x in anomaly])

if qq:

    #a, b, l, s = beta.fit(error)
    probplot(error,dist="norm", plot=plotter)
    plotter.legend()
    plotter.savefig(f"{name}_qq.png",dpi=500)
    plotter.show()

    plotter.hist(error, bins=100)
    plotter.legend()
    plotter.savefig(f"{name}_hist.png")
    plotter.show()

arr = pandas.Series(error)
means = arr.rolling(window=720).mean()
std = arr.rolling(window=720).std()
outlier = (arr > (means + 5.0 * std)) * 1.0

mark = numpy.ones(arr.shape[0]) * numpy.nan

window = 100

for i in range(window,outlier.shape[0]):
    num = window
    outliers = numpy.sum(outlier[i-window:i])
    per = outliers/num
    if per > 0.04:
        mark[i] = 1.0
    else:
        mark[i] = 0.0

plotter.plot(error, 'g-', label="Error", alpha=0.4, linewidth=0.5)
plotter.ylim(0,0.2)
plotter.xlabel("time")
plotter.ylabel("Error")
plotter.legend()
plotter.savefig(f"{name}_error_plain.png", dpi=500)
plotter.show()

fig = plotter.figure()
plotter.plot(error, 'g-', alpha=0.4, label="Error", linewidth=0.5)
plotter.plot(means, 'r-.', alpha=0.9, label="Mean", linewidth=0.5)
#plotter.plot(upper,'b-', alpha=0.2, label="Upper Bound", linewidth=0.5)
plotter.plot(means + 5.0 * std,'b--', alpha=0.9, label="Bound", linewidth=0.5)
plotter.plot(0.1 * mark,'r-', alpha=0.5, label="Anomaly")
plotter.legend()
```

```
plotter.ylim(0,0.2)
plotter.xlabel("time")
plotter.ylabel("Error")
plotter.draw()
fig.savefig(f"{name}_truetestloss.png", dpi=500)
plotter.show()



arr = pandas.Series(error)
means = arr.ewm(halflife=720).mean()
std = arr.ewm(halflife=720).std()
outlier = (arr > (means + 5.0 * std)) * 1.0

mark = numpy.ones(arr.shape[0]) * numpy.nan

window = 100

for i in range(window, outlier.shape[0]):
    num = window
    outliers = numpy.sum(outlier[i-window: i])
    per = outliers/num
    if per > 0.04:
        mark[i] = 1.0
    else:
        mark[i] = 0.0

fig = plotter.figure()
plotter.plot(error, 'g-', alpha=0.4, label="Error", linewidth=0.5)
plotter.plot(means, 'r-.', alpha=0.9, label="Mean", linewidth=0.5)
#plotter.plot(upper,'b-', alpha=0.2, label="Upper Bound", linewidth=0.5)
plotter.plot(means + 5.0 * std,'b--', alpha=0.9, label="Bound", linewidth=0.5)
plotter.plot(0.1 * mark,'r-', alpha=0.5, label="Anomaly")
plotter.legend()
plotter.ylim(0,0.2)
plotter.xlabel("time")
plotter.ylabel("Error")
plotter.draw()
fig.savefig(f"{name}_truetestloss_exp.png", dpi=500)
plotter.show()


fig.clf()
plotter.clf()
plotter.close()
error = numpy.array(error)
print(f"The mean error for {name} is {numpy.mean(error)} for length {error.shape[
```

```
        return (global_mean, bound)
```

```
In [35]: def data_test(model, dataset=test_X[0], ravel=1, write=0, name="none", window=5, bound
            test_gen = flat_generator(numpy.array([dataset]), window,0)
            error = []
            targets = []
            preds = []
            for i in range(dataset.shape[0]-(window+1)):
                _input,target = next(test_gen)
                targets.append(target.squeeze())
                if ravel:
                    _input = _input.ravel()[:,numpy.newaxis].T

                pred = model.predict(_input)
                #print(target.shape)
                #print(pred.shape)
                preds.append(pred.squeeze())
                error.append(mean_absolute_error(y_pred=pred, y_true=target))

            targets = numpy.vstack(targets)
            preds = numpy.vstack(preds)
            return plot_running_stats(error, name=name, window_size=window, bounds=bounds, qq=

            #print(error)
```

```
In [36]: def gen_test(model, dataset=test_X[0], ravel=1, write=0, name="none"):
            test_gen = flat_generator(numpy.array([dataset]), TIMESTEPS,0)
            error = []
            targets = []
            preds = []
            for i in range(2000):
                _input,target = next(test_gen)

                if i != 0:
                    #print(_input.shape)
                    _input = _input.squeeze()[1:,:]
                    #print(_input.shape)
                    _input = numpy.append(pred,_input, axis=0)[numpy.newaxis,:,:]
                    #print(_input.shape)

                targets.append(target.squeeze())
                if ravel:
                    _input = _input.ravel()[:,numpy.newaxis].T

                pred = model.predict(_input)
                #print(target.shape)
                #print(pred.shape)
                preds.append(pred.squeeze())
```

```
            error.append(mean_absolute_error(y_pred=pred, y_true=target))

        targets = numpy.vstack(targets)
        preds = numpy.vstack(preds)

        plotter.plot(error, 'g-', alpha=0.5)
        plotter.ylim(0,0.2)
        plotter.xlabel("time")
        plotter.ylabel("Error")
        plotter.savefig(f"{name}_testloss.png")
        plotter.show()
        error = numpy.array(error)
        print(numpy.mean(error))
        plotter.boxplot(error)
        plotter.ylim(0,0.2)
        plotter.xlabel("time")
        plotter.ylabel("Error")
        plotter.savefig(f"{name}_boxplot.png")
        plotter.show()
        if write:
            numpy.savetxt('loss.txt', numpy.array(error))
        true_test(model,dataset,ravel=ravel,name=name)
        #print(error)
```

```
In [37]: def test(model, ravel=1, name="none", window=20):
        print(f"---------- Beginning tests for {name} ----------")
        print(f"Testing on Disk IO begin data.")
        bounds = data_test(model, dataset=disk_io_start_late , ravel=ravel, name=(name+"_d
        print(f"Testing on Avg. load data.")
        data_test(model, dataset=avg_load, ravel=ravel, name=(name+"_avg_load_"), window=w
        print(f"Testing on app change early data.")
        data_test(model, dataset=app_change_early, ravel=ravel, name=(name+"_app_change_ea
        print(f"Testing on Normal data.")
        data_test(model, dataset=normal_test, ravel=ravel, name=(name+"_normal_"), window=
        #print(f"Testing on Idle early data.")
        #data_test(model, dataset=idle_early, ravel=ravel, name=(name+"_idle_early_"), wi
        print("="*20)
        print("\r\n\r\n")
```

1.11 Train Models

```
In [38]: X = train_X
```

1.11.1 Linear Regression

2 steps

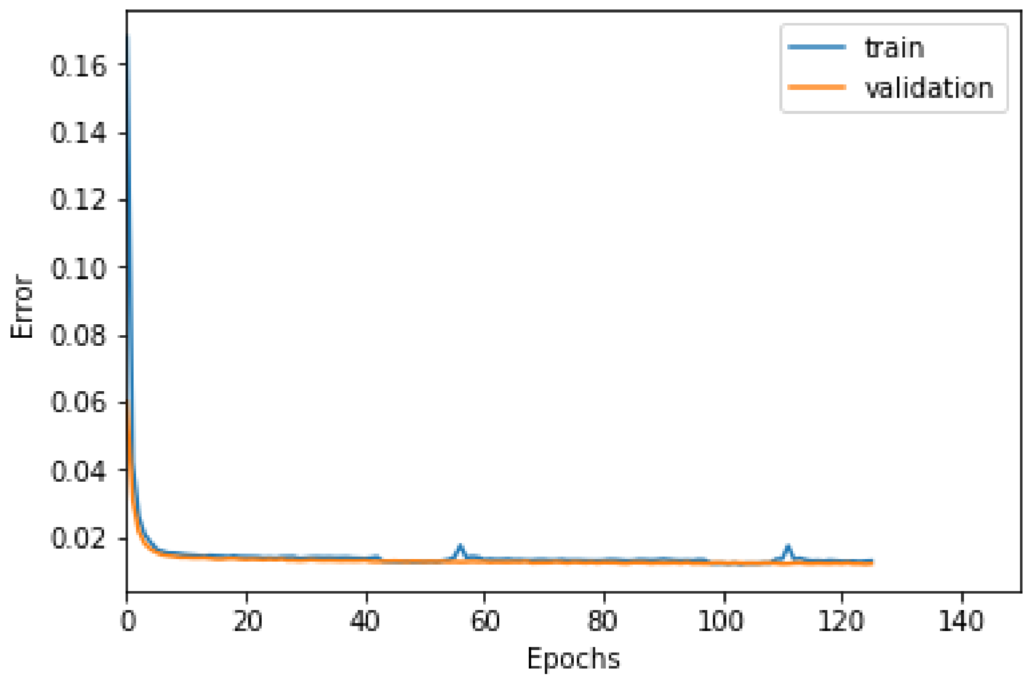
```
In [39]: TIMESTEPS = 2
         DIM = 29
```

```
tgen = flat_generator(X, TIMESTEPS)
vgen = flat_generator(val_X, TIMESTEPS)
name = "lin2"
```

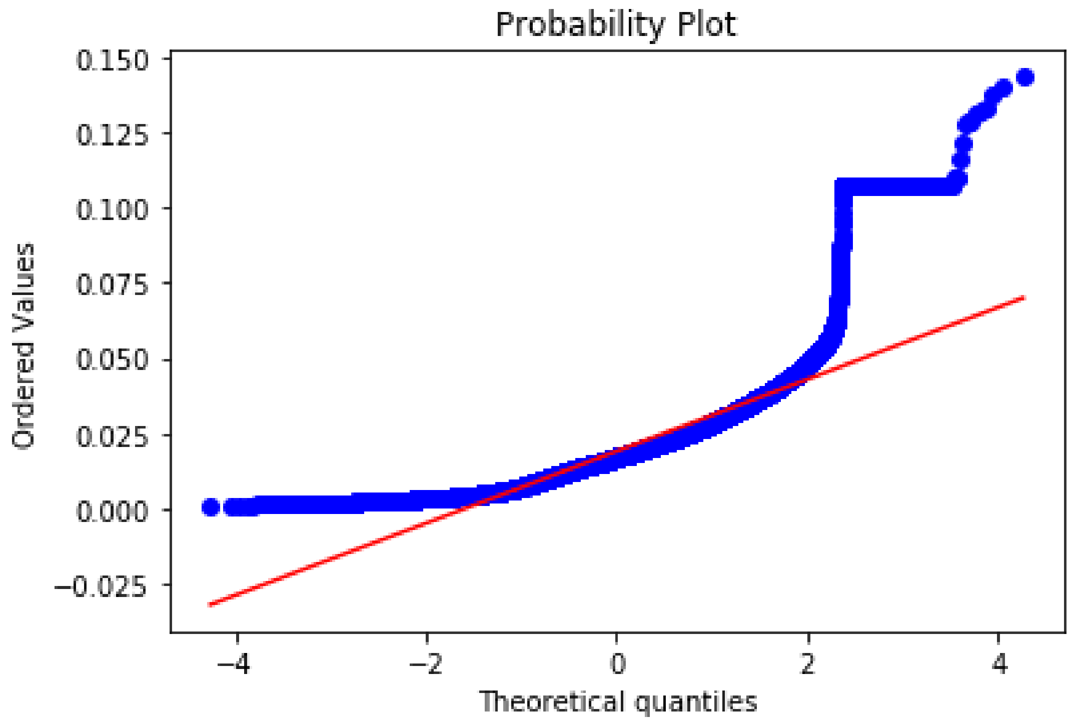
```
In [40]: input_layer = Input(shape=(TIMESTEPS*DIM,))
         output = Dense(DIM, activation='sigmoid')(input_layer)
```

```
In [41]: model = Model(input_layer, output)
         model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])
```

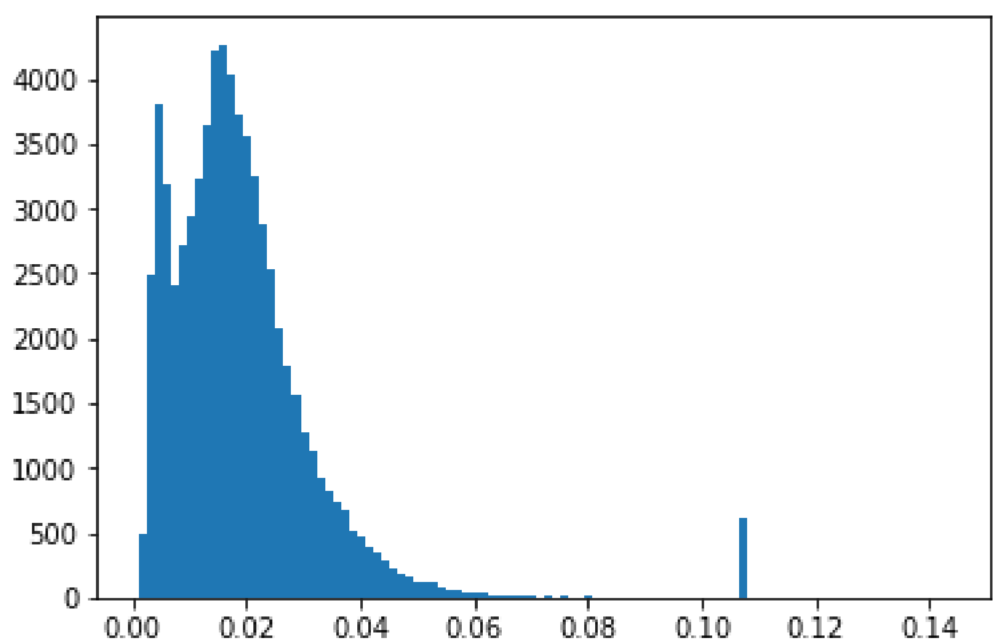
```
In [42]: train(model, tgen, vgen, name=name)
         test(model, name=name, window=TIMESTEPS)
```



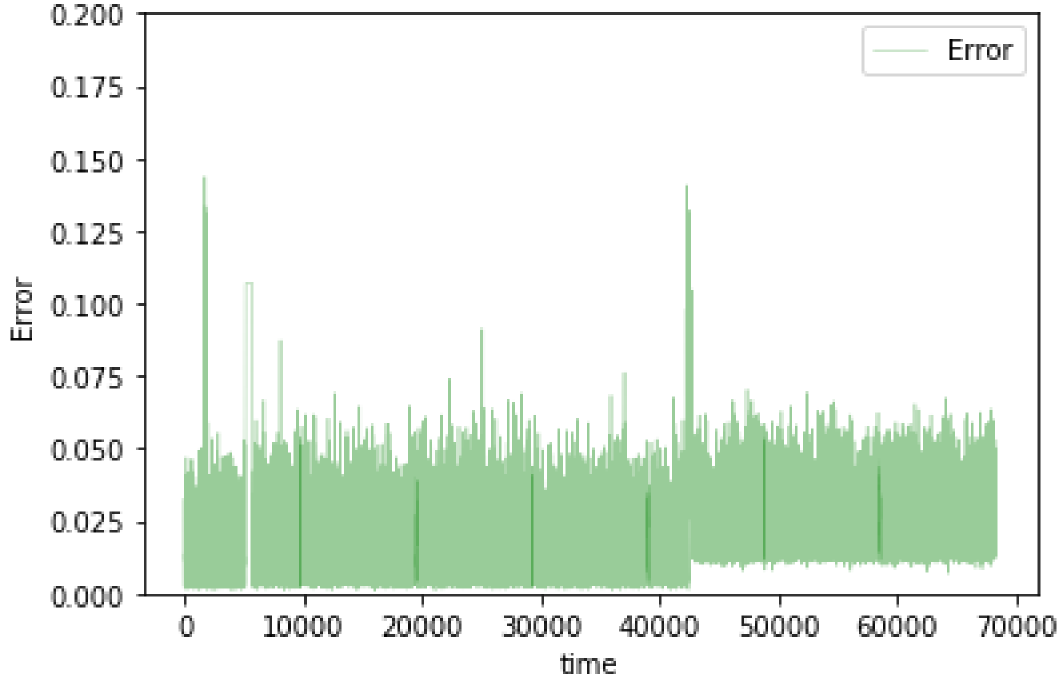

```
Training loss for final epoch is 0.012928565362934023
Validation loss for final epoch is 0.012101519779535011
---------- Beginning tests for lin2 ----------
Testing on Disk IO begin data.
```

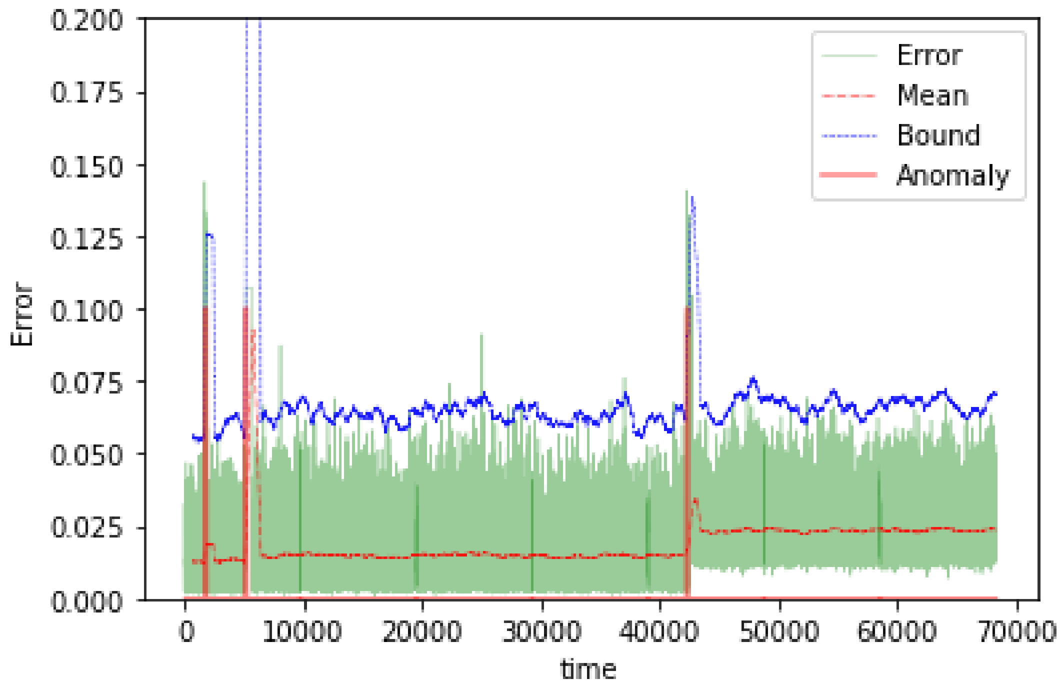


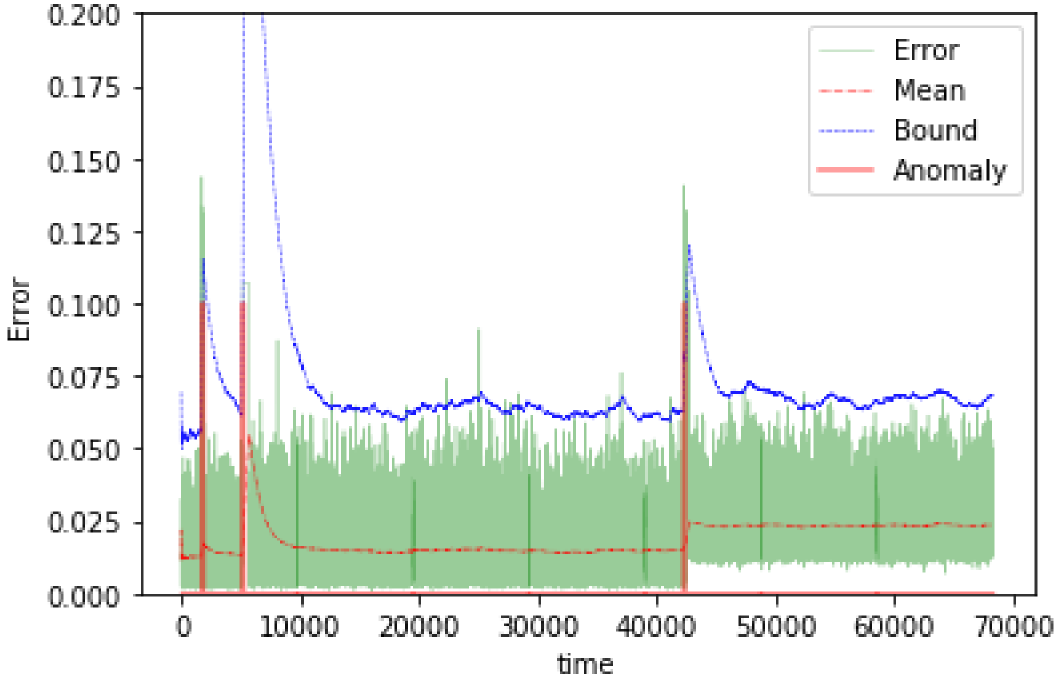
Probability Plot
Ordered Values
Theoretical quantiles
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−4
−2
0
2
4



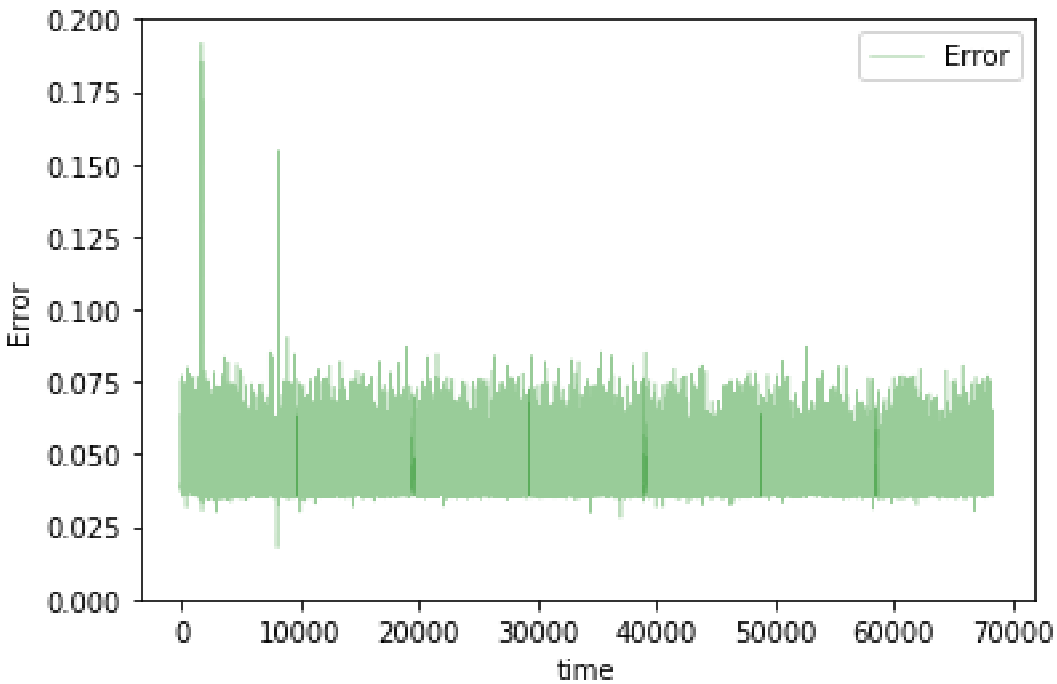
4000
3500
3000
2500
2000
1500
1000
500
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

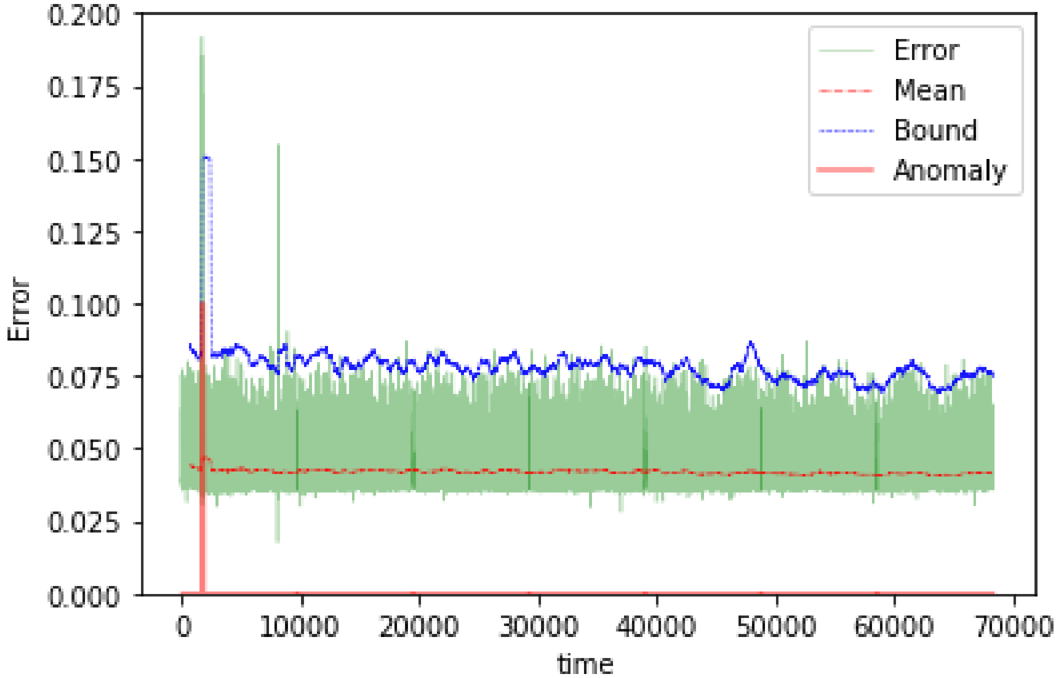

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

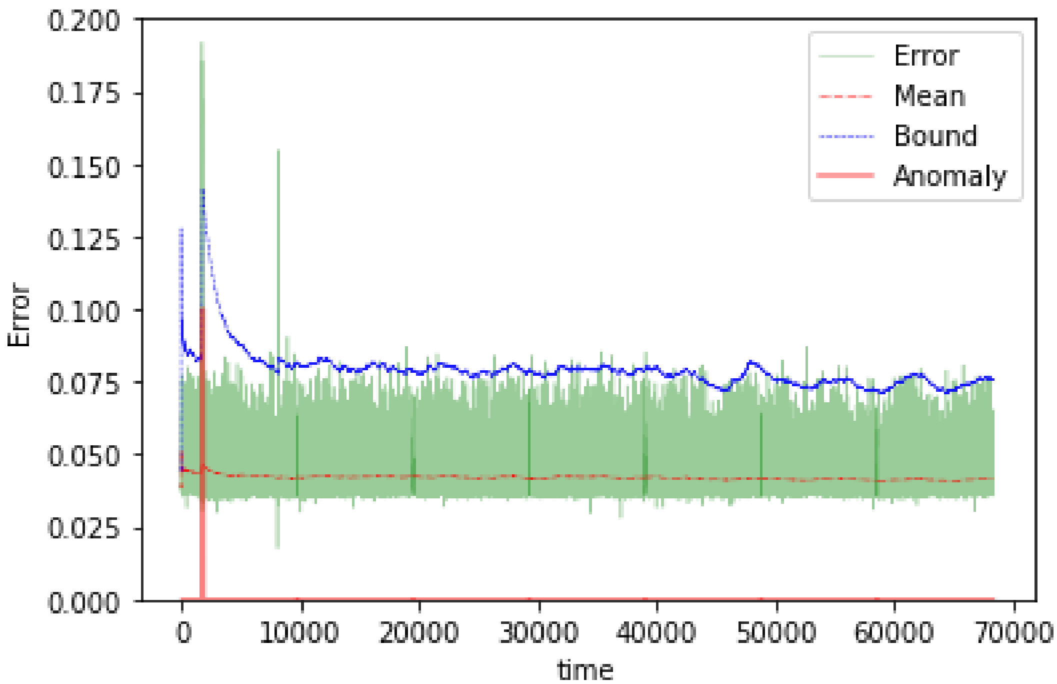



```
The mean error for lin2_disk_IO_start_ is 0.018950646547926227 for length 68297
Testing on Avg. load data.
```



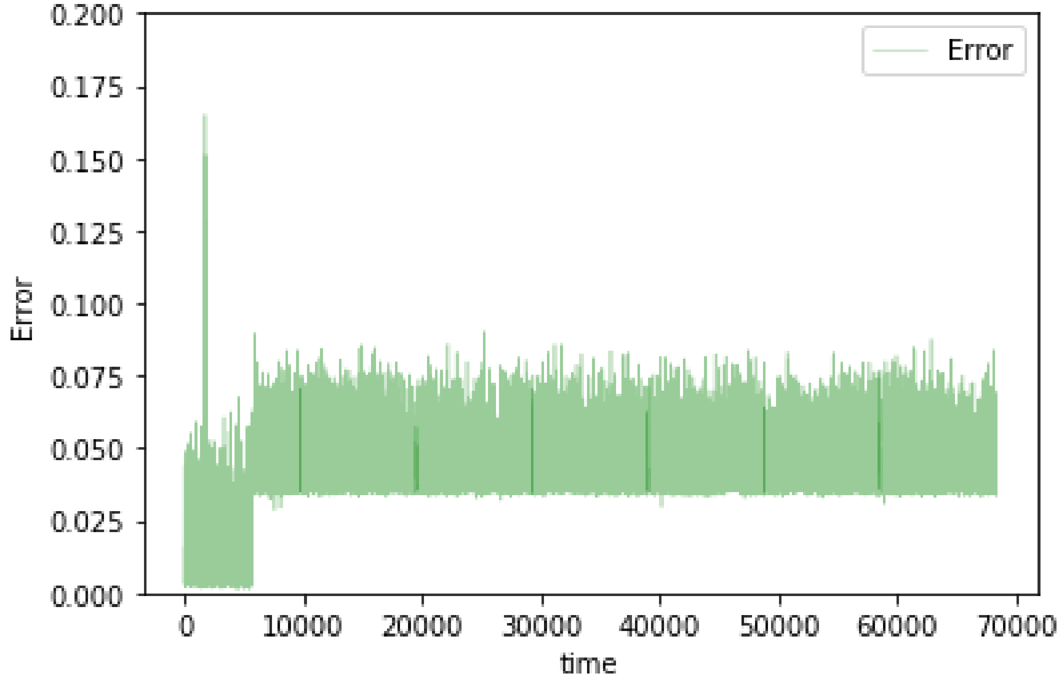


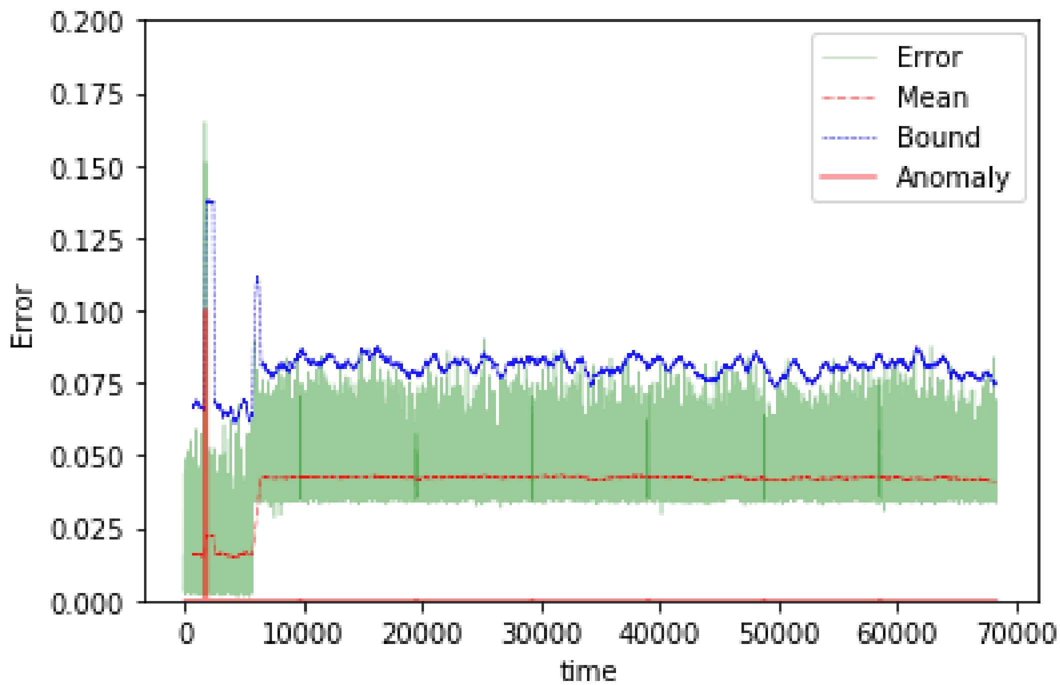
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

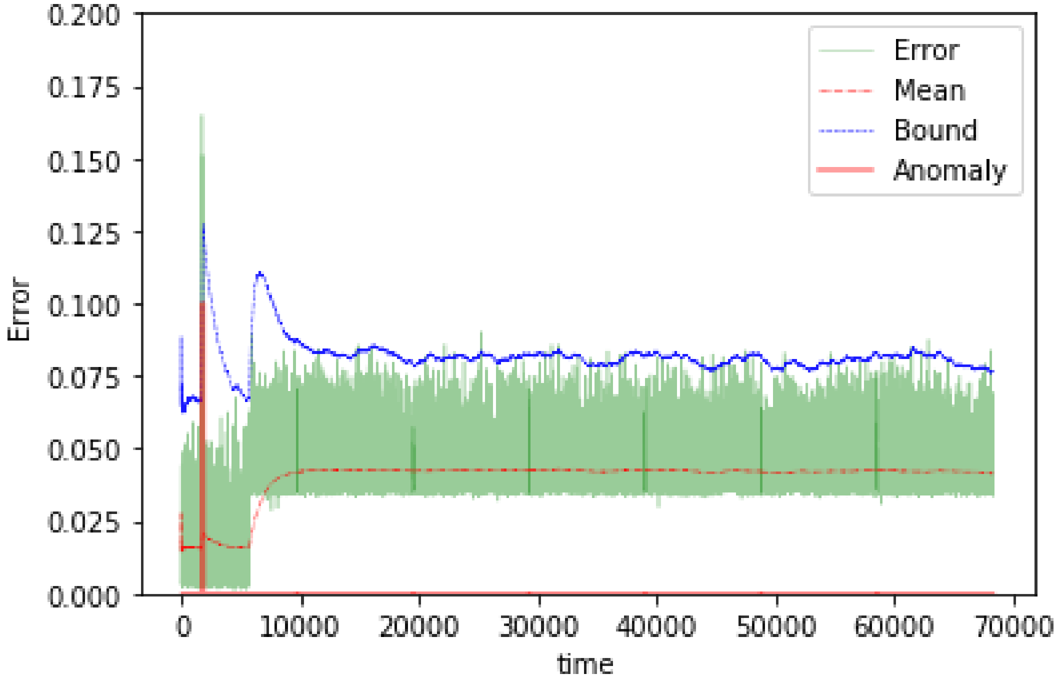


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

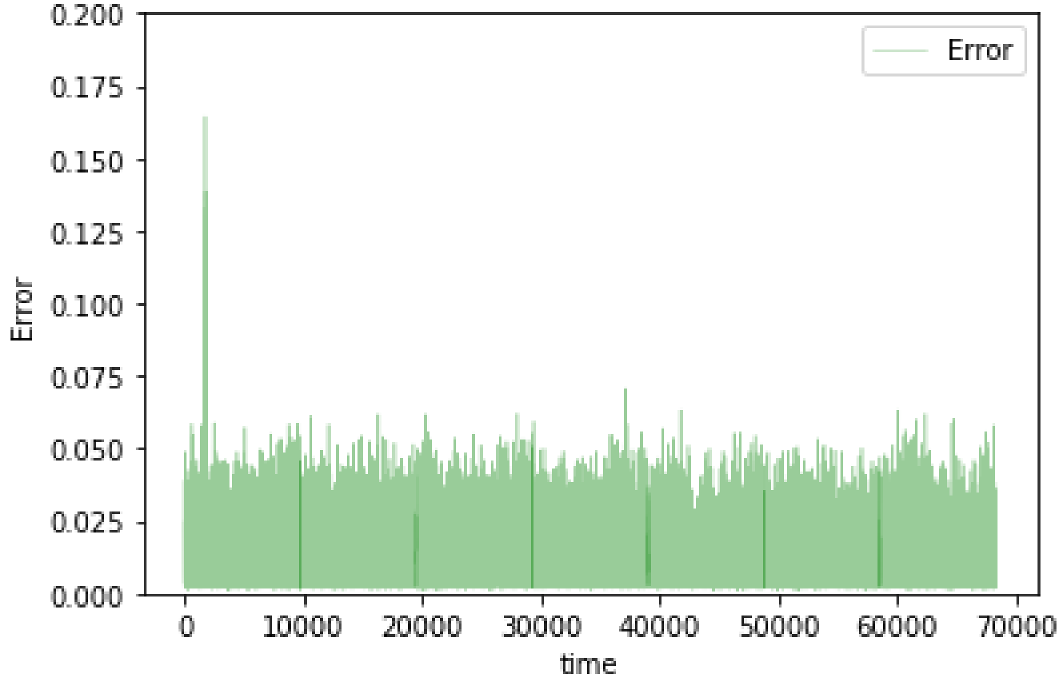
```
The mean error for lin2_avg_load_ is 0.041943856984552386 for length 68297
Testing on app change early data.
```

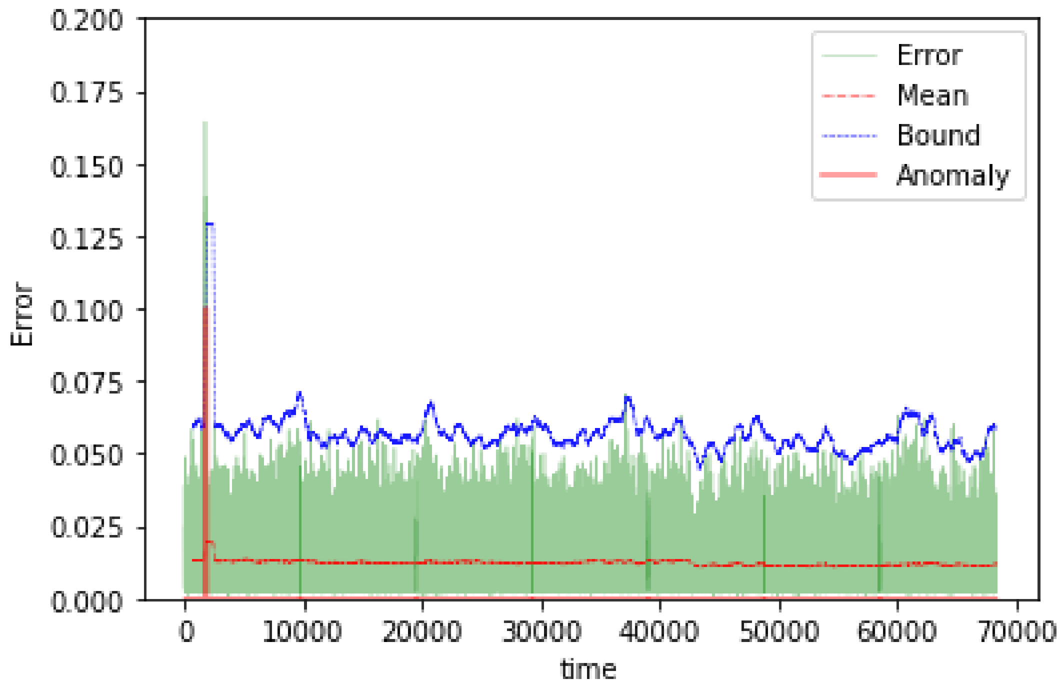


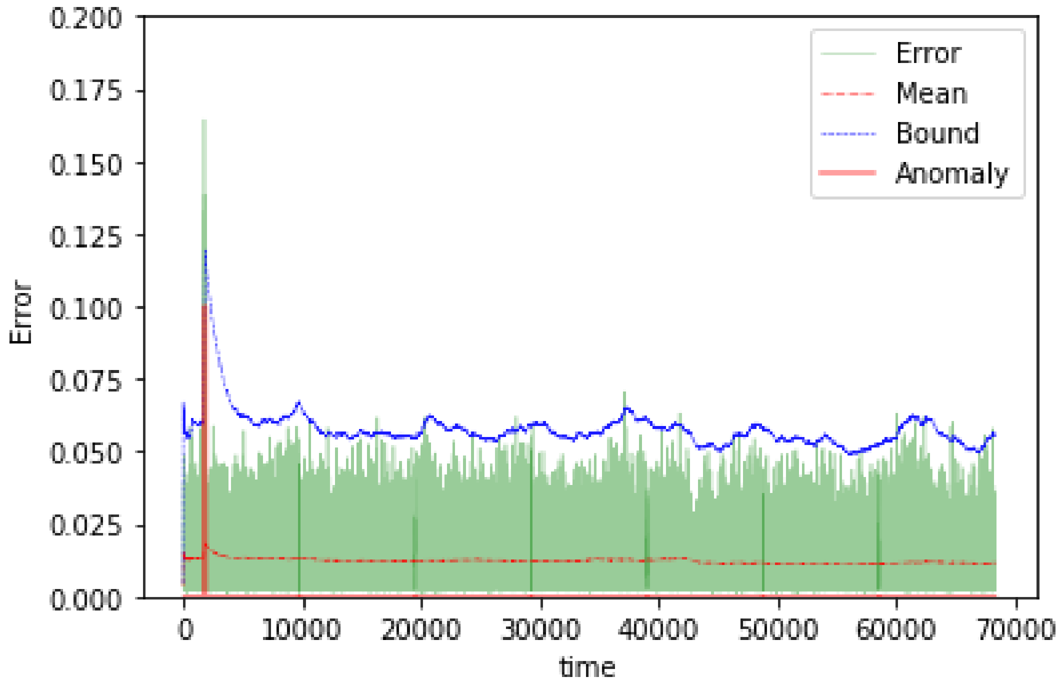





```
The mean error for lin2_app_change_early_ is 0.04016179985655575 for length 68297
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

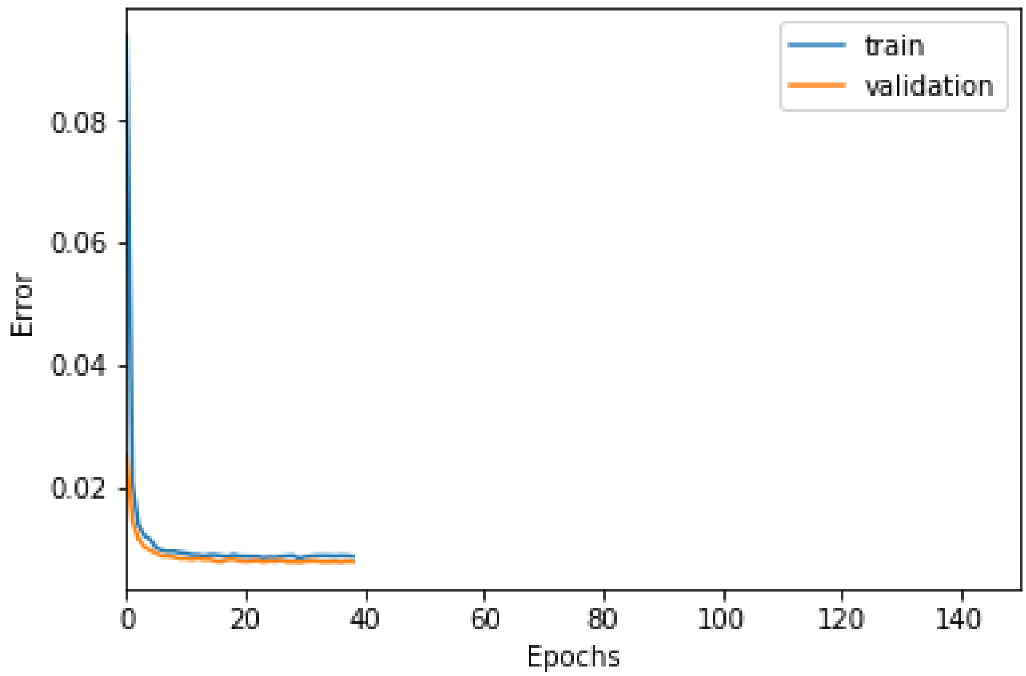

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



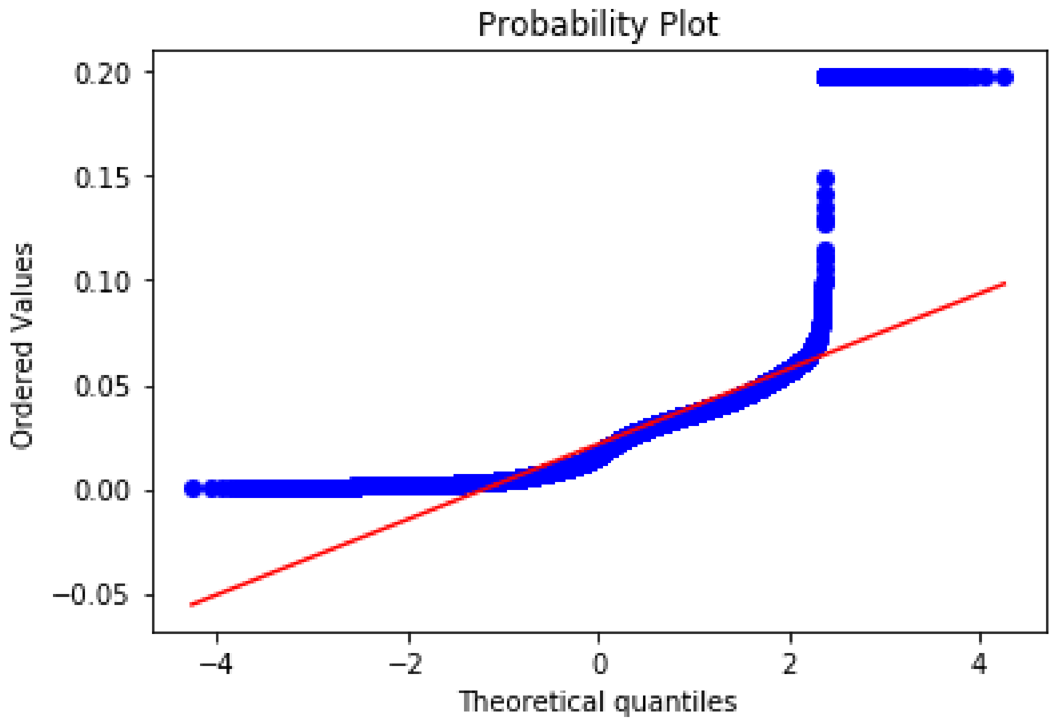

```
The mean error for lin2_normal_ is 0.012259594105293292 for length 68297
====================
```

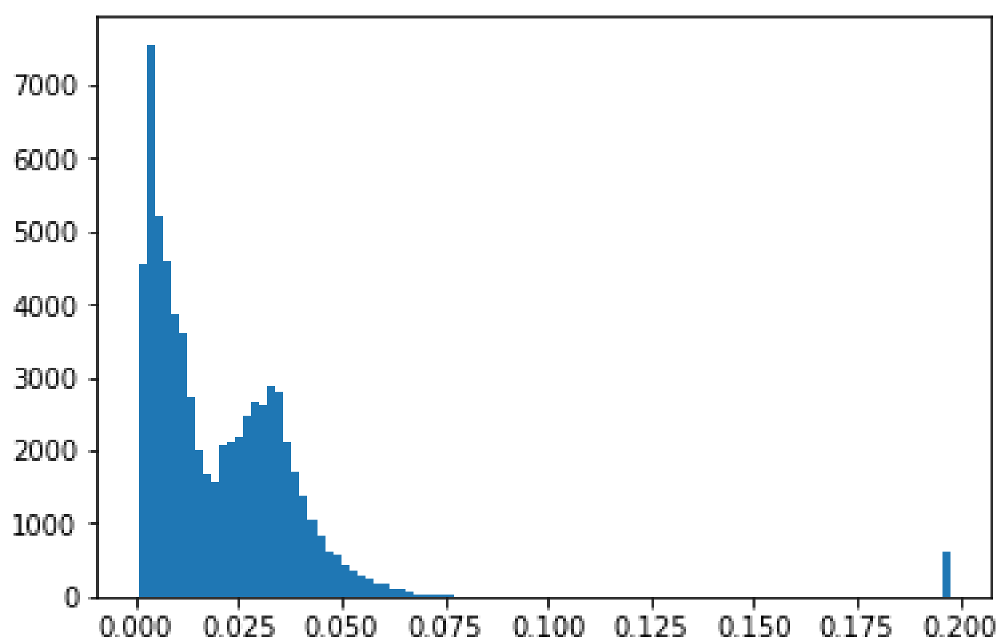
5 steps

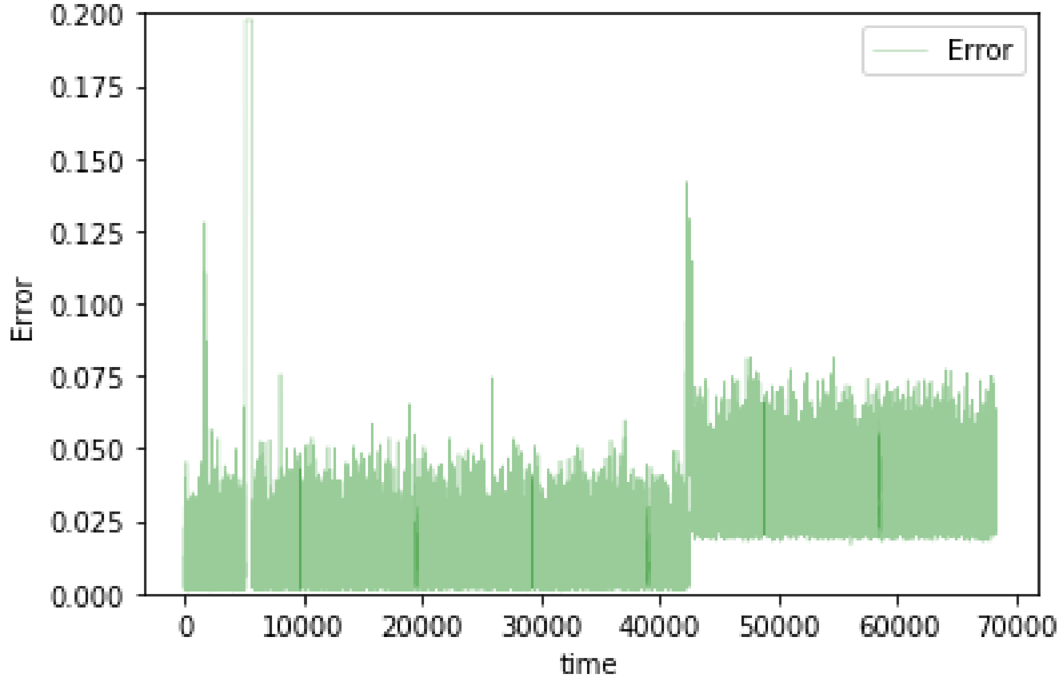
```
In [43]: TIMESTEPS = 5
         DIM = 29
         tgen = flat_generator(X, TIMESTEPS)
         vgen = flat_generator(val_X, TIMESTEPS)
         name = "lin5"

In [44]: input_layer = Input(shape=(TIMESTEPS*DIM,))
         output = Dense(DIM, activation='sigmoid')(input_layer)

In [45]: model = Model(input_layer, output)
         model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [46]: train(model, tgen, vgen, name=name)
         test(model, name=name, window=TIMESTEPS)
```

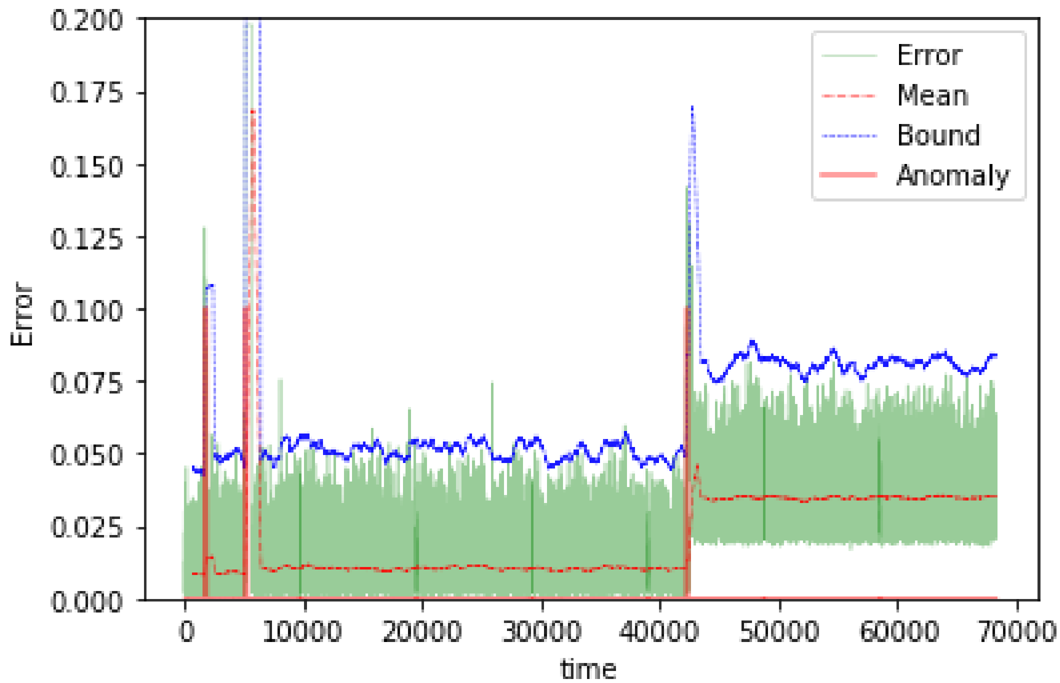


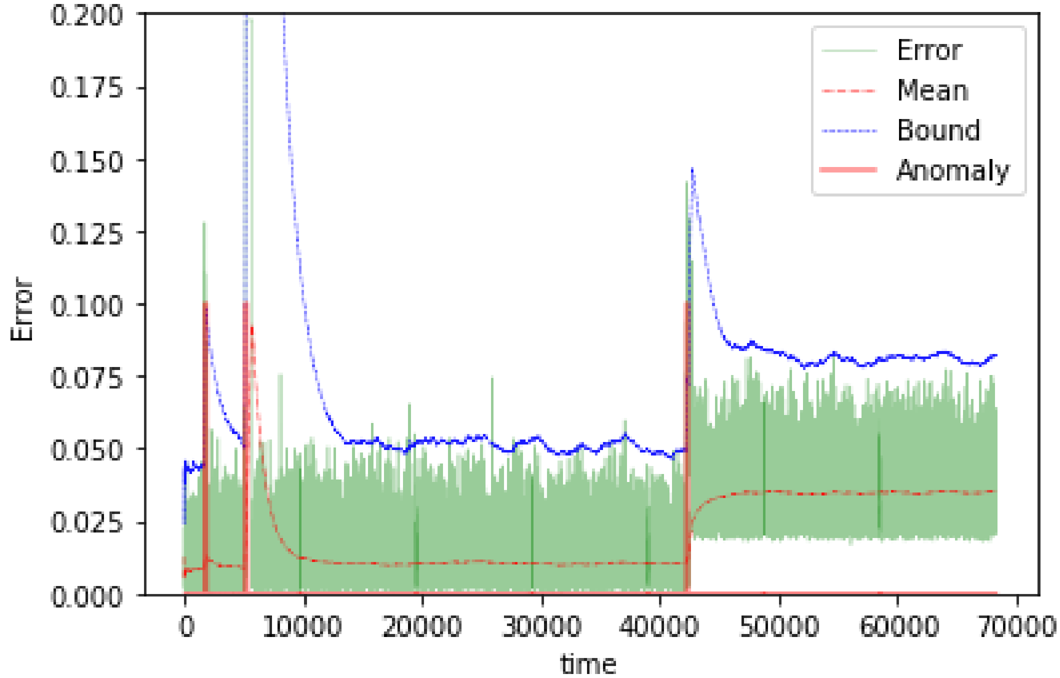

```
Training loss for final epoch is 0.008623824689886532
Validation loss for final epoch is 0.007751765090739354
---------- Beginning tests for lin5 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.20
0.15
0.10
0.05
0.00
−0.05
−4
−2
0
2
4

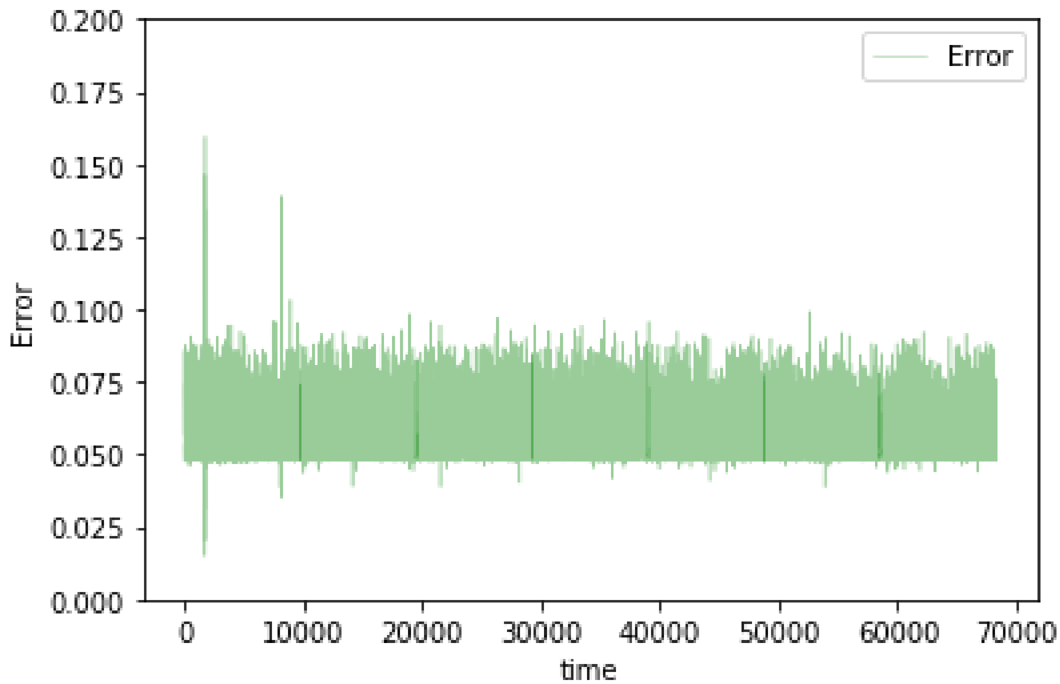

7000
6000
5000
4000
3000
2000
1000
0
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200

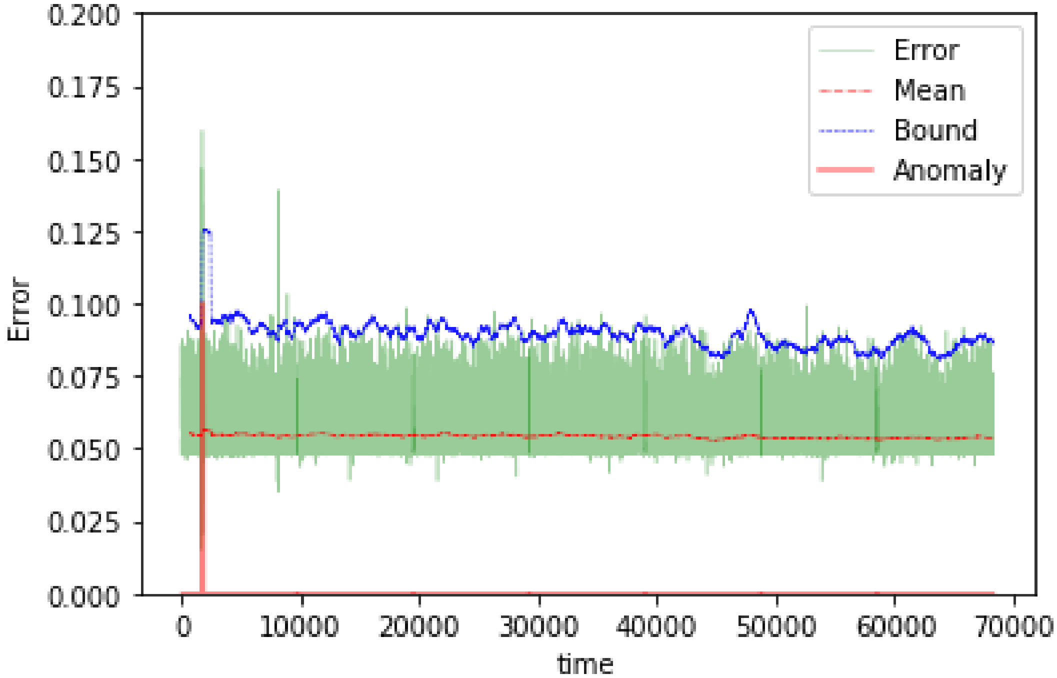

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error

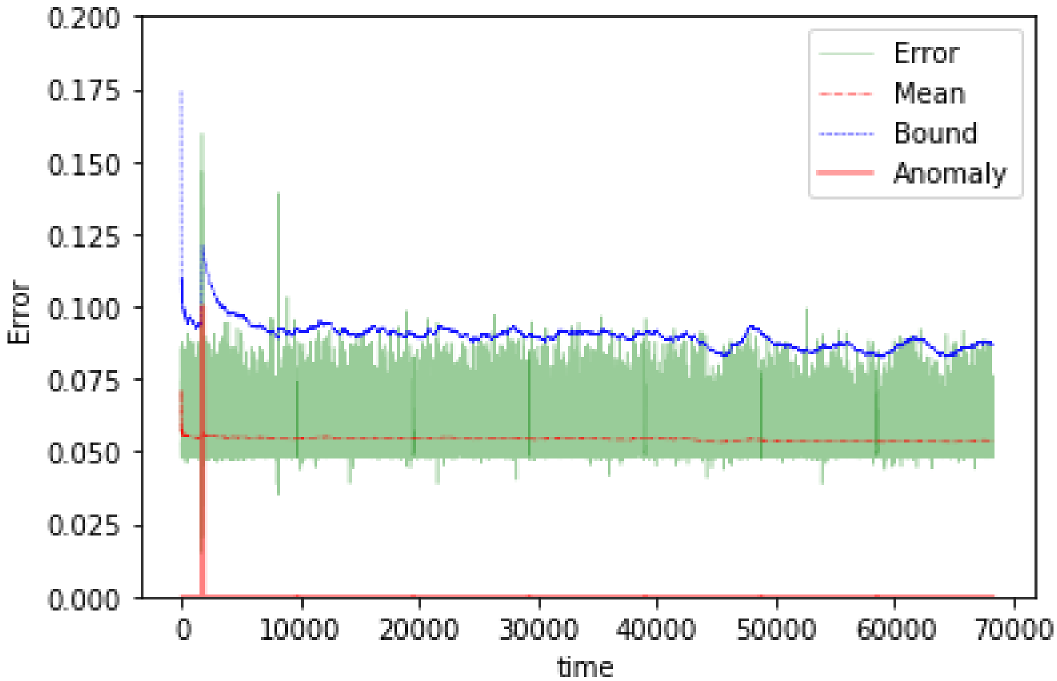

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly



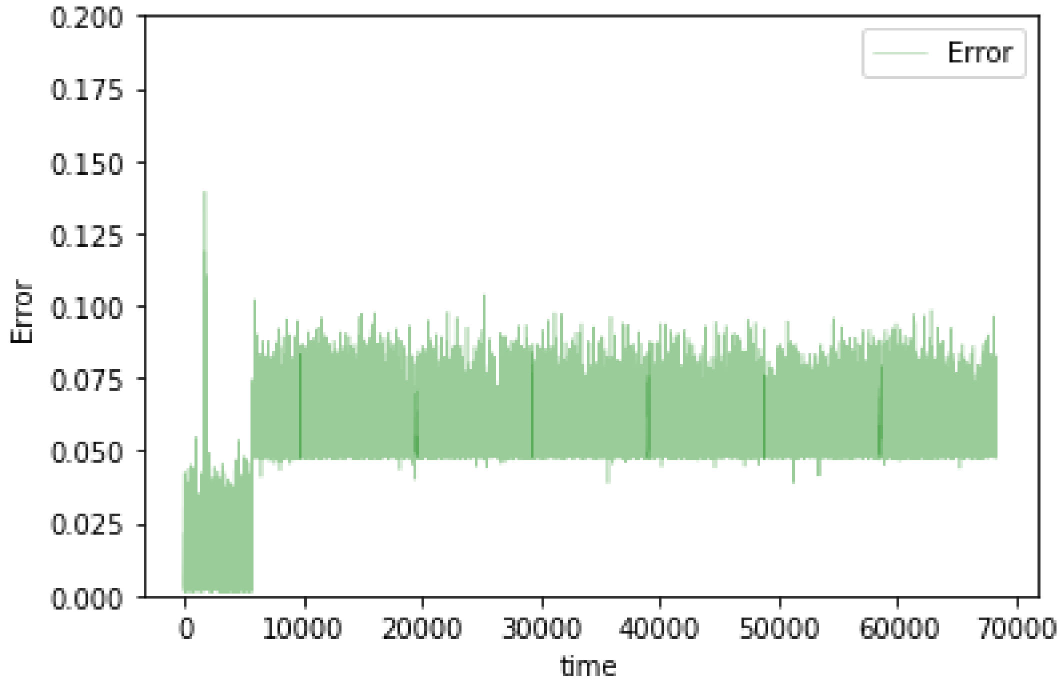

```
The mean error for lin5_disk_IO_start_ is 0.021346507201983056 for length 68294
Testing on Avg. load data.
```

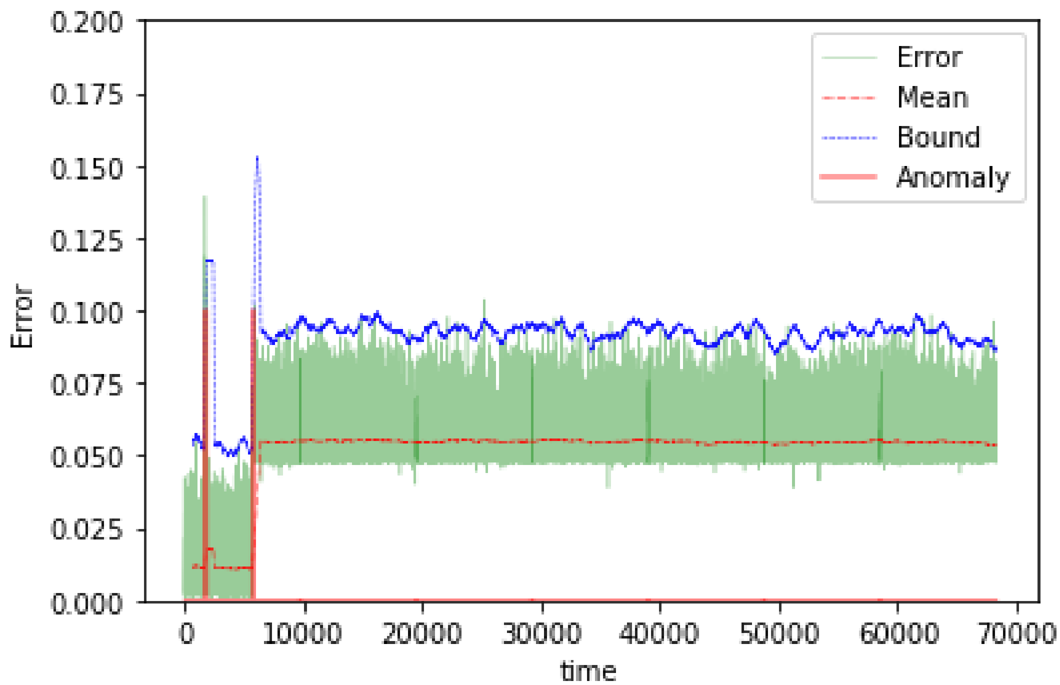


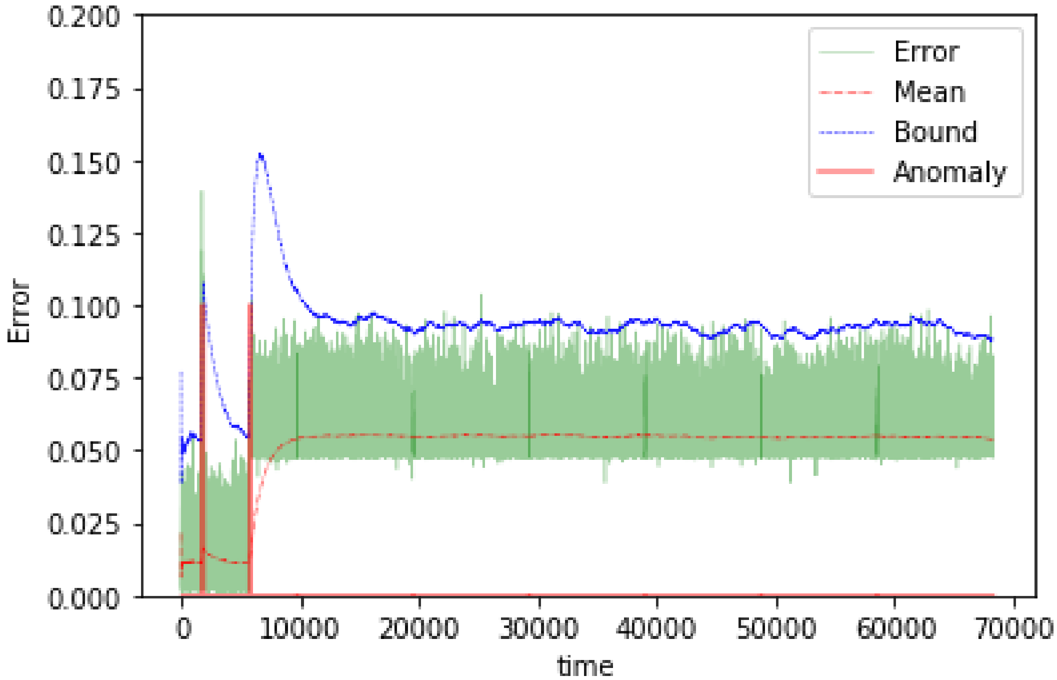

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

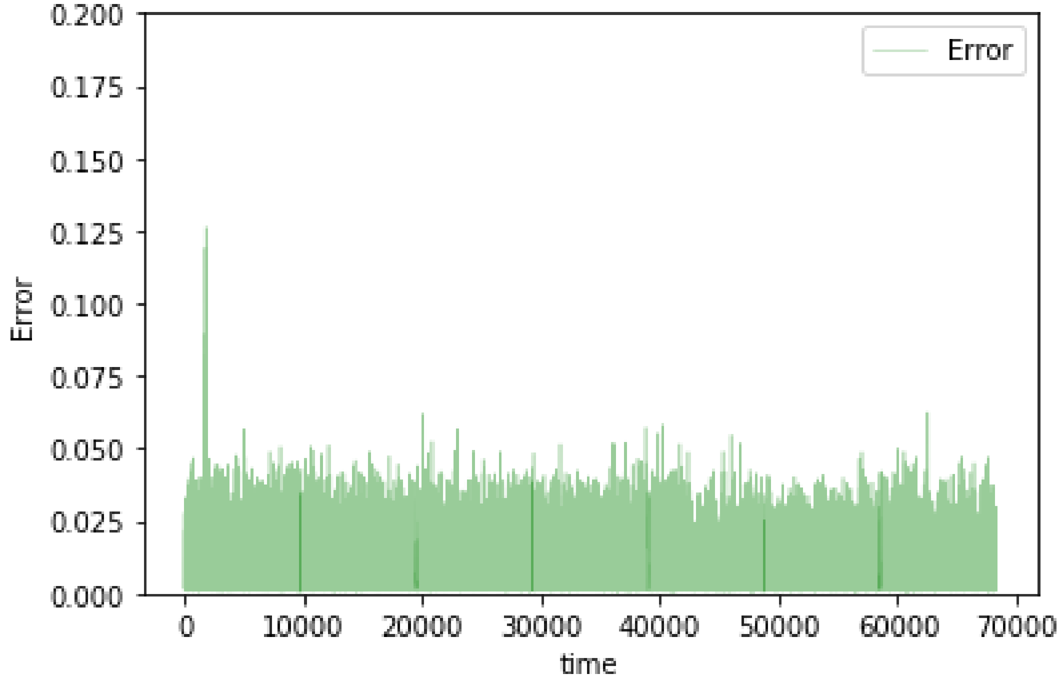
```
The mean error for lin5_avg_load_ is 0.054234096514334466 for length 68294
Testing on app change early data.
```

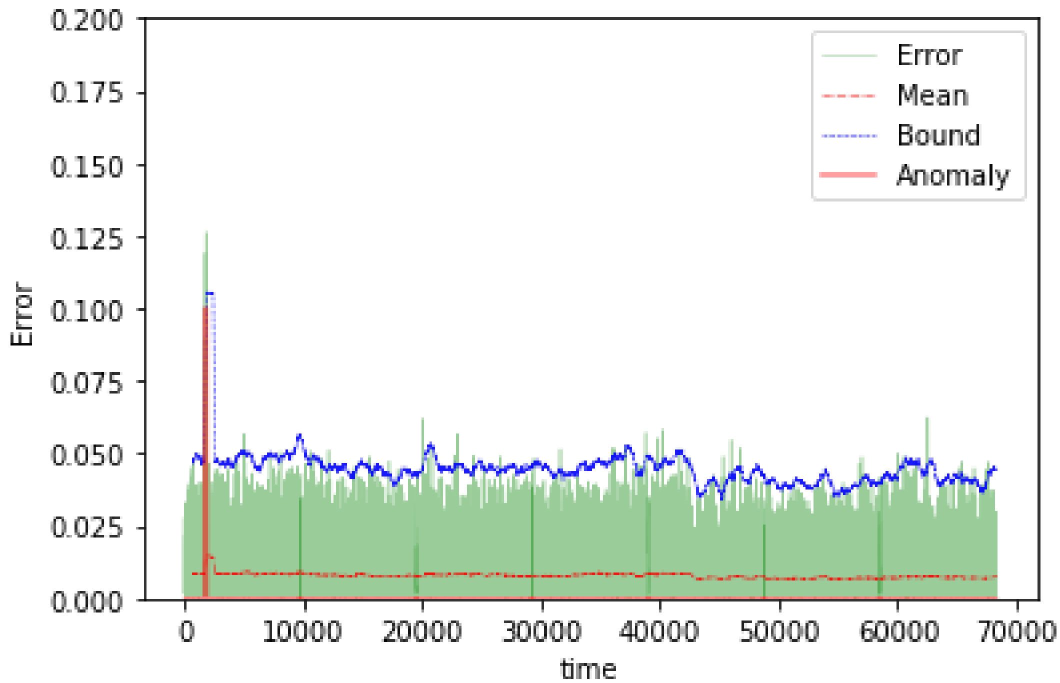


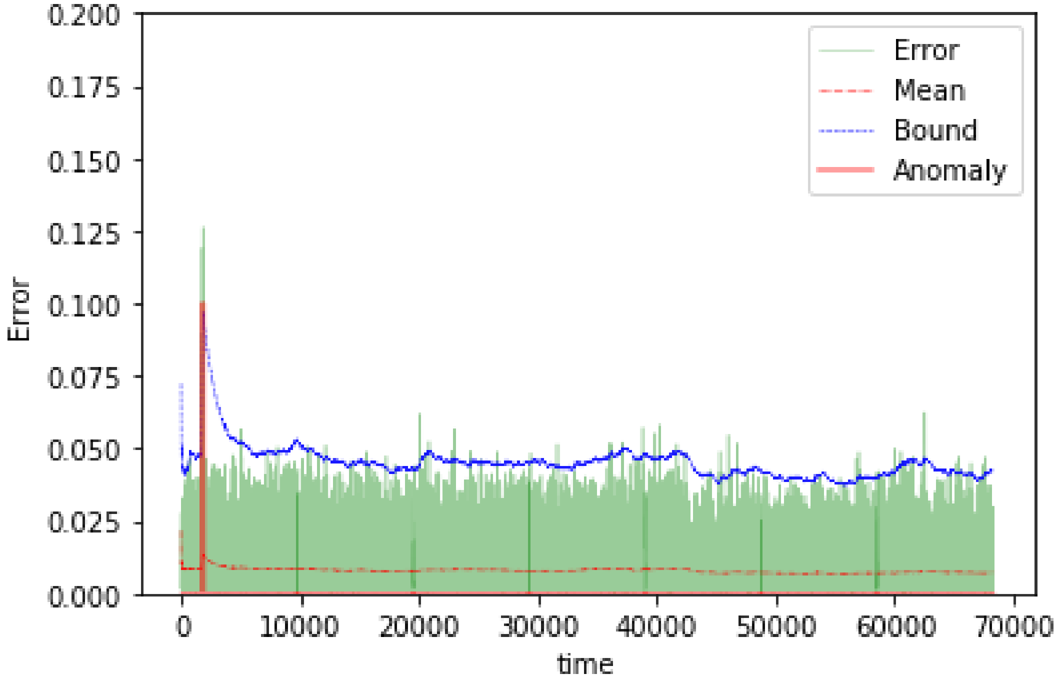





```
The mean error for lin5_app_change_early_ is 0.0511666120448845 for length 68294
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

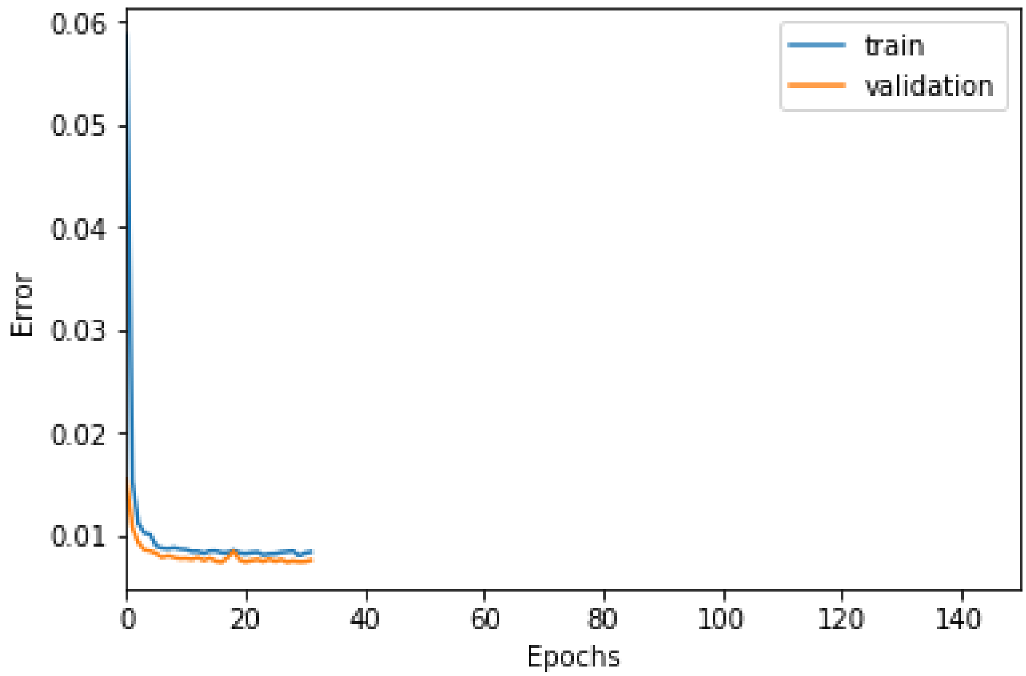

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



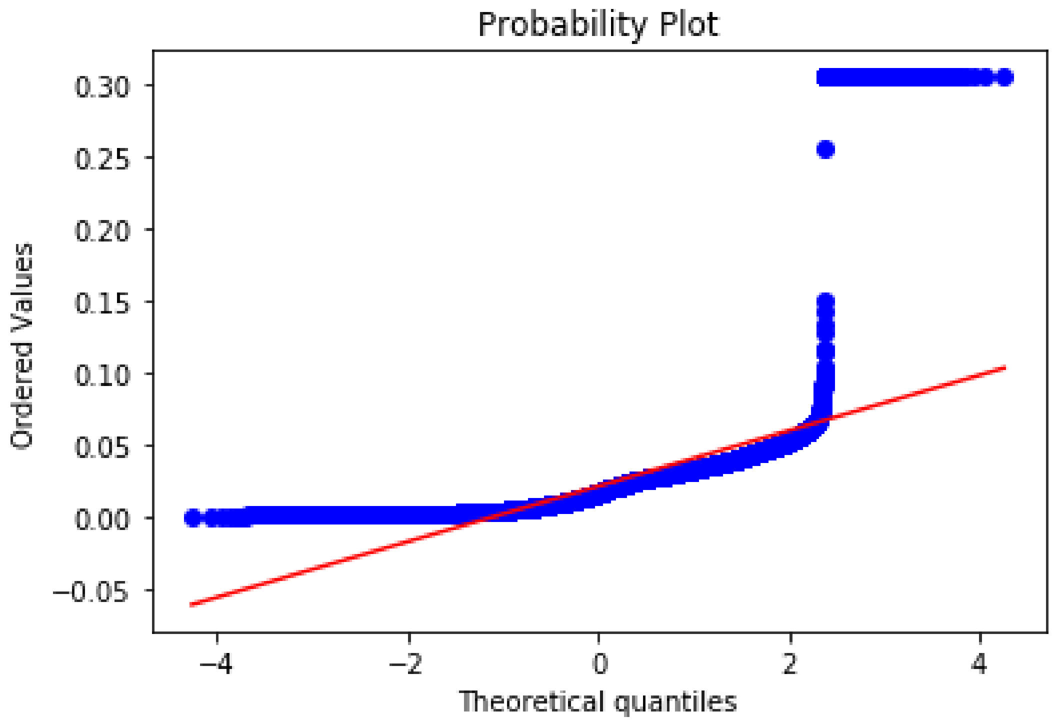

```
The mean error for lin5_normal_ is 0.007881495414756872 for length 68294
====================
```

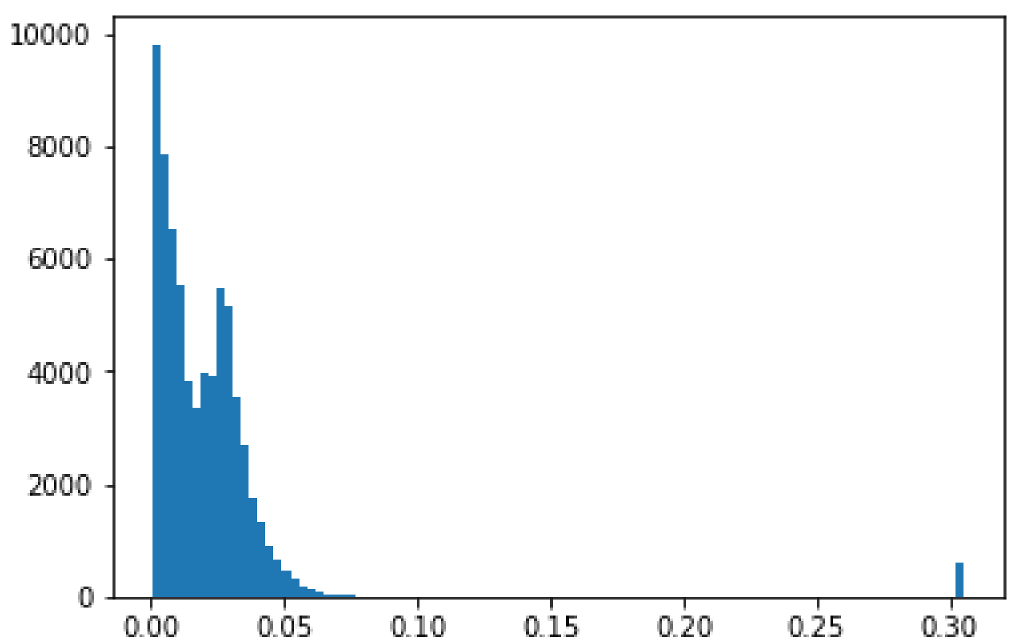
10 steps

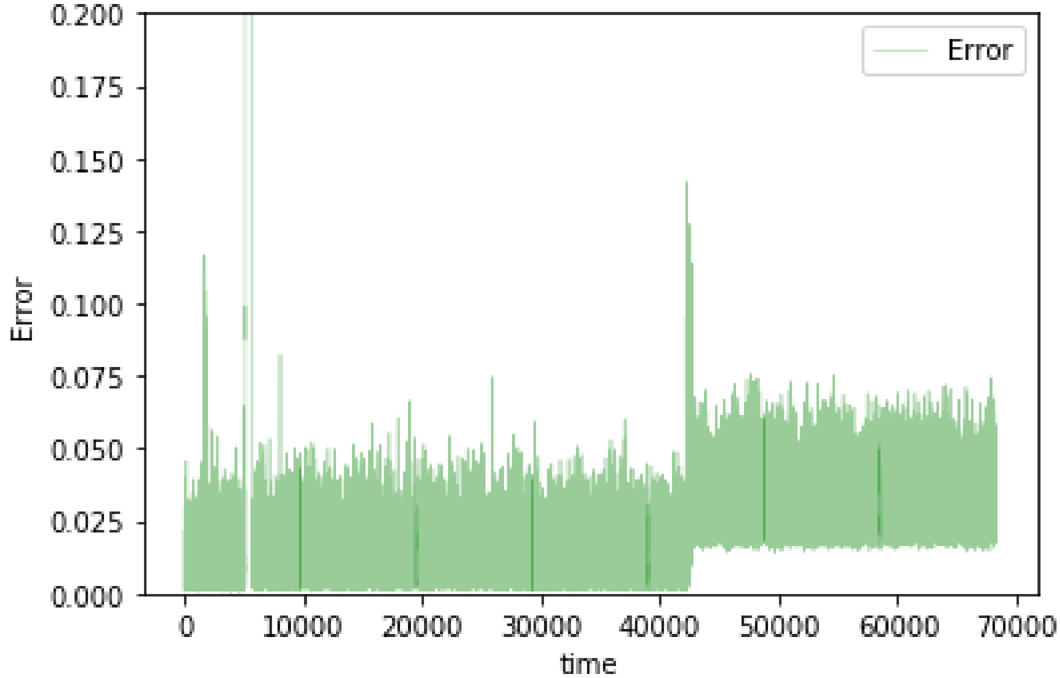
```
In [47]: TIMESTEPS = 10
         DIM = 29
         tgen = flat_generator(X, TIMESTEPS)
         vgen = flat_generator(val_X, TIMESTEPS)
         name = "lin10"

In [48]: input_layer = Input(shape=(TIMESTEPS*DIM,))
         output = Dense(DIM, activation='sigmoid')(input_layer)

In [49]: model = Model(input_layer, output)
         model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [50]: train(model, tgen, vgen, name=name)
         test(model, name=name, window=TIMESTEPS)
```



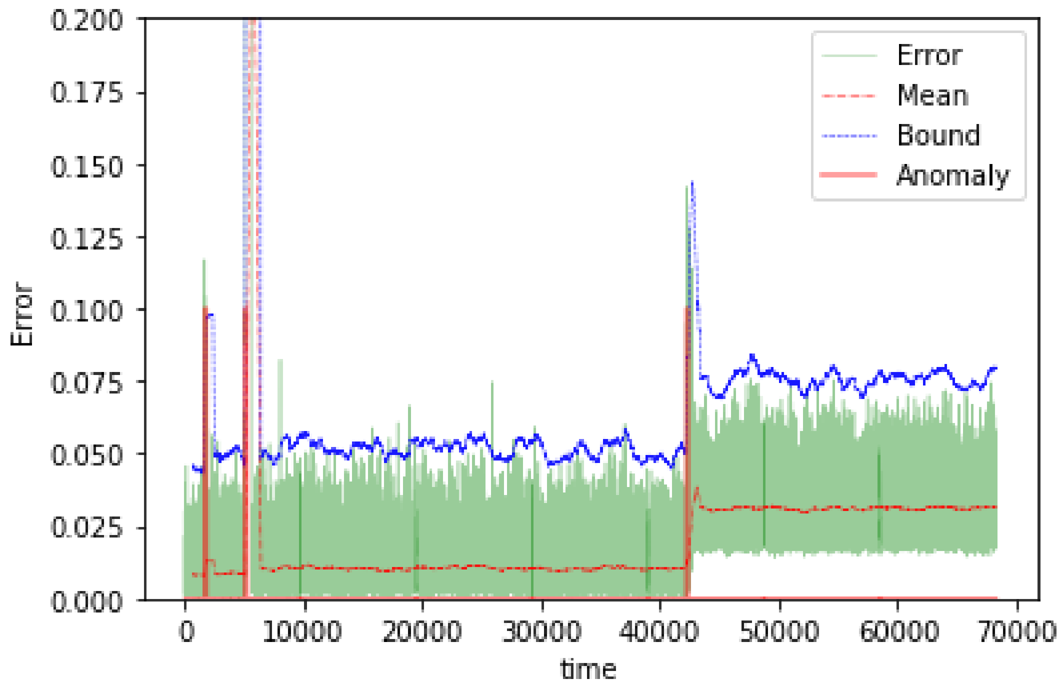

```
Training loss for final epoch is 0.008360213179839775
Validation loss for final epoch is 0.007554968993877992
---------- Beginning tests for lin10 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

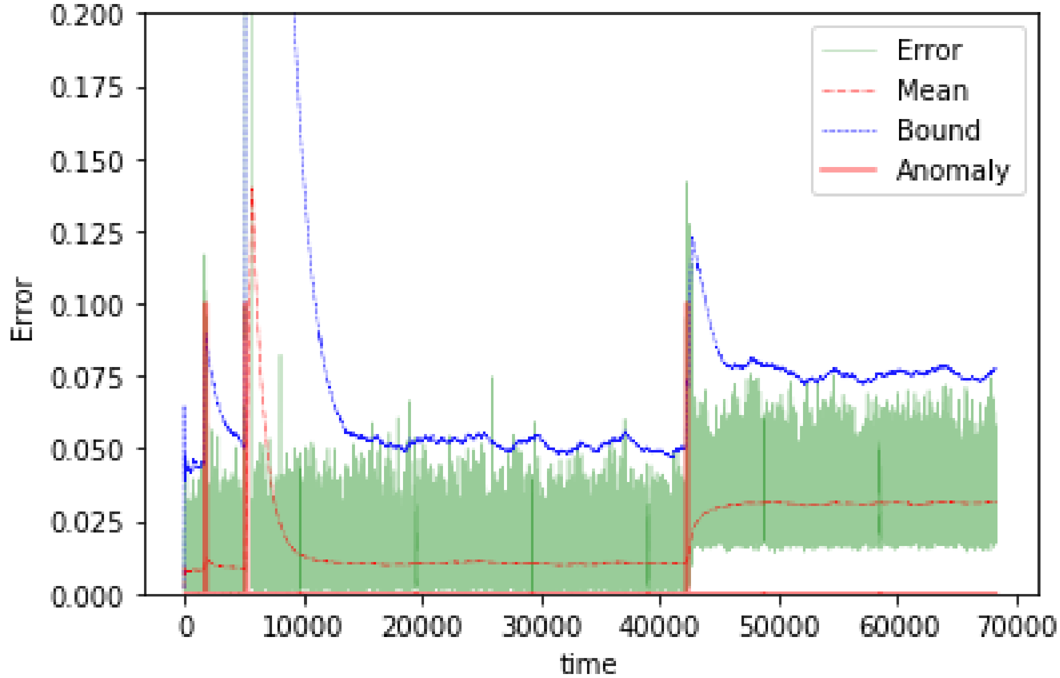

10000
8000
6000
4000
2000
0
0.00
0.05
0.10
0.15
0.20
0.25
0.30



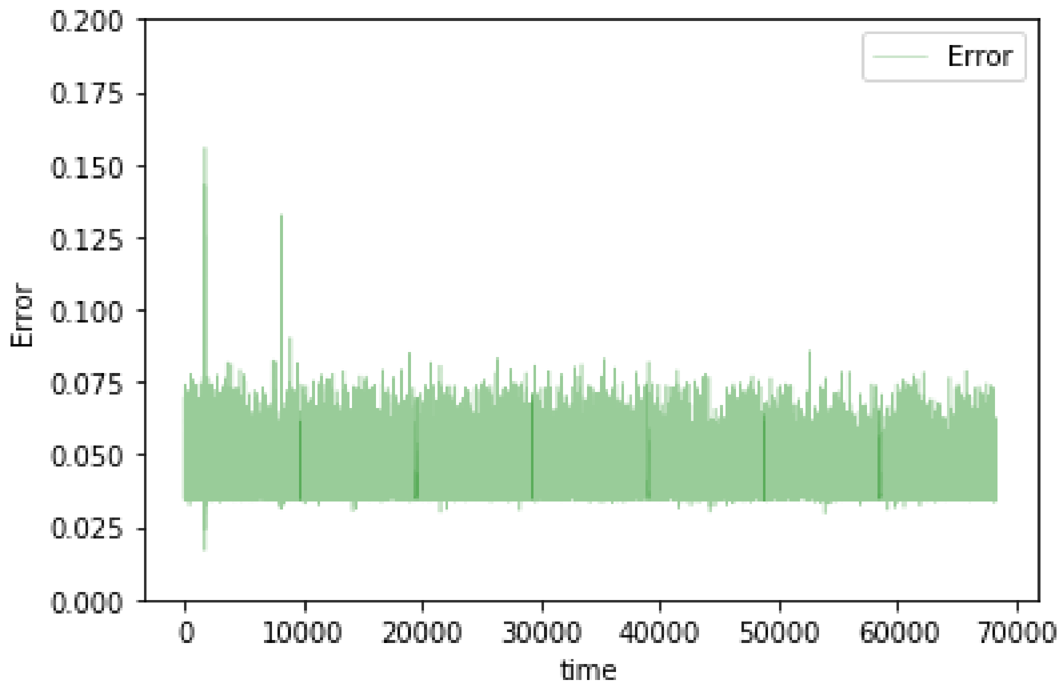
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

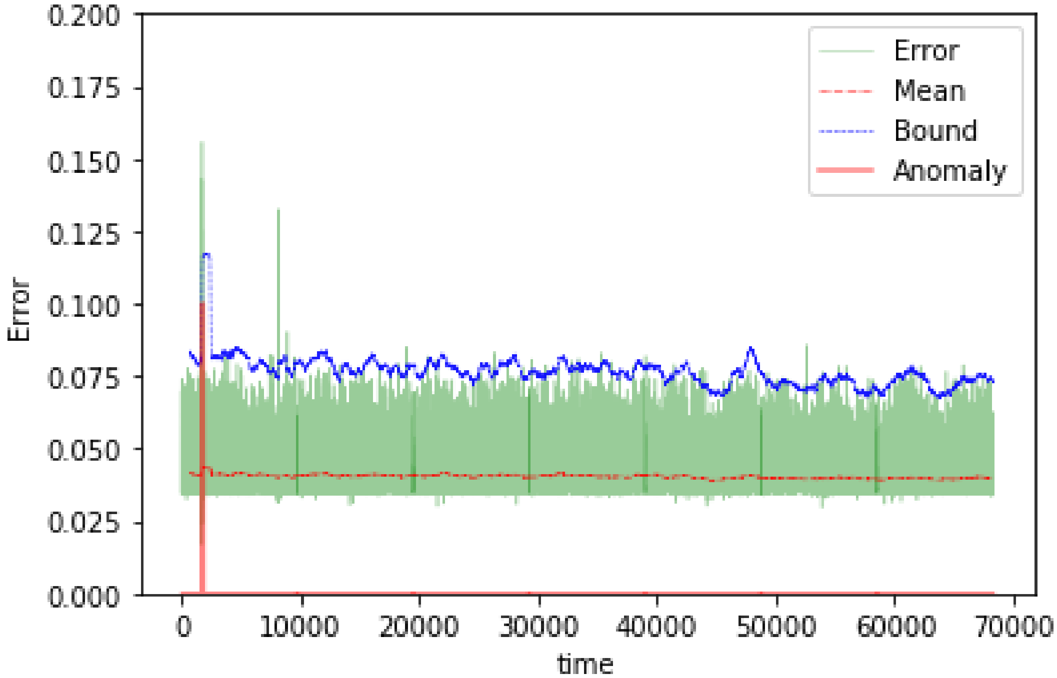


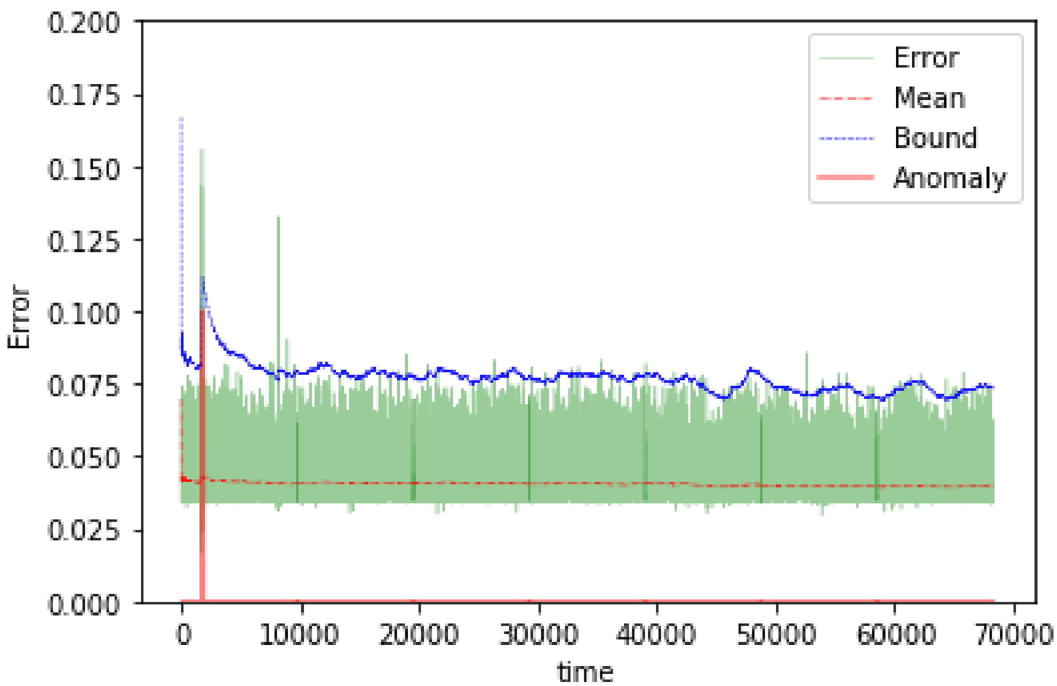
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



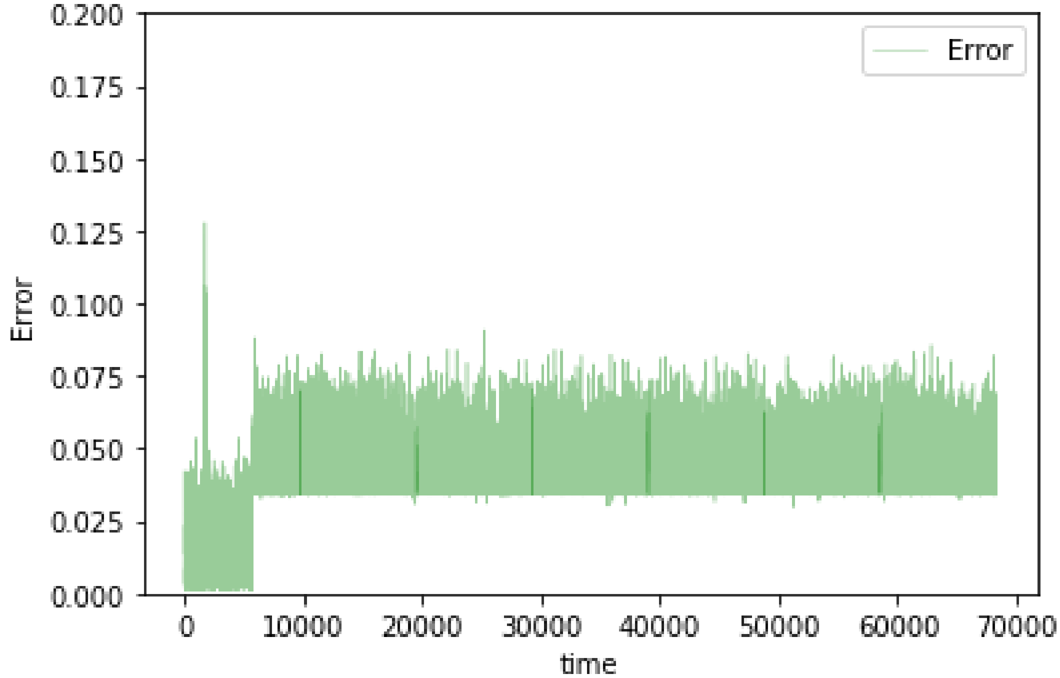
The mean error for lin10_disk_IO_start_ is 0.020840498627911 89 for length 68289
Testing on Avg. load data.

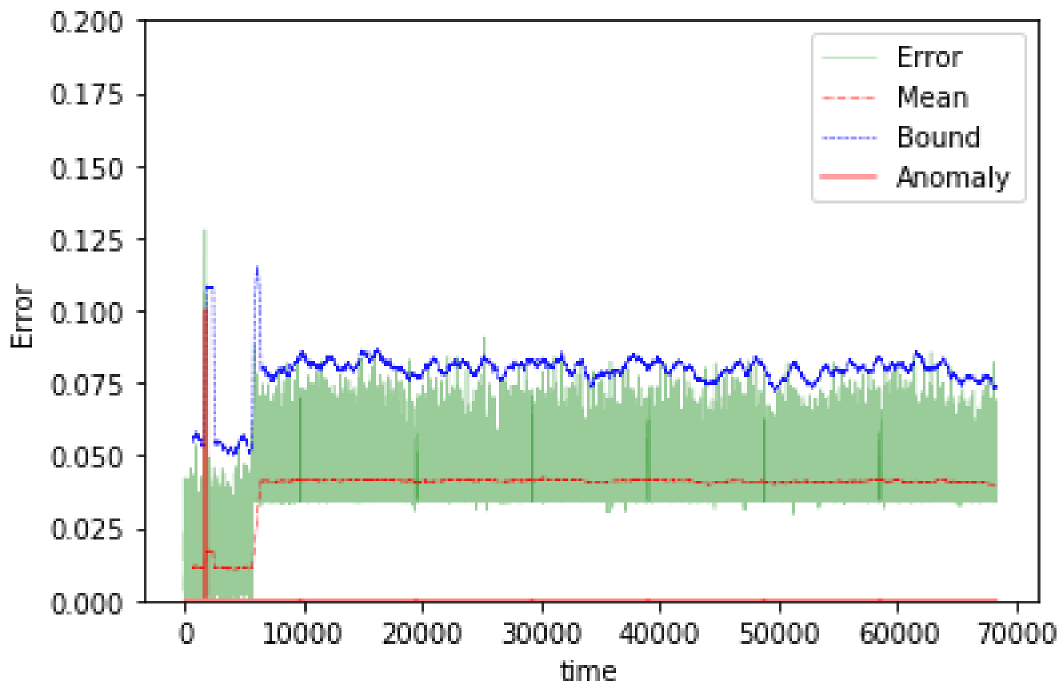


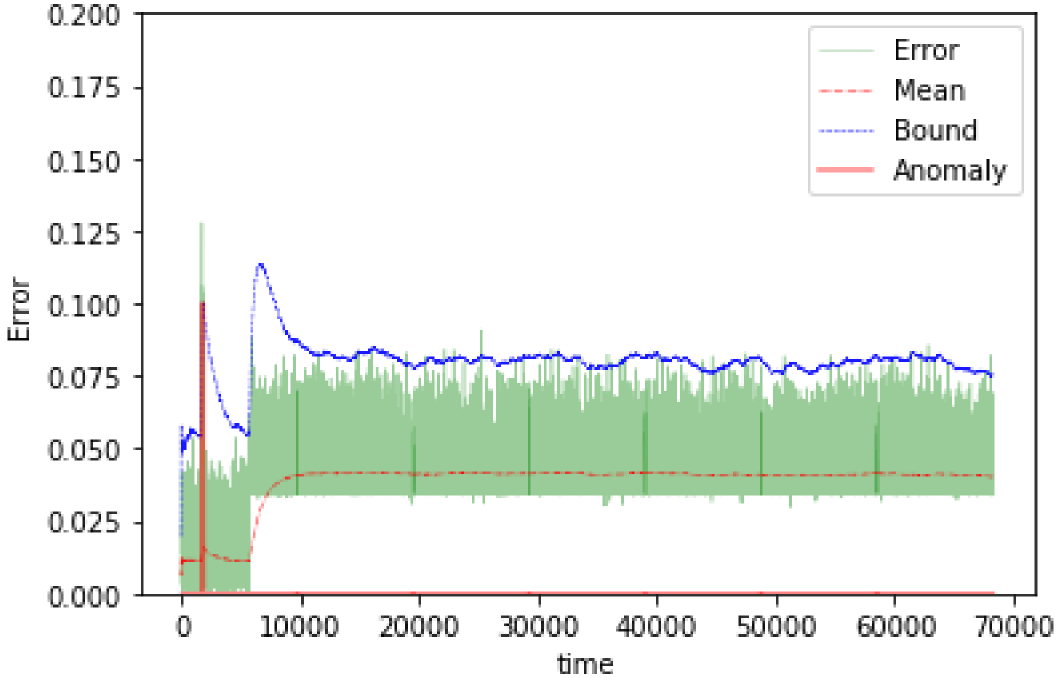

Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000


Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

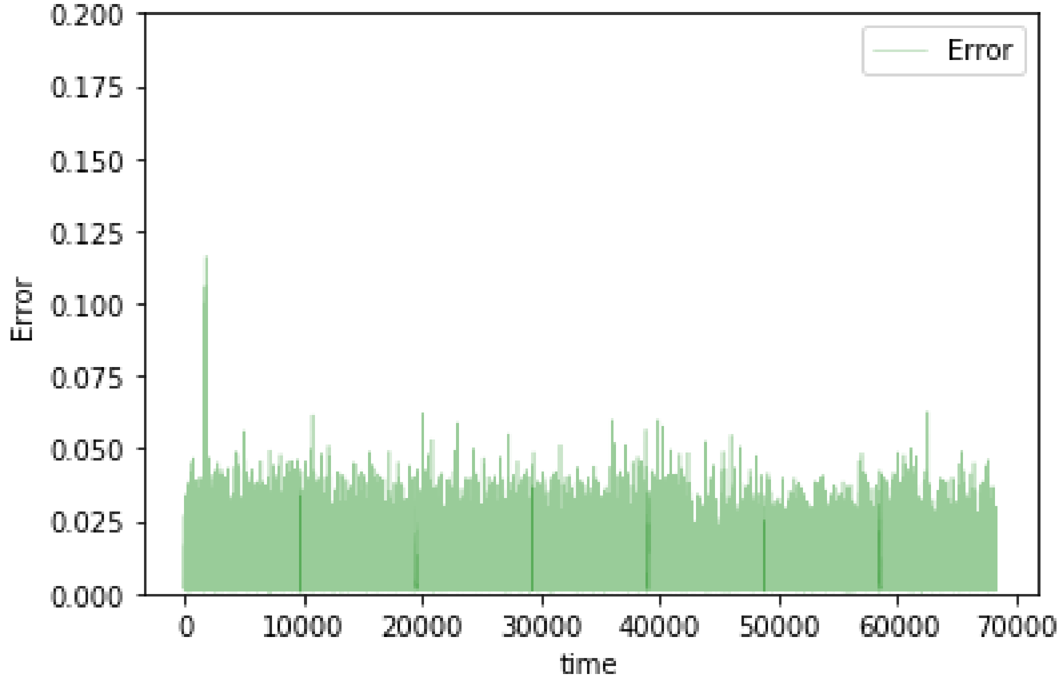
```
The mean error for lin10_avg_load_ is 0.04041410534421001 for length 68289
Testing on app change early data.
```

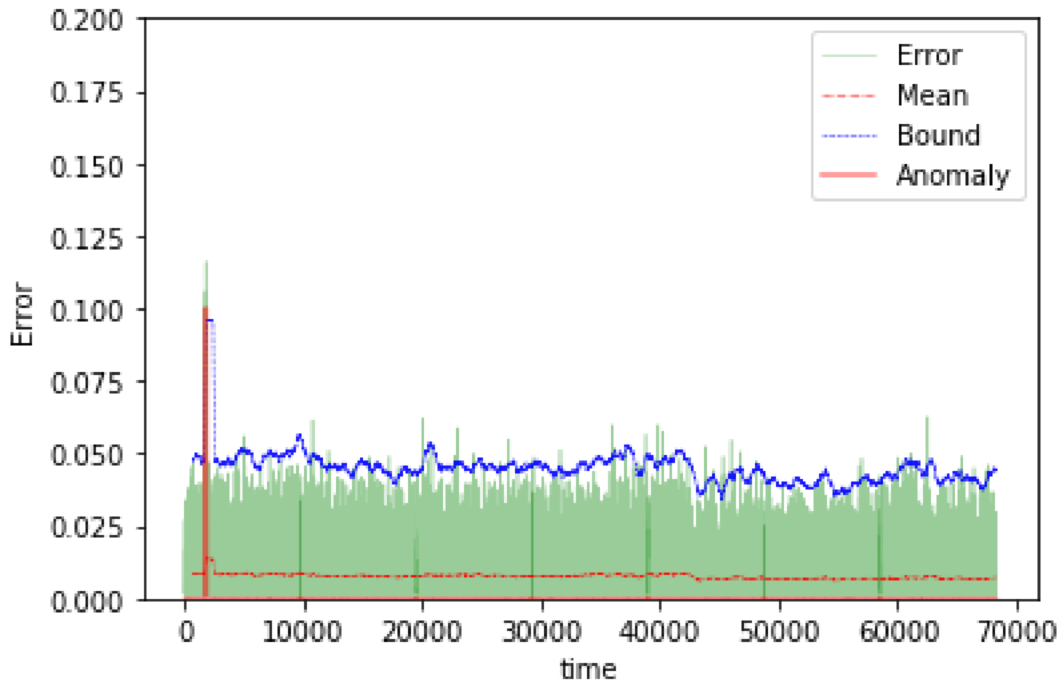


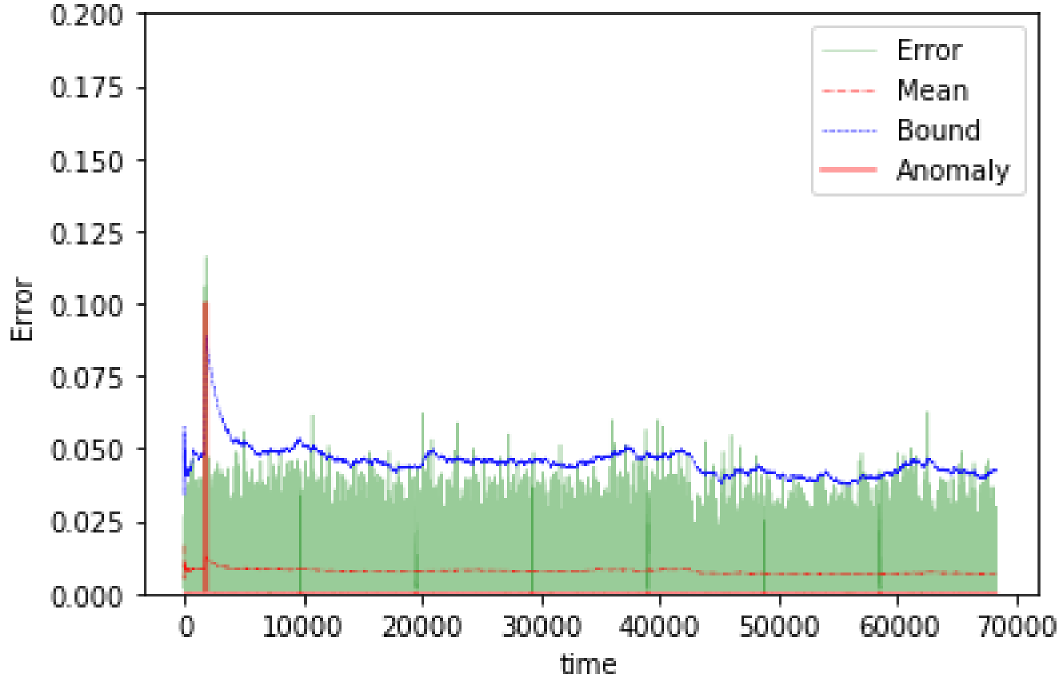





```
The mean error for lin10_app_change_early_ is 0.03874384335913257 for length 68289
Testing on Normal data.
```


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error

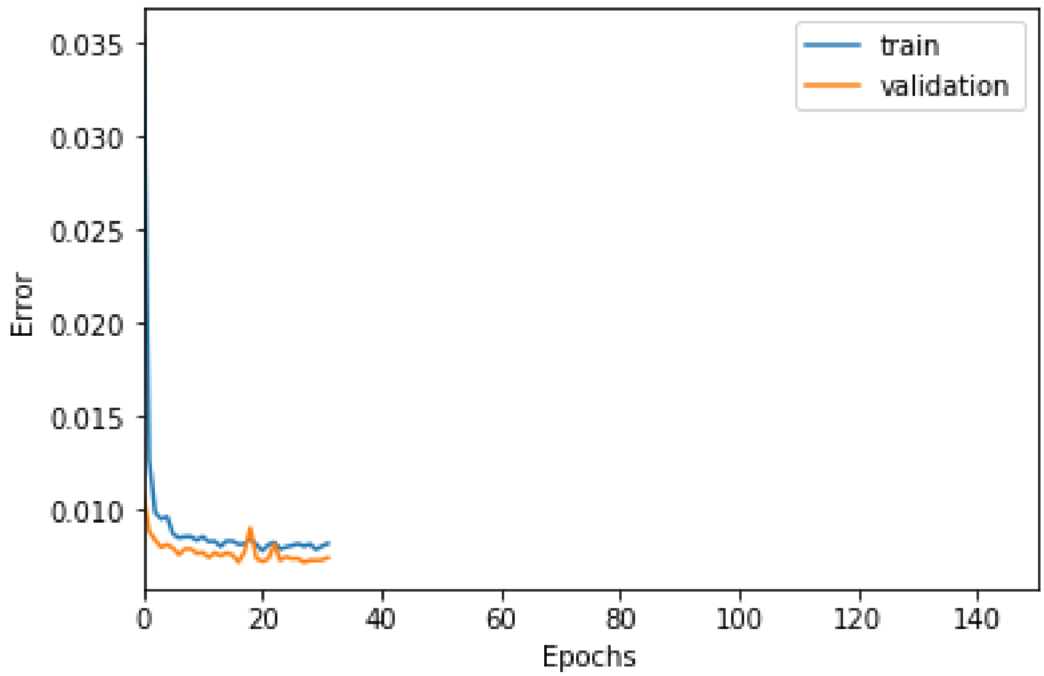

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly



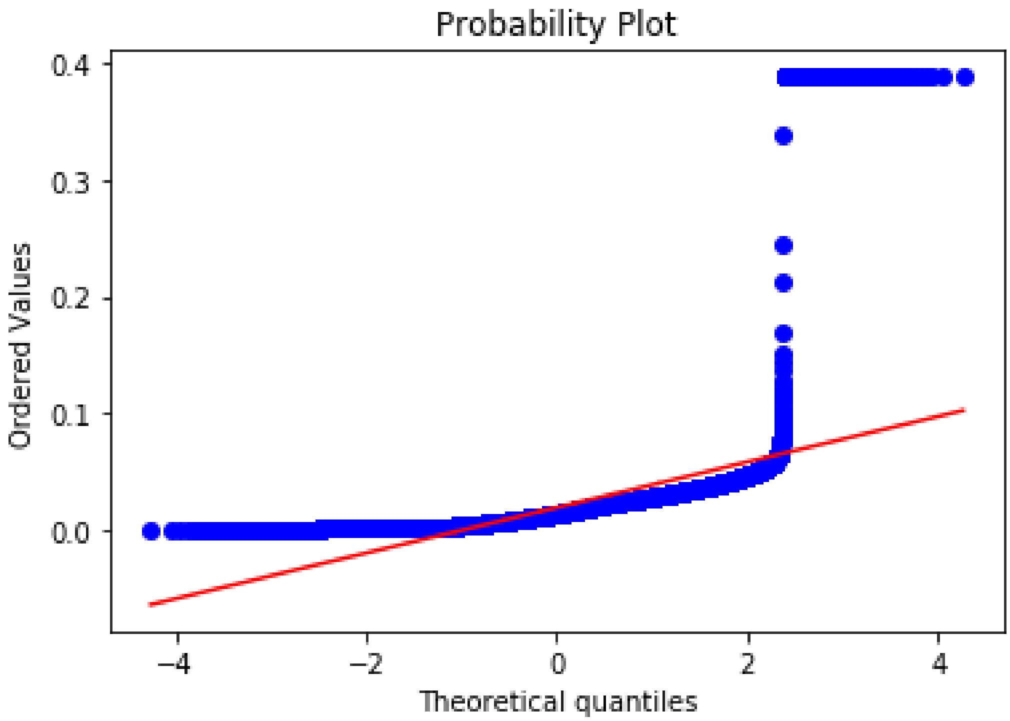

```
The mean error for lin10_normal_ is 0.007633873036824338 for length 68289
====================
```

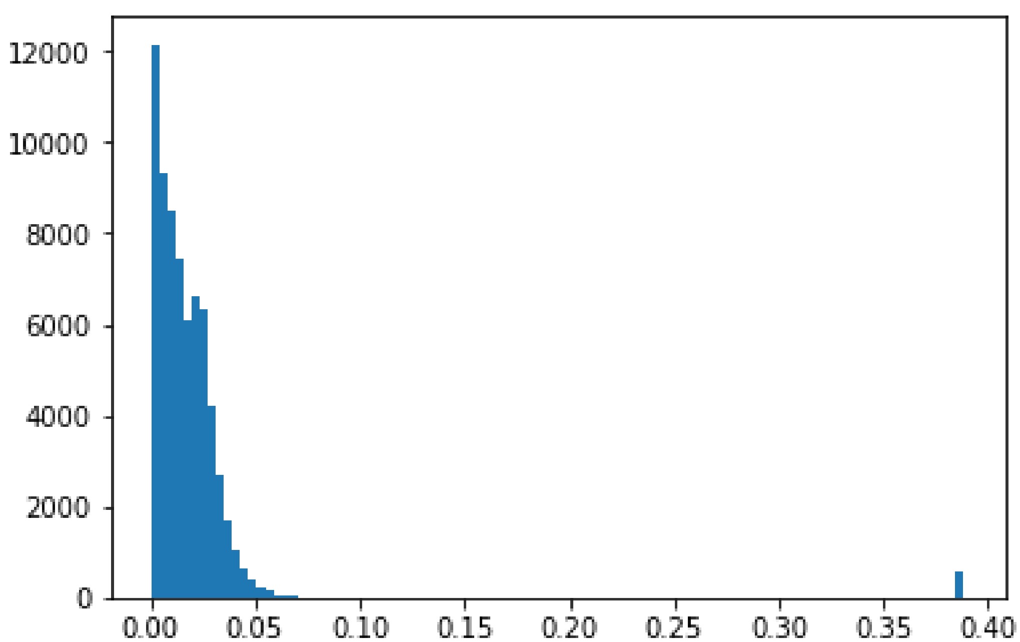
20 steps

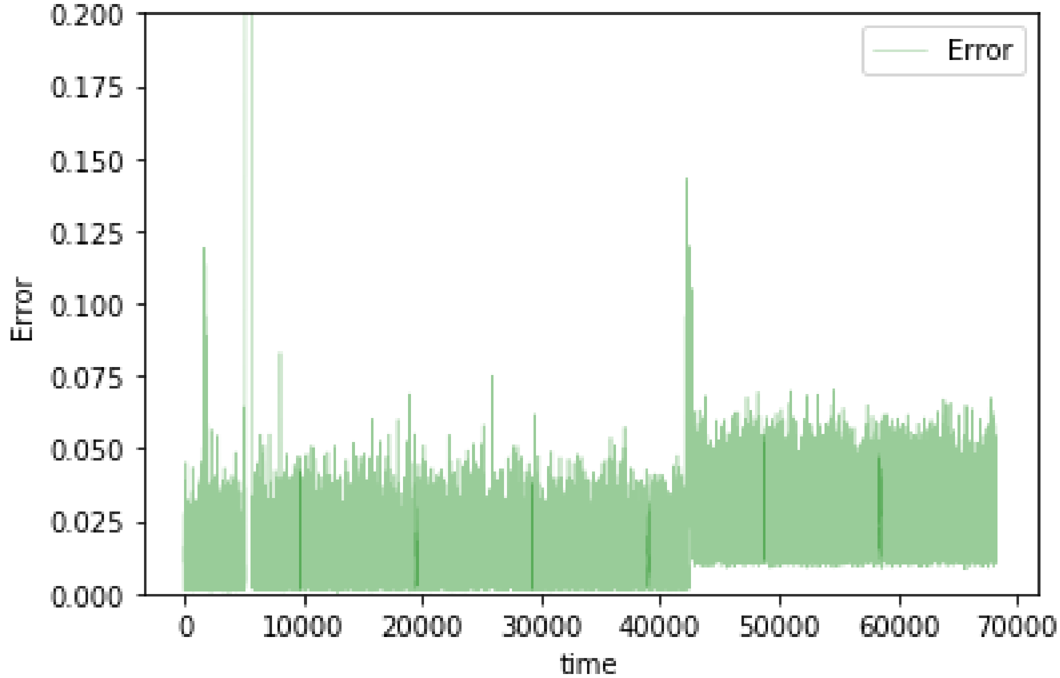
```
In [51]: TIMESTEPS = 20
         DIM = 29
         tgen = flat_generator(X, TIMESTEPS)
         vgen = flat_generator(val_X, TIMESTEPS)
         name = "lin20"

In [52]: input_layer = Input(shape=(TIMESTEPS*DIM,))
         output = Dense(DIM, activation='sigmoid')(input_layer)

In [53]: model = Model(input_layer, output)
         model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [54]: train(model, tgen, vgen, name=name)
         test(model, name=name, window=TIMESTEPS)
```

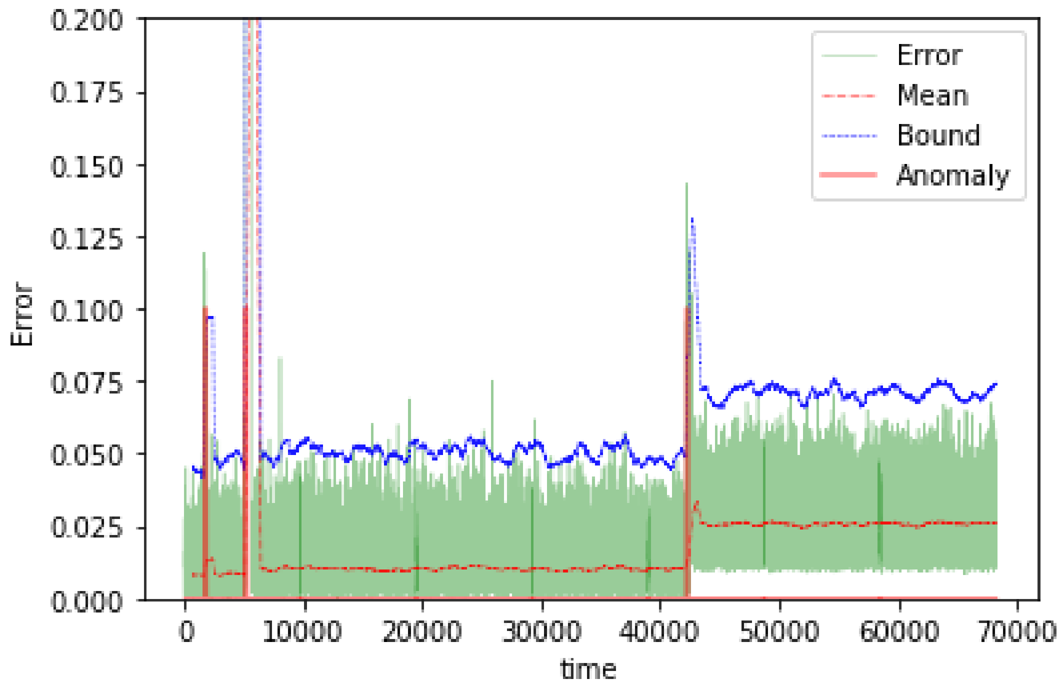


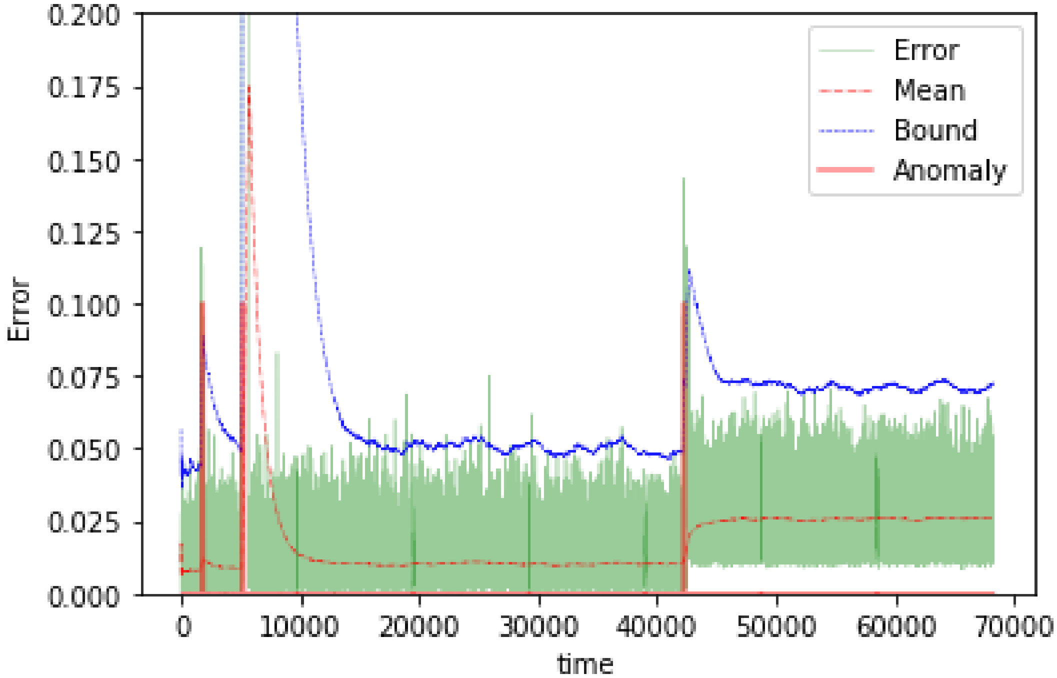

```
Training loss for final epoch is 0.008188069149153307
Validation loss for final epoch is 0.007428524158429354
---------- Beginning tests for lin20 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.4
0.3
0.2
0.1
0.0
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

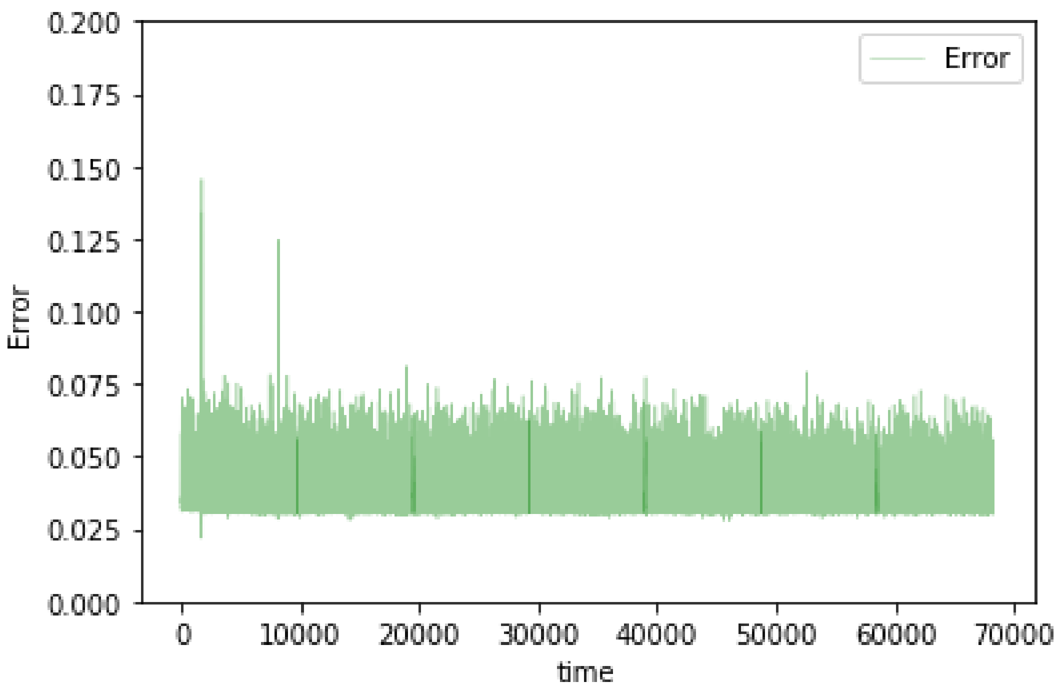

12000
10000
8000
6000
4000
2000
0
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40

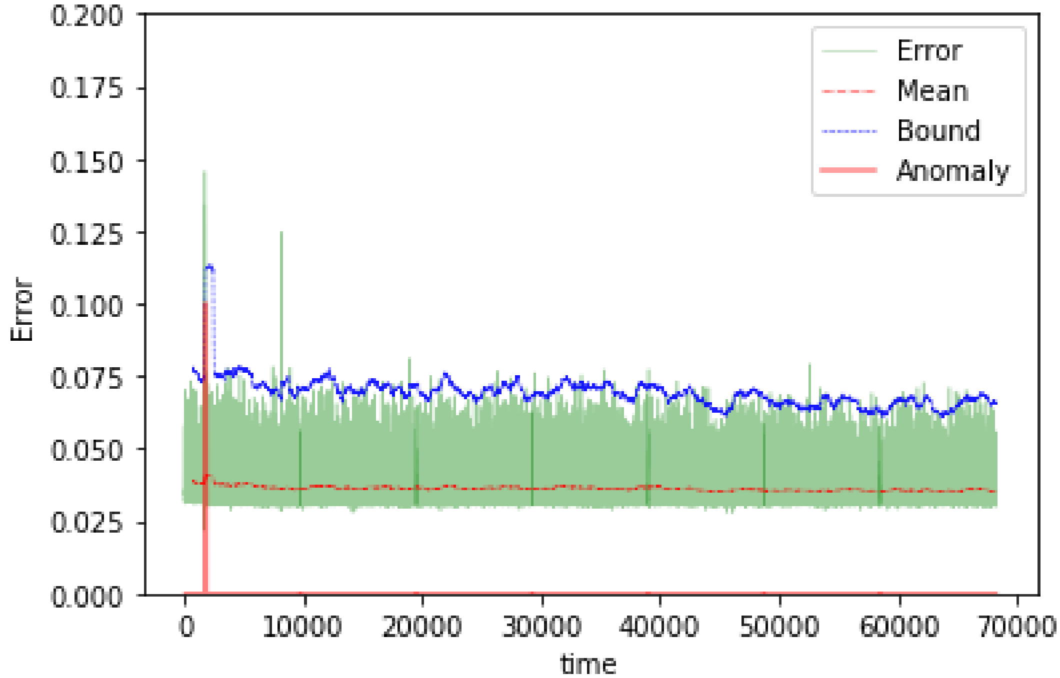

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

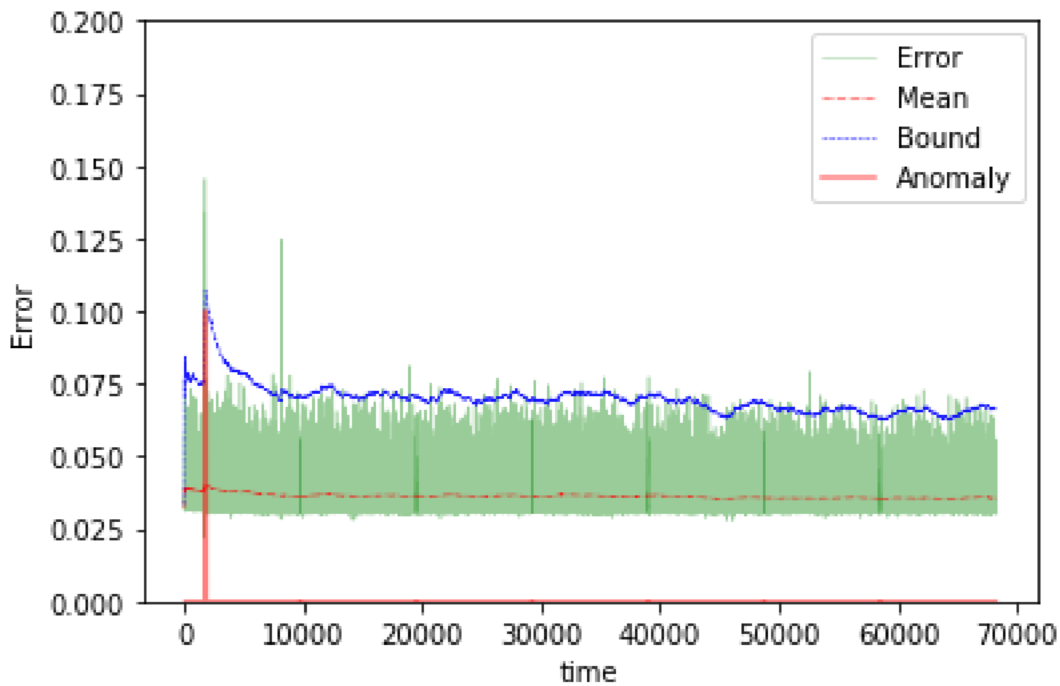

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



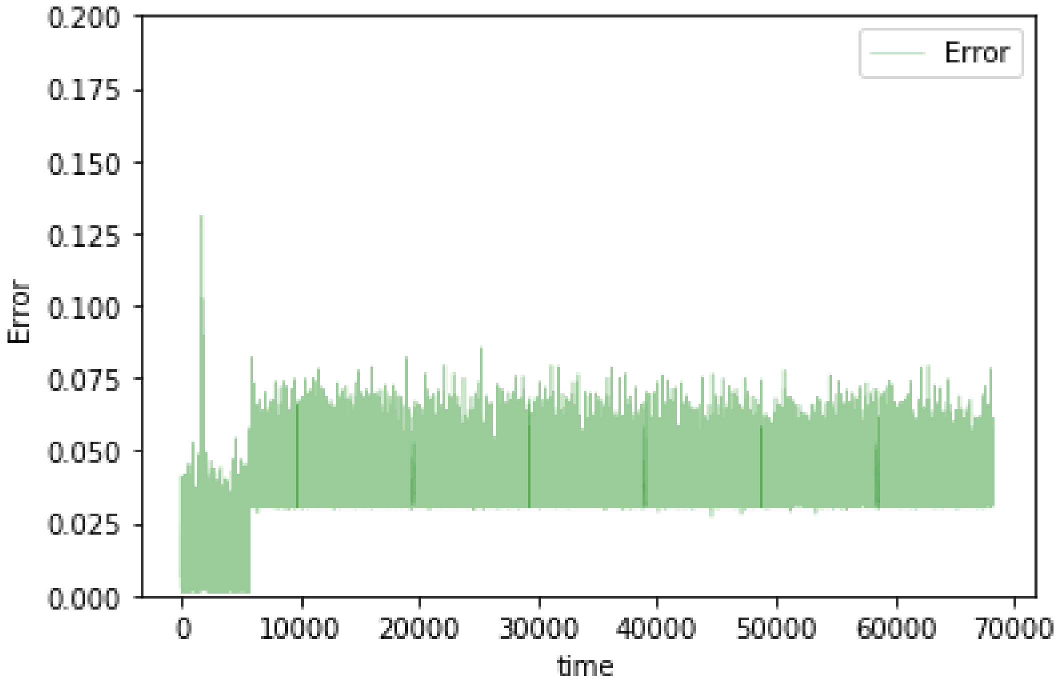

```
The mean error for lin20_disk_IO_start_ is 0.01941877791987124 for length 68279
Testing on Avg. load data.
```

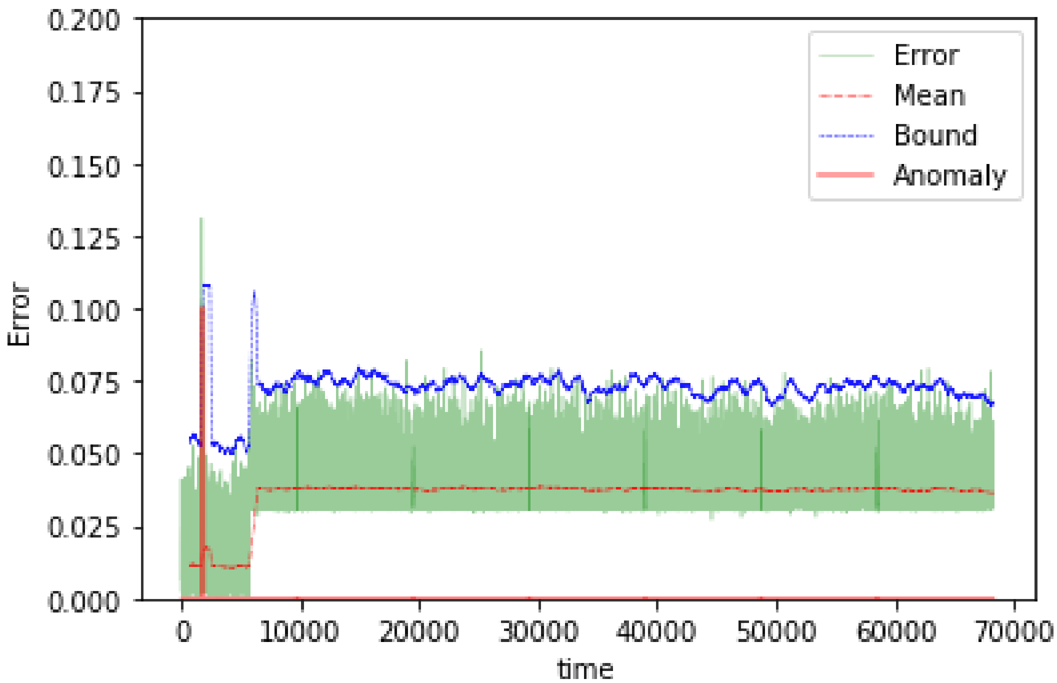


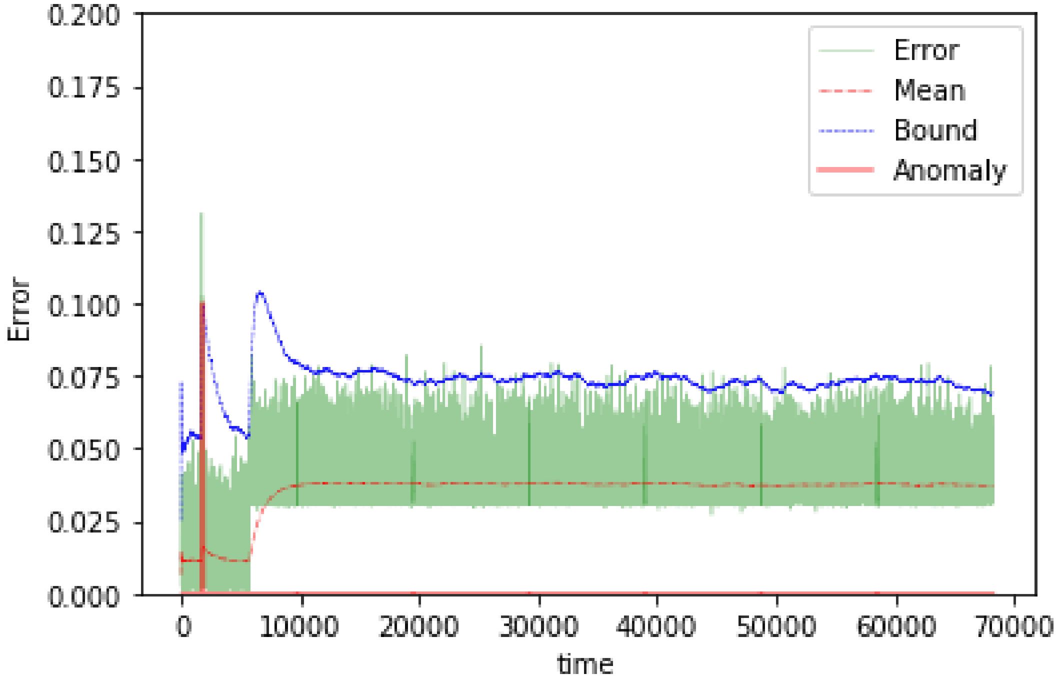

Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000


Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

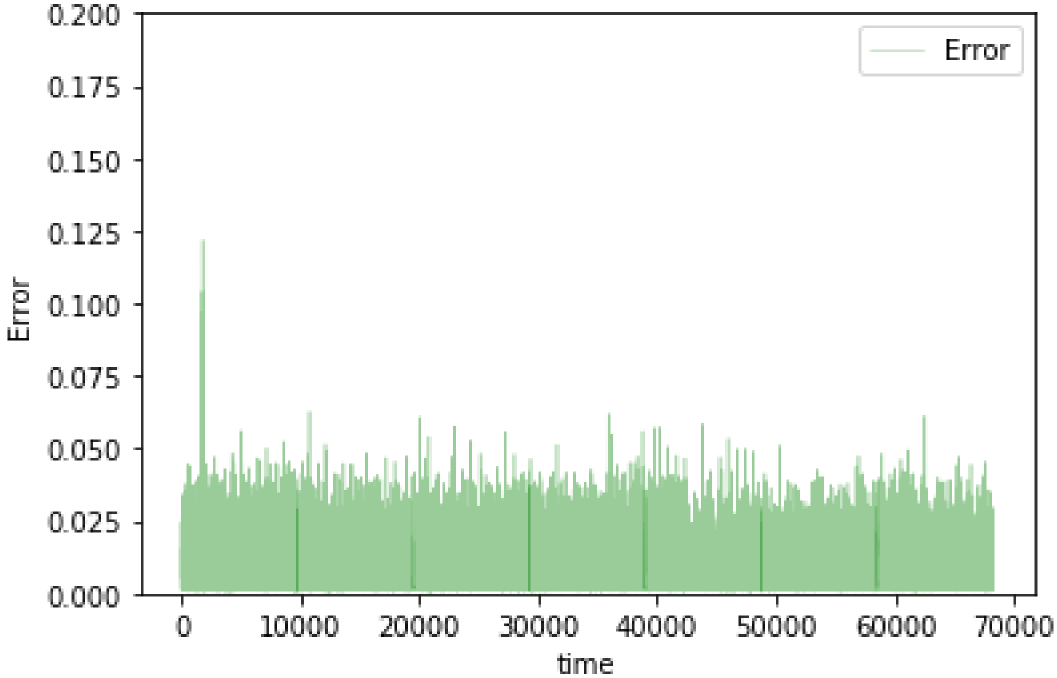
```
The mean error for lin20_avg_load_ is 0.0362672815182159 for length 68279
Testing on app change early data.
```

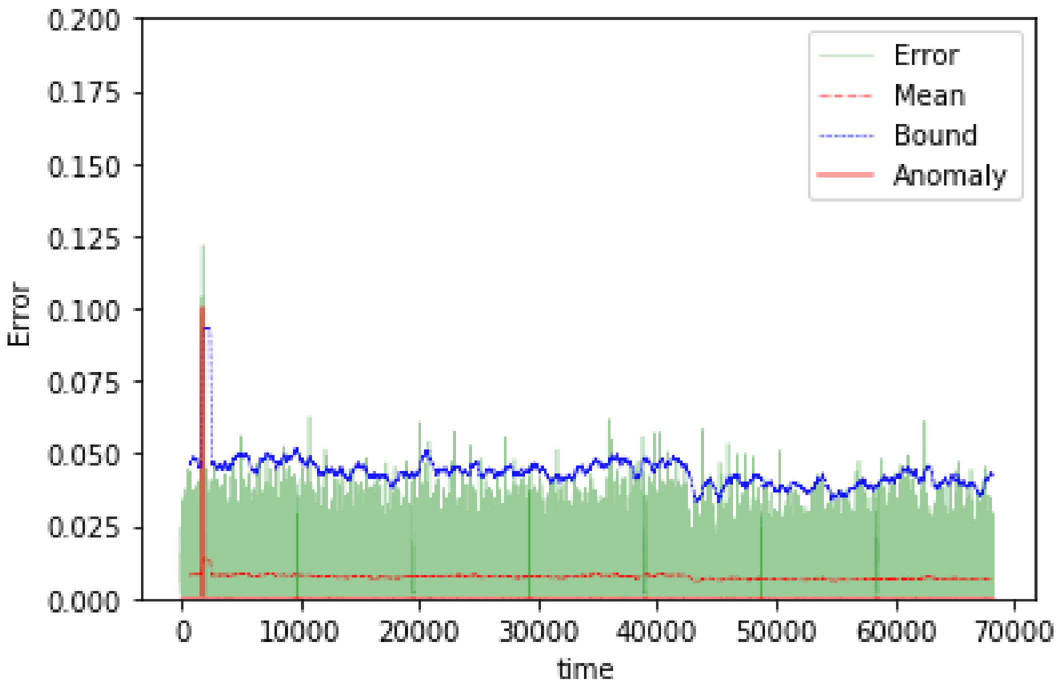


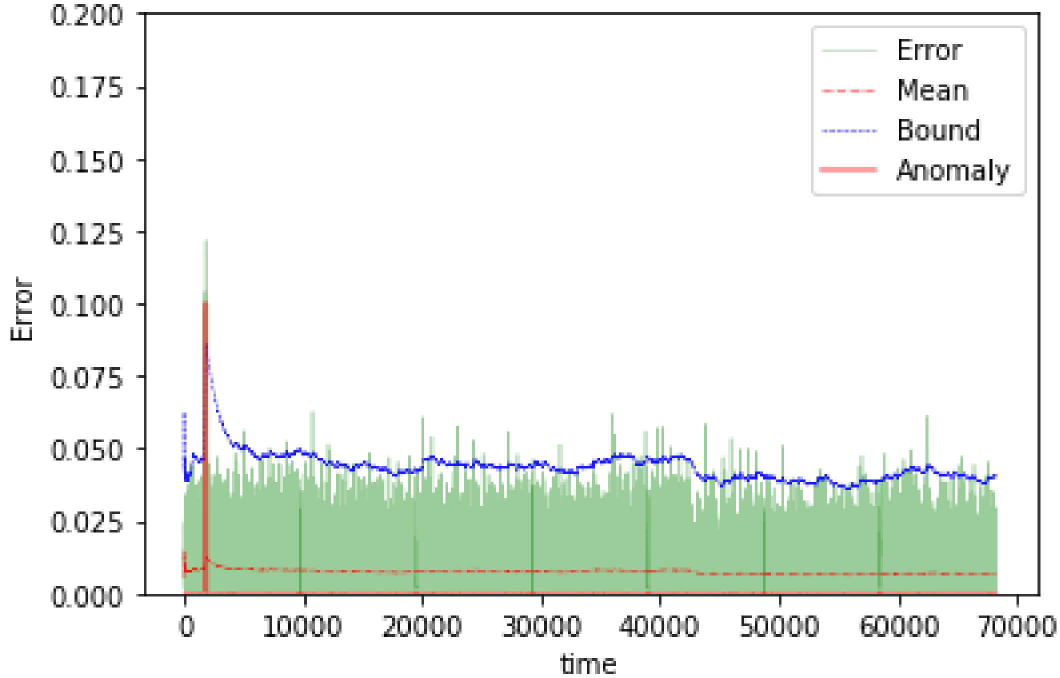





```
The mean error for lin20_app_change_early_ is 0.035553063447885364 for length 68279
Testing on Normal data.
```


Error
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

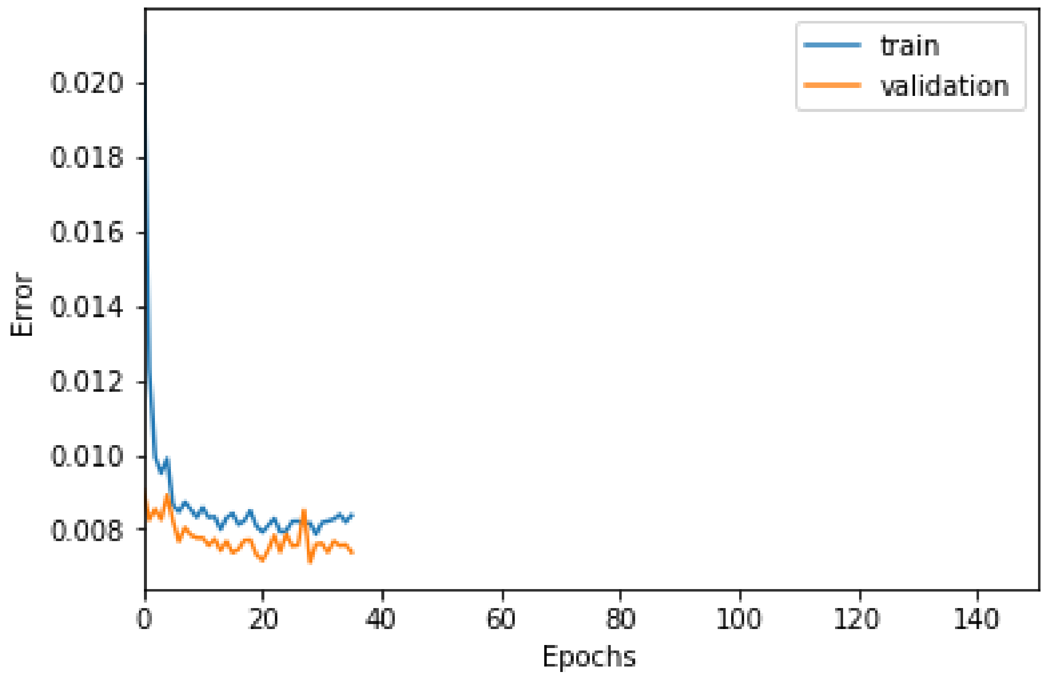

Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000



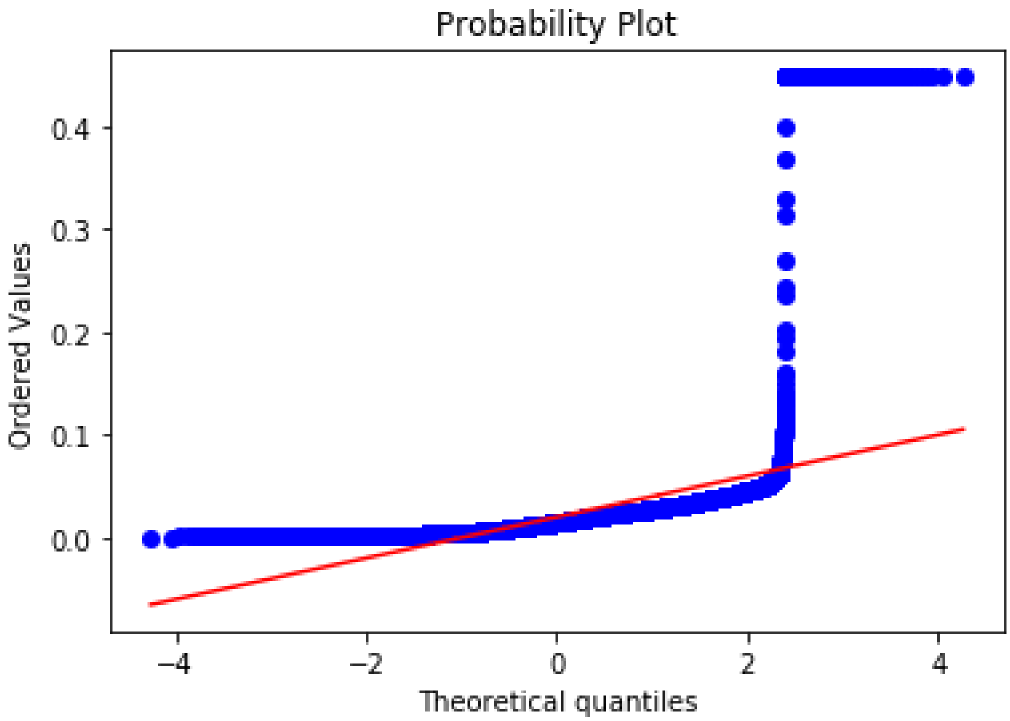

```
The mean error for lin20_normal_ is 0.0074365174910556485 for length 68279
====================
```

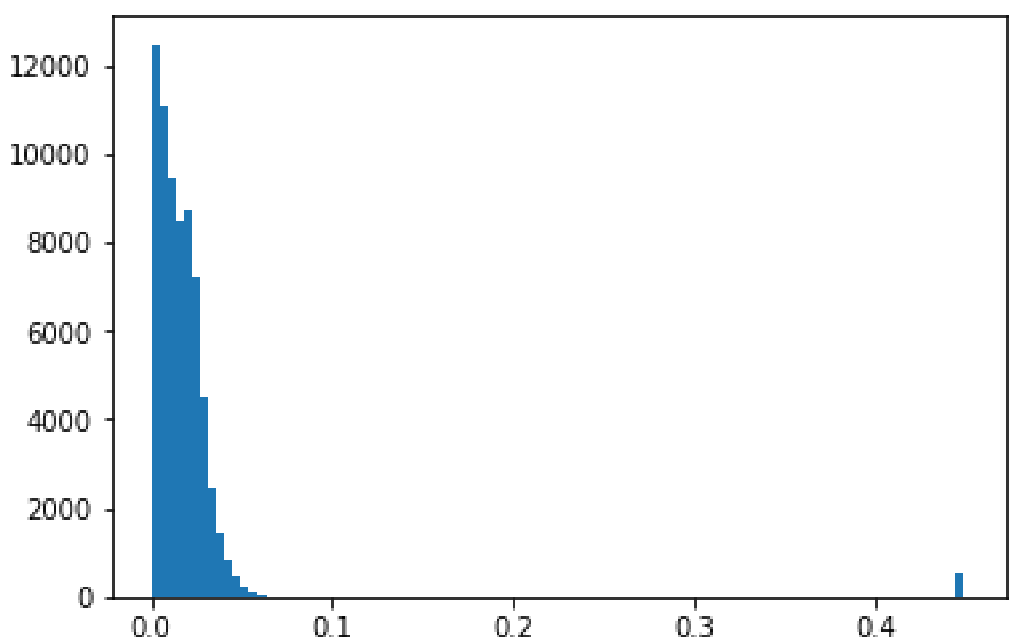
50 steps

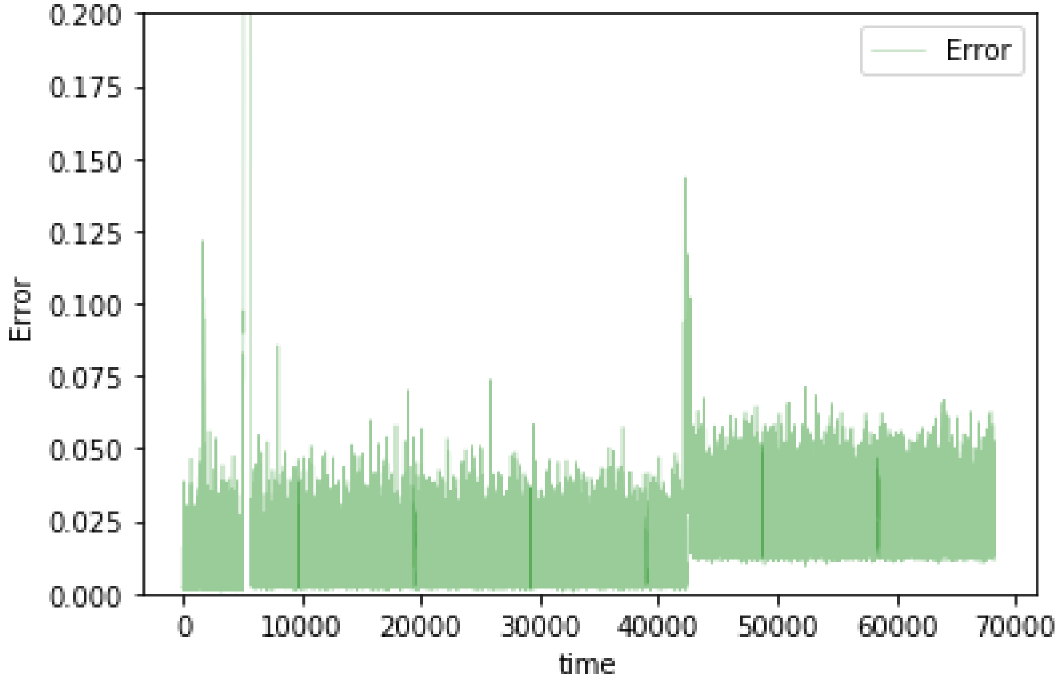
```
In [55]: TIMESTEPS = 50
         DIM = 29
         tgen = flat_generator(X, TIMESTEPS)
         vgen = flat_generator(val_X, TIMESTEPS)
         name = "lin50"

In [56]: input_layer = Input(shape=(TIMESTEPS*DIM,))
         output = Dense(DIM, activation='sigmoid')(input_layer)

In [57]: model = Model(input_layer, output)
         model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [58]: train(model, tgen, vgen, name=name)
         test(model, name=name, window=TIMESTEPS)
```



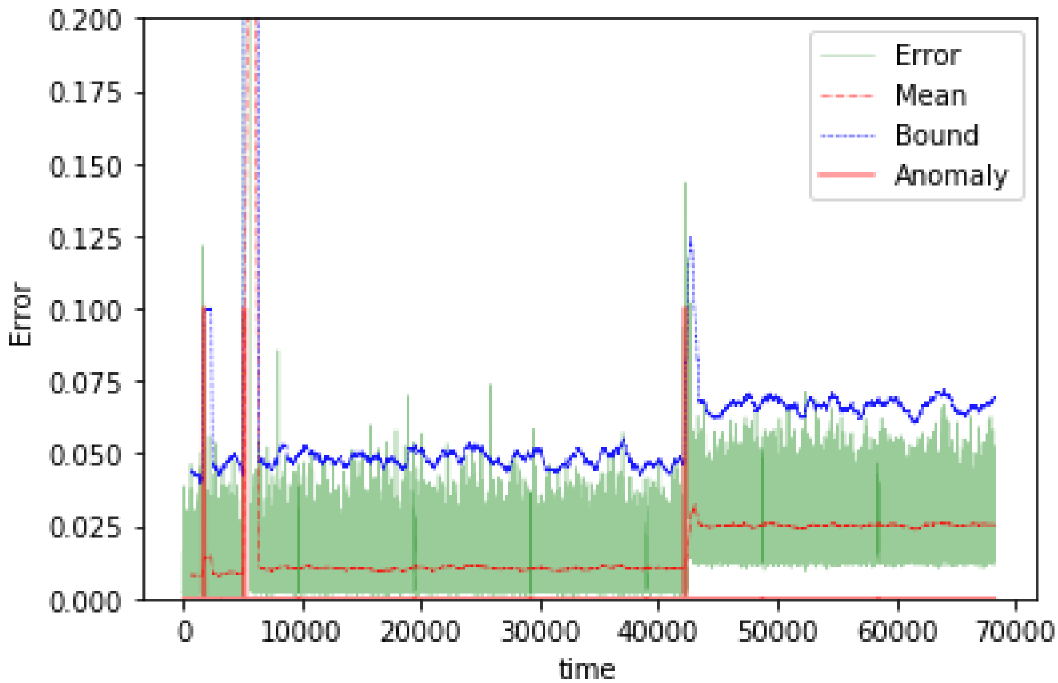

```
Training loss for final epoch is 0.008371333268121816
Validation loss for final epoch is 0.007378307749284431
---------- Beginning tests for lin50 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
0.4
0.3
0.2
0.1
0.0
−4
−2
0
2
4
Theoretical quantiles

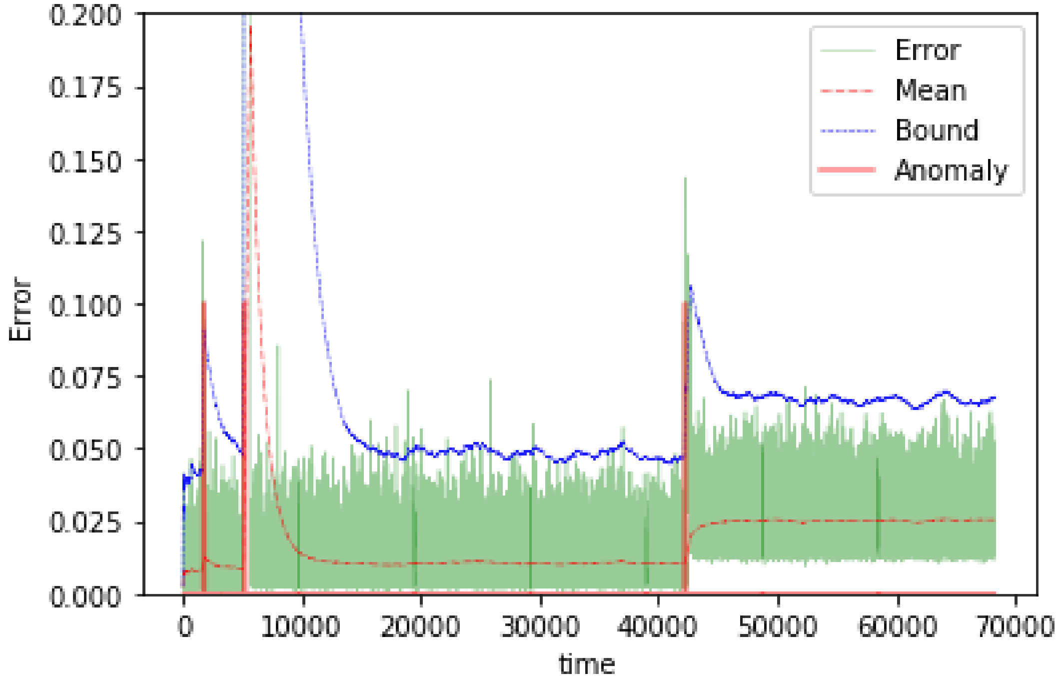

12000
10000
8000
6000
4000
2000
0
0.0
0.1
0.2
0.3
0.4



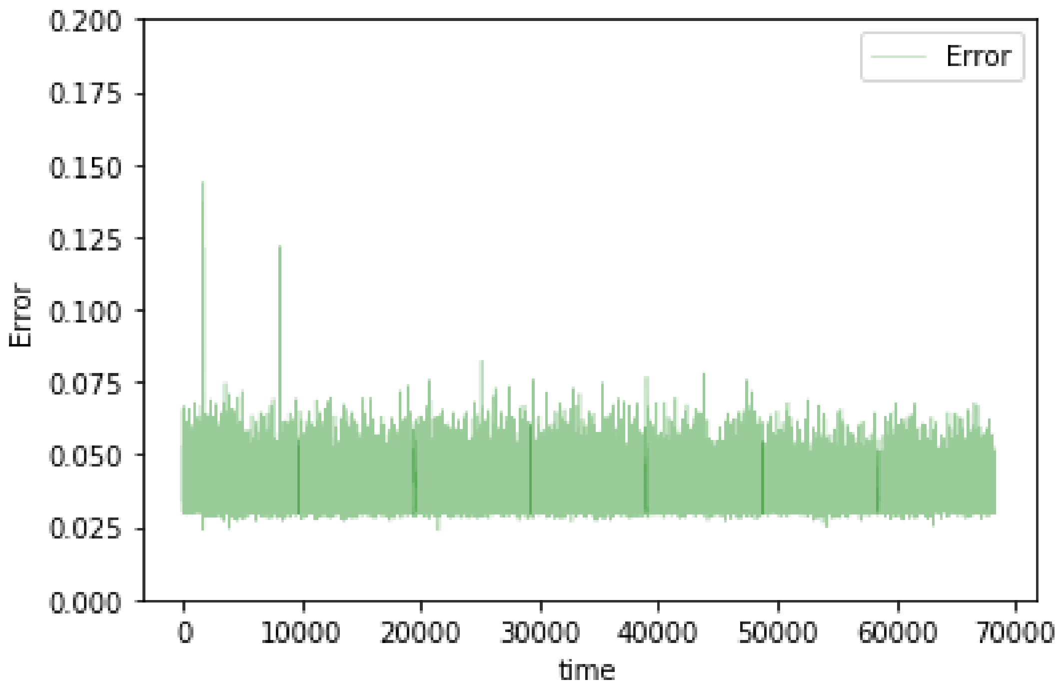
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

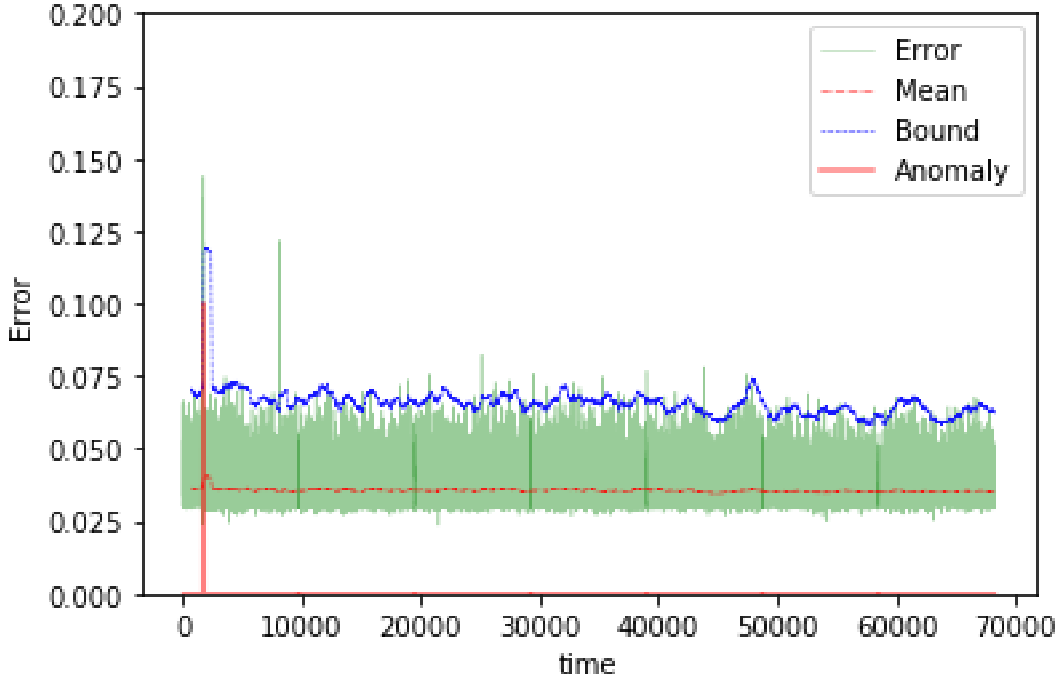


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

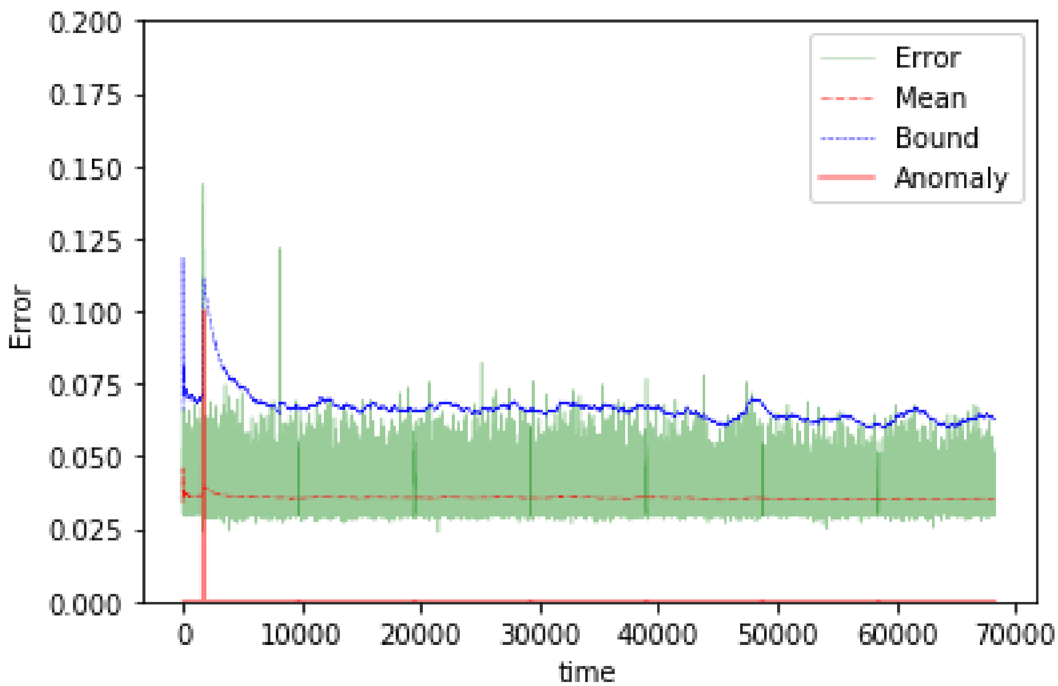



```
The mean error for lin50_disk_IO_start_ is 0.019730904649480466 for length 68249
Testing on Avg. load data.
```



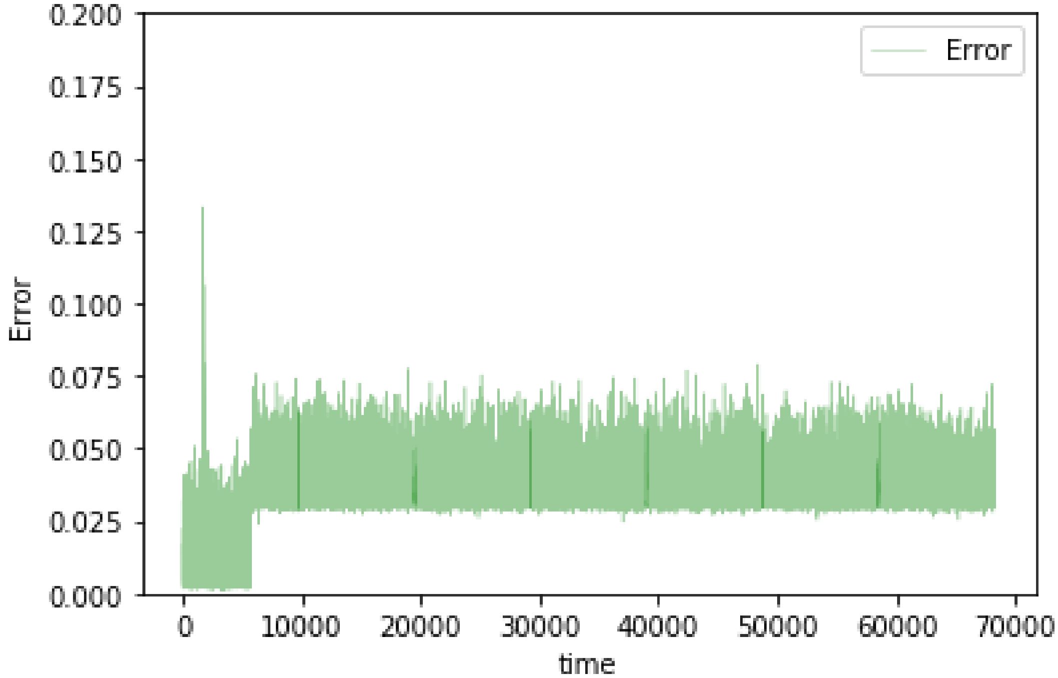


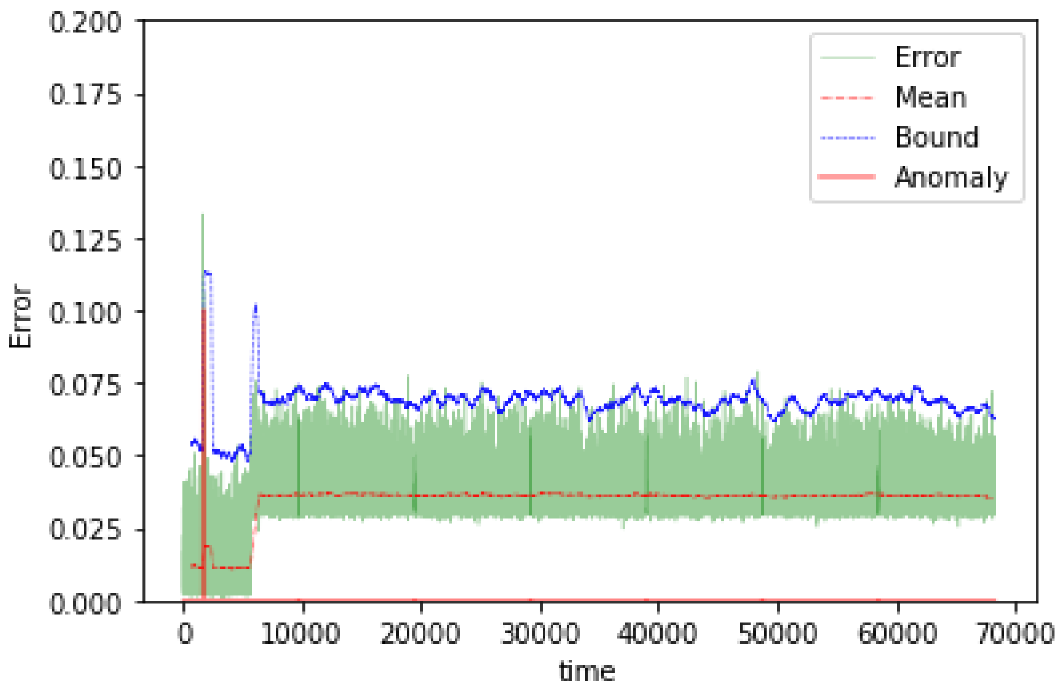
Error
Mean
Bound
Anomaly
Error
time

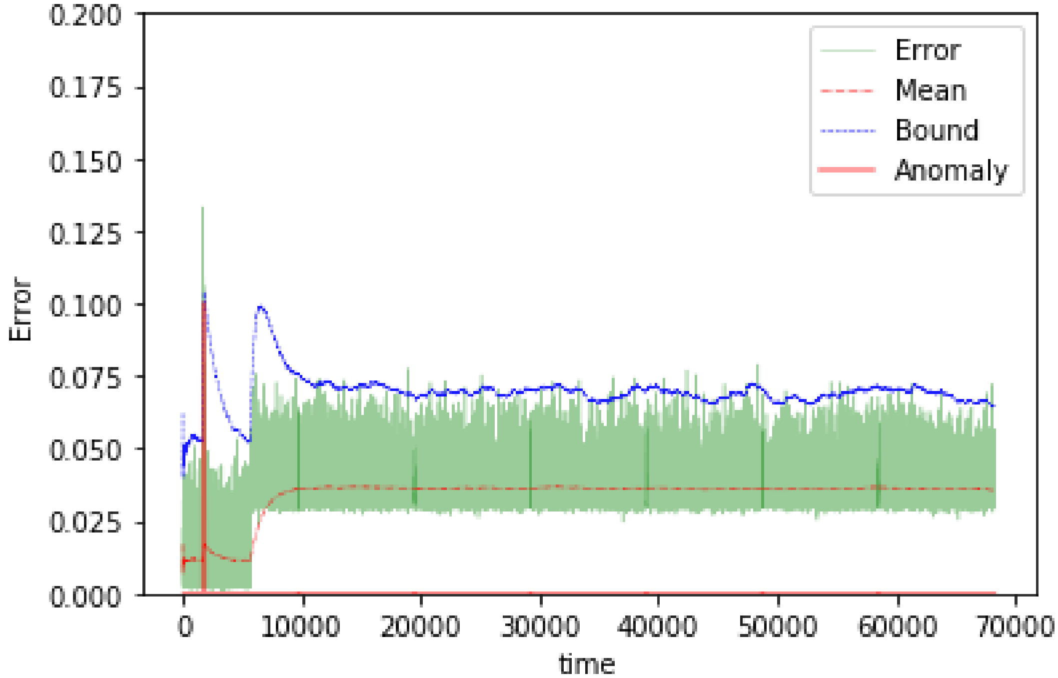


Error
Mean
Bound
Anomaly
Error
time

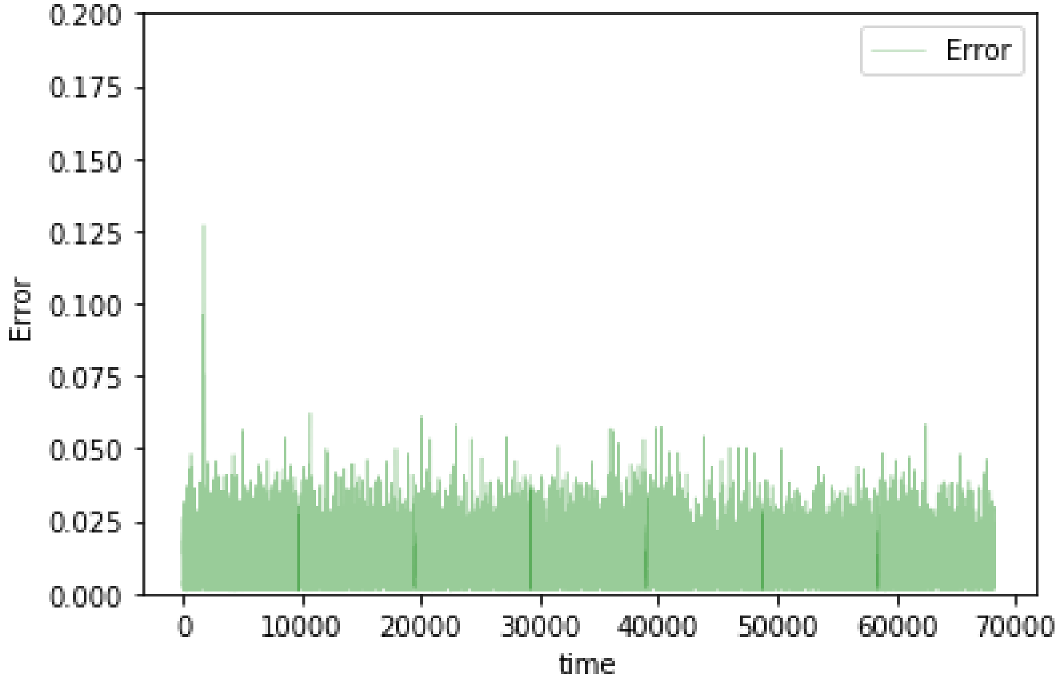
```
The mean error for lin50_avg_load_ is 0.03569381854670514 for length 68249
Testing on app change early data.
```

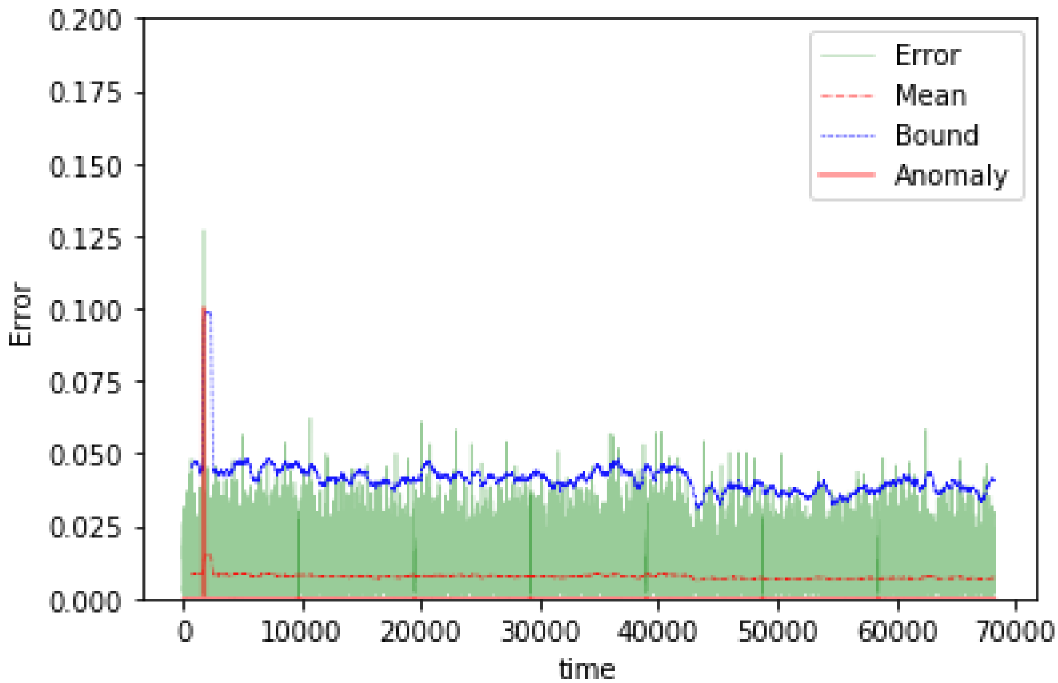


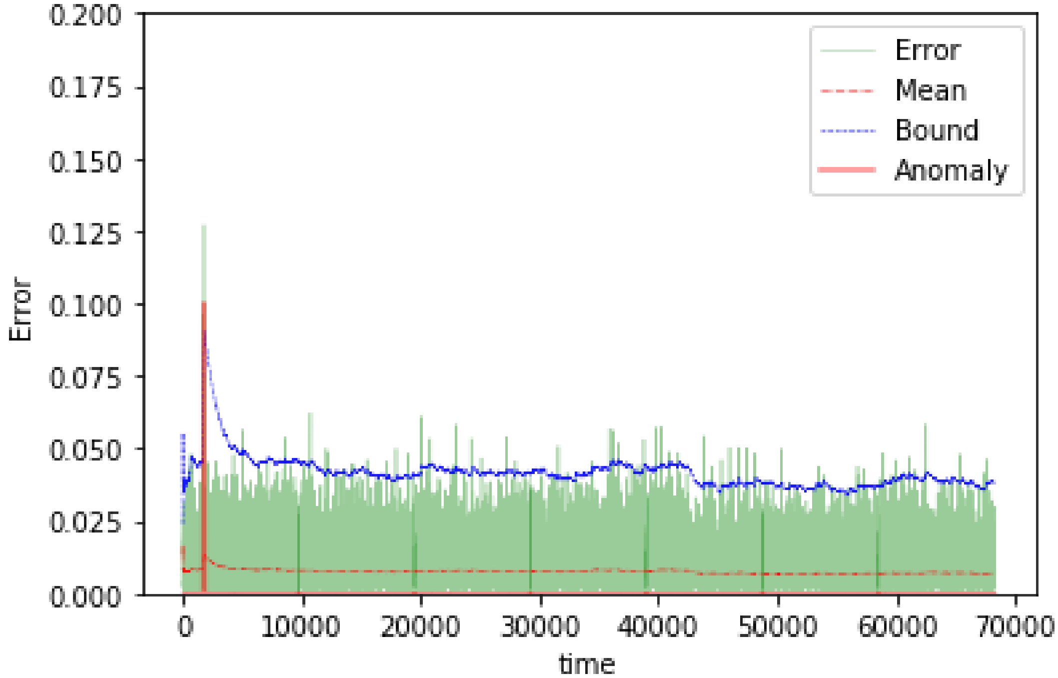





```
The mean error for lin50_app_change_early_ is 0.034286619888392565 for length 68249
Testing on Normal data.
```


Error
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

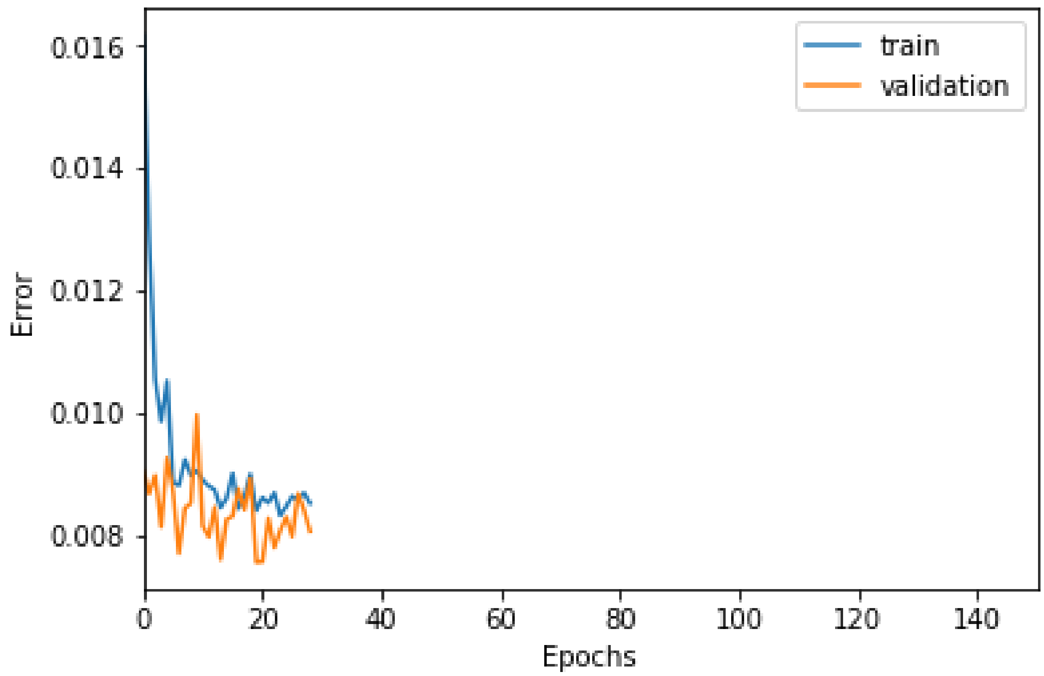

Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000



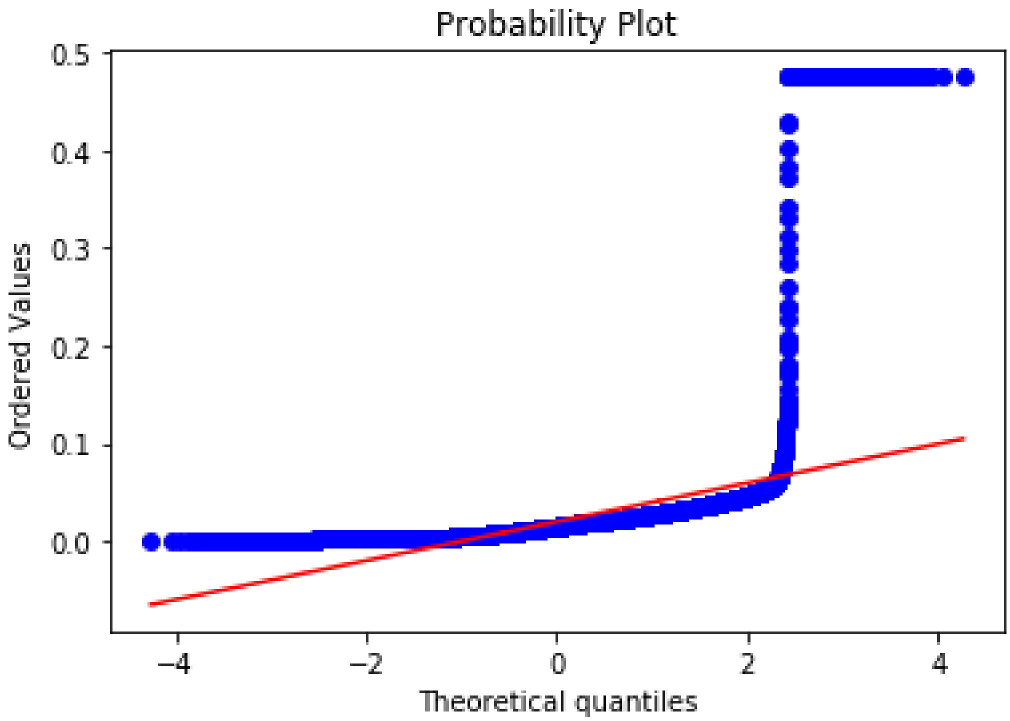

```
The mean error for lin50_normal_ is 0.007567797619566114 for length 68249
====================
```

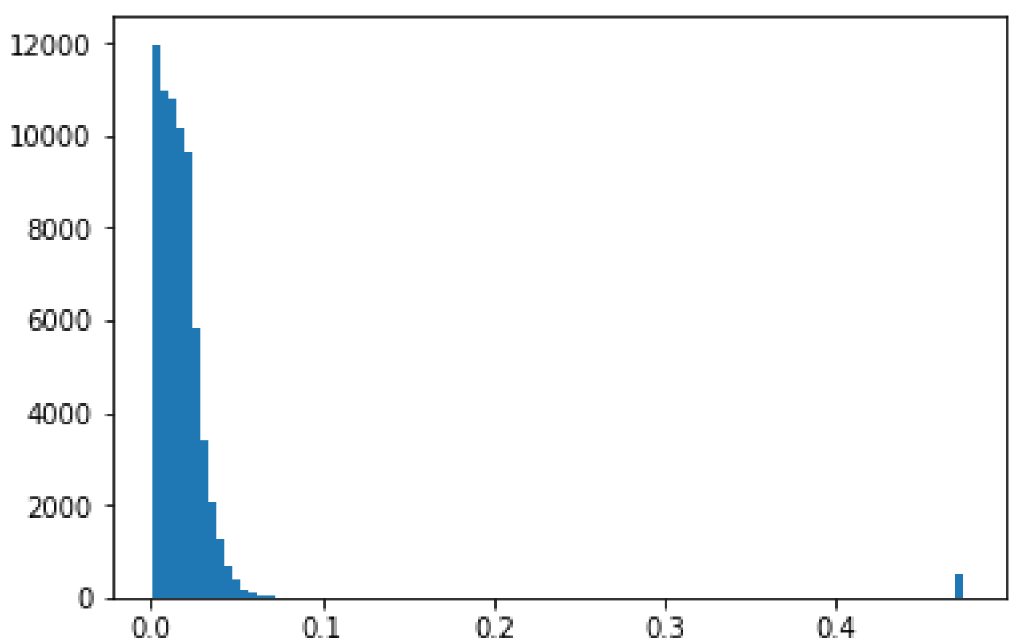
100 steps

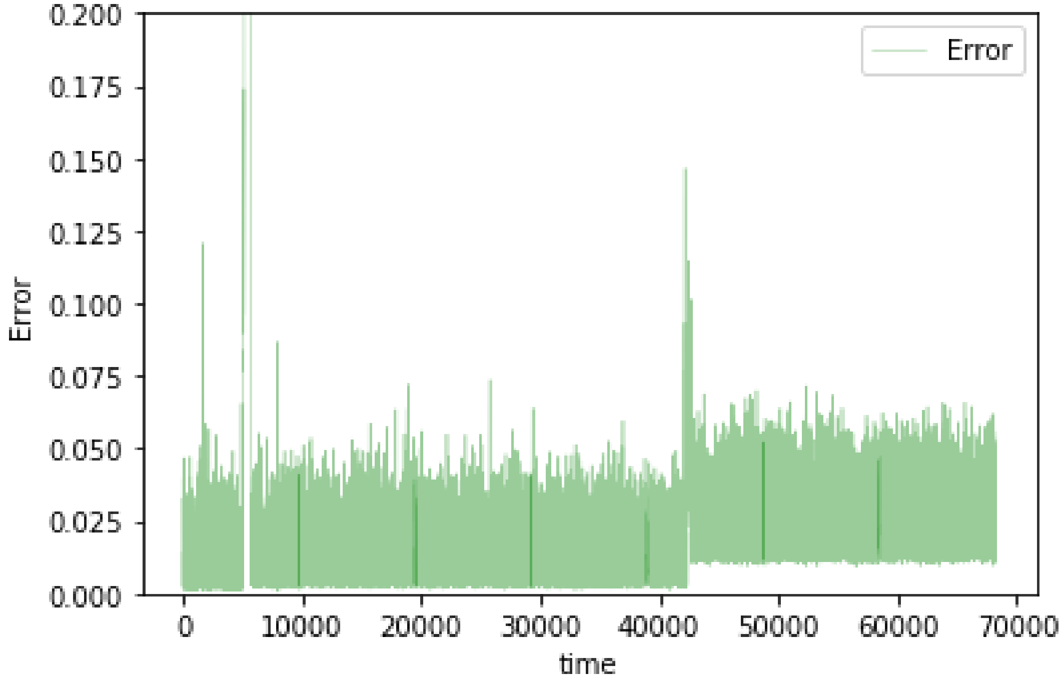
```
In [59]: TIMESTEPS = 100
         DIM = 29
         tgen = flat_generator(X, TIMESTEPS)
         vgen = flat_generator(val_X, TIMESTEPS)
         name = "lin100"

In [60]: input_layer = Input(shape=(TIMESTEPS*DIM,))
         output = Dense(DIM, activation='sigmoid')(input_layer)

In [61]: model = Model(input_layer, output)
         model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [62]: train(model, tgen, vgen, name=name)
         test(model, name=name, window=TIMESTEPS)
```



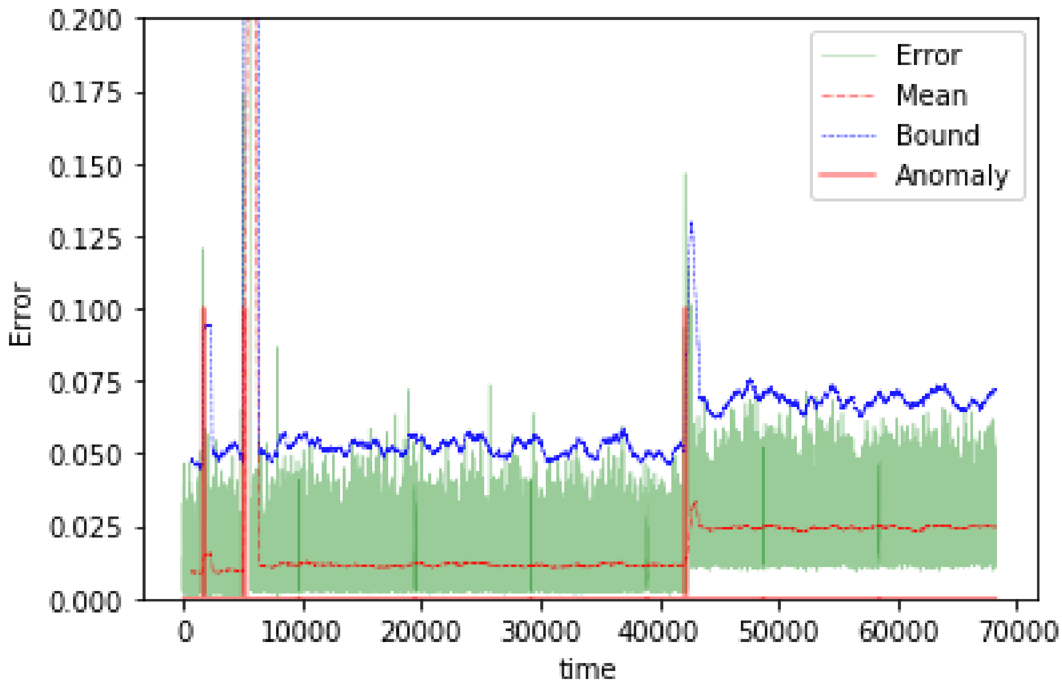

```
Training loss for final epoch is 0.00852404024277348
Validation loss for final epoch is 0.008063269310630857
---------- Beginning tests for lin100 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.5
0.4
0.3
0.2
0.1
0.0
−4
−2
0
2
4

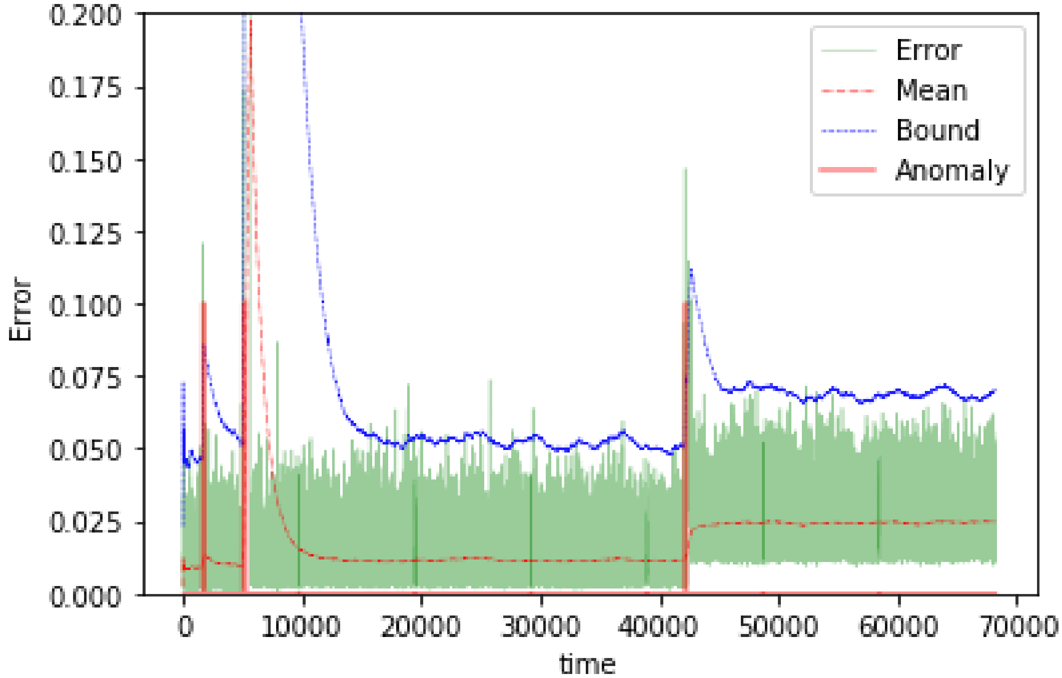

12000
10000
8000
6000
4000
2000
0
0.0
0.1
0.2
0.3
0.4



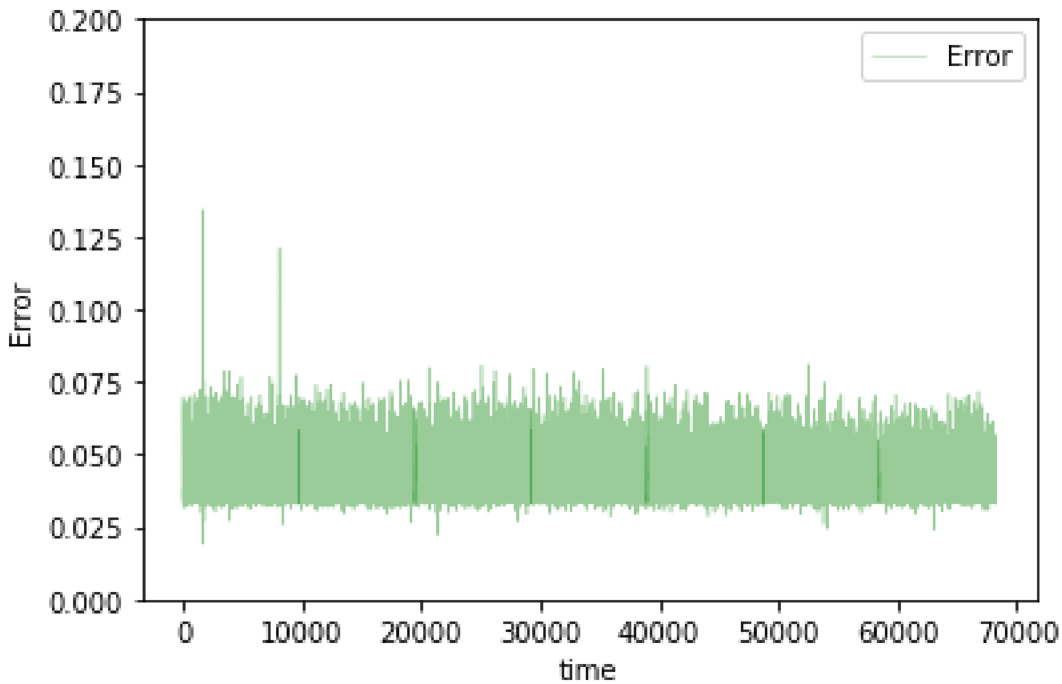
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

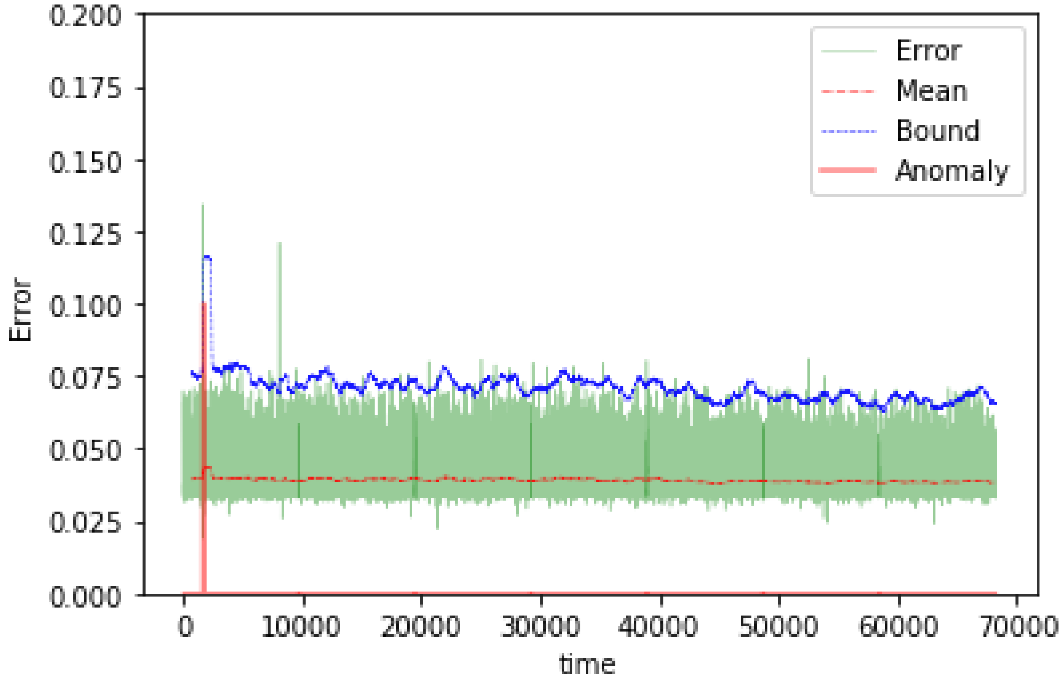


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

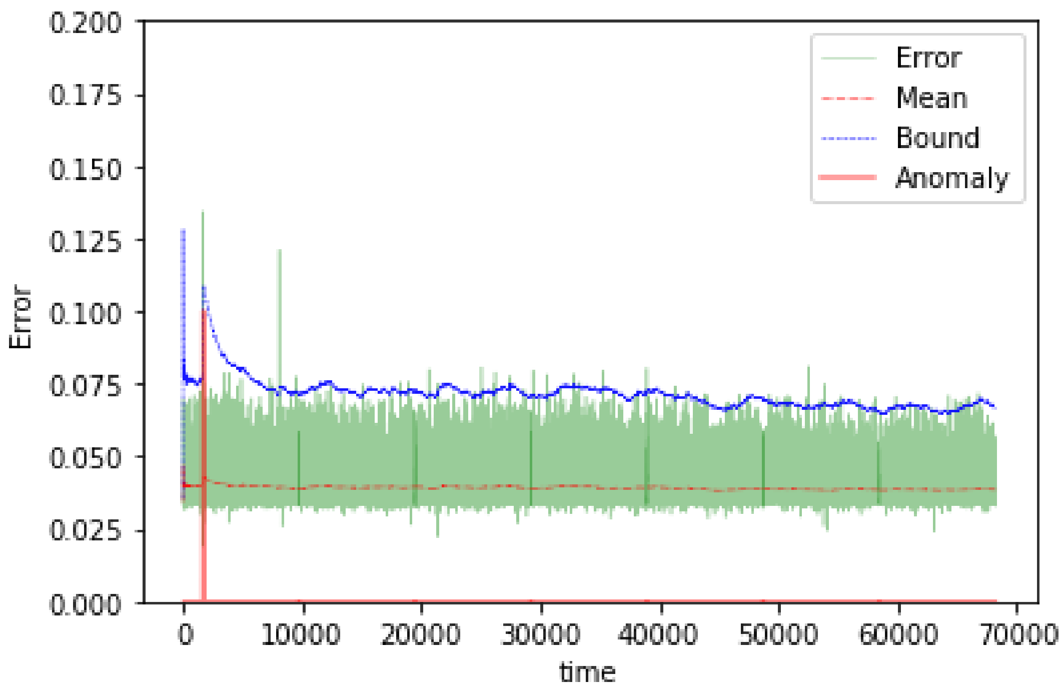


The mean error for lin100_disk_IO_start_ is 0.020098491686056756 for length 68199
Testing on Avg. load data.



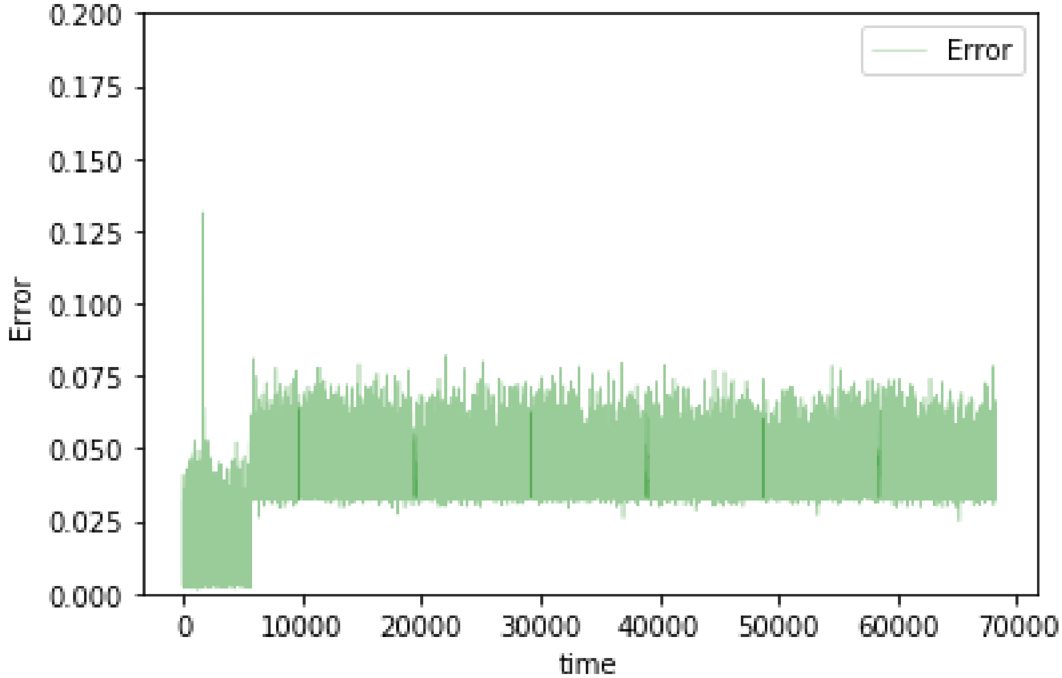


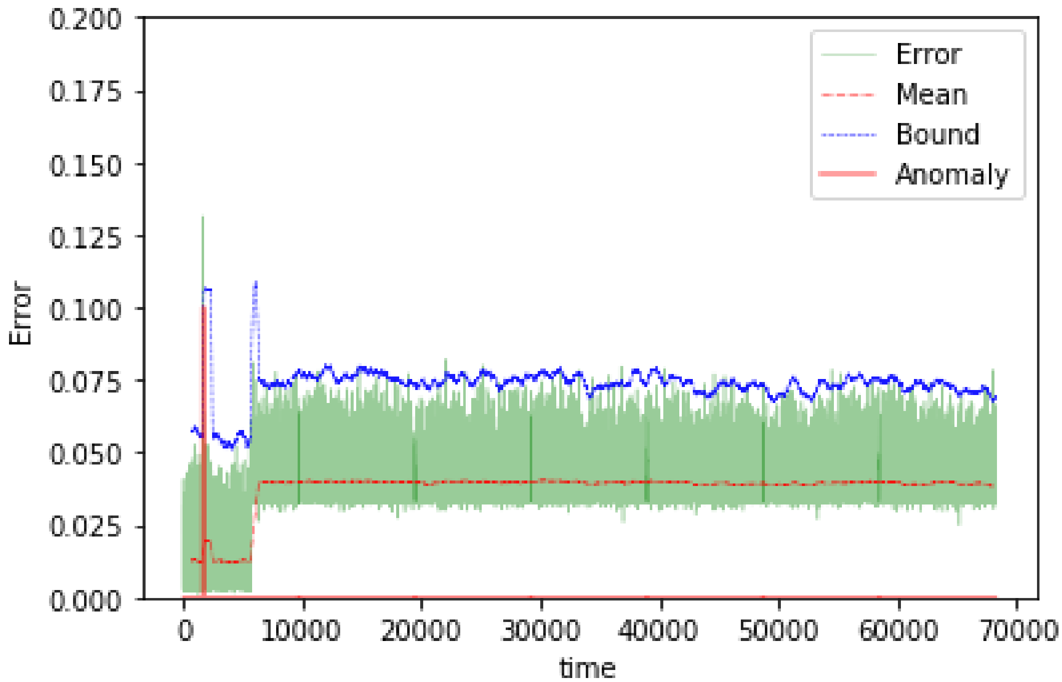
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

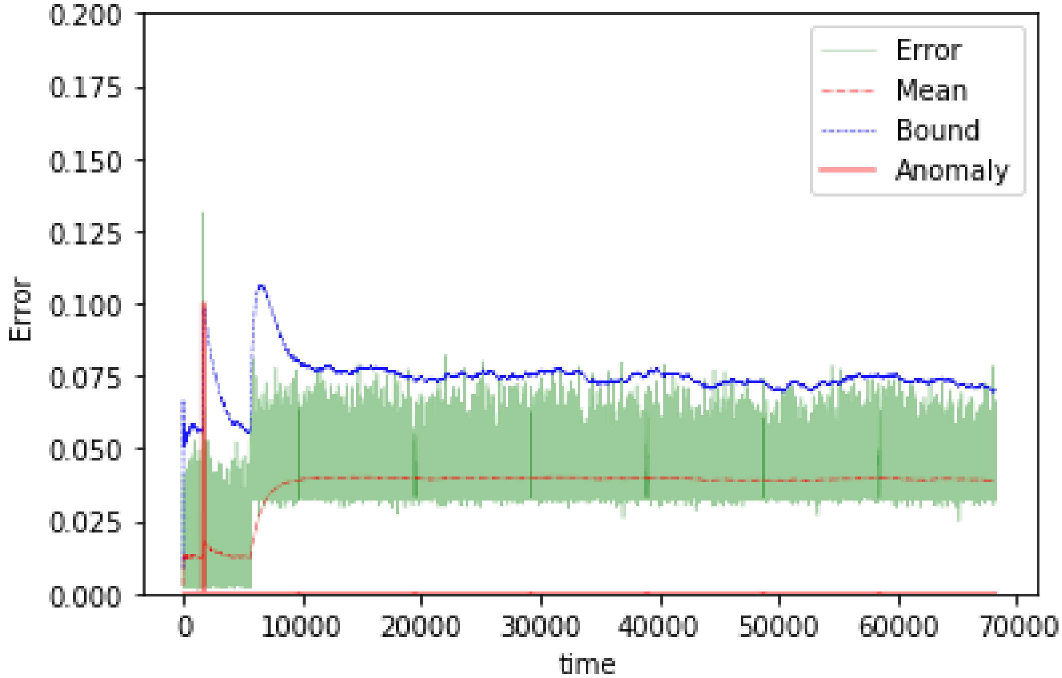


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

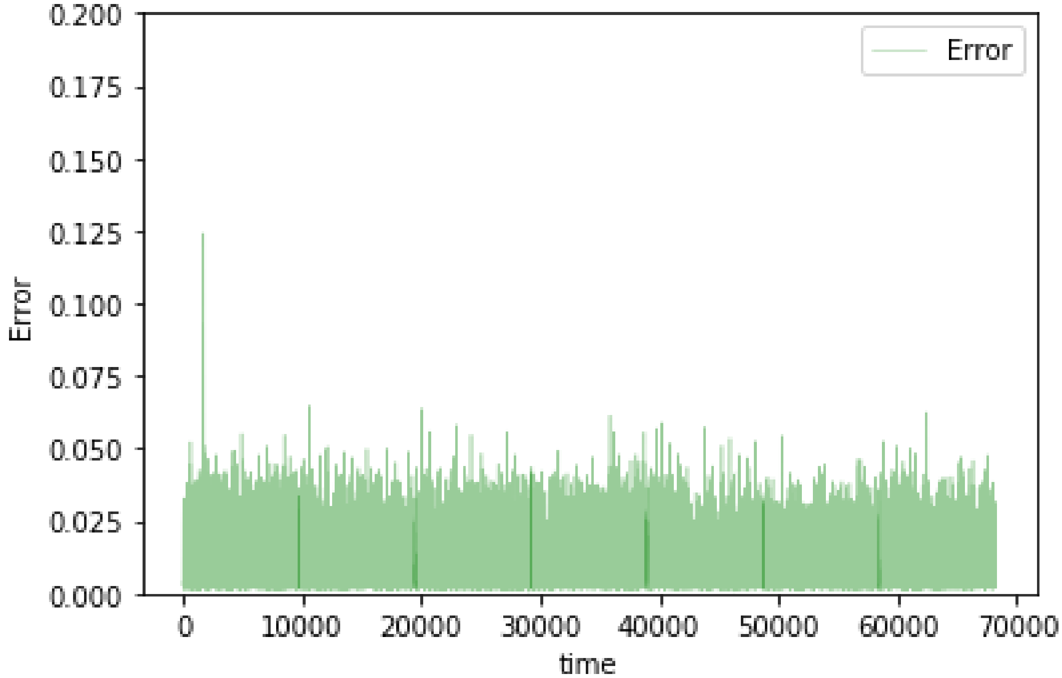
```
The mean error for lin100_avg_load_ is 0.03913561276849554 for length 68199
Testing on app change early data.
```

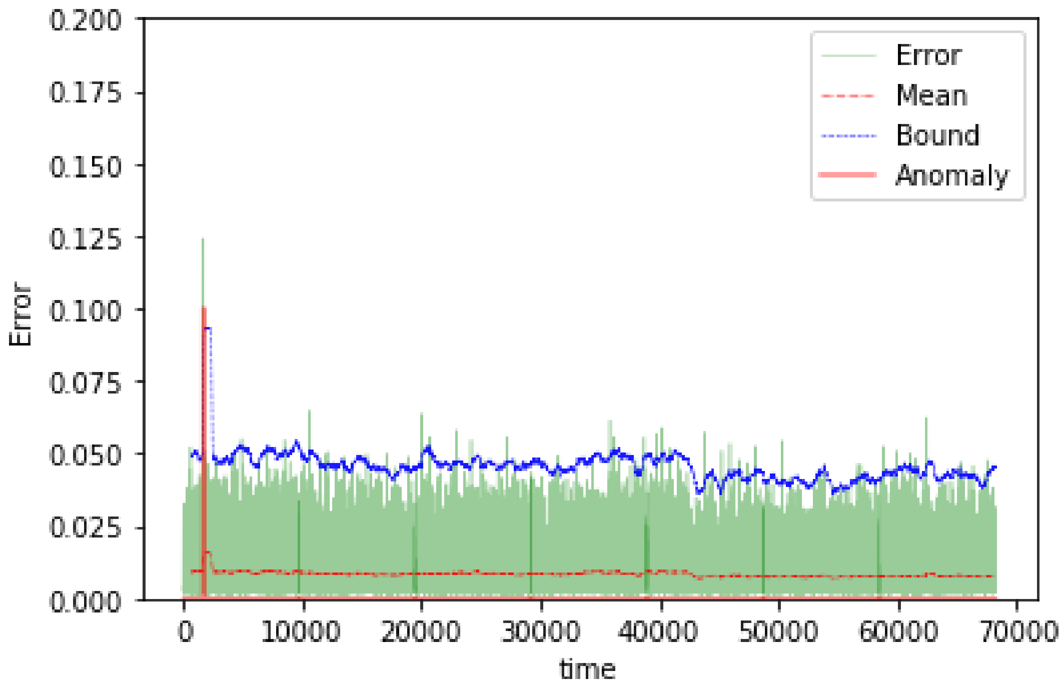


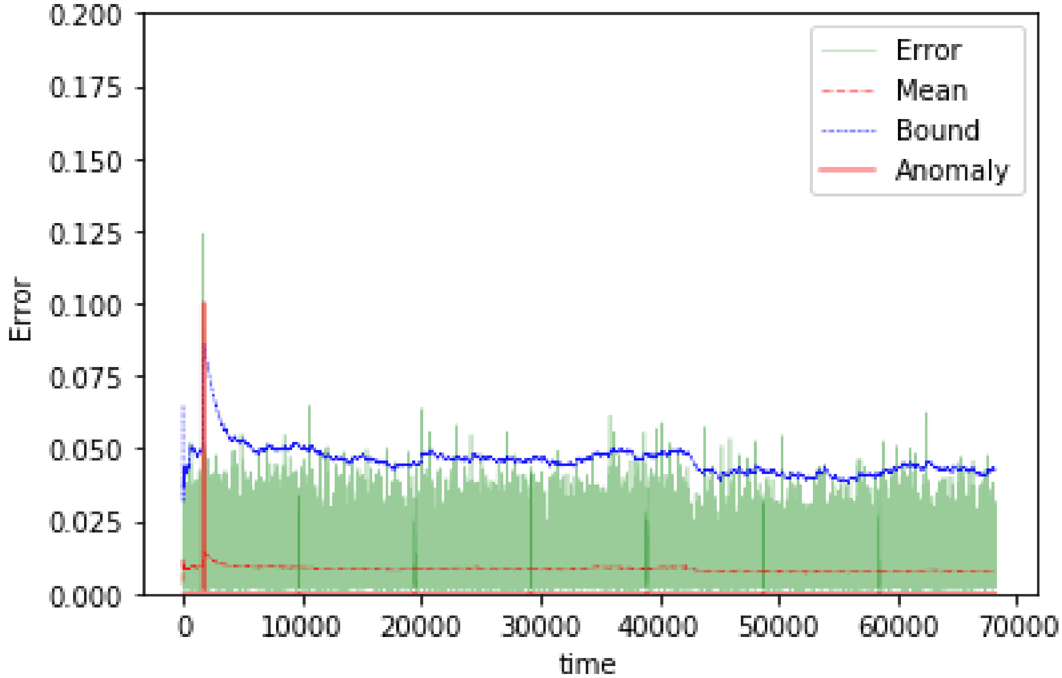





```
The mean error for lin100_app_change_early_ is 0.037432093117054914 for length 68199
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

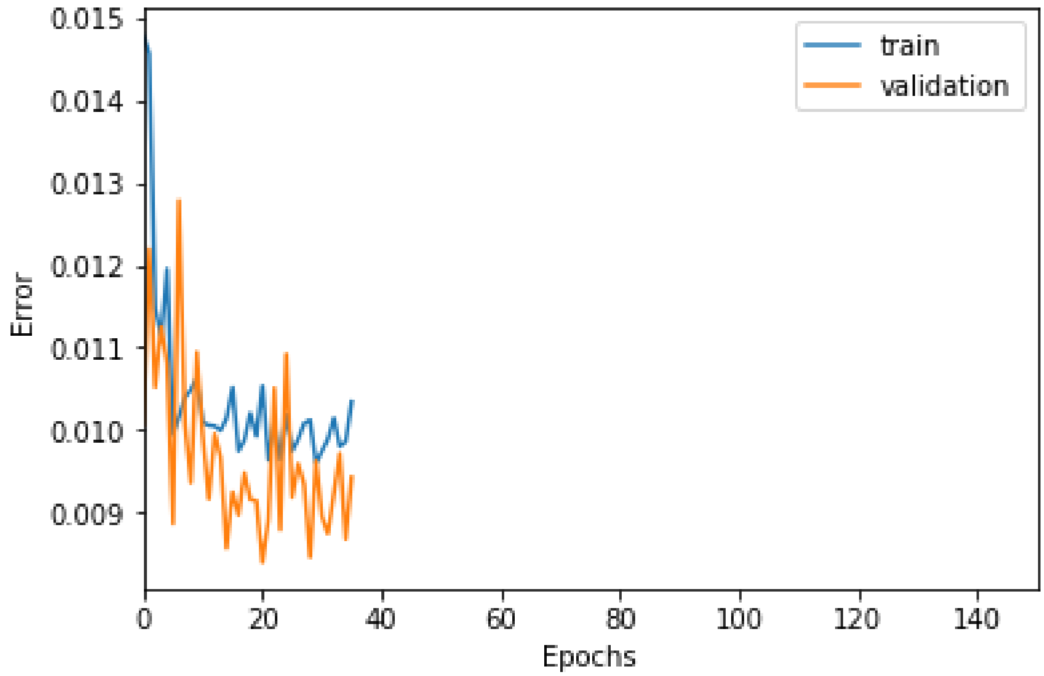

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly



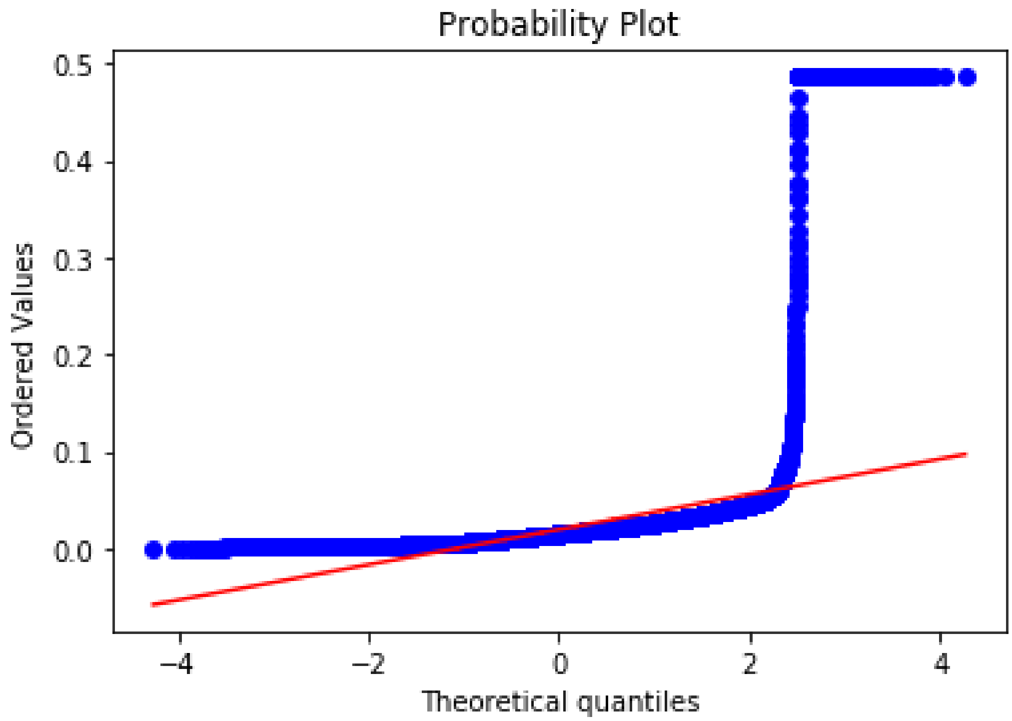

```
The mean error for lin100_normal_ is 0.008396999142431824 for length 68199
====================
```

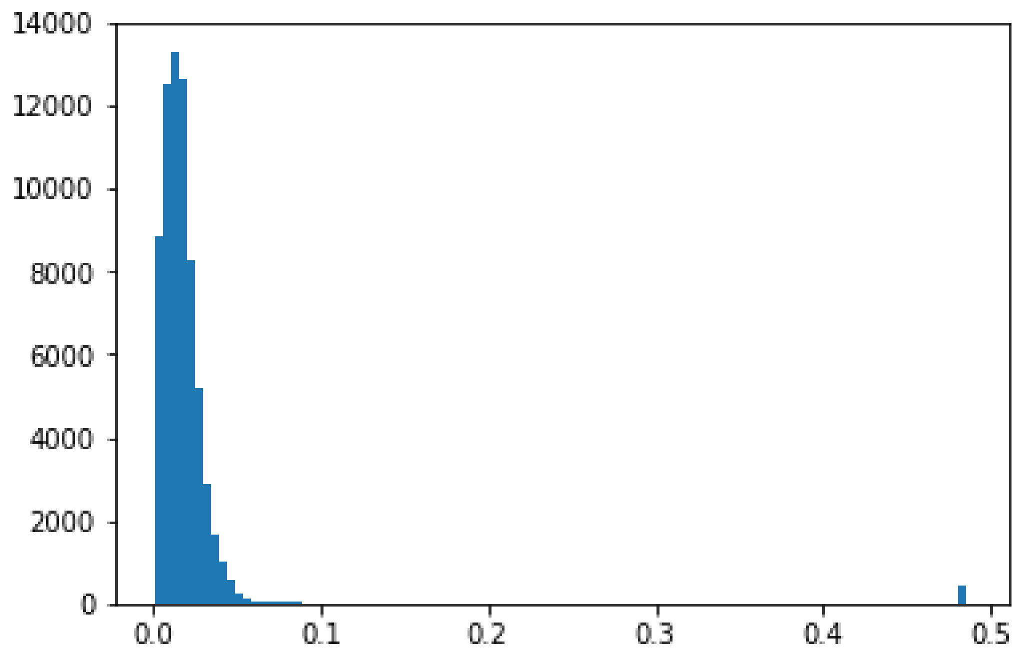
200 steps

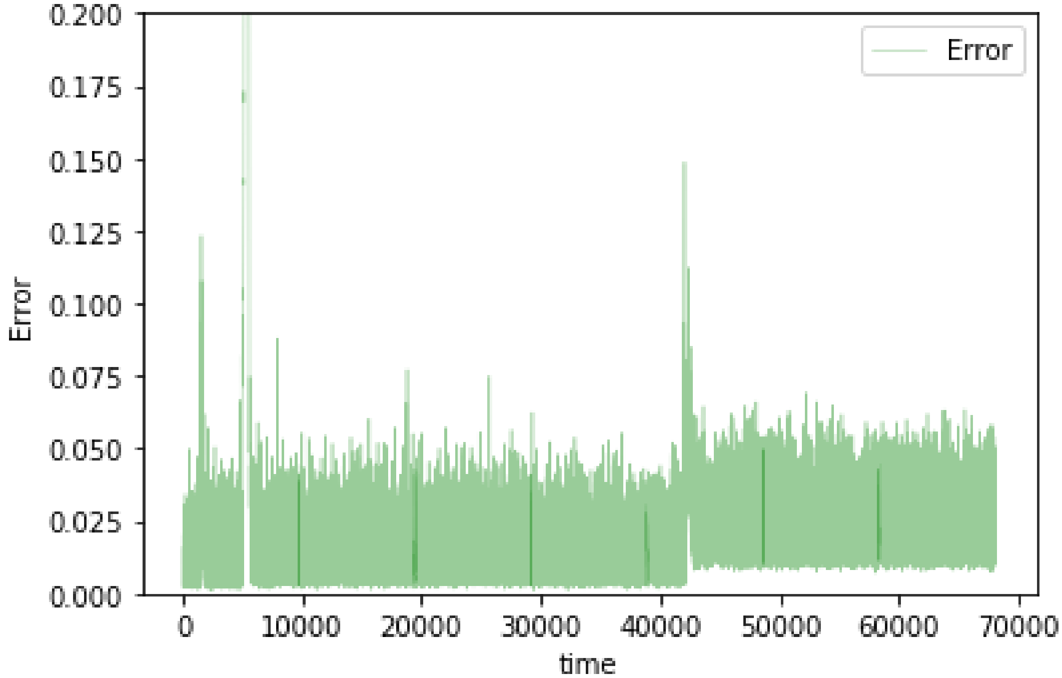
```
In [63]: TIMESTEPS = 200
         DIM = 29
         tgen = flat_generator(X, TIMESTEPS)
         vgen = flat_generator(val_X, TIMESTEPS)
         name = "lin200"

In [64]: input_layer = Input(shape=(TIMESTEPS*DIM,))
         output = Dense(DIM, activation='sigmoid')(input_layer)

In [65]: model = Model(input_layer, output)
         model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [66]: train(model, tgen, vgen, name=name)
         test(model, name=name, window=TIMESTEPS)
```

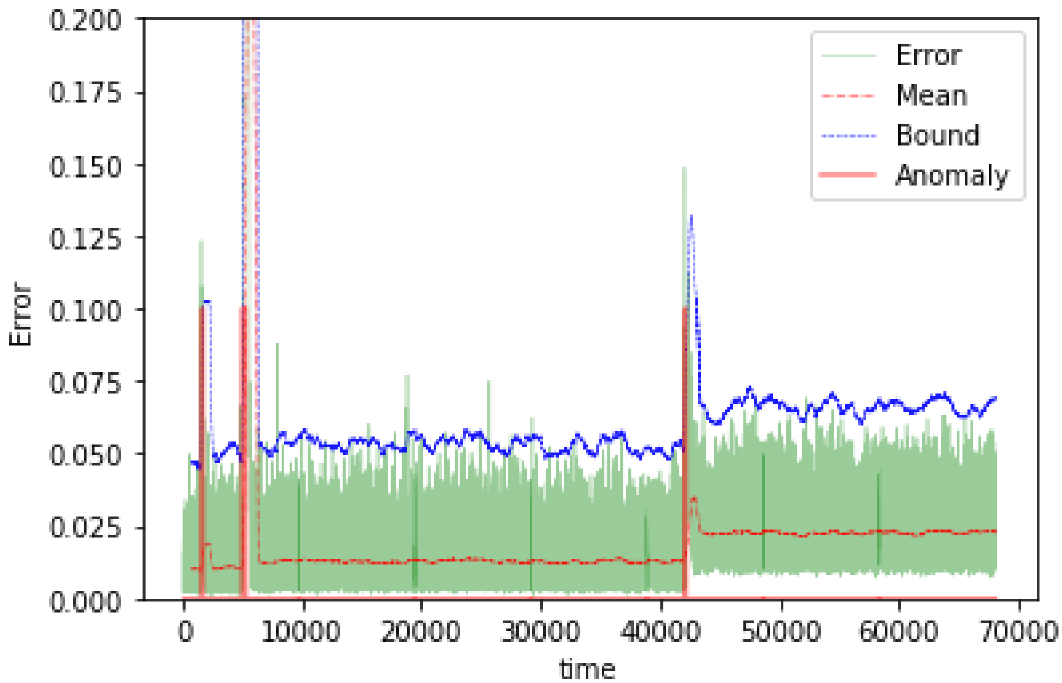


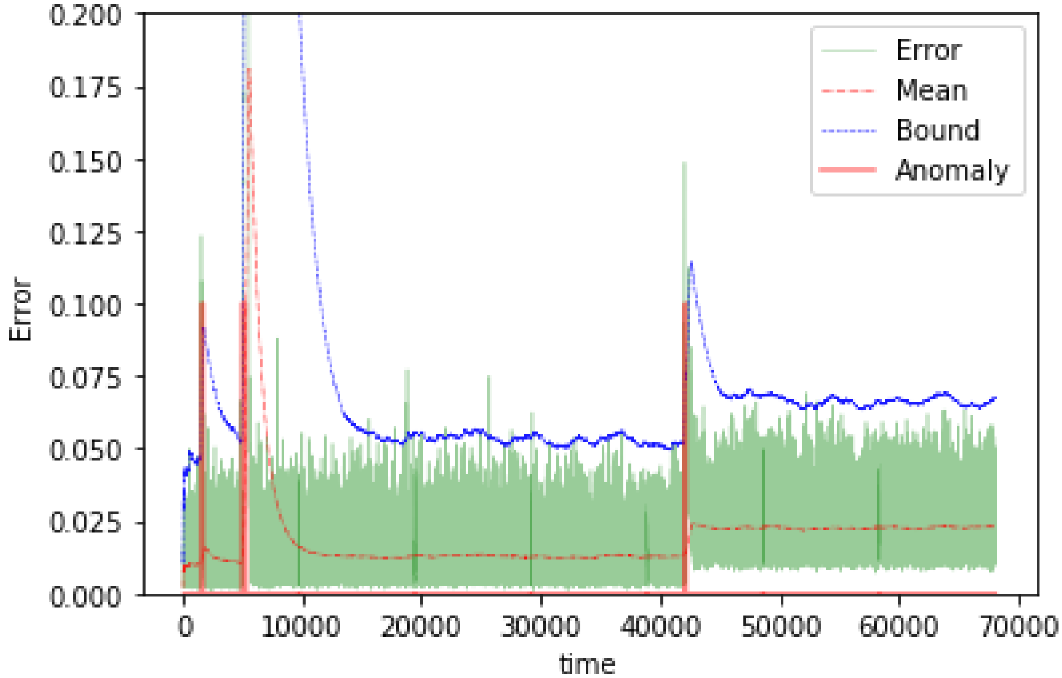

```
Training loss for final epoch is 0.010344892457593232
Validation loss for final epoch is 0.00943154941406101
---------- Beginning tests for lin200 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.5
0.4
0.3
0.2
0.1
0.0
−4
−2
0
2
4

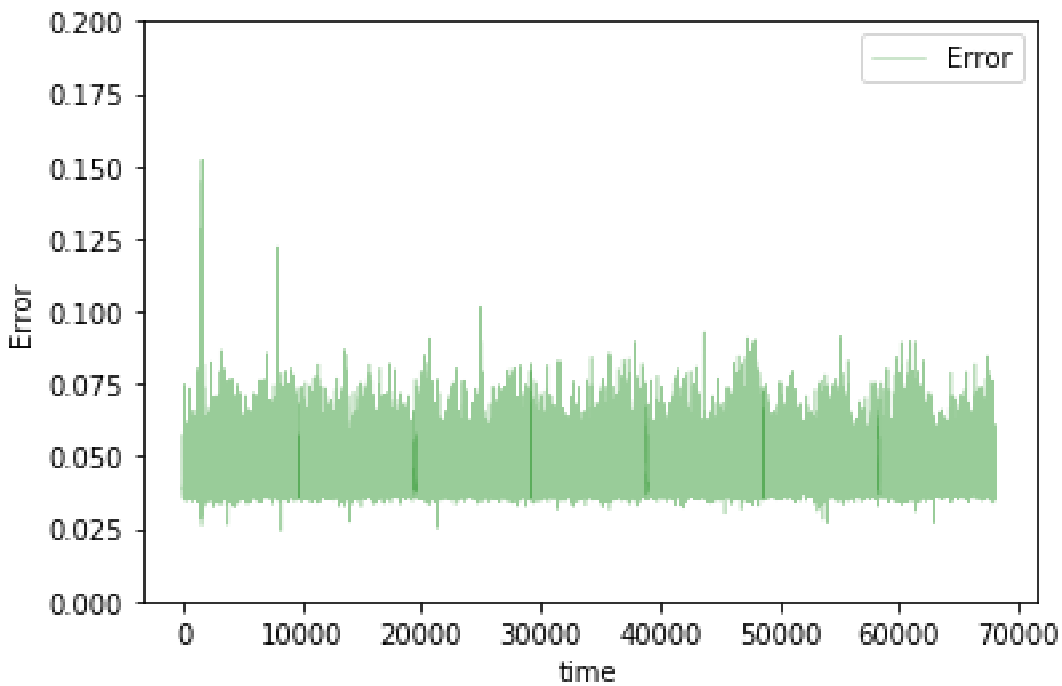

14000
12000
10000
8000
6000
4000
2000
0
0.0
0.1
0.2
0.3
0.4
0.5

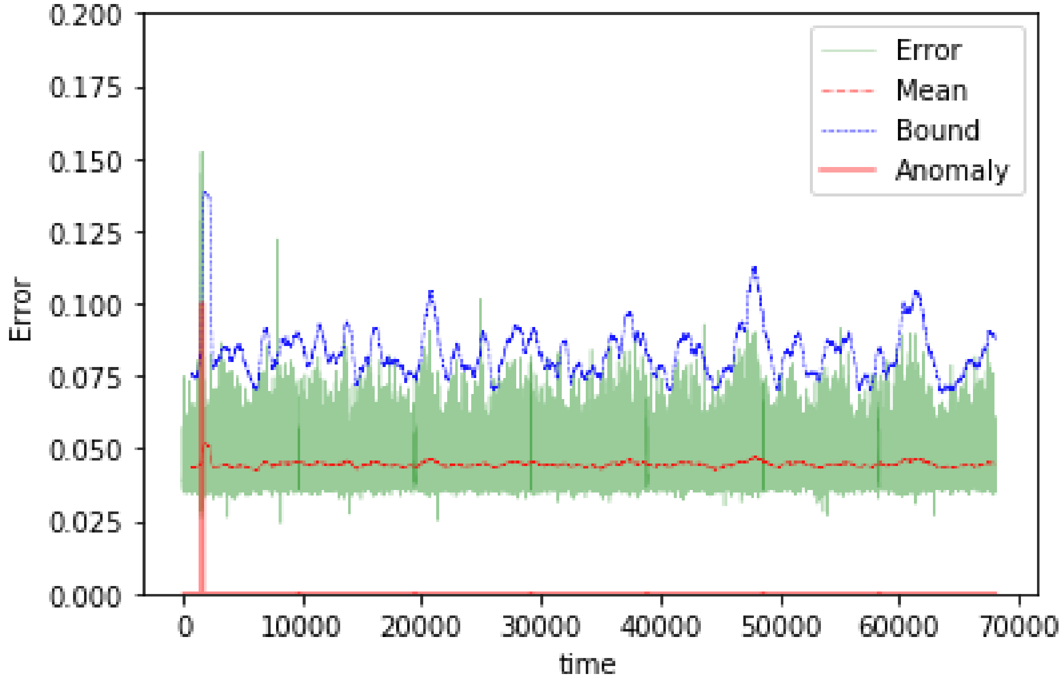

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

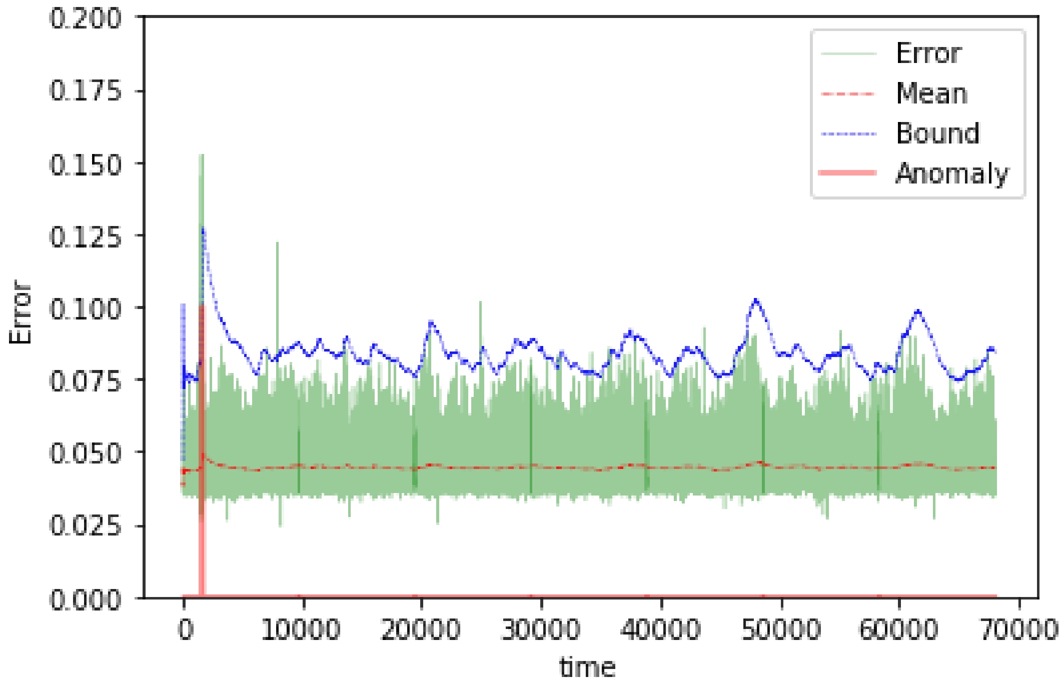

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



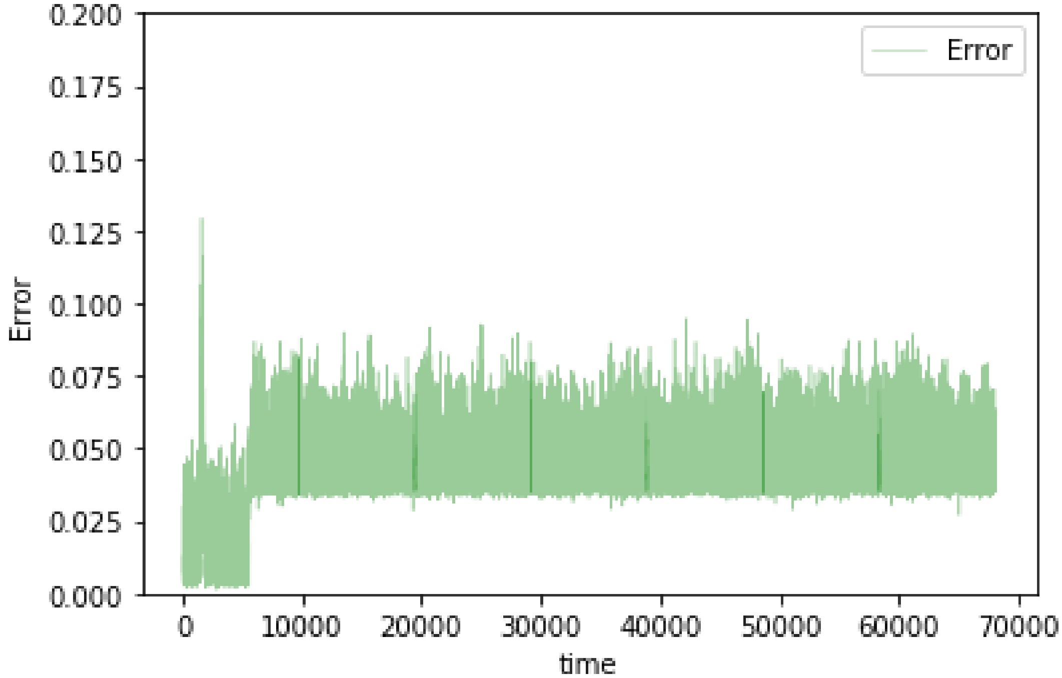

```
The mean error for lin200_disk_IO_start_ is 0.019885709737504444 for length 68099
Testing on Avg. load data.
```

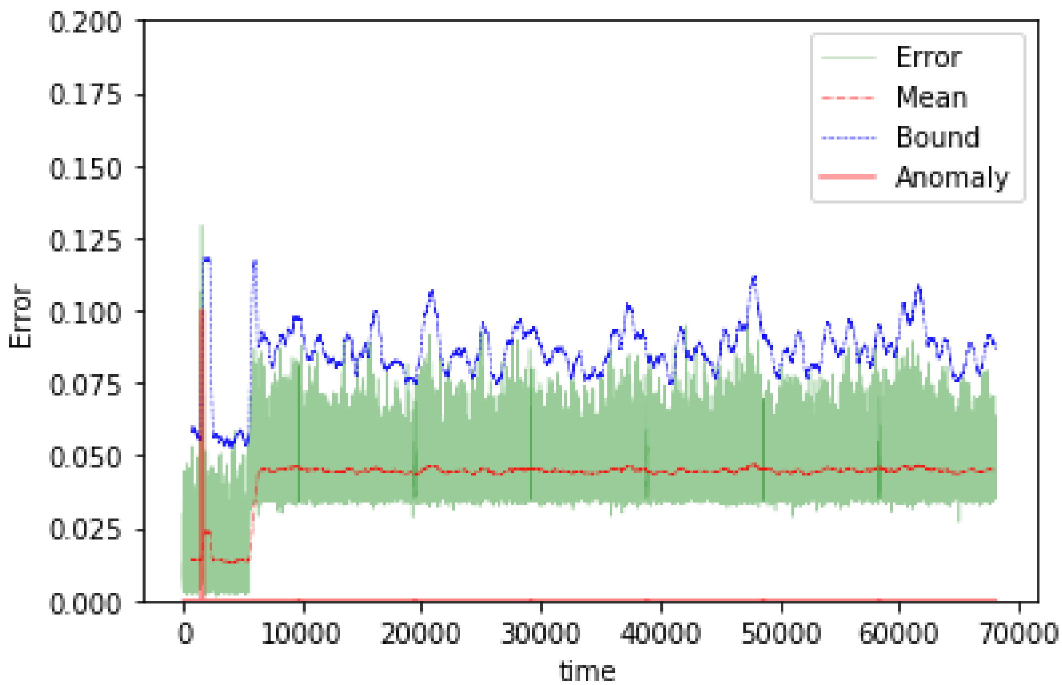


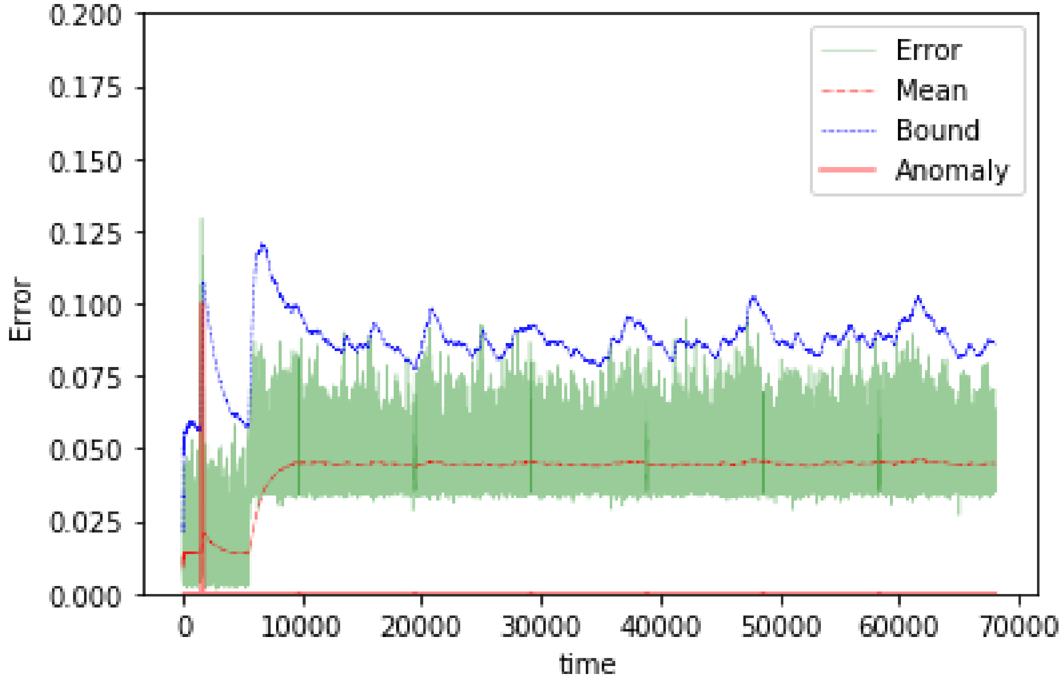

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

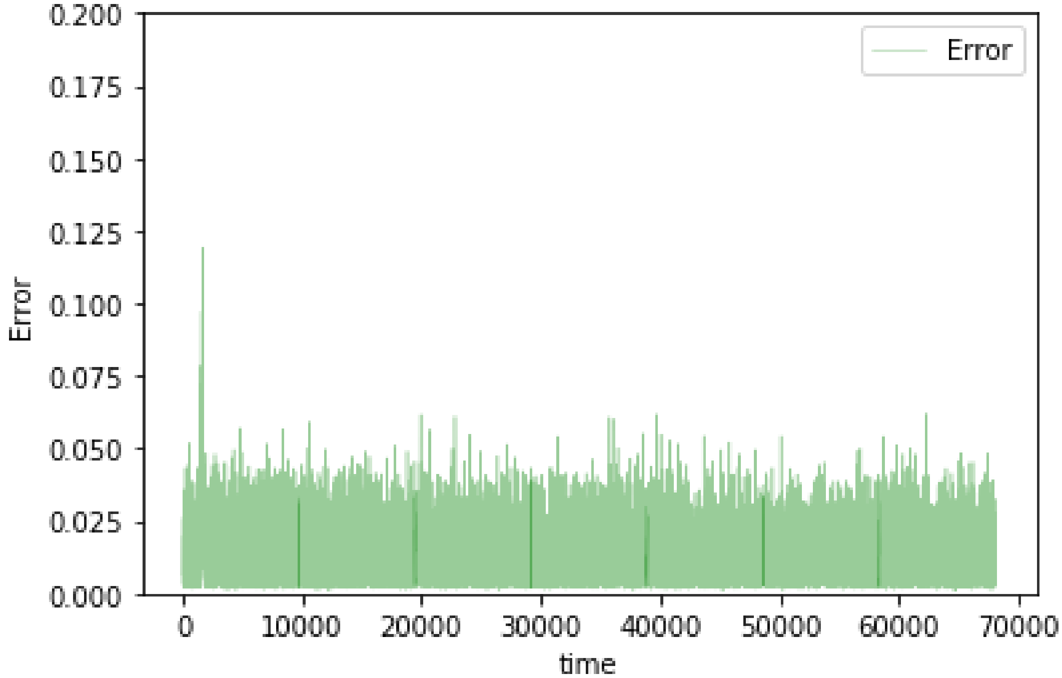
```
The mean error for lin200_avg_load_ is 0.04455745507057327 for length 68099
Testing on app change early data.
```



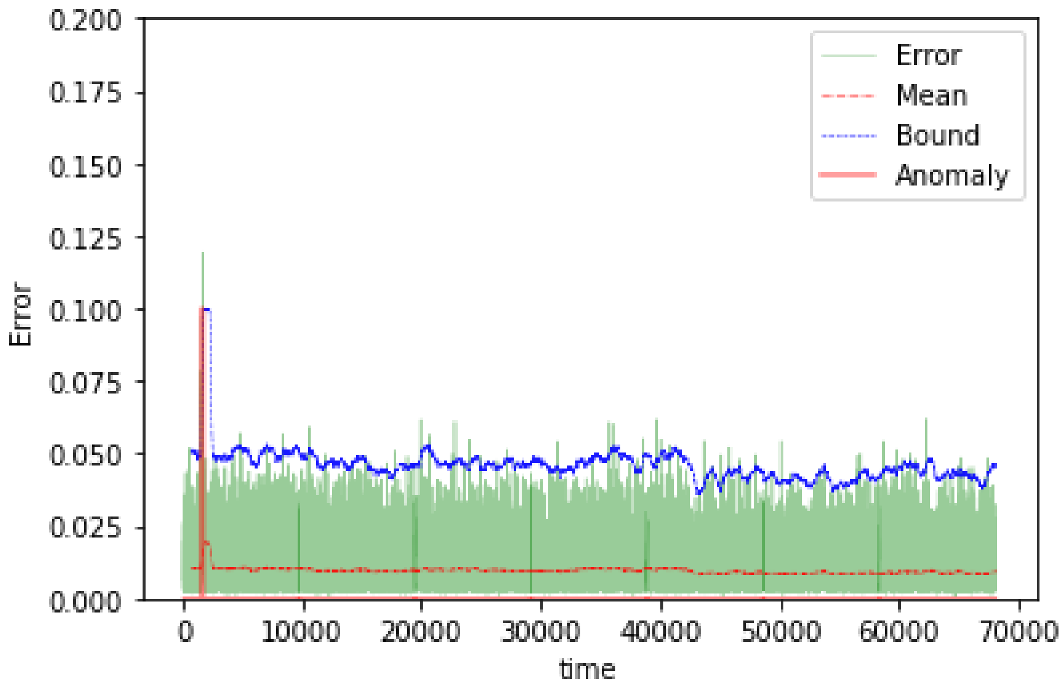




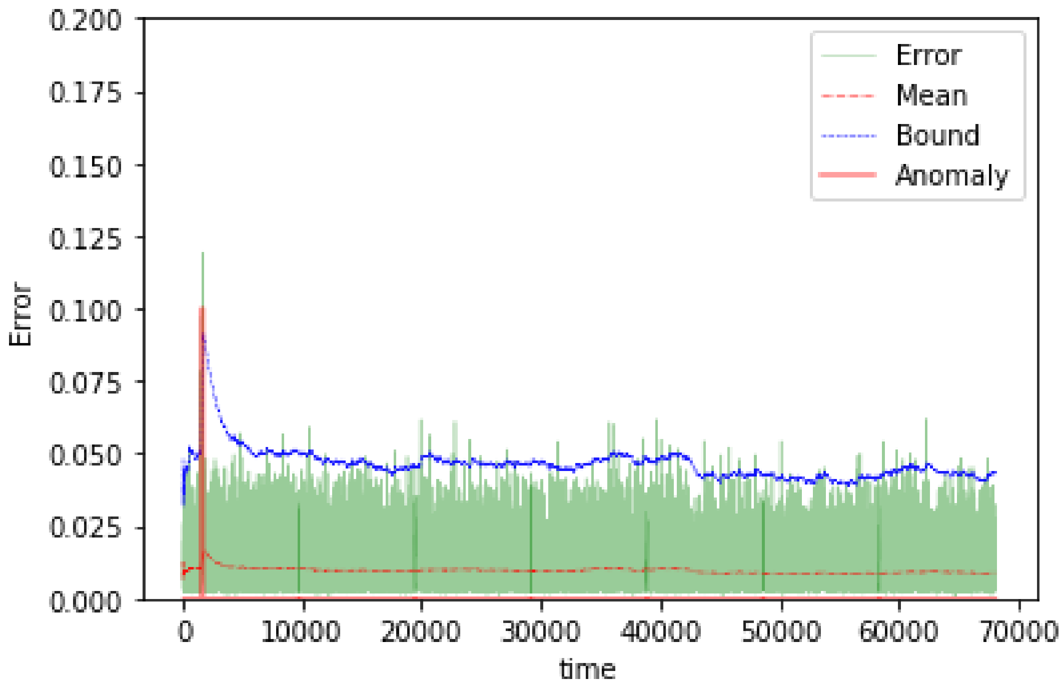

```
The mean error for lin200_app_change_early_ is 0.04239730269308195 for length 68099
Testing on Normal data.
```



Error
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000



Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

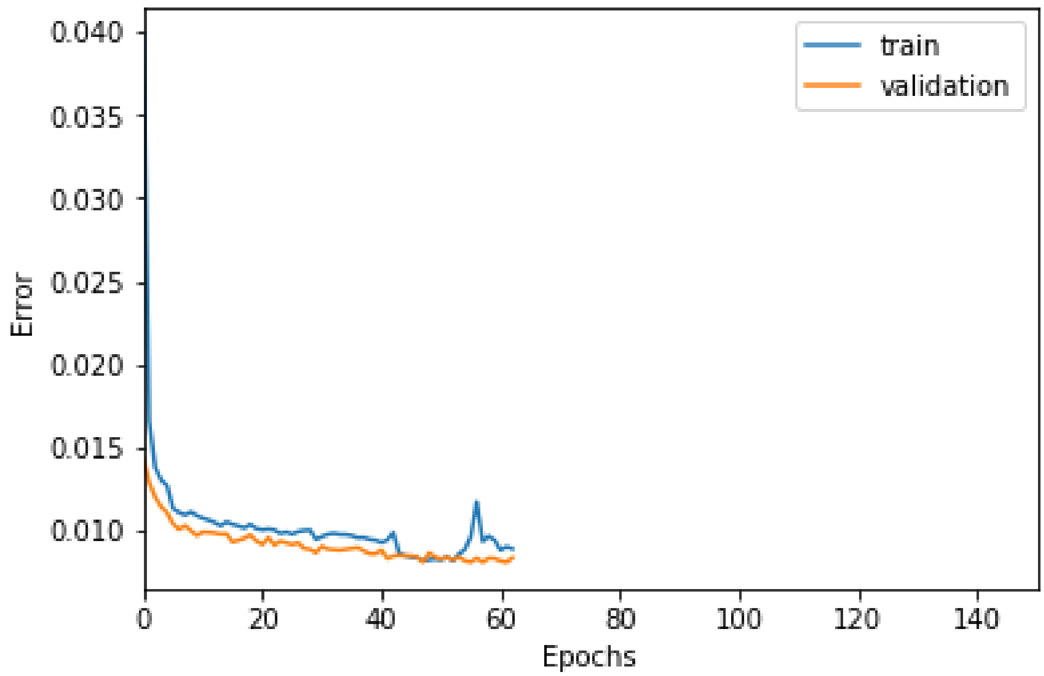



```
The mean error for lin200_normal_ is 0.009611244292339805 for length 68099
====================
```

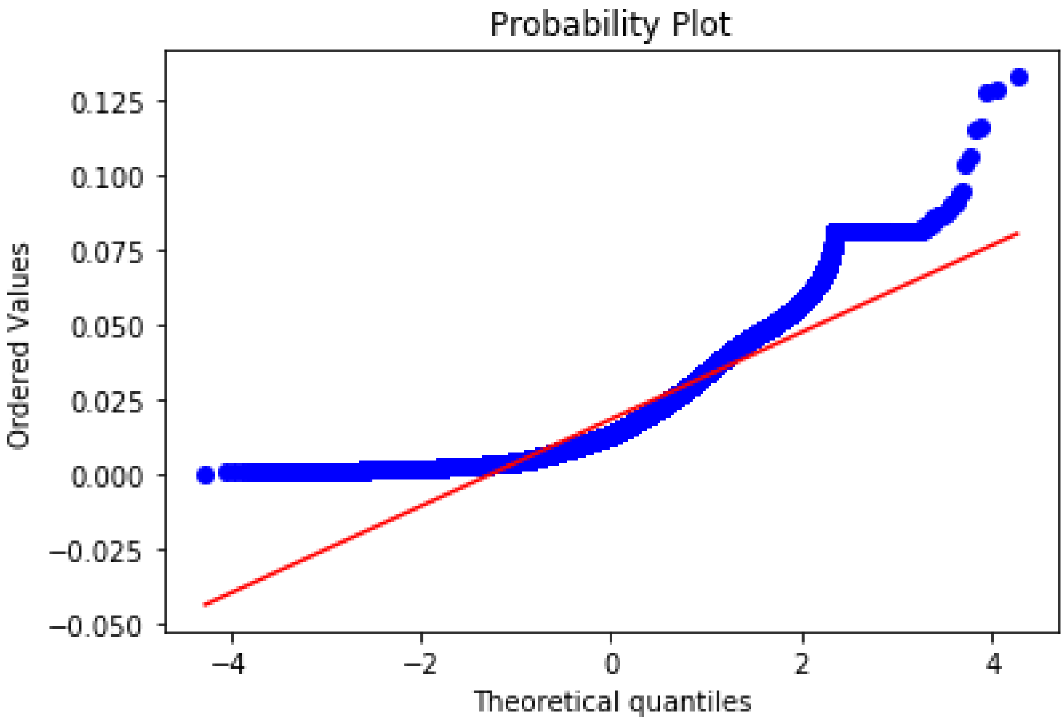
1.11.2 NN with 1 hidden layer

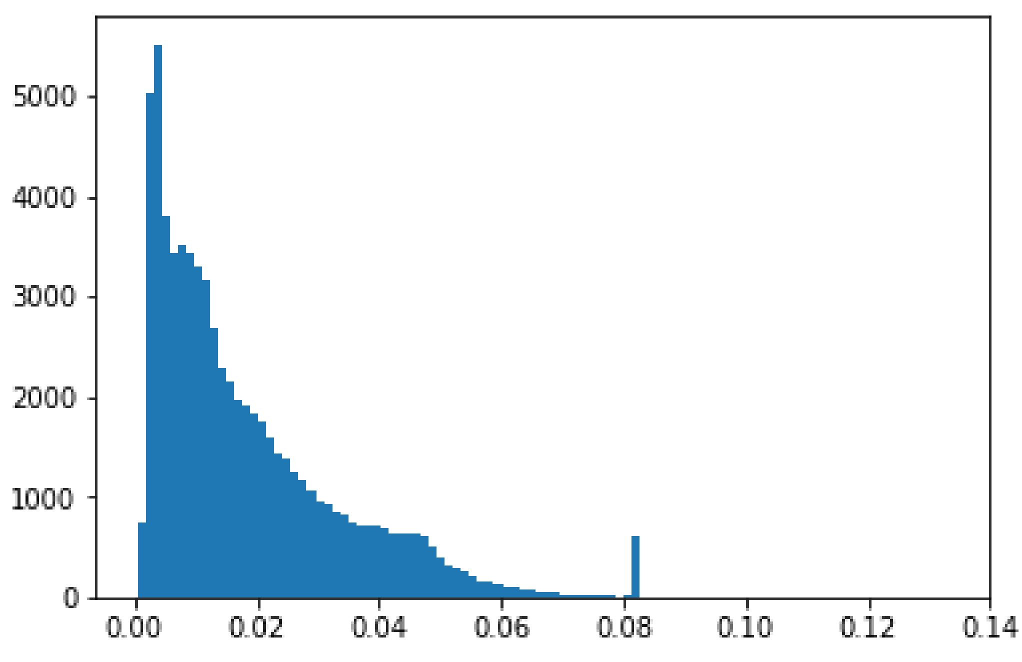
2 steps

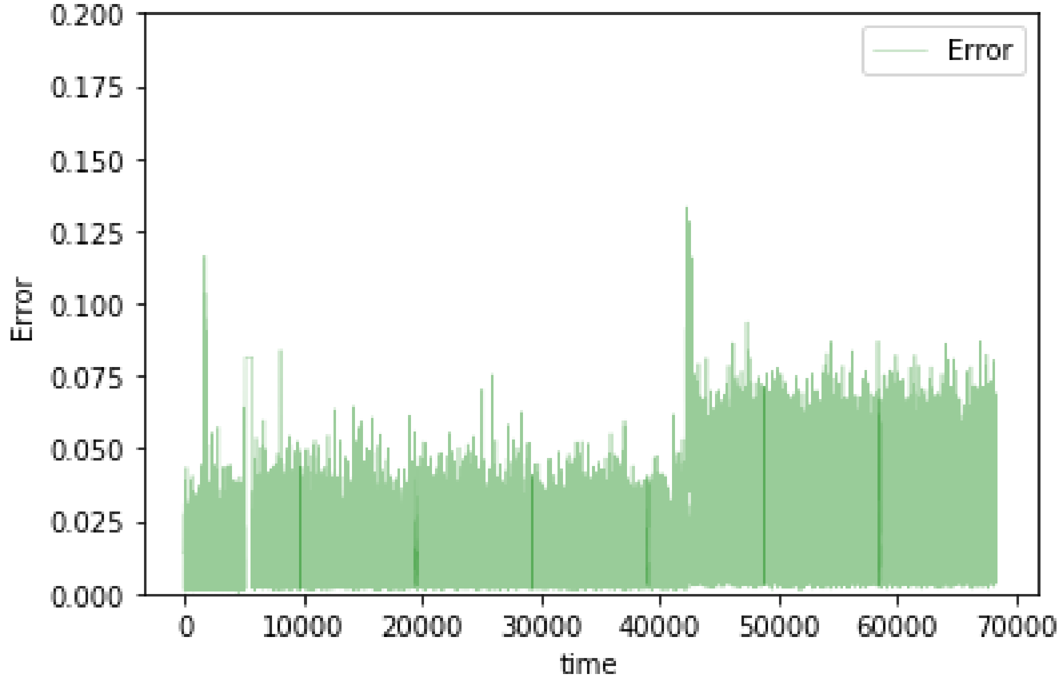
```
In [67]: TIMESTEPS = 2
         DIM = 29
         tgen = flat_generator(X, TIMESTEPS)
         vgen = flat_generator(val_X, TIMESTEPS)
         name = "nn1_2"

In [68]: input_layer = Input(shape=(TIMESTEPS*DIM,))
         hidden = Dense(100, activation='relu')(input_layer)
         output = Dense(DIM, activation='sigmoid')(hidden)

In [69]: model = Model(input_layer, output)
         model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [70]: train(model, tgen, vgen, name=name)
         test(model, name=name, window=TIMESTEPS)
```



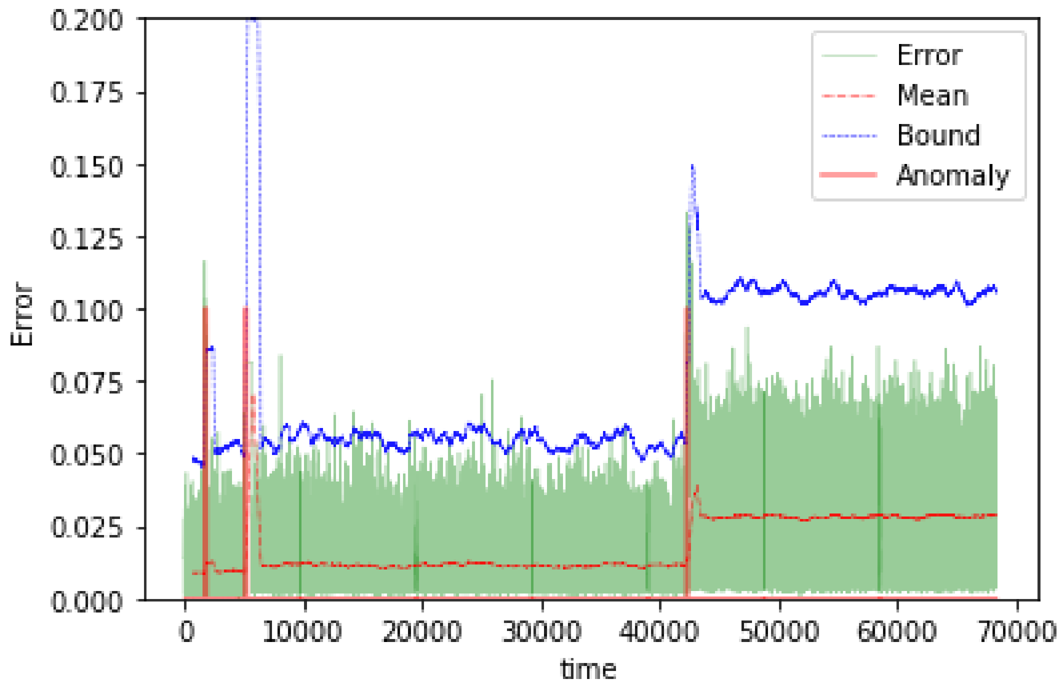

```
Training loss for final epoch is 0.008892452337779105
Validation loss for final epoch is 0.008361250429064966
---------- Beginning tests for nn1_2 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−4
−2
0
2
4

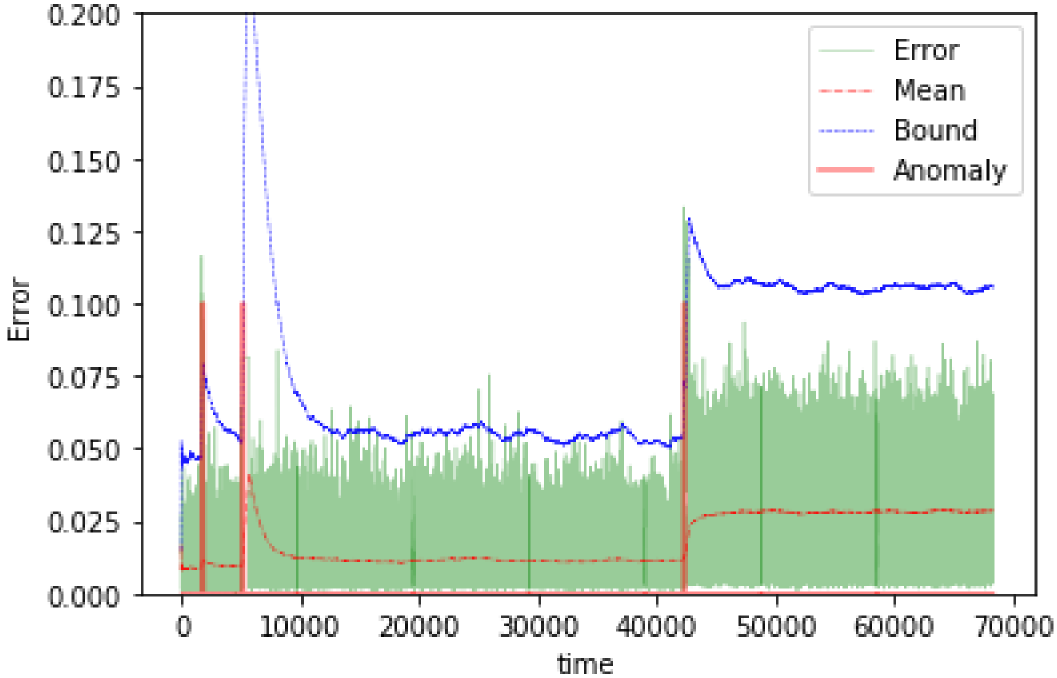

5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14



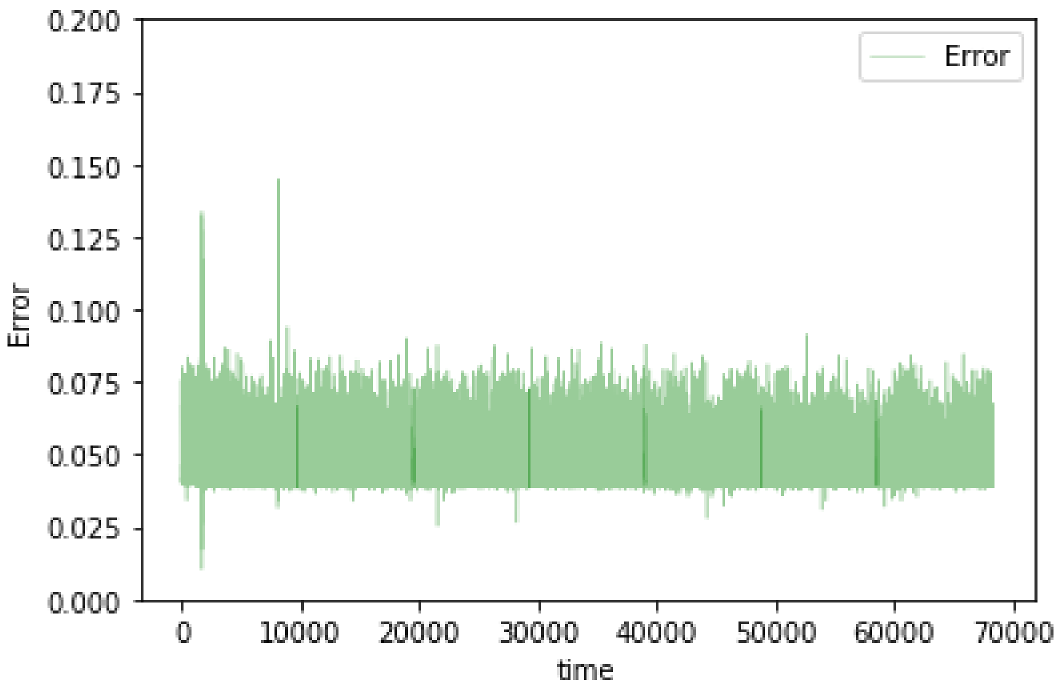
Error
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

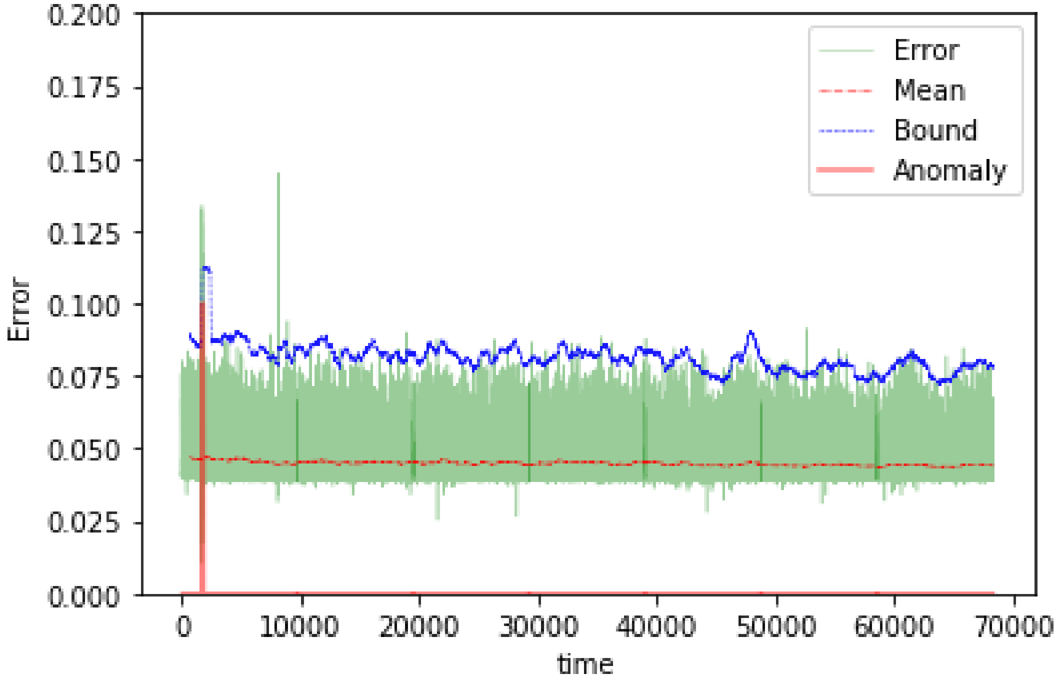


Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

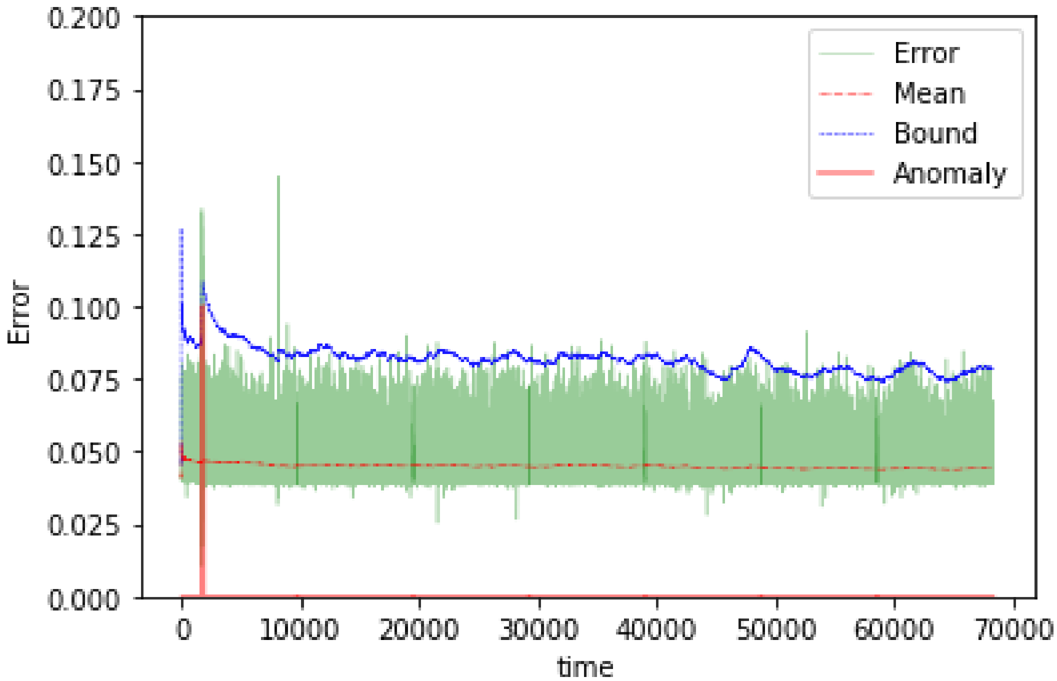



```
The mean error for nn1_2_disk_IO_start_ is 0.018403414306870633 for length 68297
Testing on Avg. load data.
```



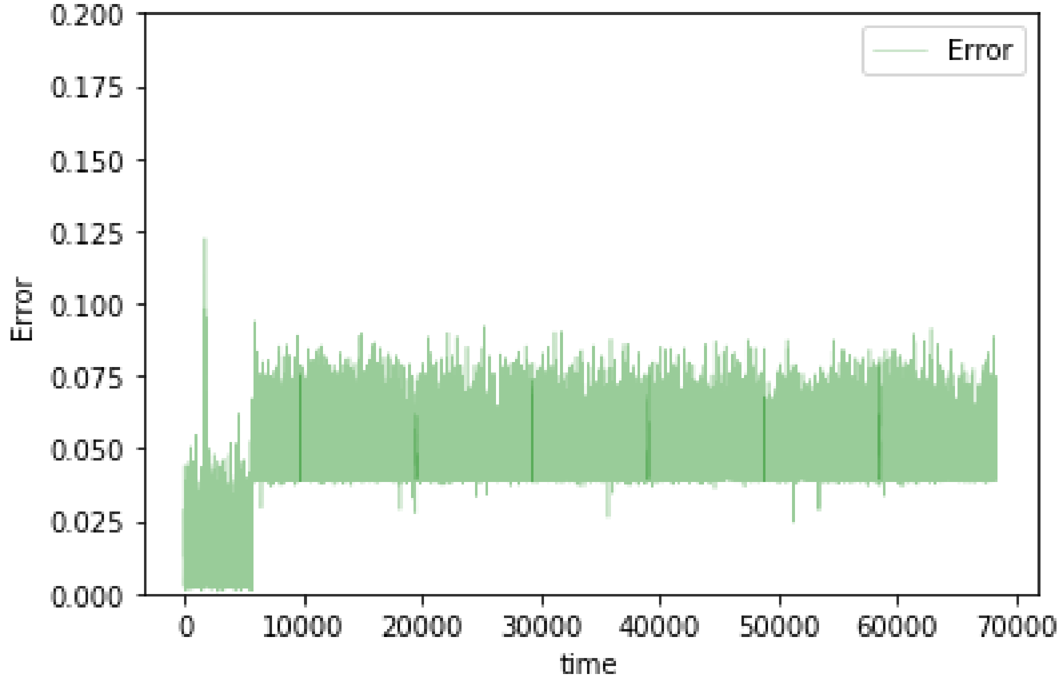


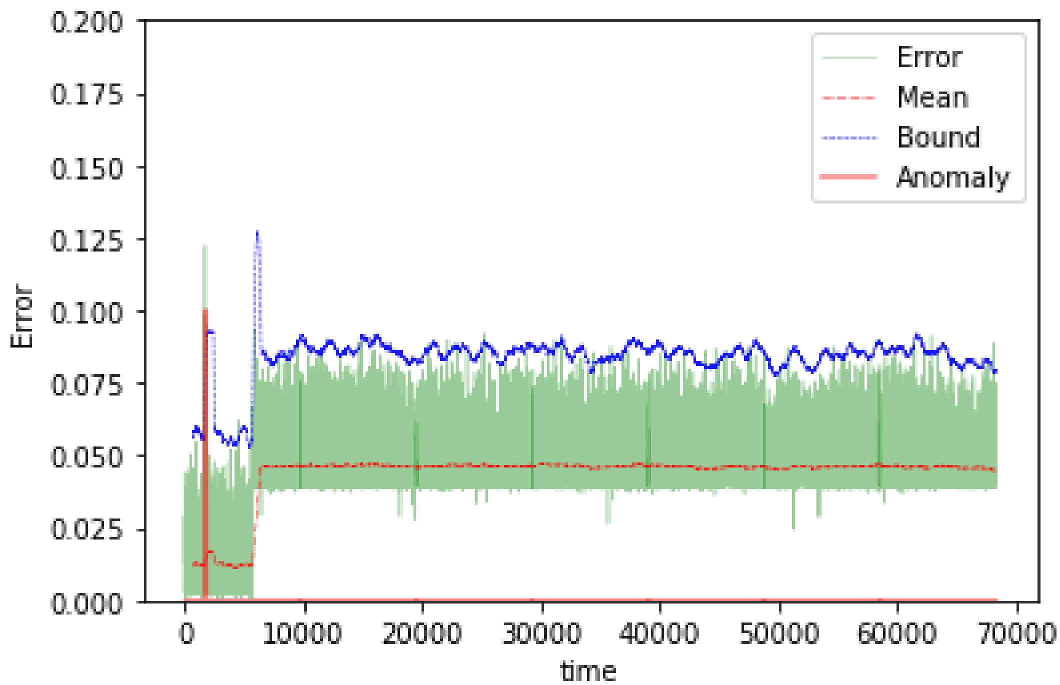
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

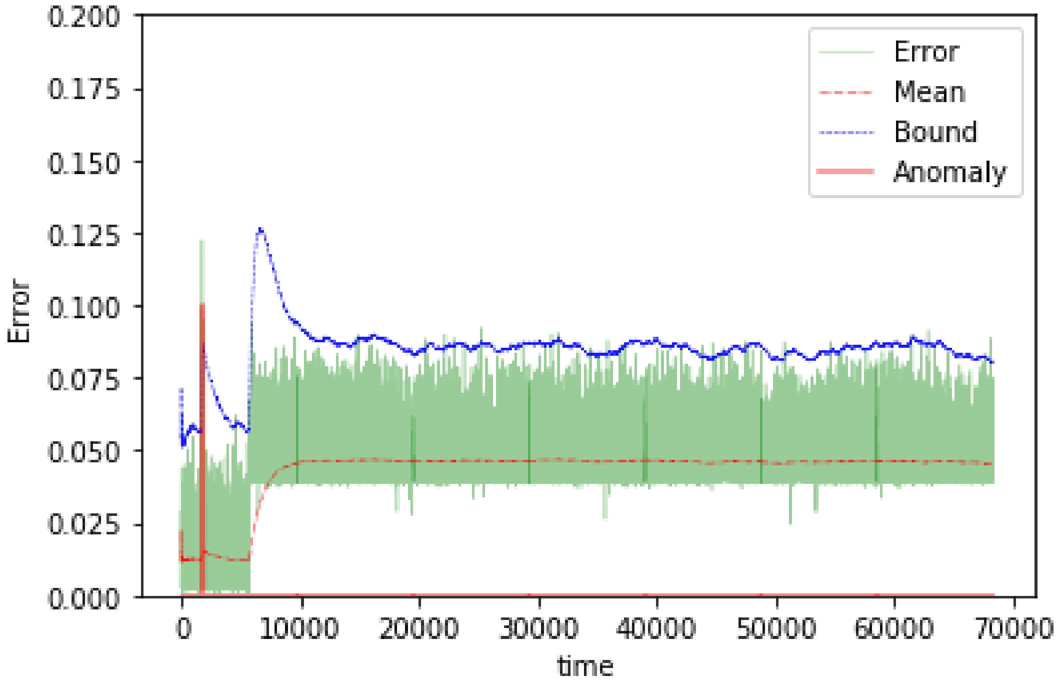


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

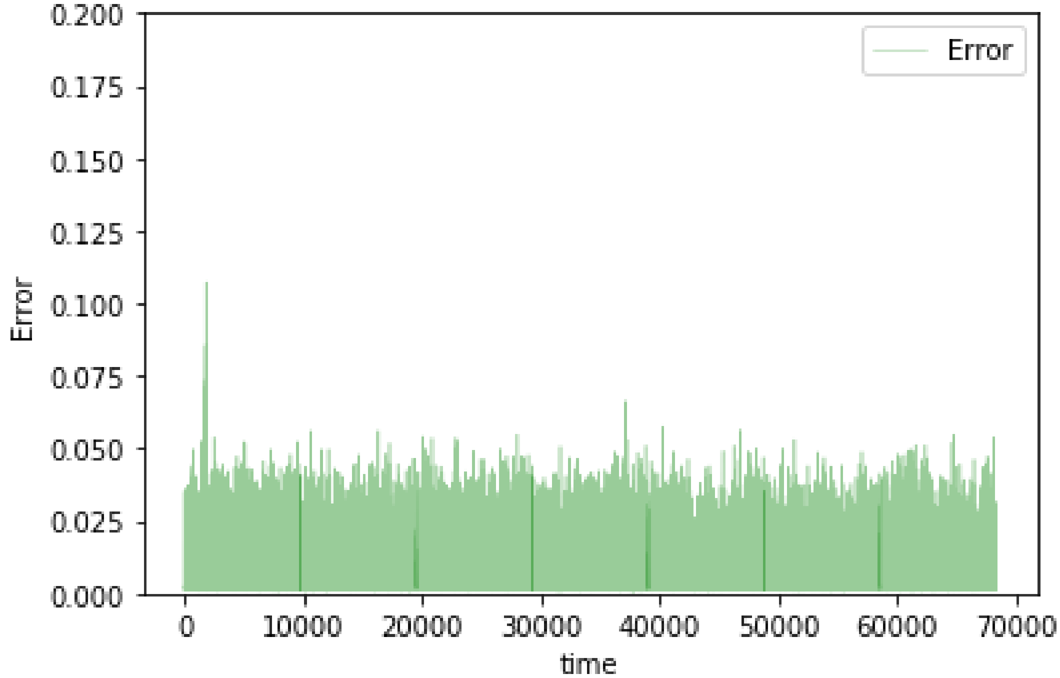
```
The mean error for nn1_2_avg_load_ is 0.044887494557099776 for length 68297
Testing on app change early data.
```

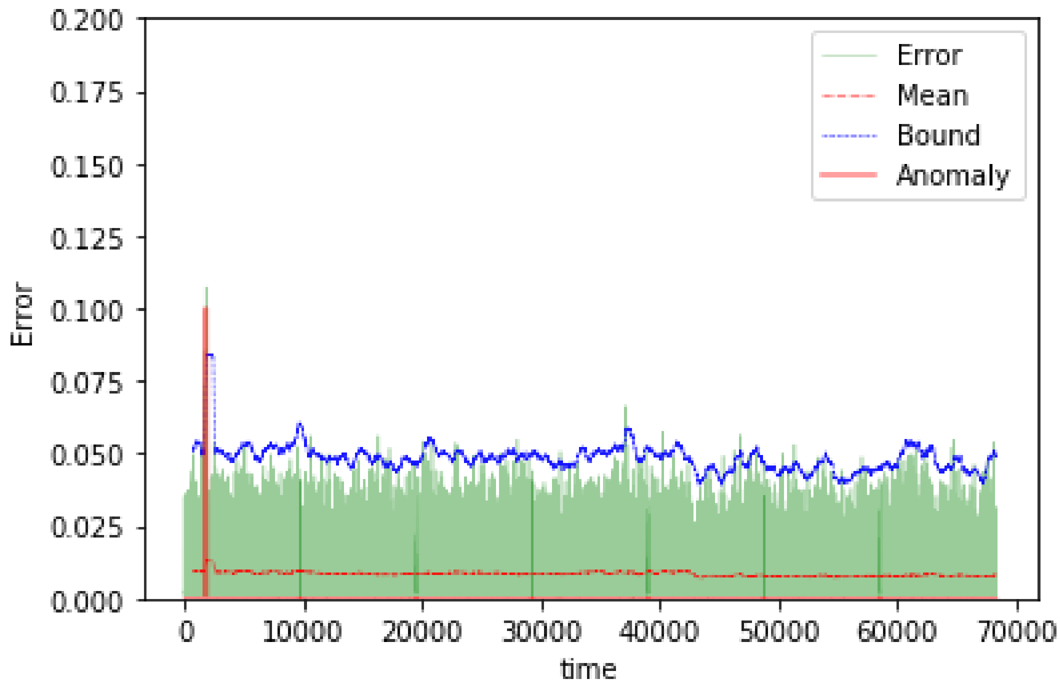


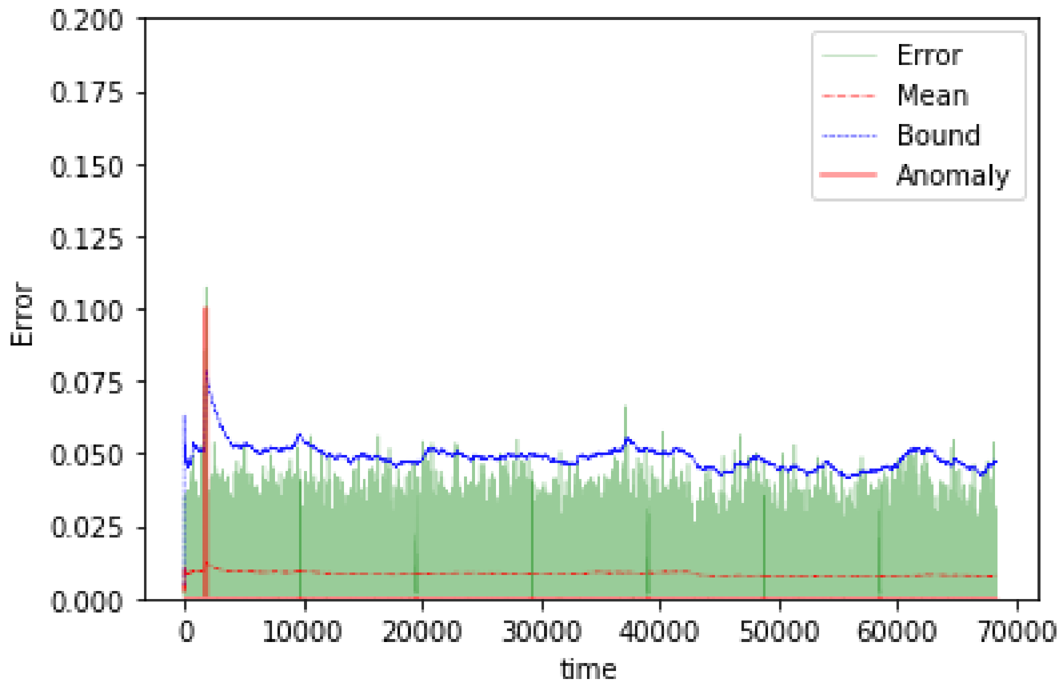





```
The mean error for nn1_2_app_change_early_ is 0.04342041837538587 for length 68297
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

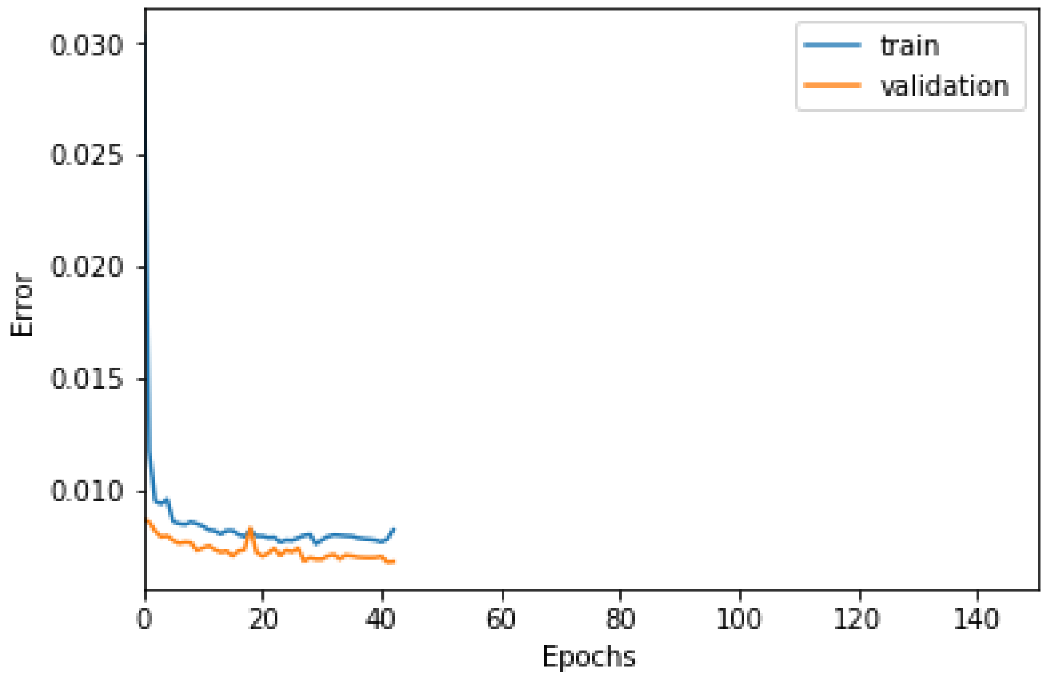

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



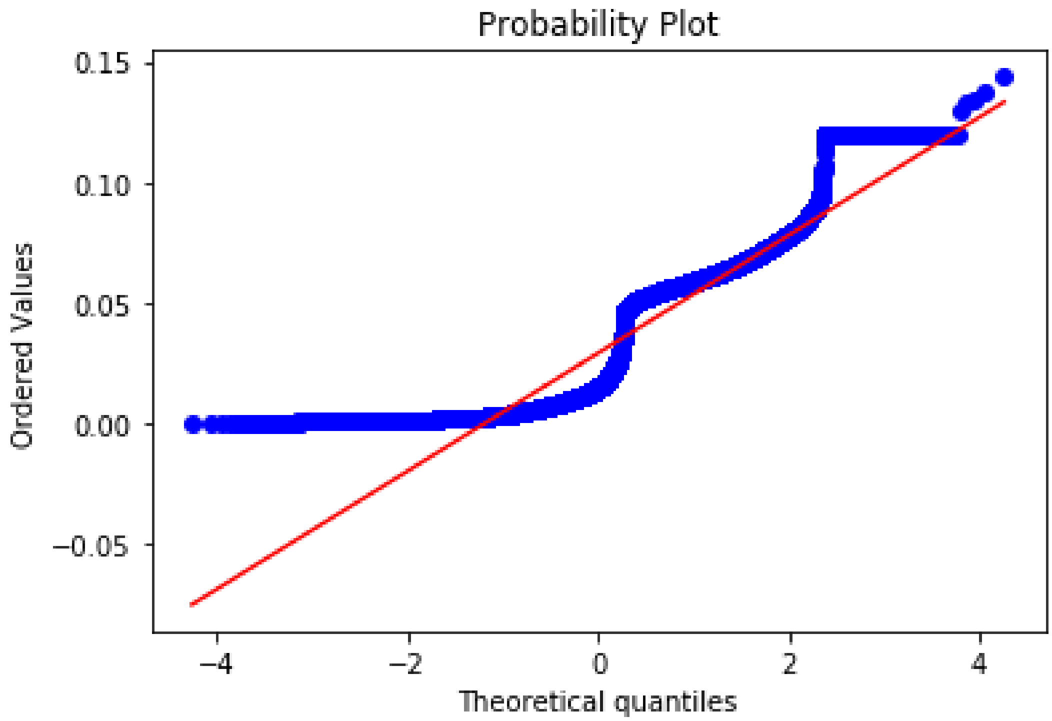

```
The mean error for nn1_2_normal_ is 0.008474720048524557 for length 68297
====================
```

5 steps

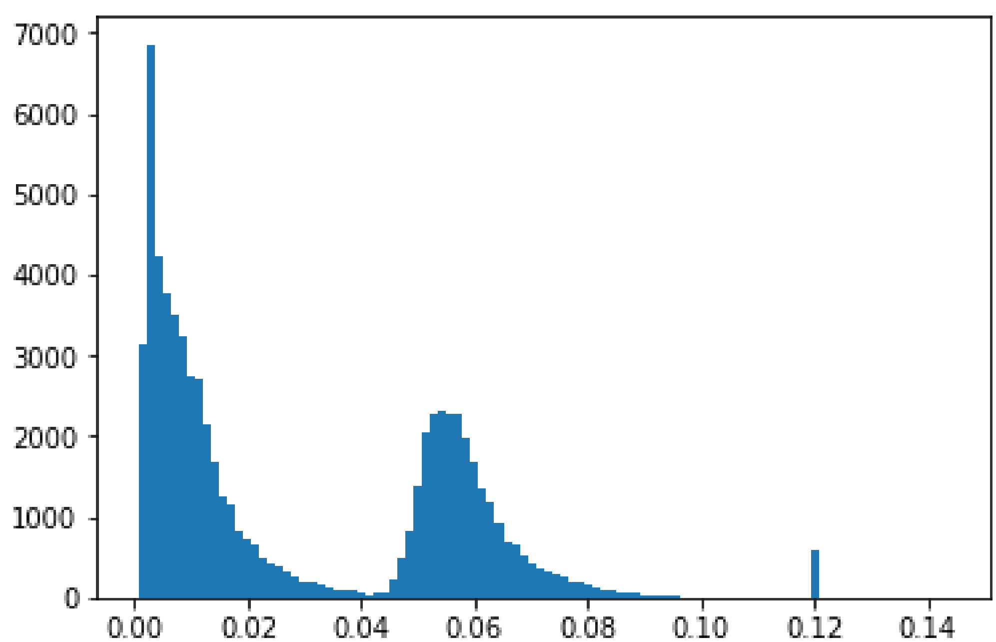
```
In [71]: TIMESTEPS = 5
         DIM = 29
         tgen = flat_generator(X, TIMESTEPS)
         vgen = flat_generator(val_X, TIMESTEPS)
         name = "nn1_5"

In [72]: input_layer = Input(shape=(TIMESTEPS*DIM,))
         hidden = Dense(100, activation='relu')(input_layer)
         output = Dense(DIM, activation='sigmoid')(hidden)

In [73]: model = Model(input_layer, output)
         model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [74]: train(model, tgen, vgen, name=name)
         test(model, name=name, window=TIMESTEPS)
```



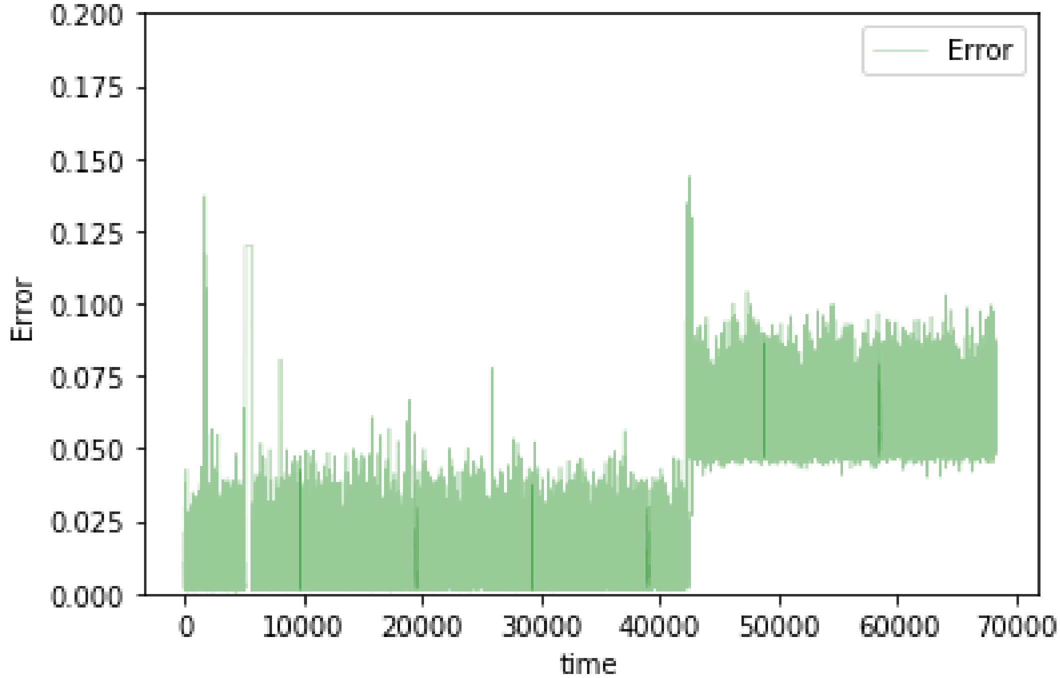

```
Training loss for final epoch is 0.008181585344835185
Validation loss for final epoch is 0.006756556435022503
---------- Beginning tests for nn1_5 ----------
Testing on Disk IO begin data.
```



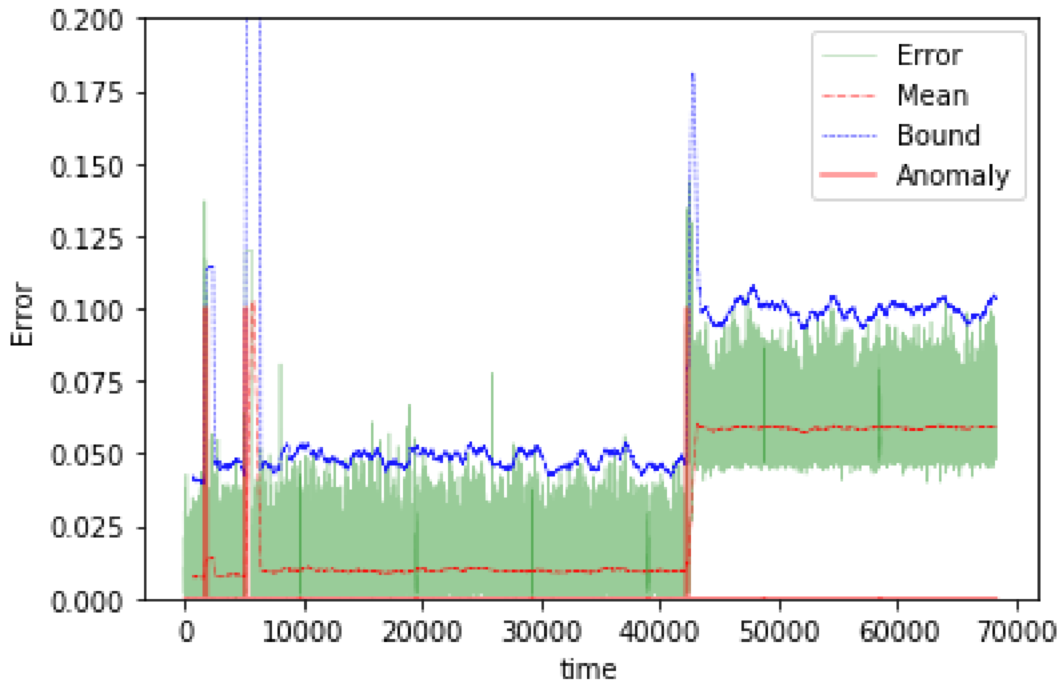
Probability Plot
0.15
0.10
0.05
0.00
−0.05
Ordered Values
−4
−2
0
2
4
Theoretical quantiles



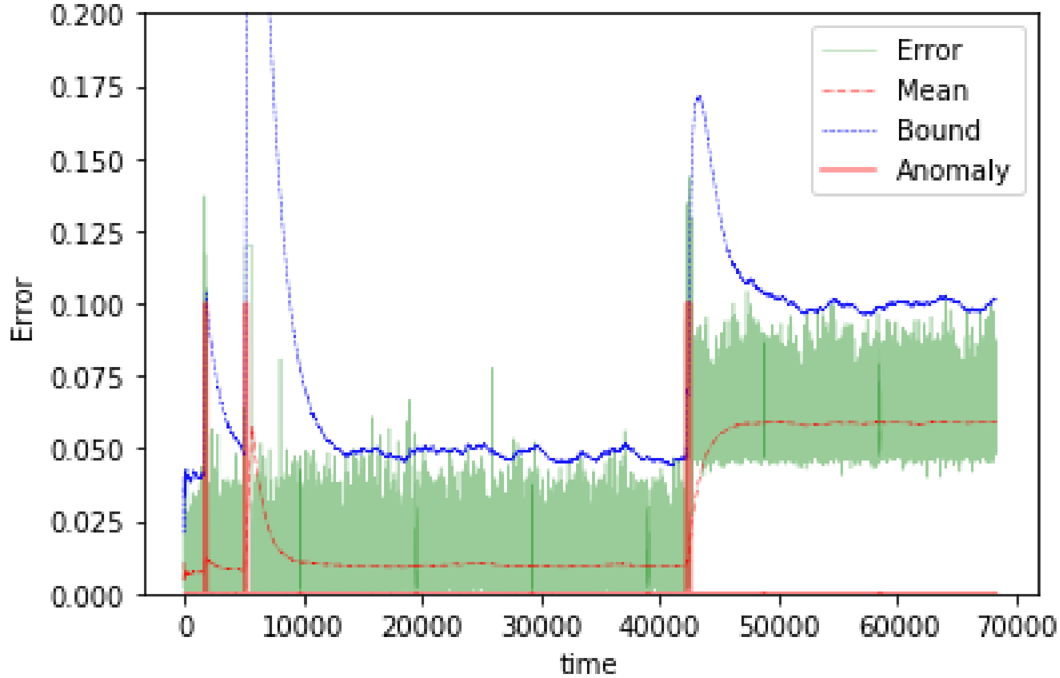
7000
6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14



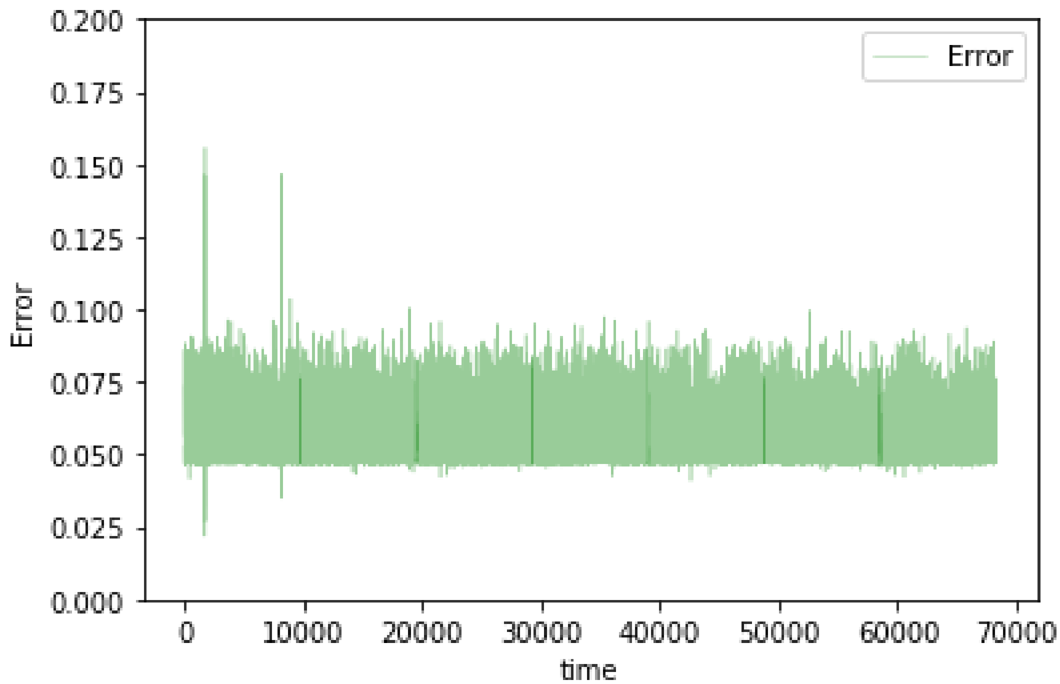
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

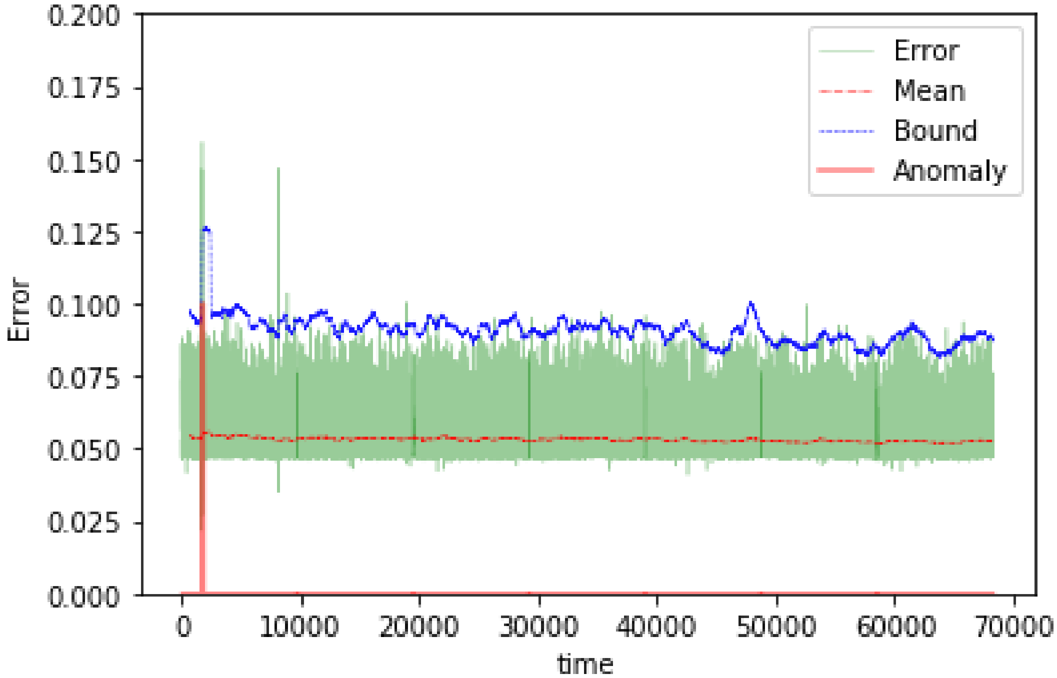


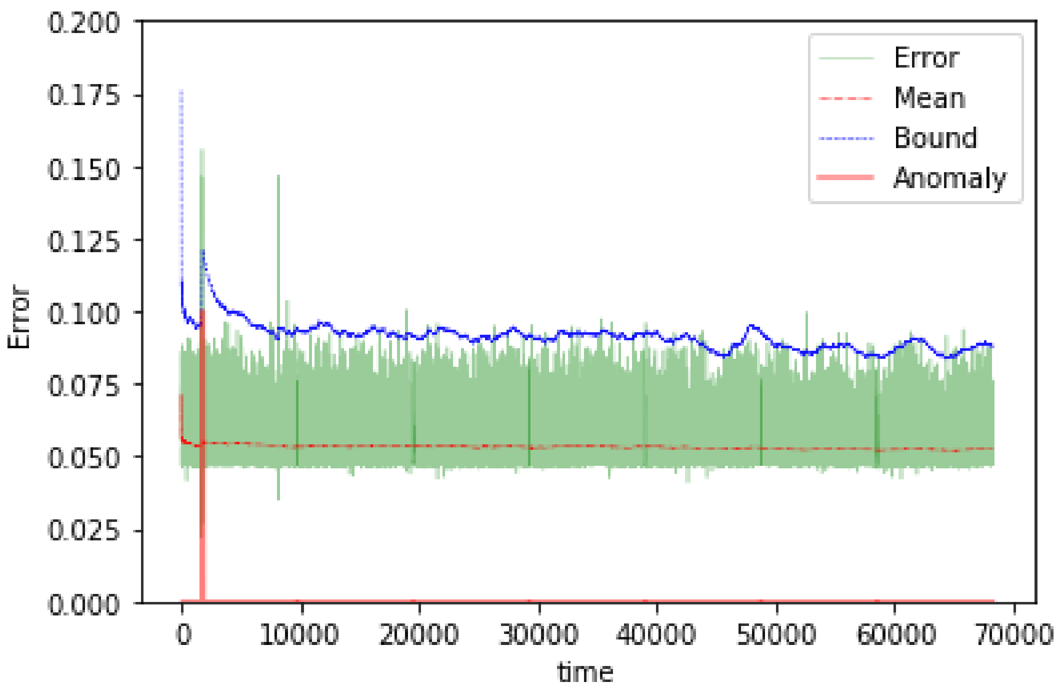
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



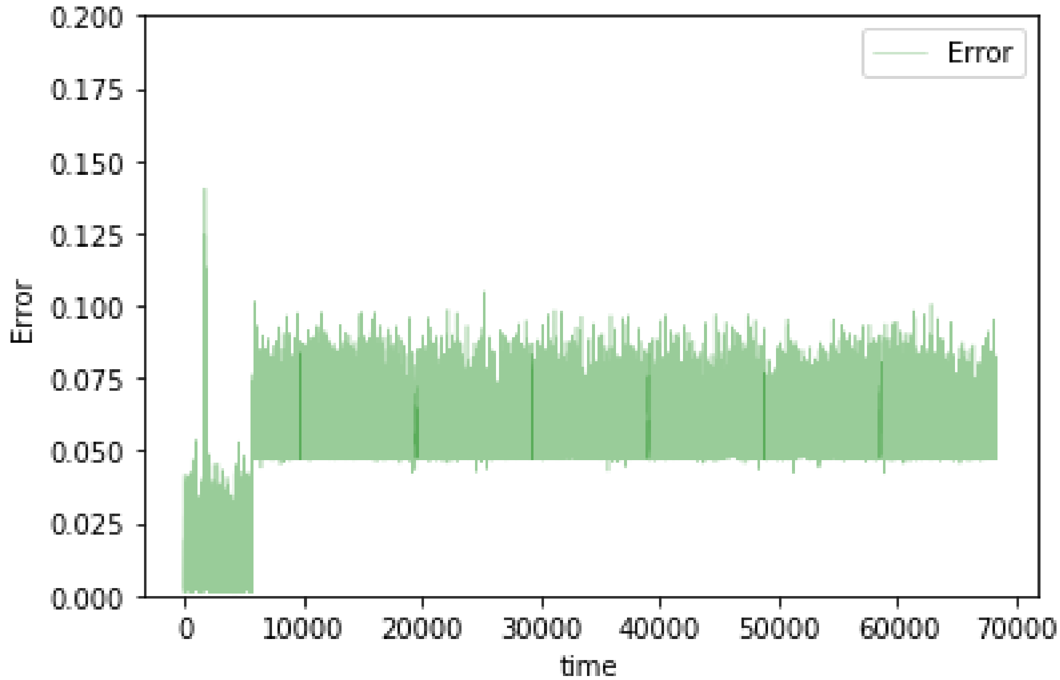

```
The mean error for nn1_5_disk_IO_start_ is 0.029208292395209123 for length 68294
Testing on Avg. load data.
```

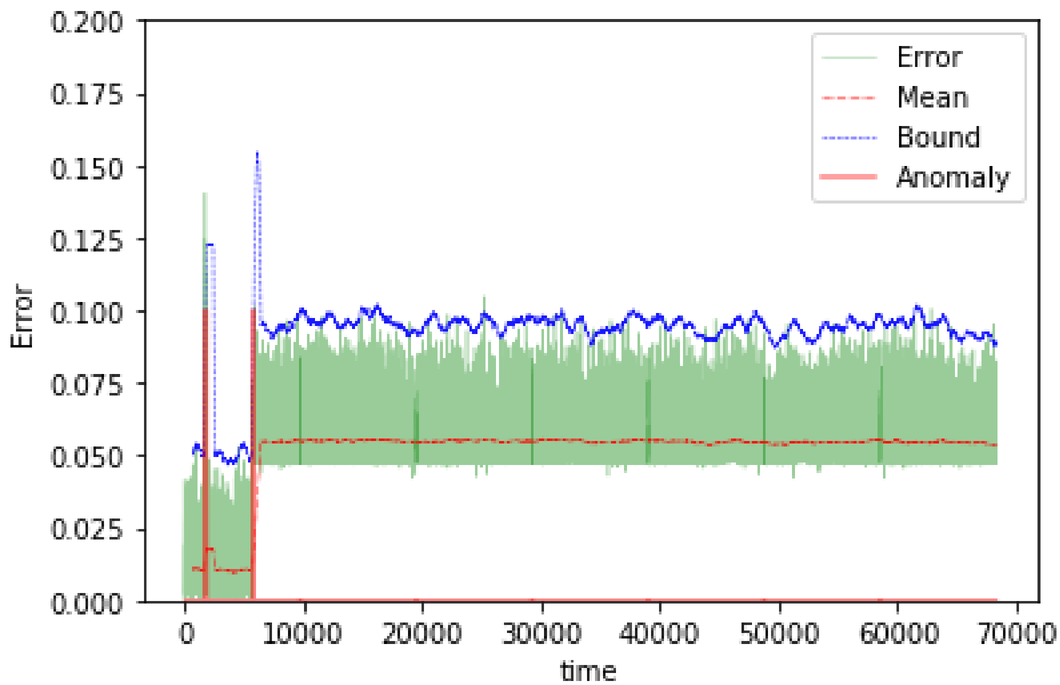


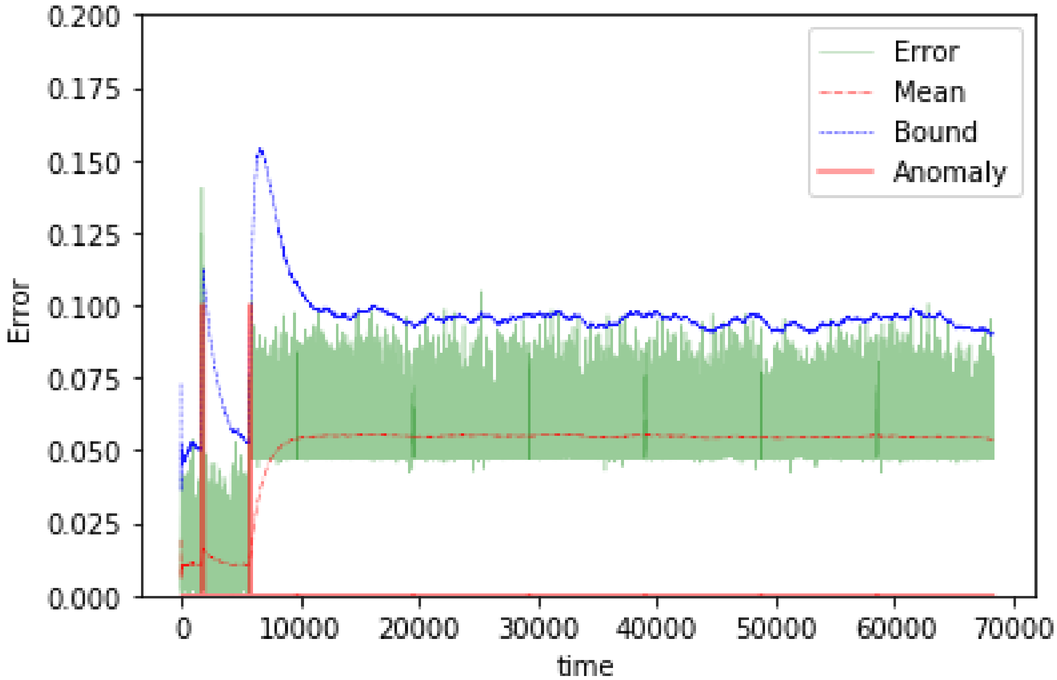

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

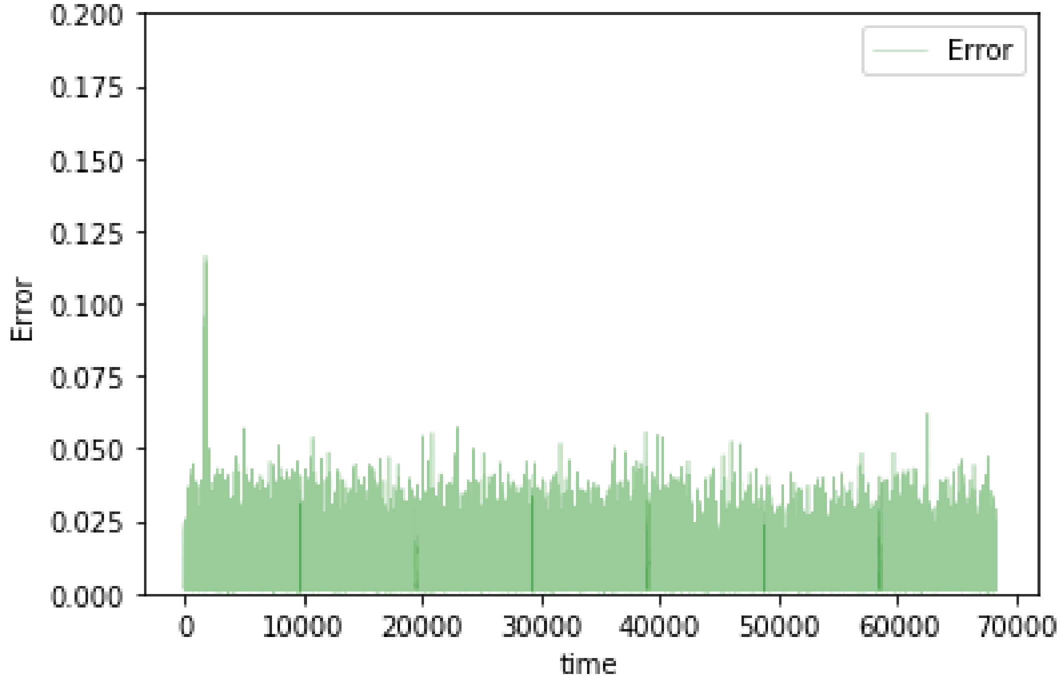
```
The mean error for nn1_5_avg_load_ is 0.05315398711596679 for length 68294
Testing on app change early data.
```

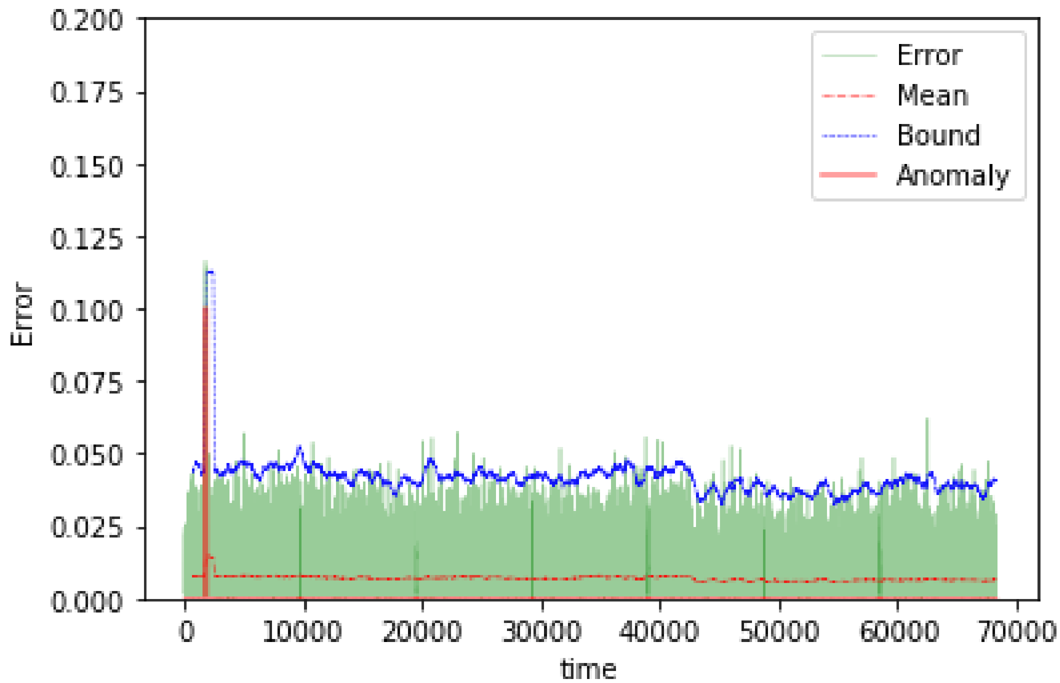


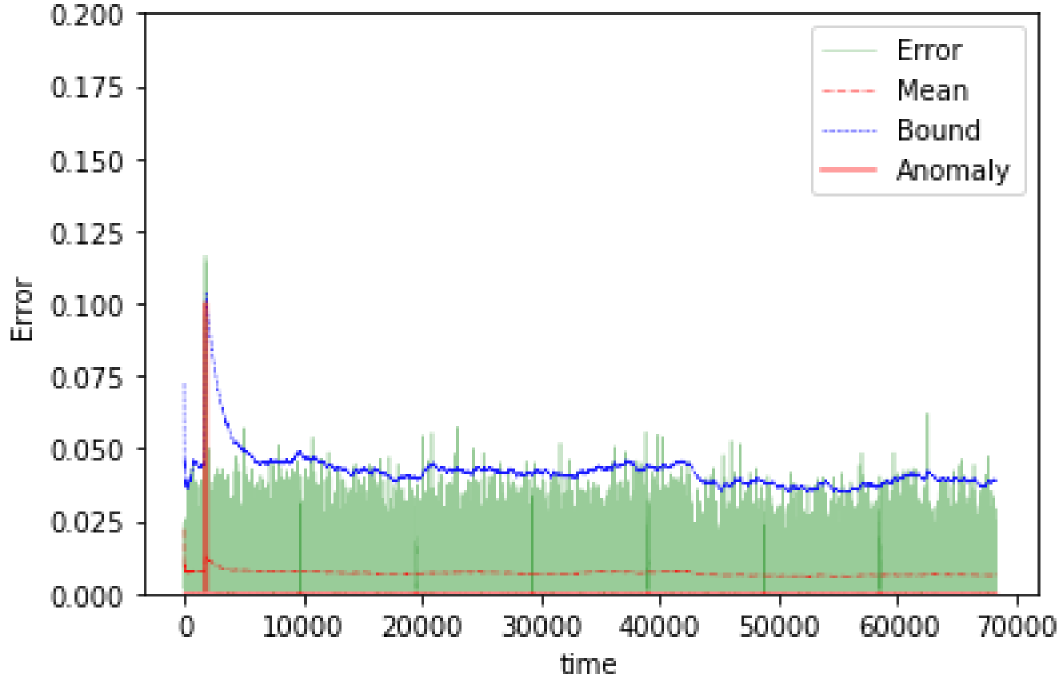





```
The mean error for nn1_5_app_change_early_ is 0.05115941328373264 for length 68294
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

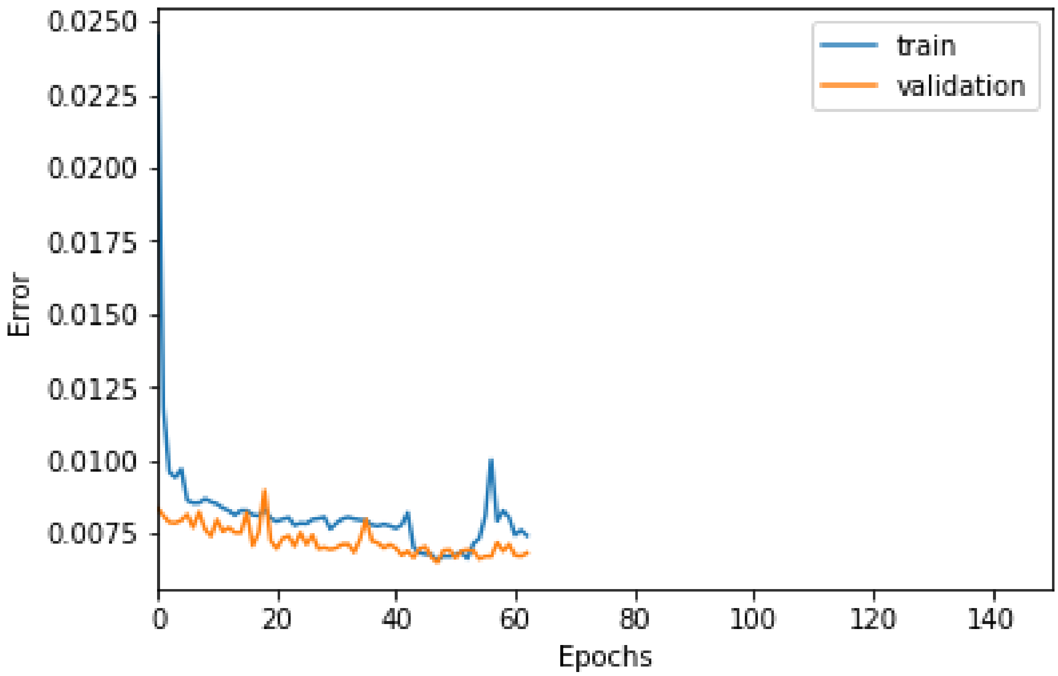

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



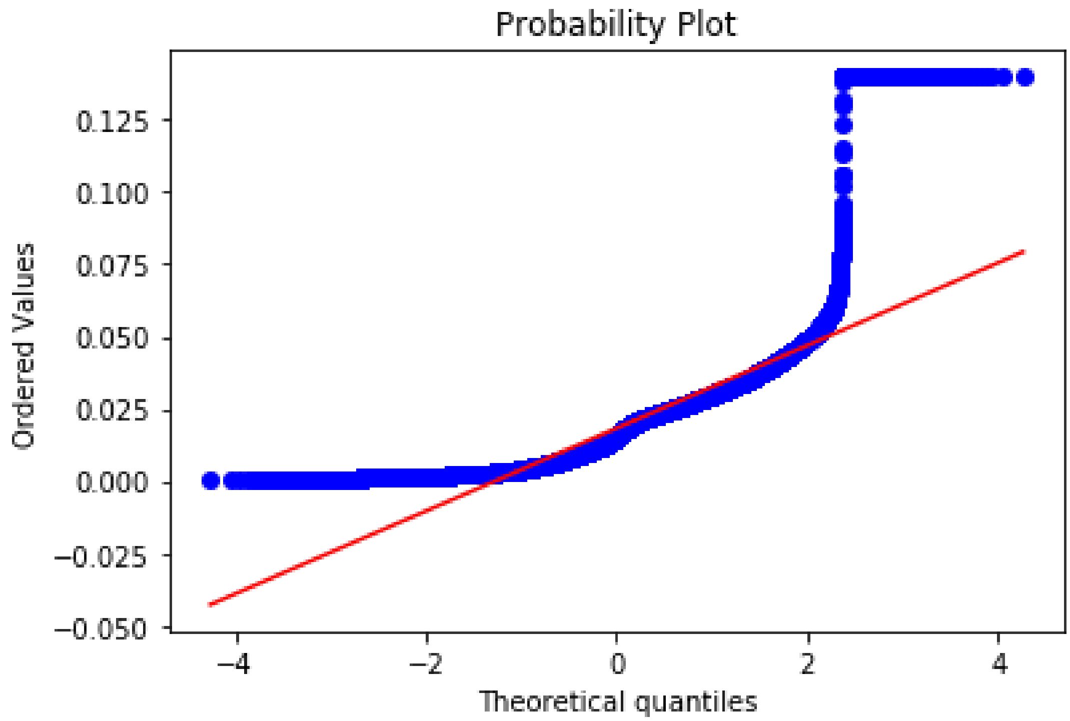

```
The mean error for nn1_5_normal_ is 0.006968521305129605 for length 68294
====================
```

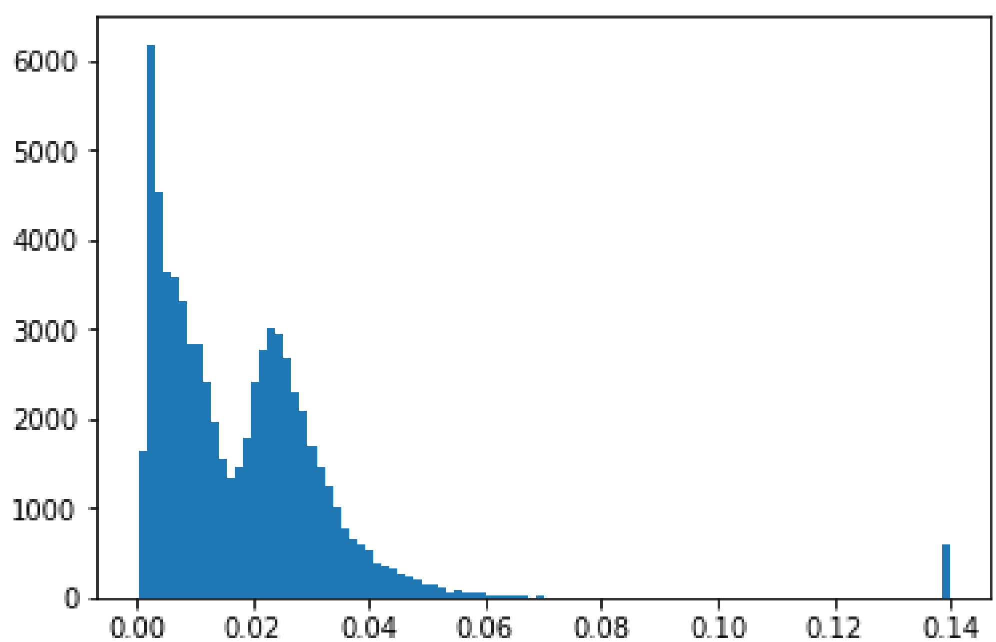
10 steps

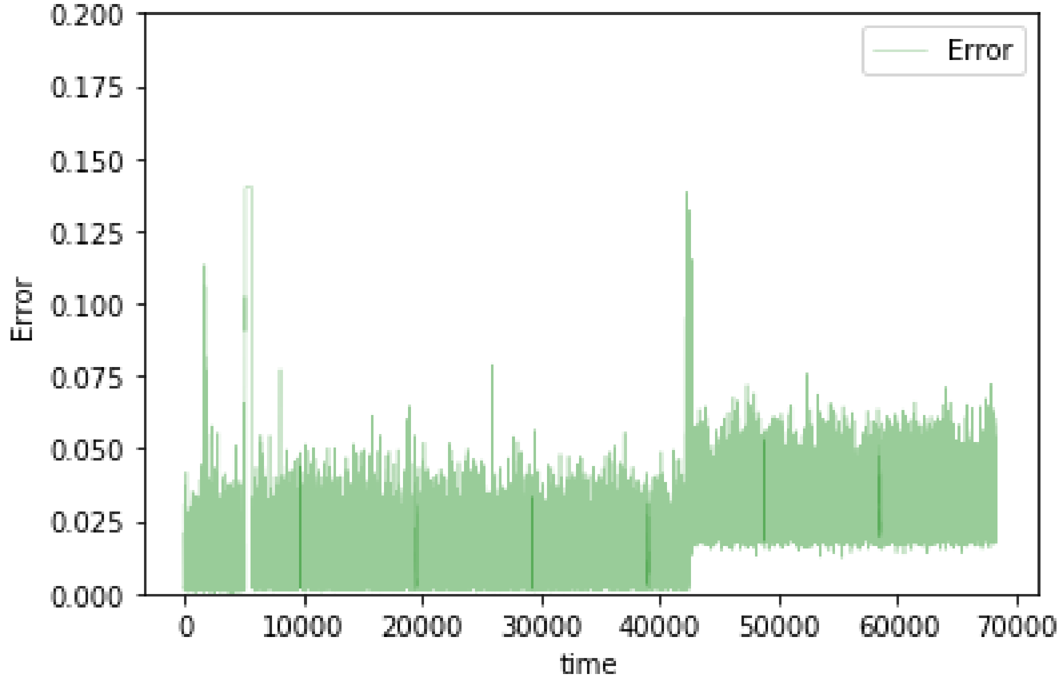
```
In [75]: TIMESTEPS = 10
         DIM = 29
         tgen = flat_generator(X, TIMESTEPS)
         vgen = flat_generator(val_X, TIMESTEPS)
         name = "nn1_10"

In [76]: input_layer = Input(shape=(TIMESTEPS*DIM,))
         hidden = Dense(100, activation='relu')(input_layer)
         output = Dense(DIM, activation='sigmoid')(hidden)

In [77]: model = Model(input_layer, output)
         model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [78]: train(model, tgen, vgen, name=name)
         test(model, name=name, window=TIMESTEPS)
```

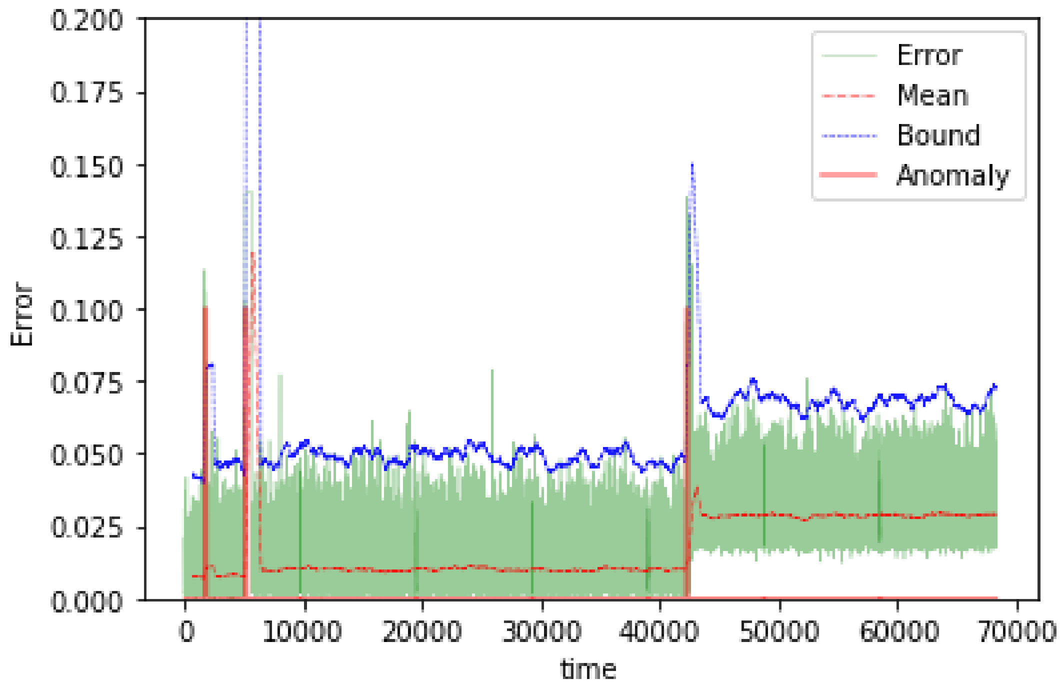


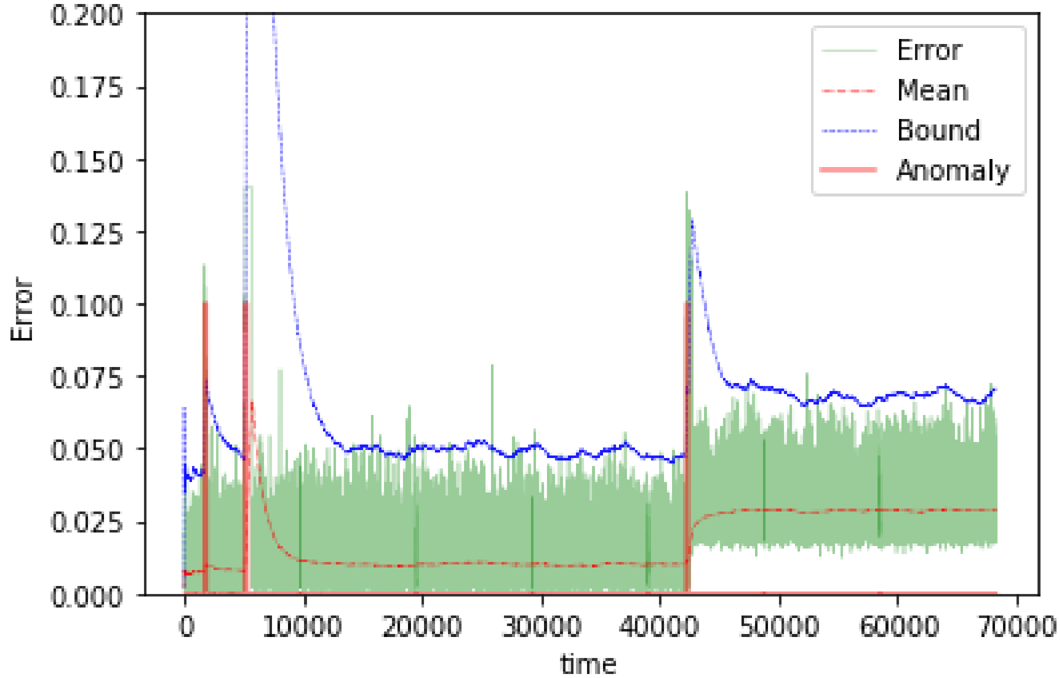

```
Training loss for final epoch is 0.007421169995213858
Validation loss for final epoch is 0.006824114109040238
---------- Beginning tests for nn1_10 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−4
−2
0
2
4

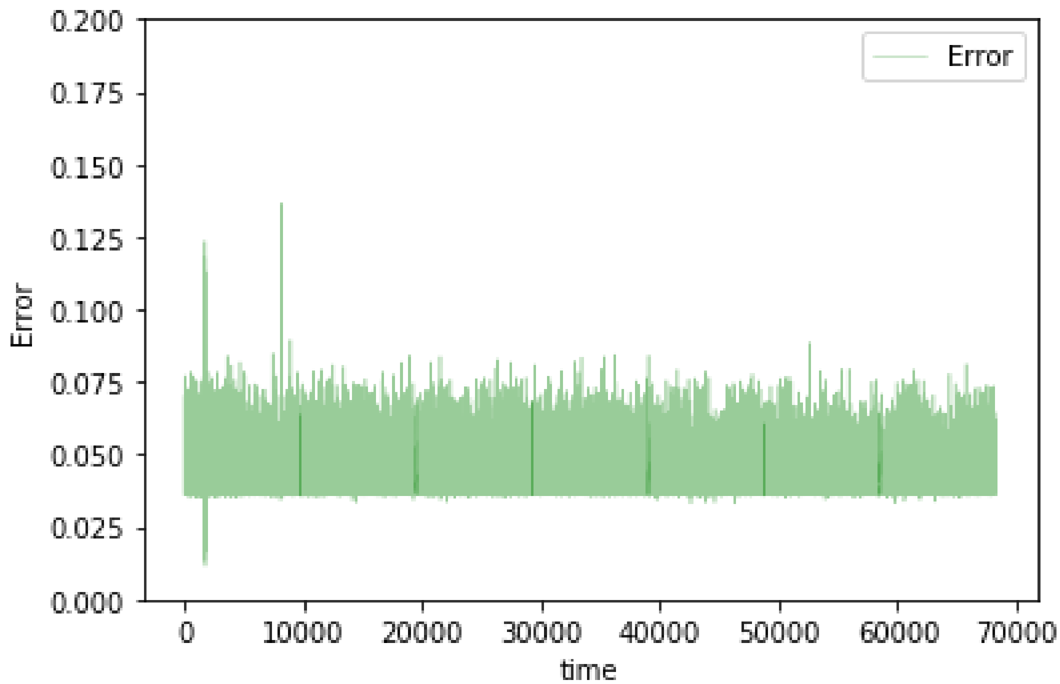

6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

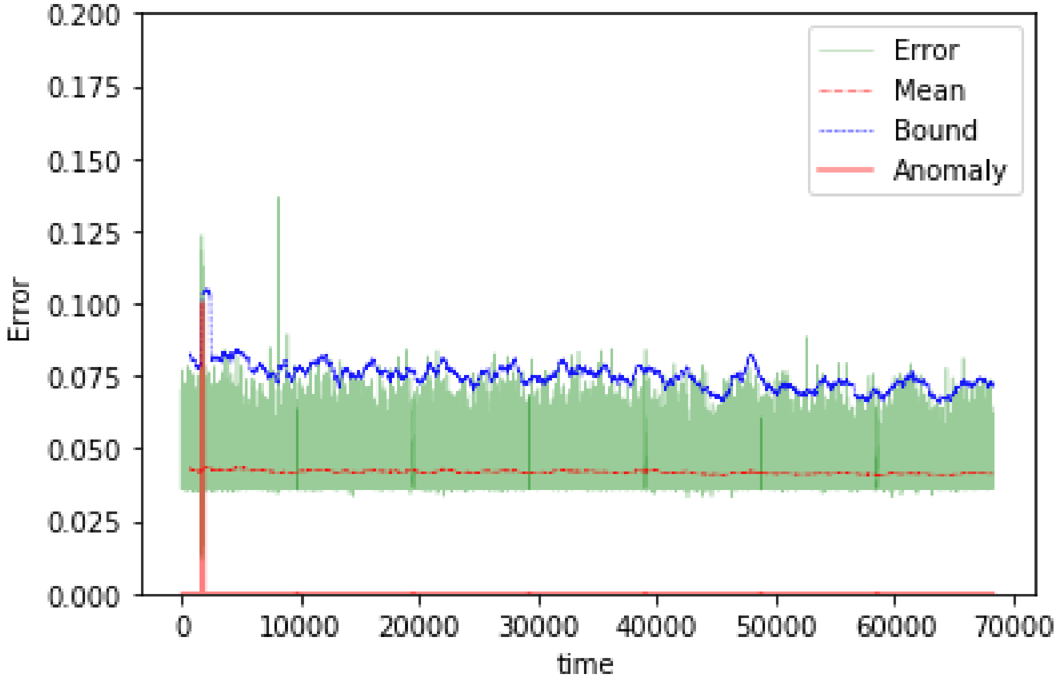

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

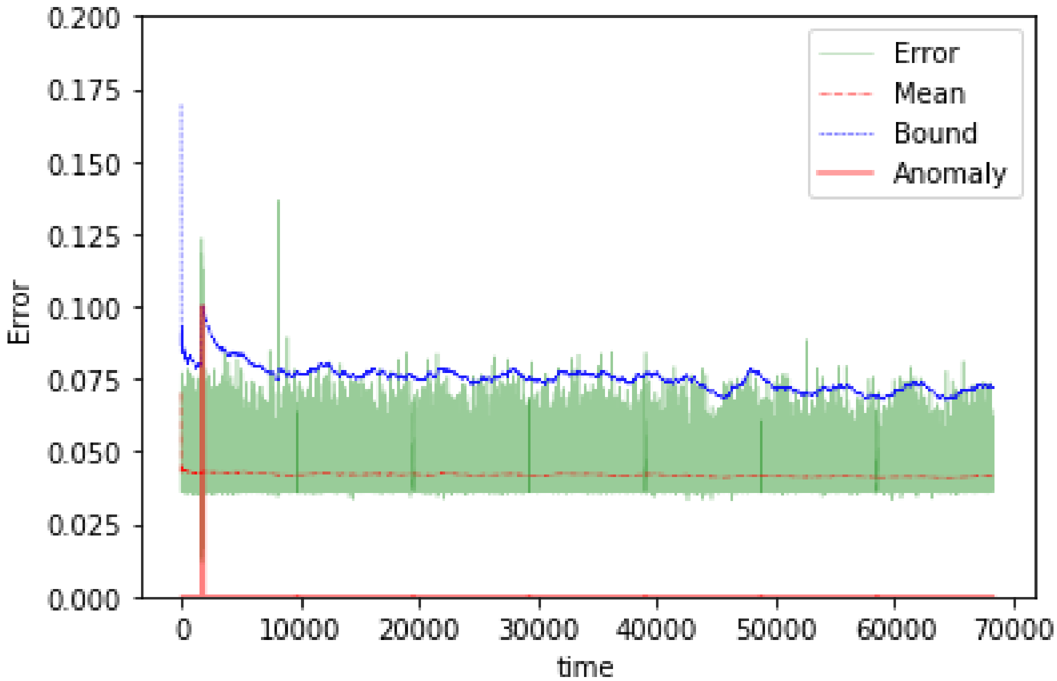

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



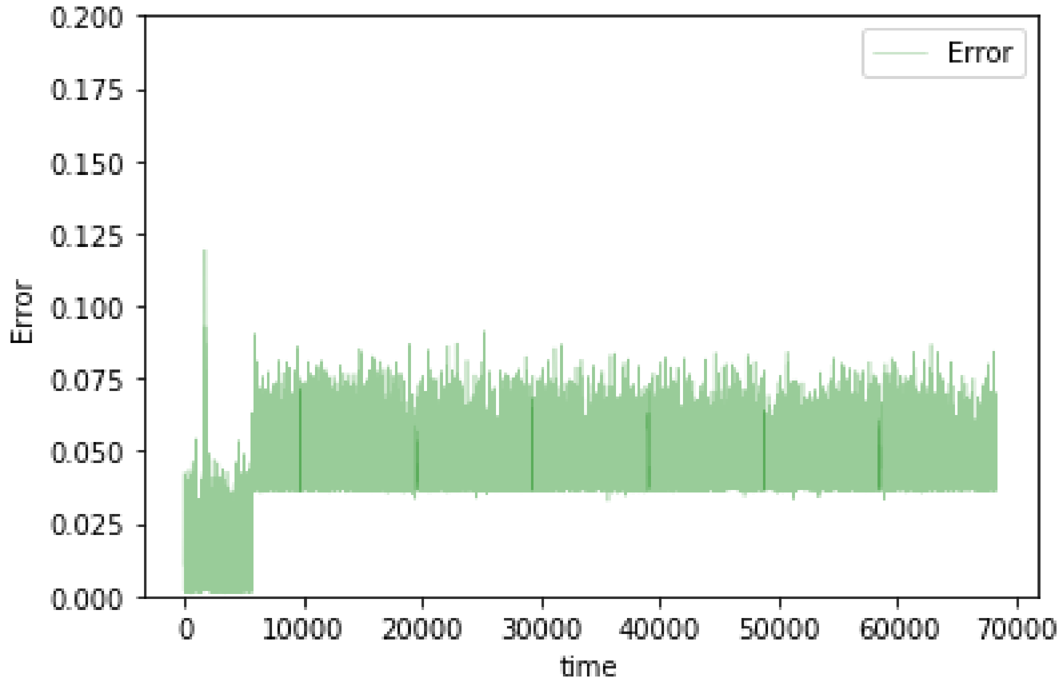

```
The mean error for nn1_10_disk_IO_start_ is 0.01822856948656902 for length 68289
Testing on Avg. load data.
```

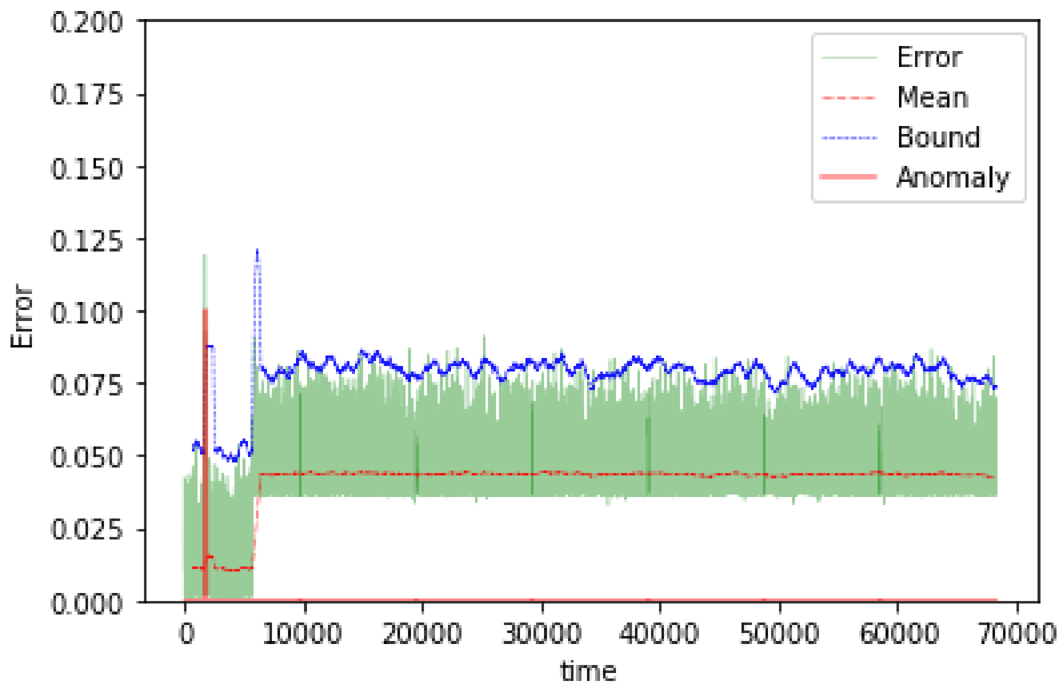


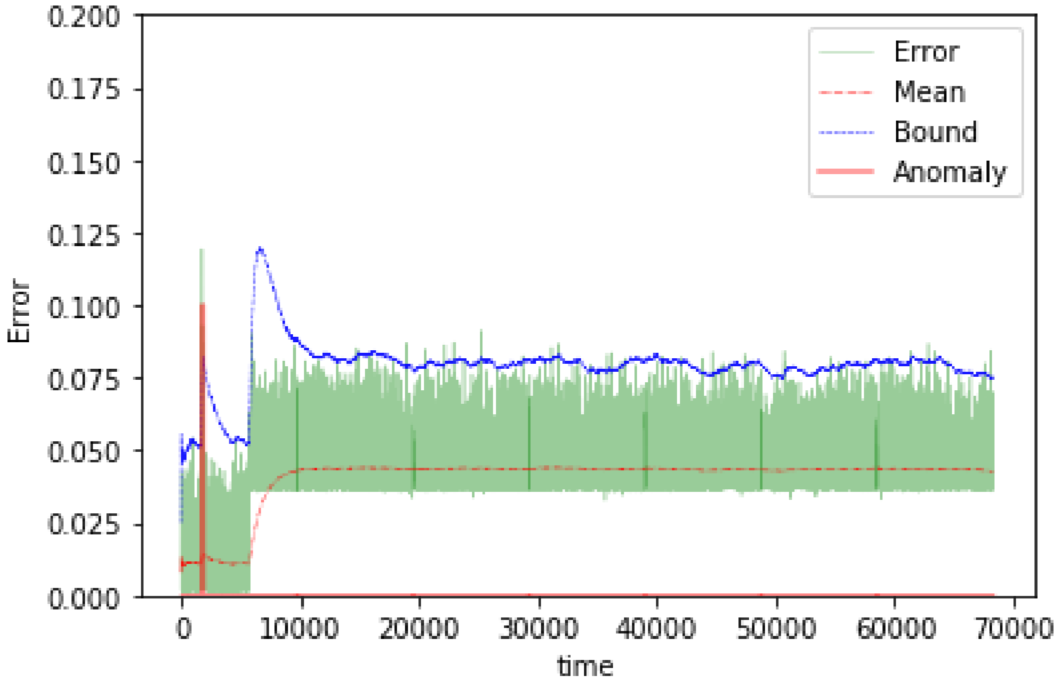

Error
Mean
Bound
Anomaly
Error
time


Error
Mean
Bound
Anomaly
Error
time

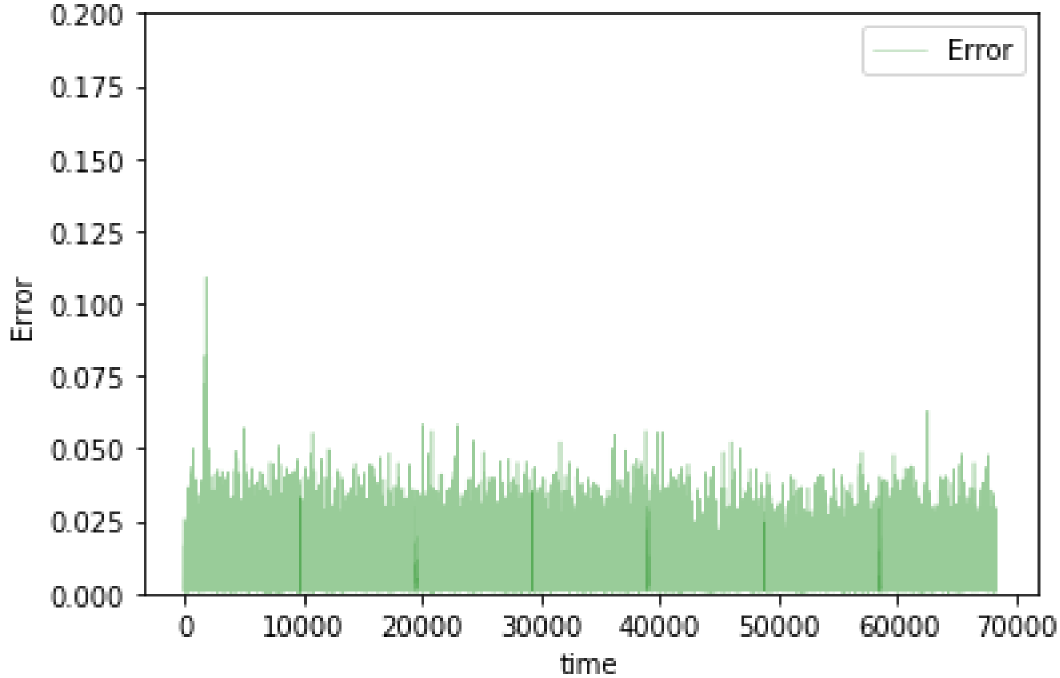
```
The mean error for nn1_10_avg_load_ is 0.041858292285943136 for length 68289
Testing on app change early data.
```



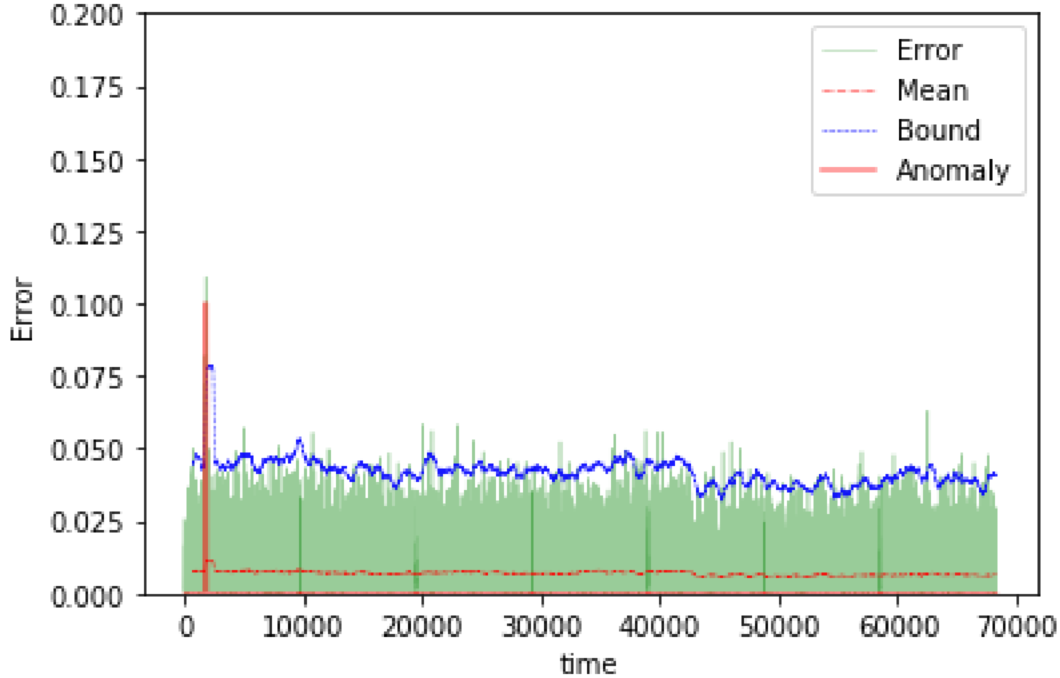




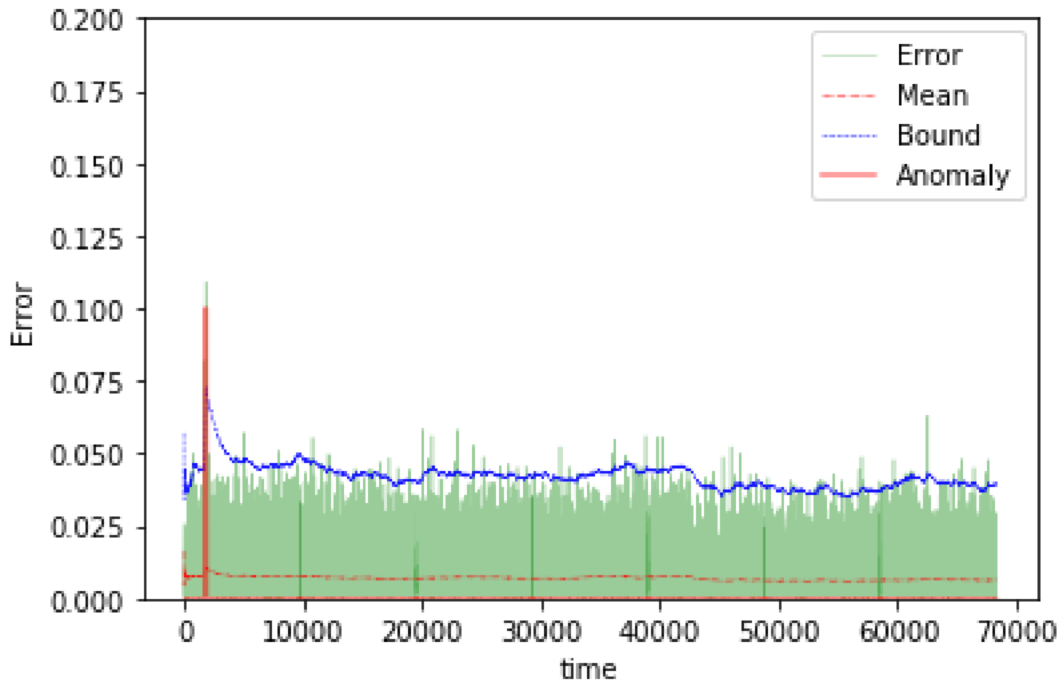

```
The mean error for nn1_10_app_change_early_ is 0.040859405738585715 for length 68289
Testing on Normal data.
```



Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



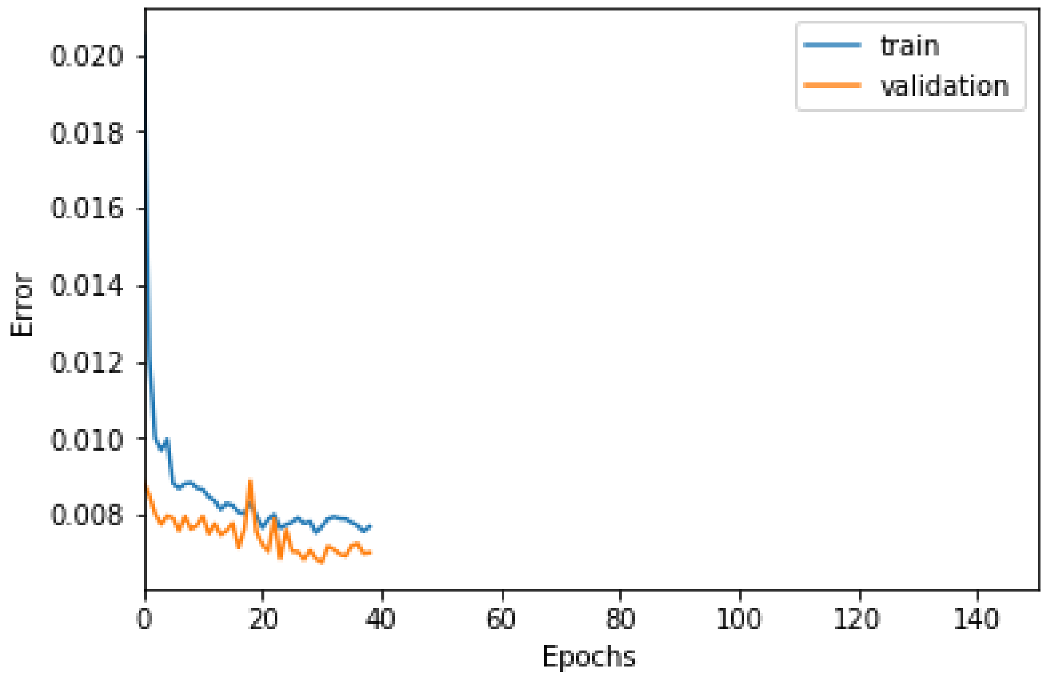
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



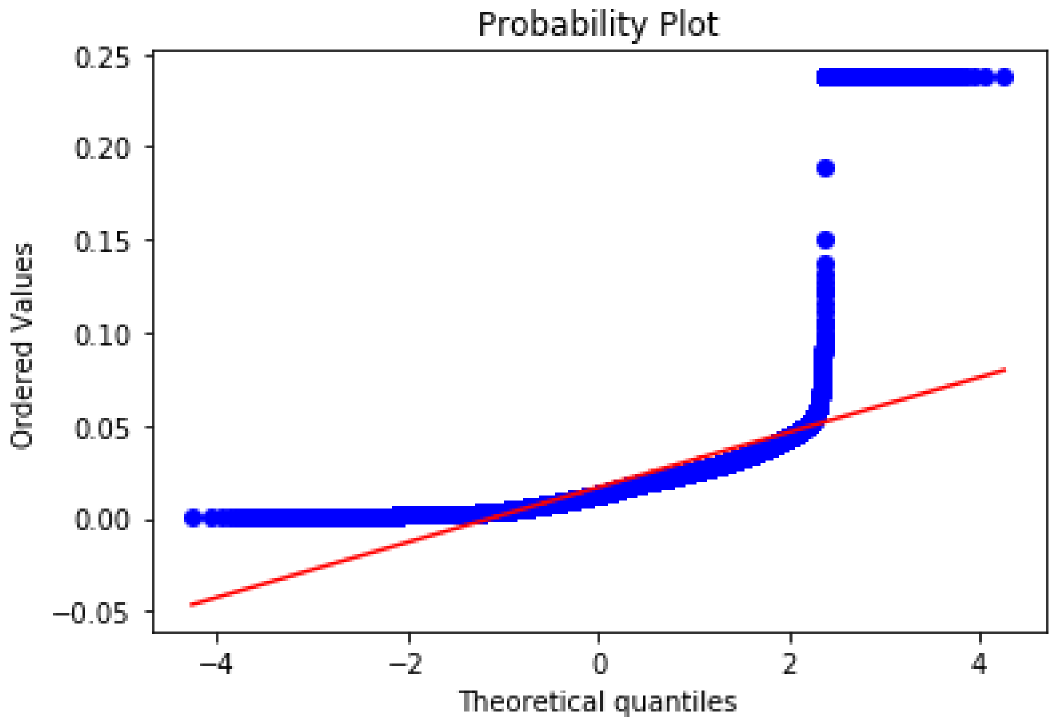

```
The mean error for nn1_10_normal_ is 0.0069526114440992895 for length 68289
====================
```

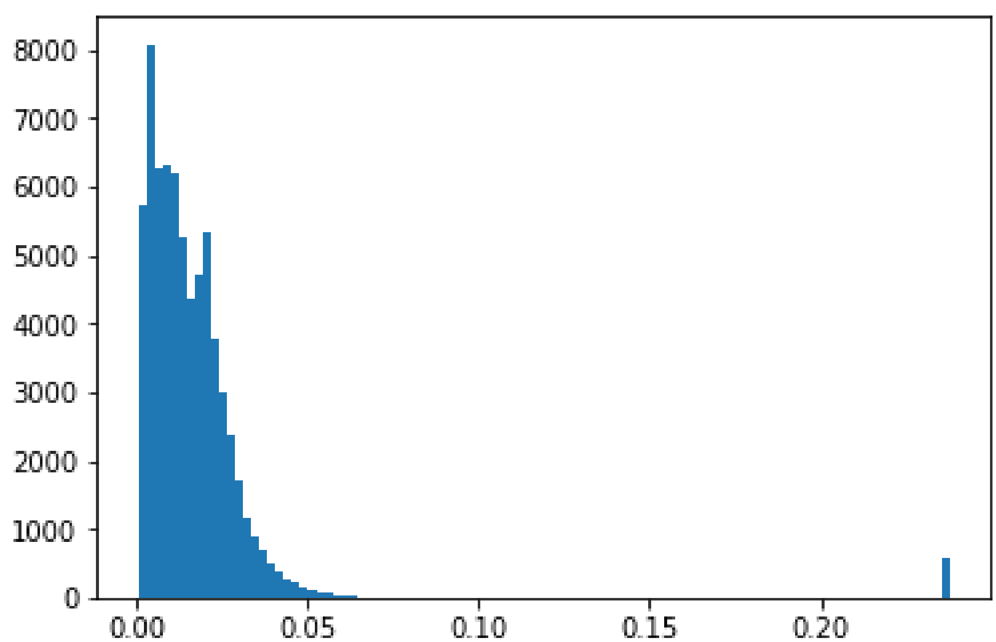
20 steps

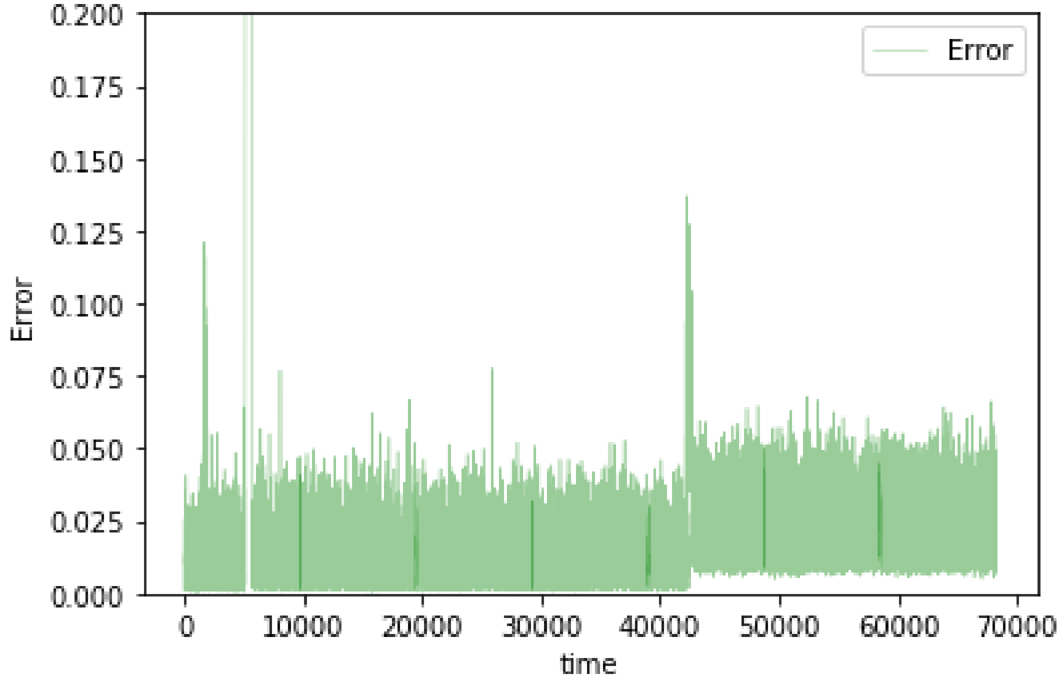
```
In [79]: TIMESTEPS = 20
         DIM = 29
         tgen = flat_generator(X, TIMESTEPS)
         vgen = flat_generator(val_X, TIMESTEPS)
         name = "nn1_20"

In [80]: input_layer = Input(shape=(TIMESTEPS*DIM,))
         hidden = Dense(100,activation='relu')(input_layer)
         output = Dense(DIM, activation='sigmoid')(hidden)

In [81]: model = Model(input_layer, output)
         model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [82]: train(model, tgen, vgen, name=name)
         test(model, name=name, window=TIMESTEPS)
```



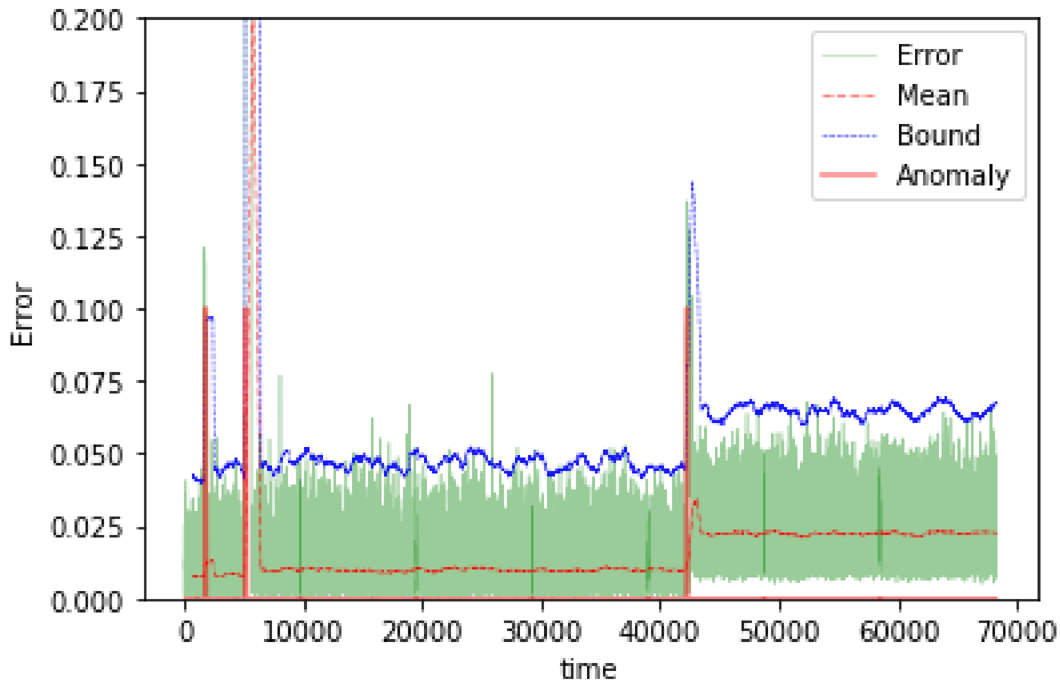

```
Training loss for final epoch is 0.0076809632552322
Validation loss for final epoch is 0.007003320118645206
---------- Beginning tests for nn1_20 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−4
−2
0
2
4

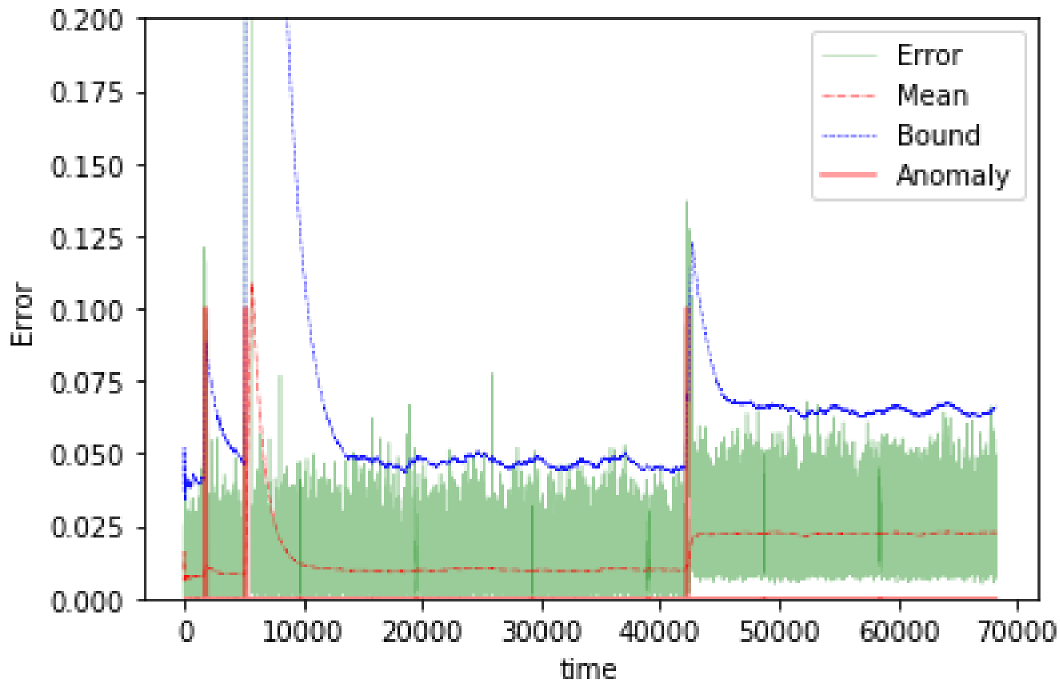

8000
7000
6000
5000
4000
3000
2000
1000
0
0.00
0.05
0.10
0.15
0.20



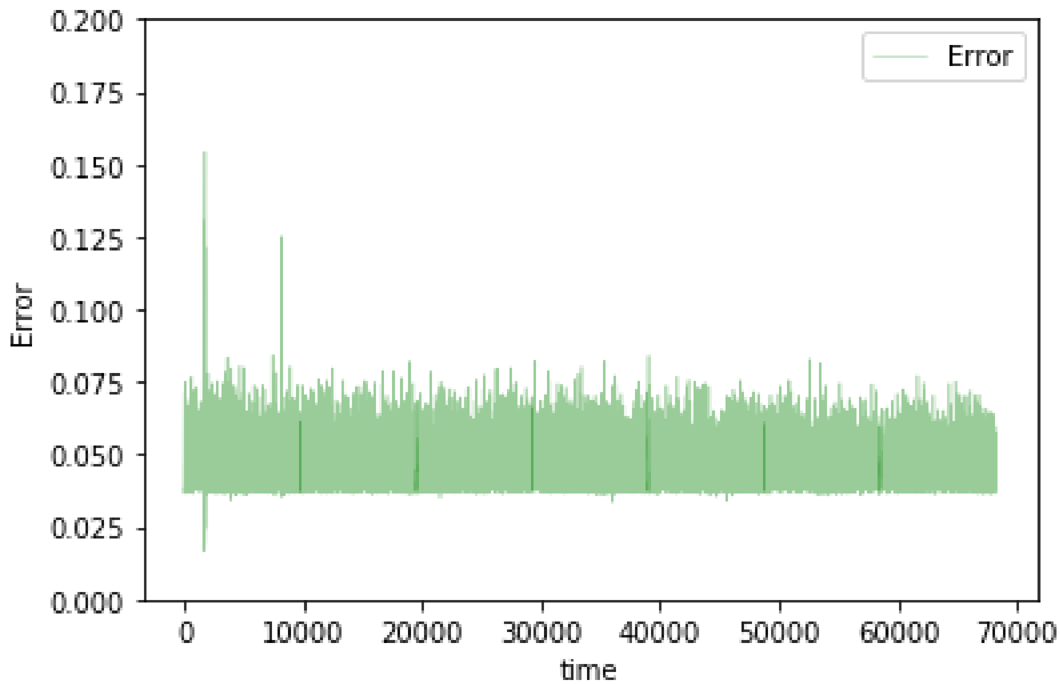
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

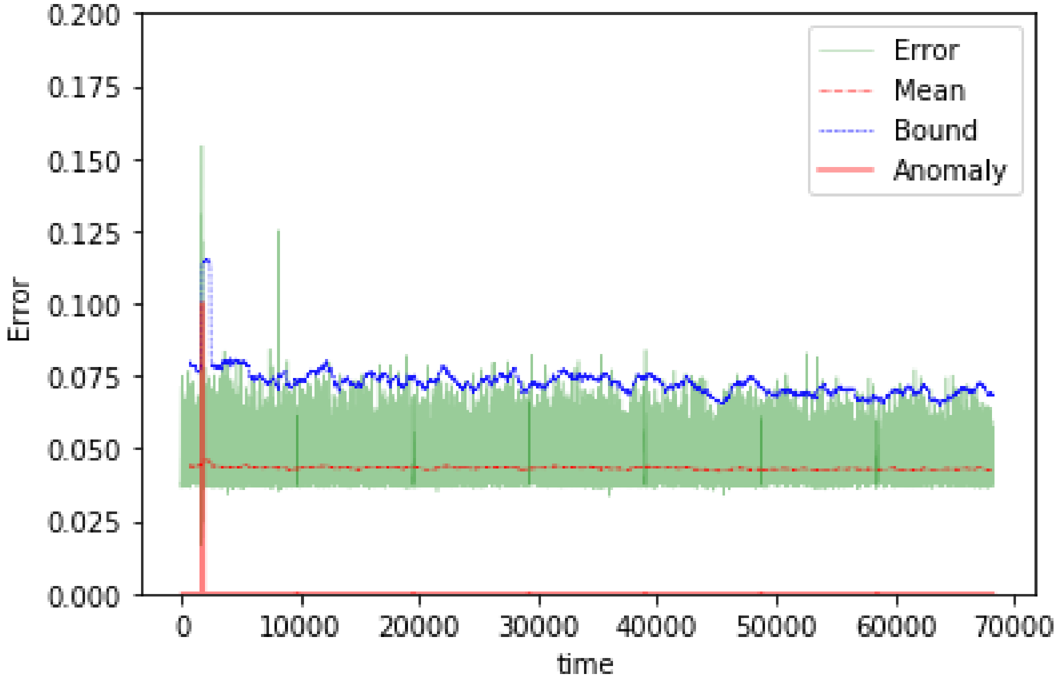


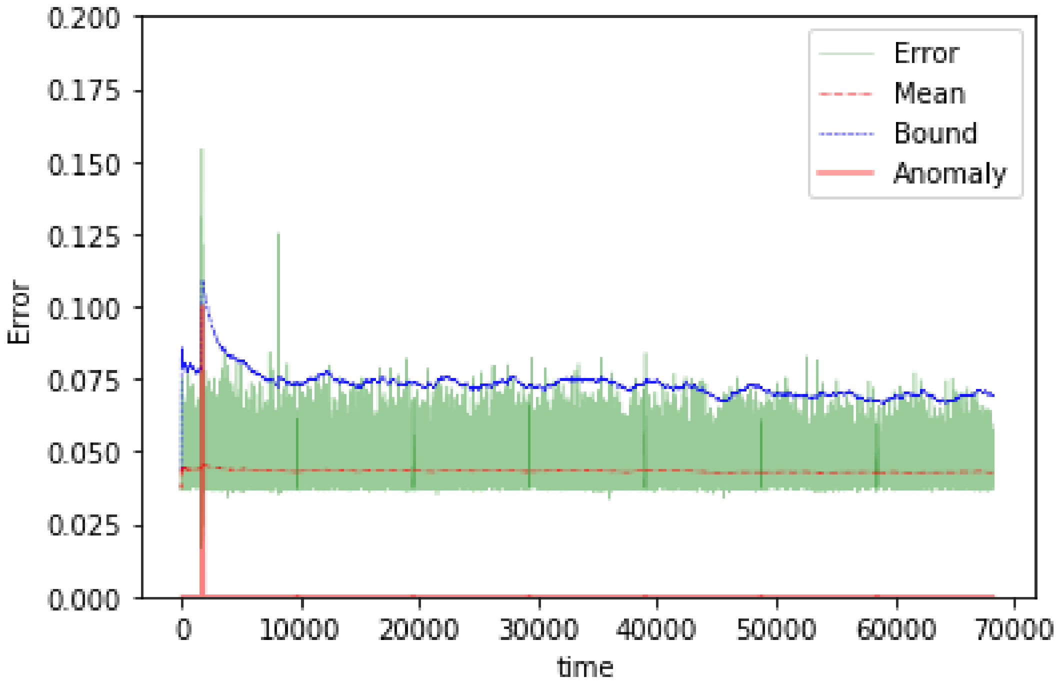
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



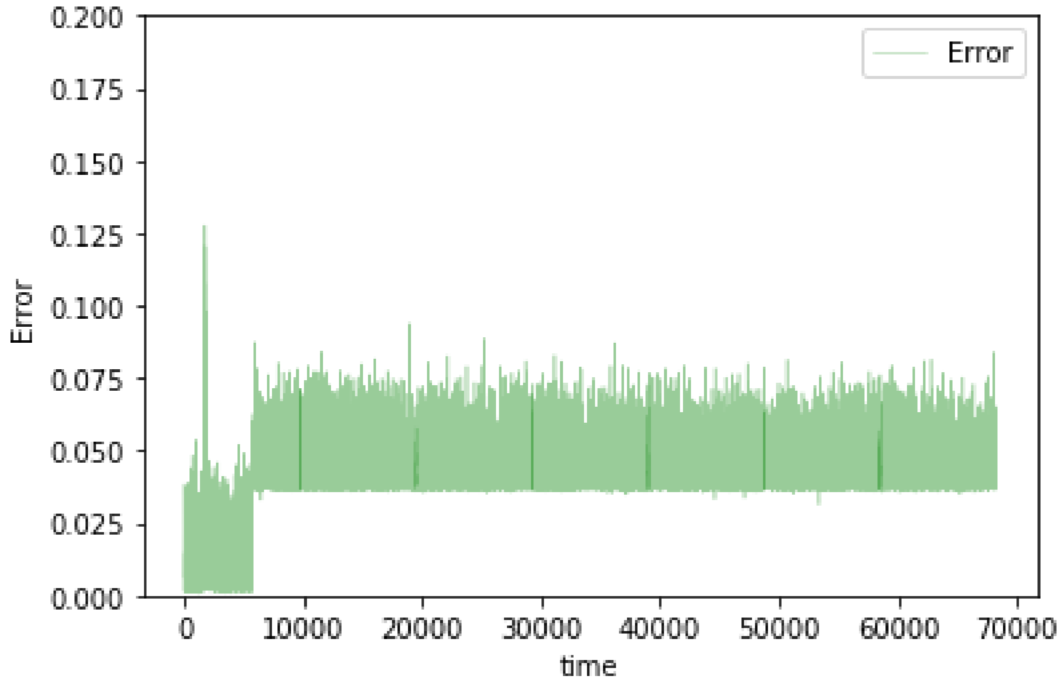

```
The mean error for nn1_20_disk_IO_start_ is 0.016673588497141966 for length 68279
Testing on Avg. load data.
```

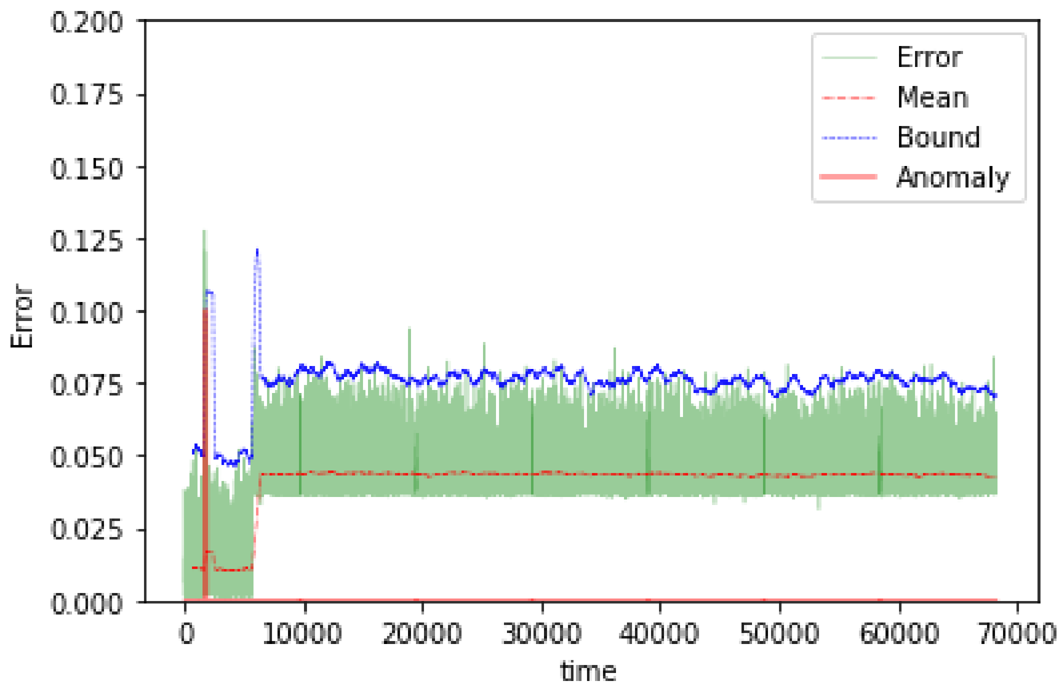


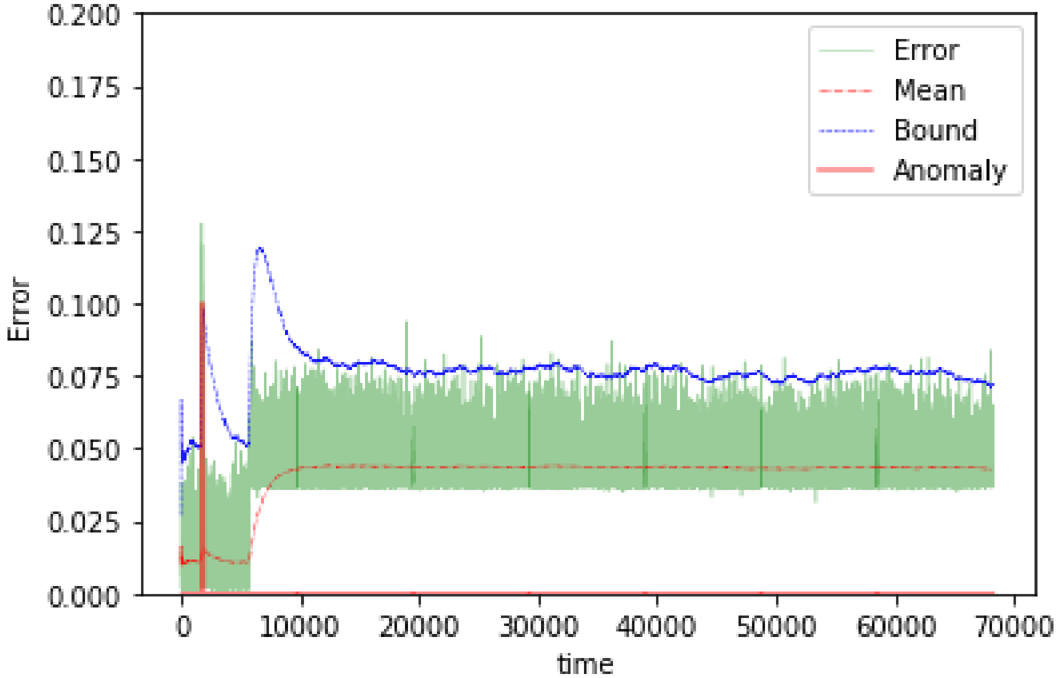

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

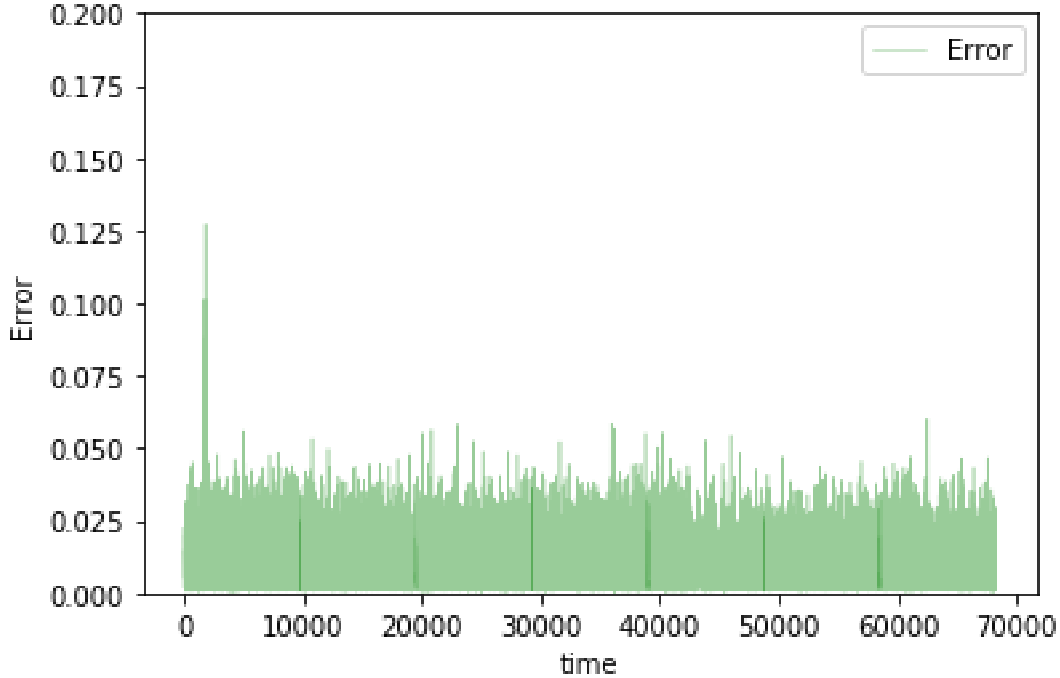
```
The mean error for nn1_20_avg_load_ is 0.04324083160607533 for length 68279
Testing on app change early data.
```

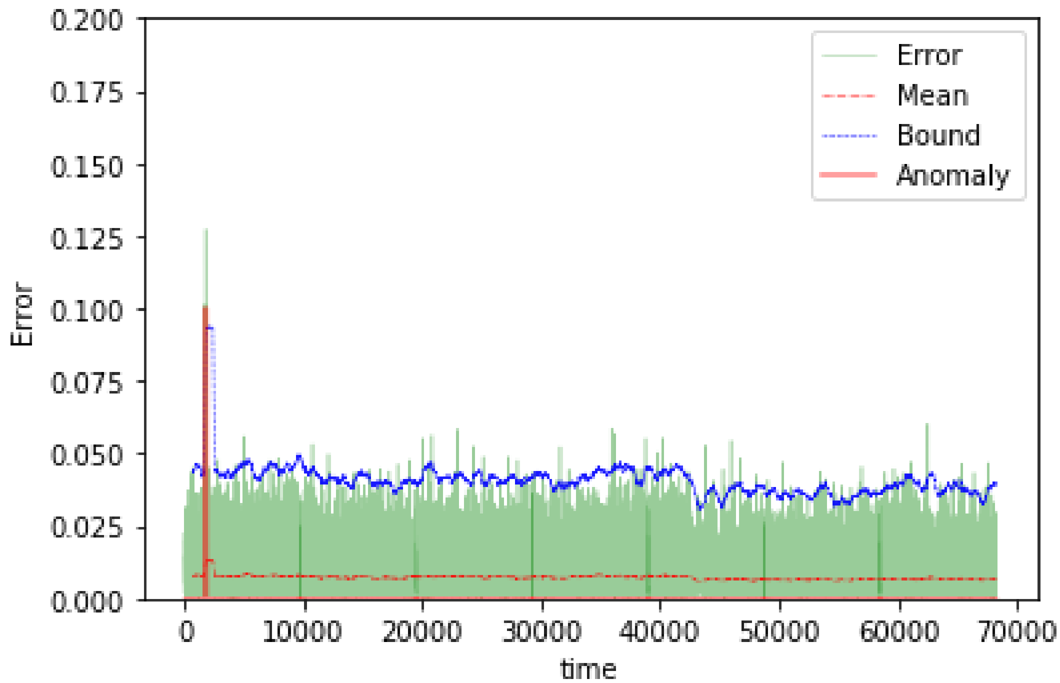


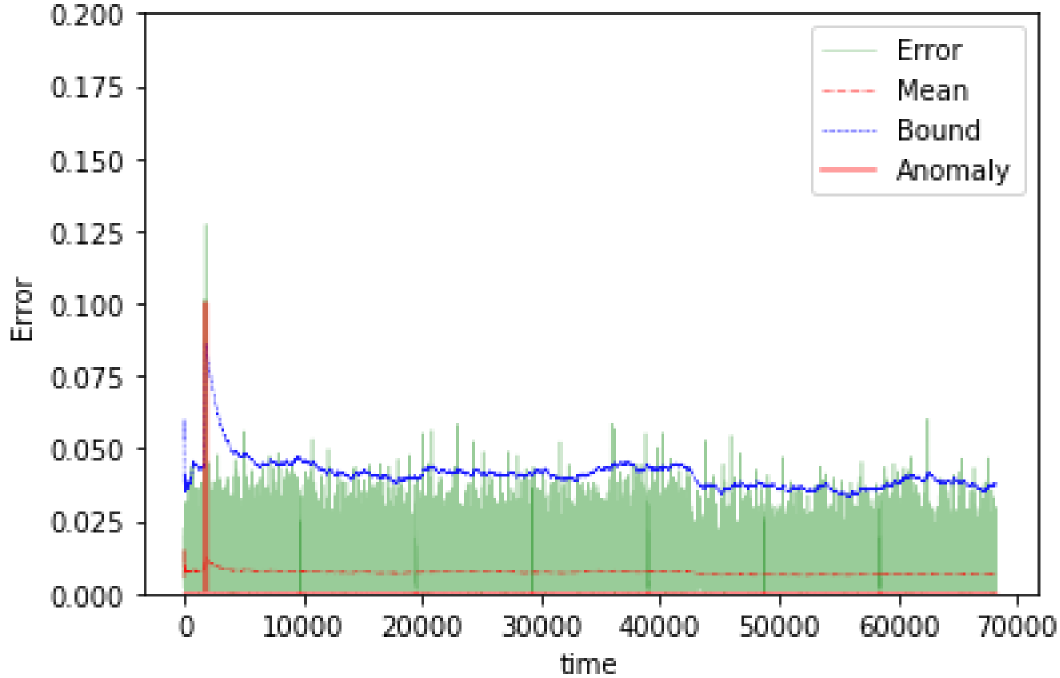





```
The mean error for nn1_20_app_change_early_ is 0.04084225237956082 for length 68279
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

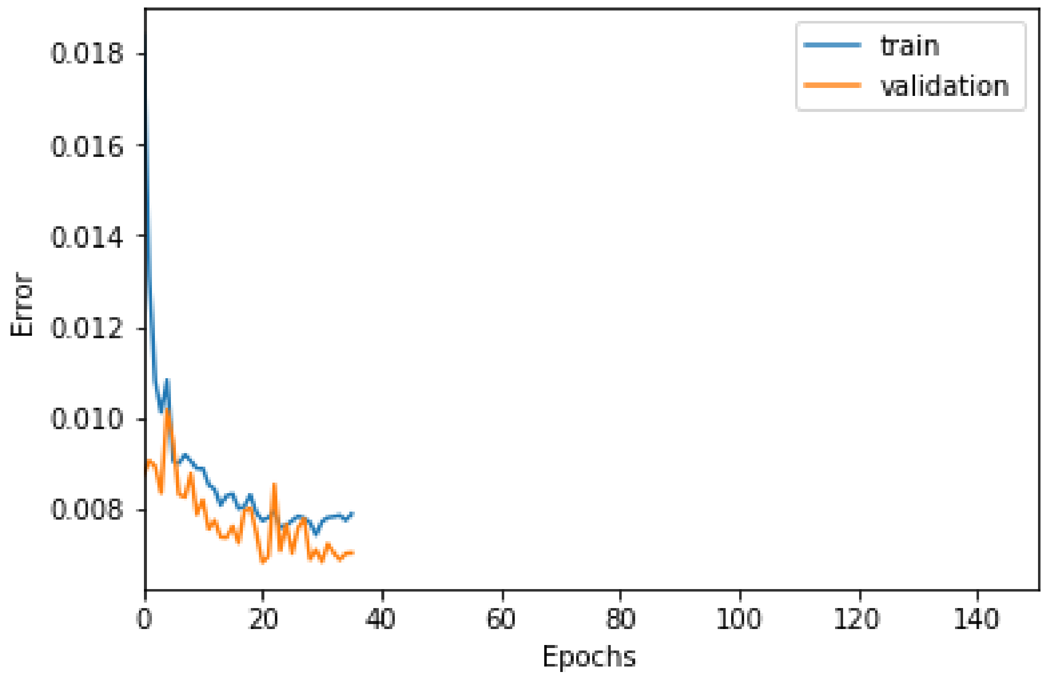

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



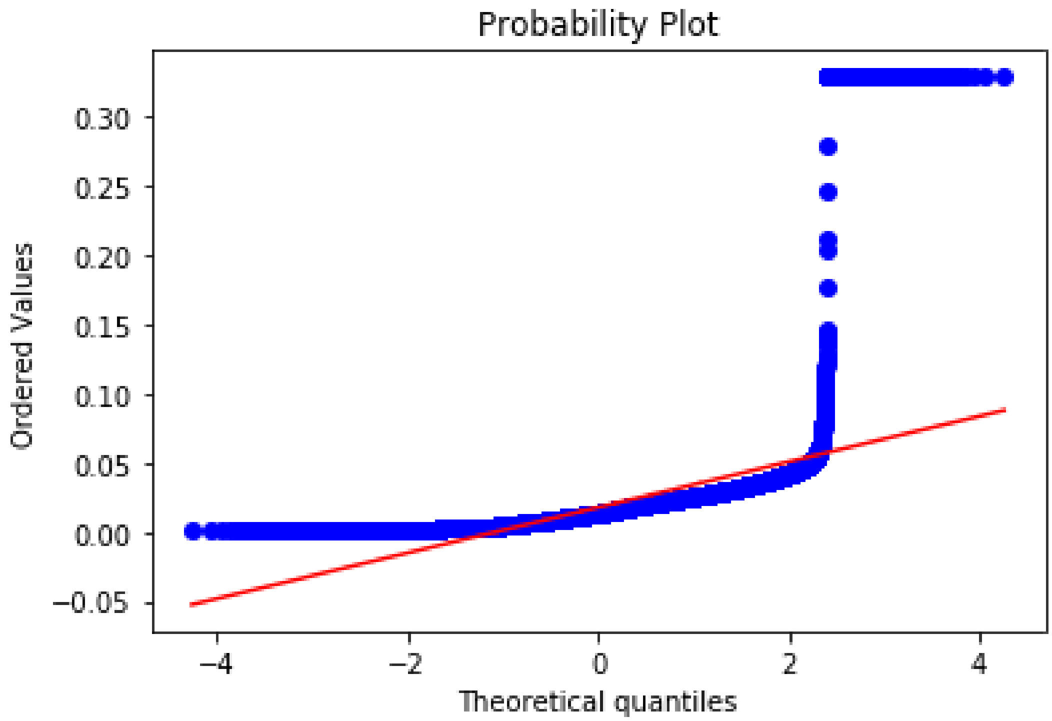

```
The mean error for nn1_20_normal_ is 0.007236831576064556 for length 68279
====================
```

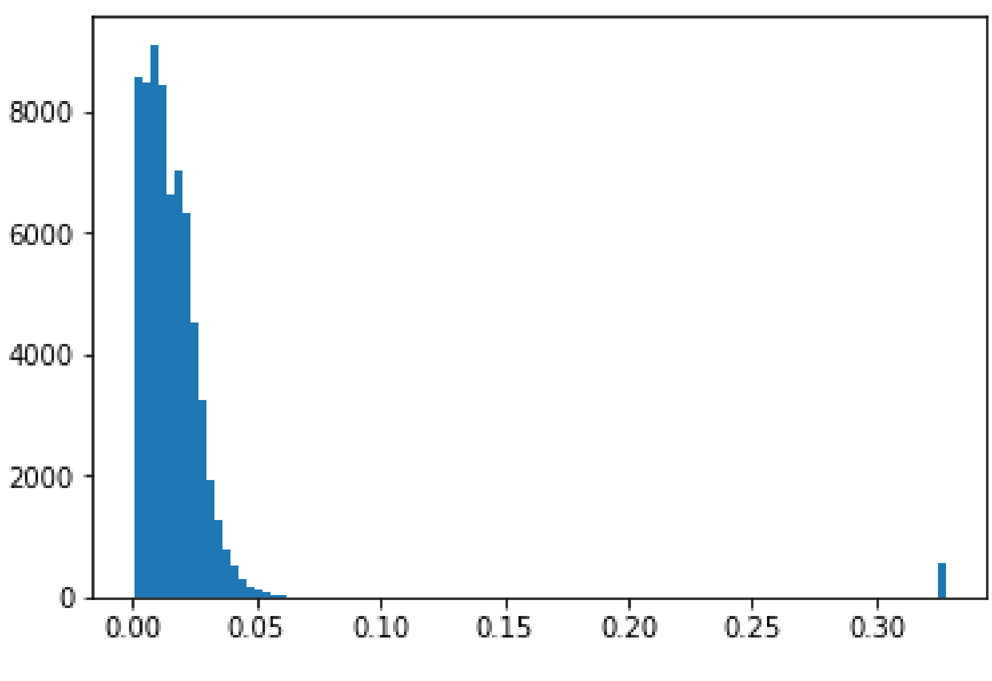
50 steps

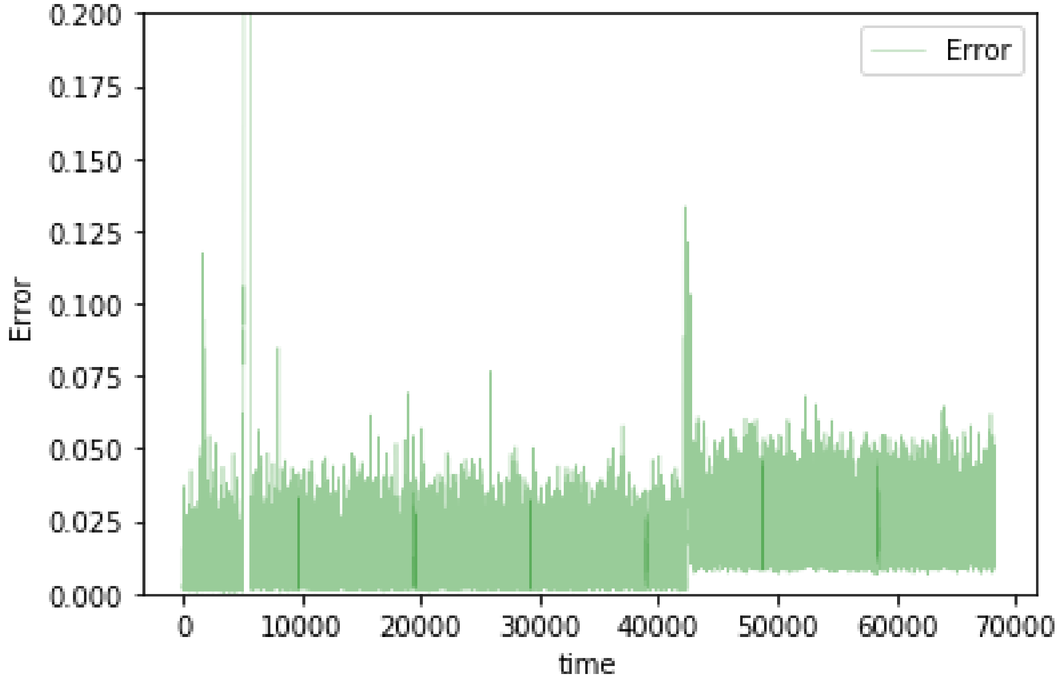
```
In [83]: TIMESTEPS = 50
         DIM = 29
         tgen = flat_generator(X, TIMESTEPS)
         vgen = flat_generator(val_X, TIMESTEPS)
         name = "nn1_50"

In [84]: input_layer = Input(shape=(TIMESTEPS*DIM,))
         hidden = Dense(100,activation='relu')(input_layer)
         output = Dense(DIM, activation='sigmoid')(hidden)

In [85]: model = Model(input_layer, output)
         model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [86]: train(model, tgen, vgen, name=name)
         test(model, name=name, window=TIMESTEPS)
```

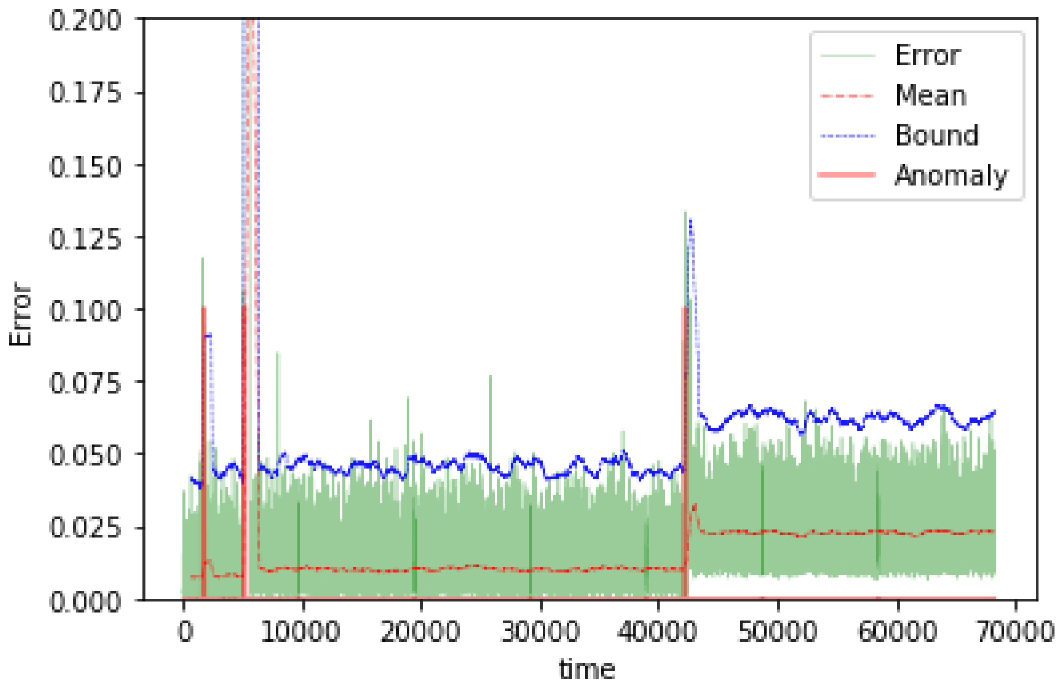


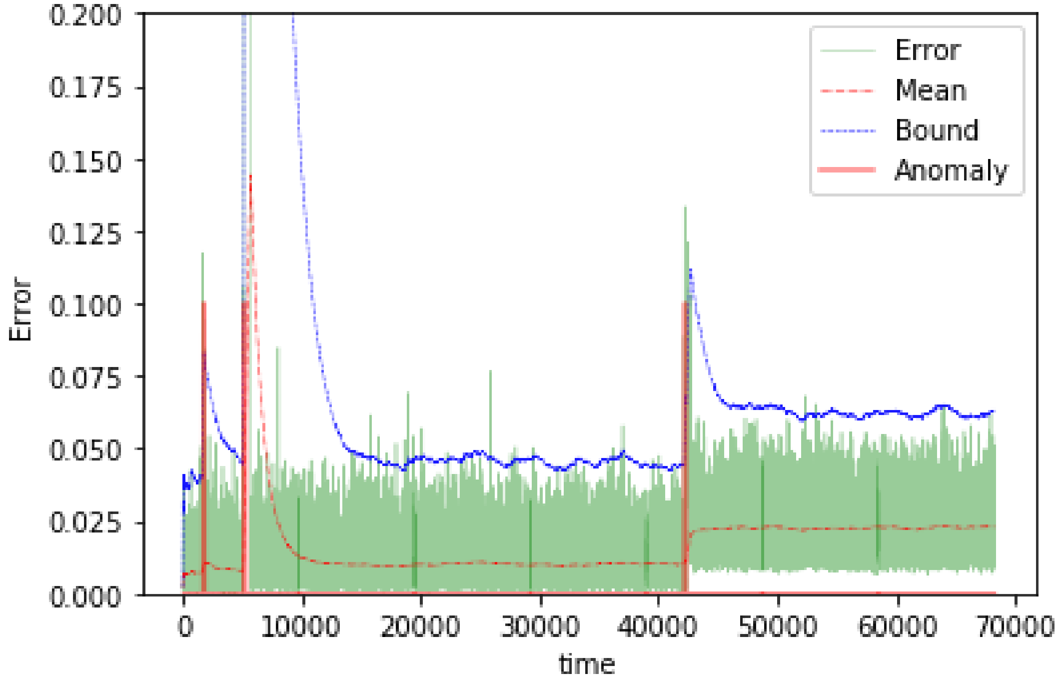

```
Training loss for final epoch is 0.007885990605223924
Validation loss for final epoch is 0.00703394577500876
---------- Beginning tests for nn1_50 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−4
−2
0
2
4

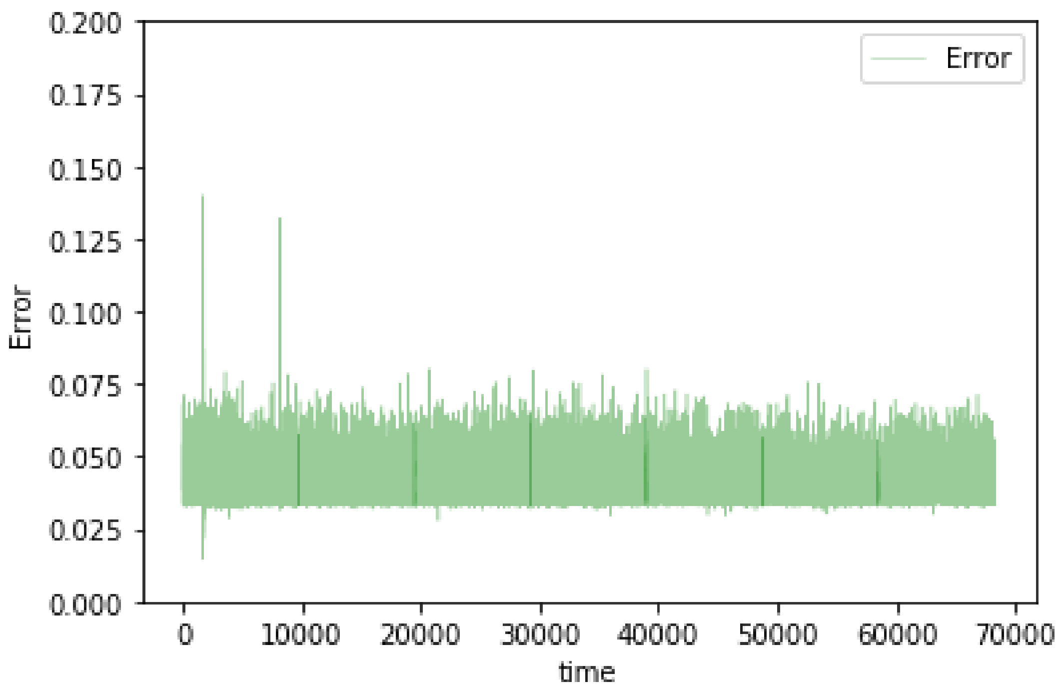

8000
6000
4000
2000
0
0.00
0.05
0.10
0.15
0.20
0.25
0.30

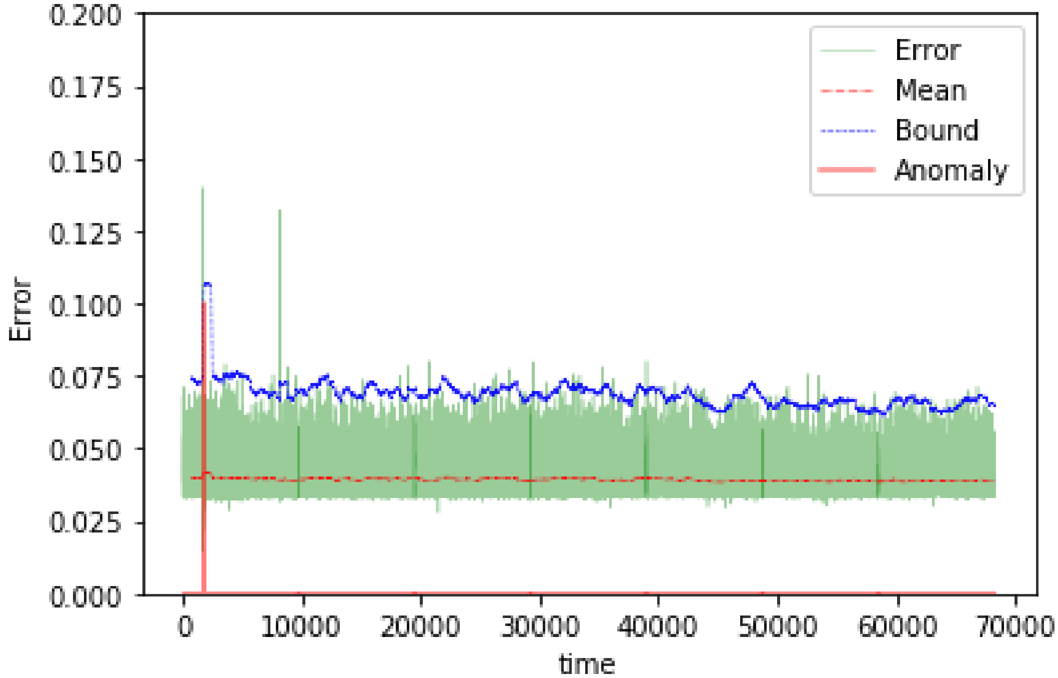

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

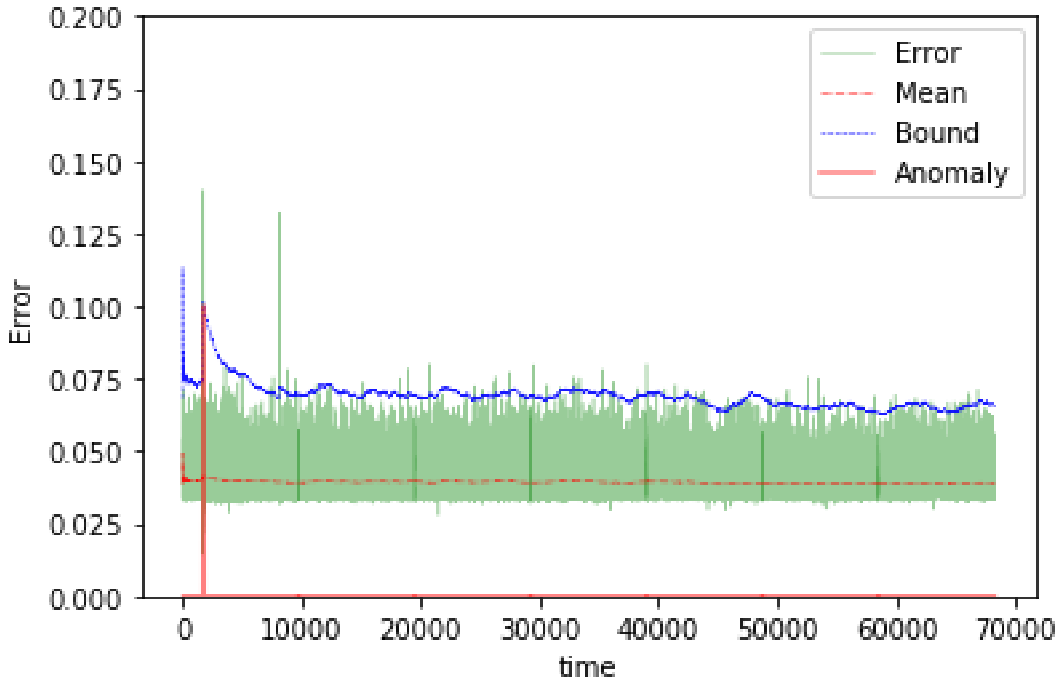

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



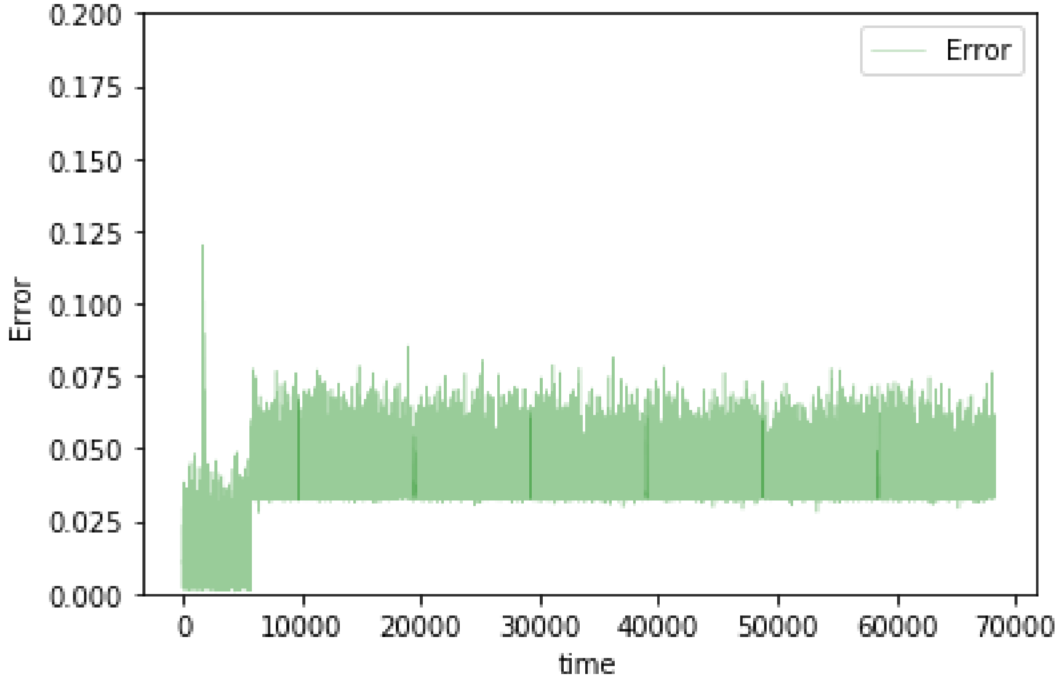

```
The mean error for nn1_50_disk_IO_start_ is 0.01754085910611449 for length 68249
Testing on Avg. load data.
```

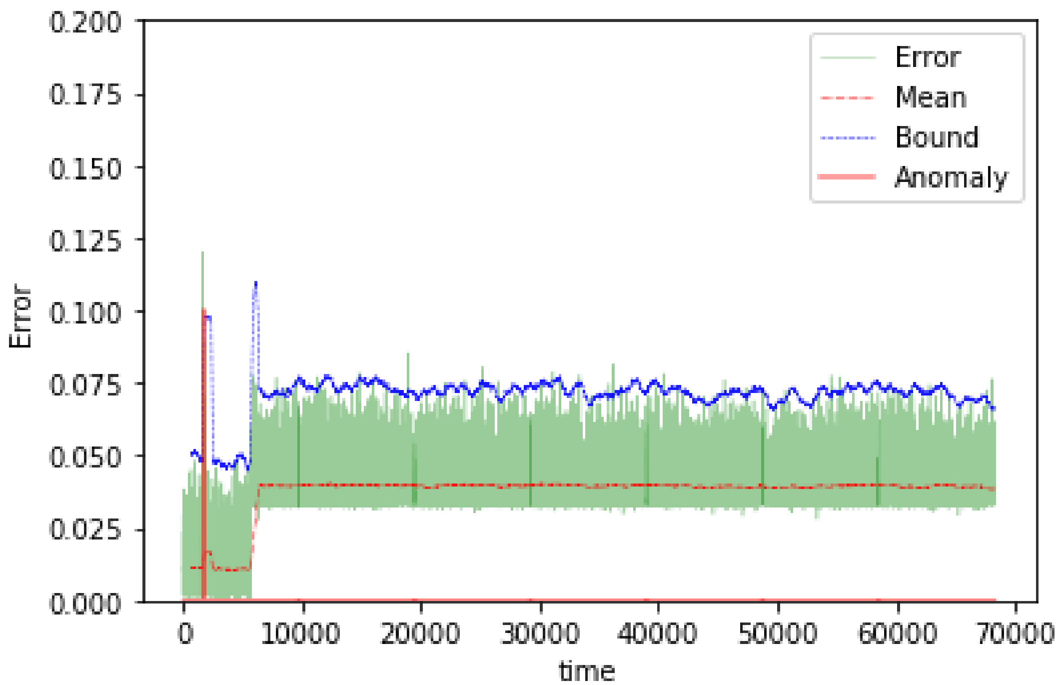


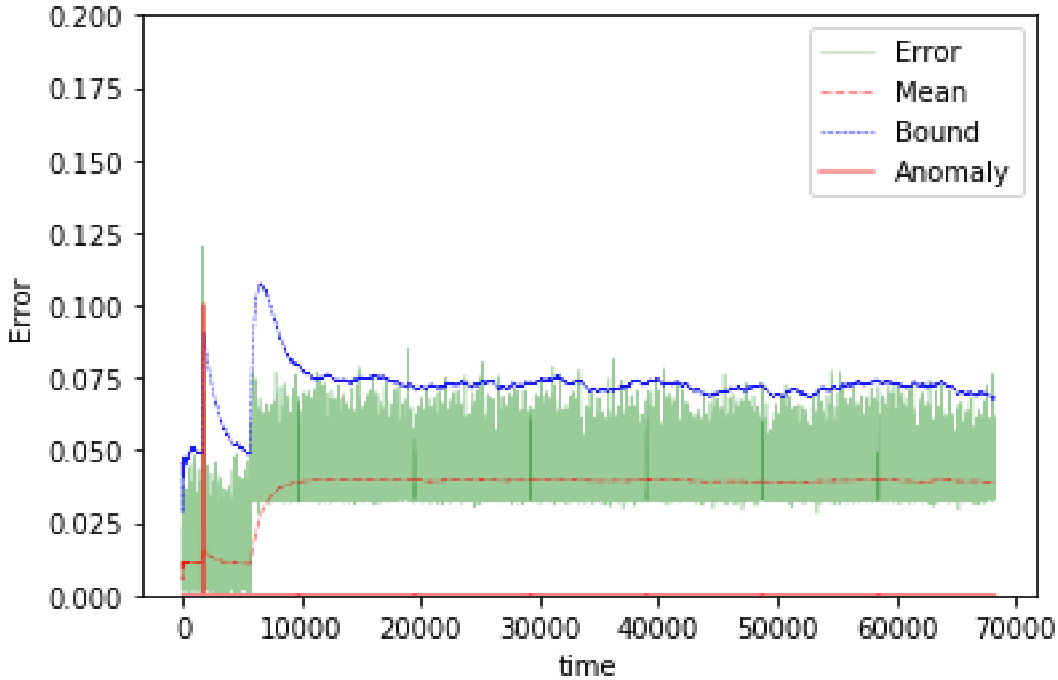

Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000


Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

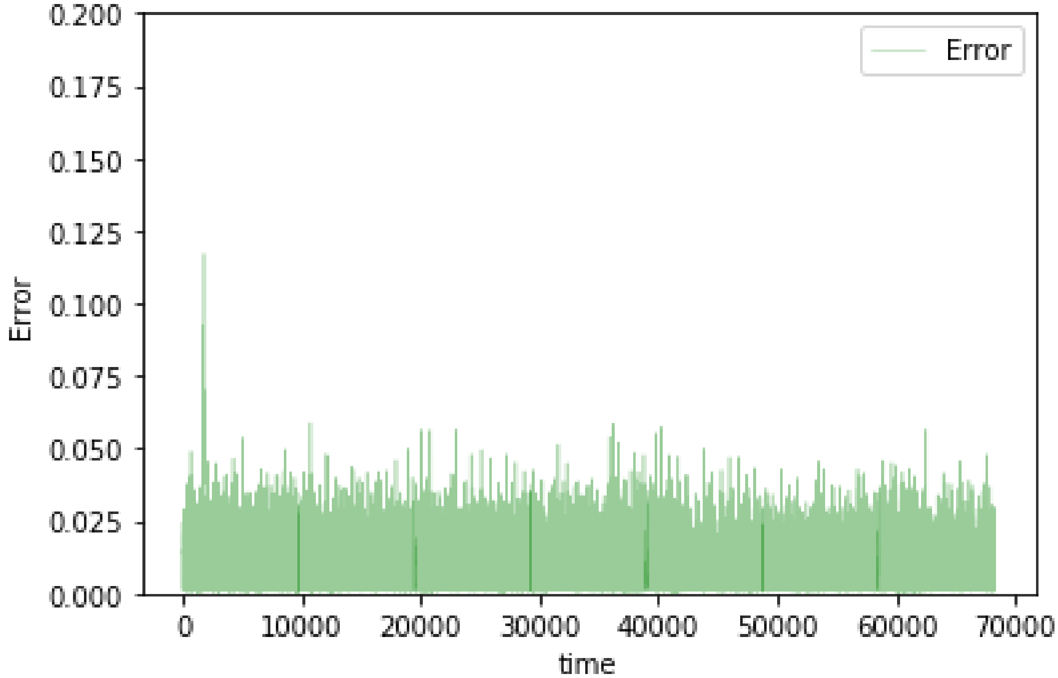
```
The mean error for nn1_50_avg_load_ is 0.039297765406708506 for length 68249
Testing on app change early data.
```

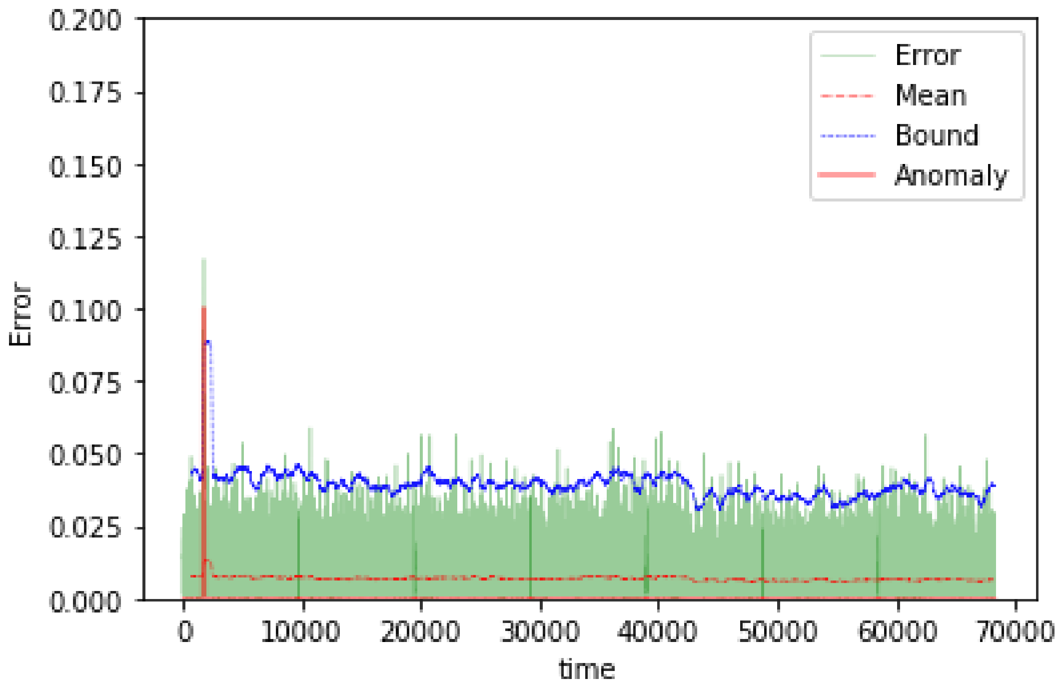


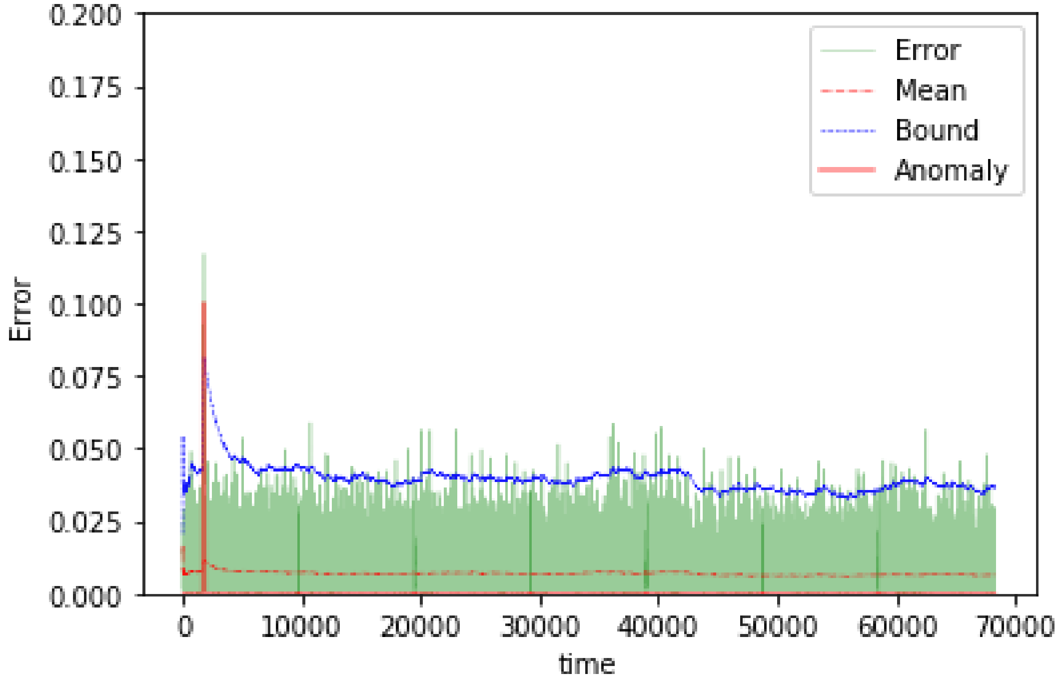





```
The mean error for nn1_50_app_change_early_ is 0.0371985712667431 for length 68249
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

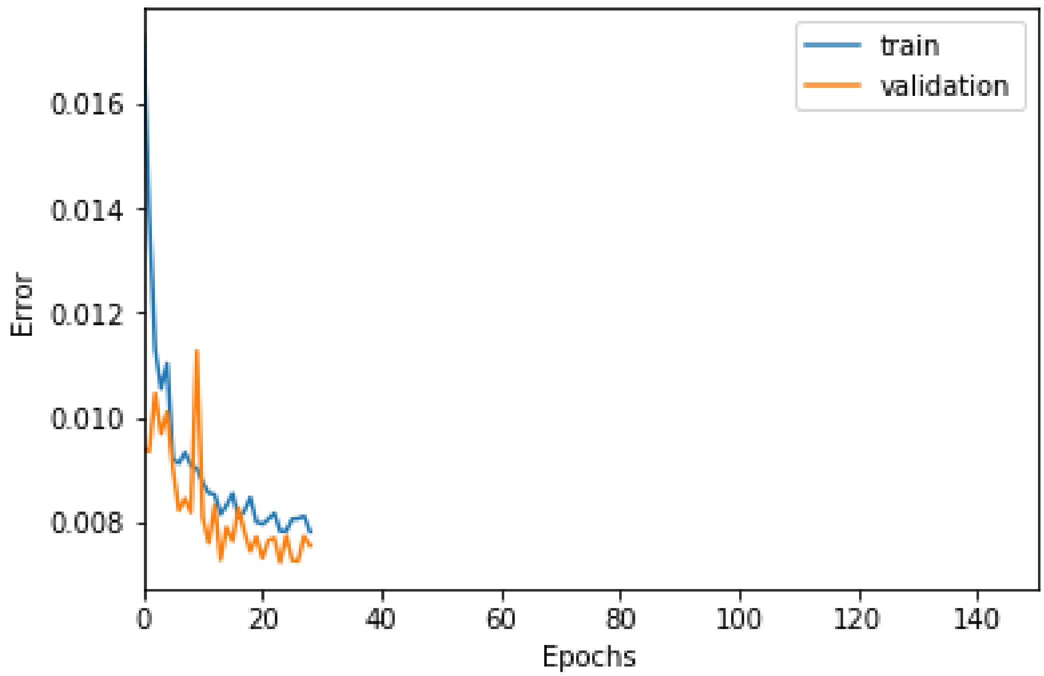

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



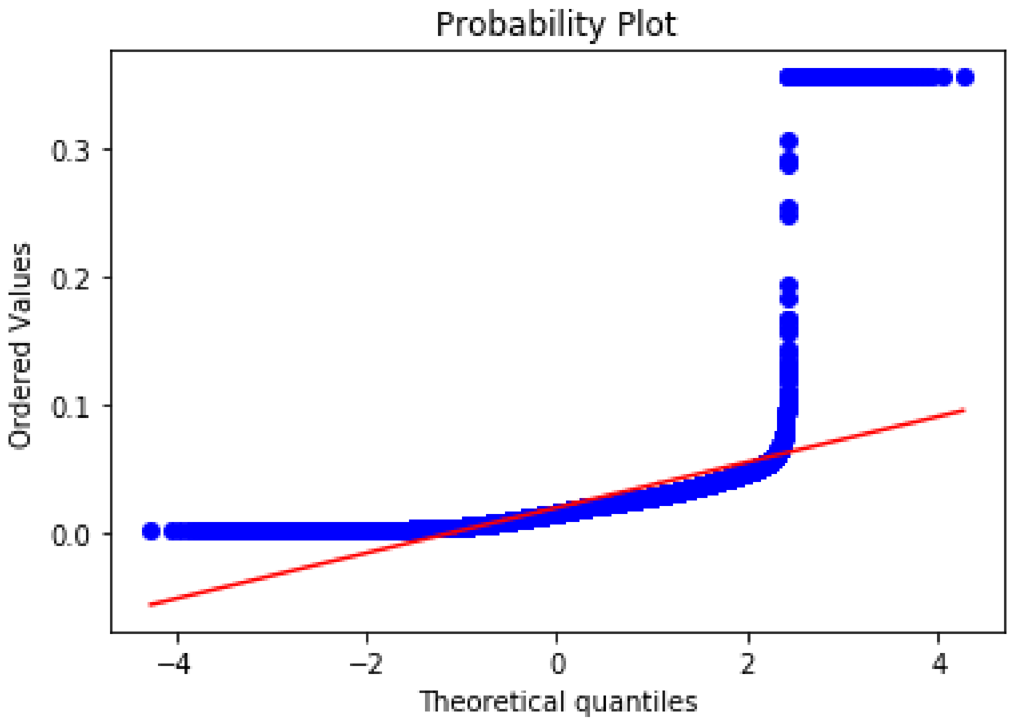

```
The mean error for nn1_50_normal_ is 0.006860234535571892 for length 68249
====================
```

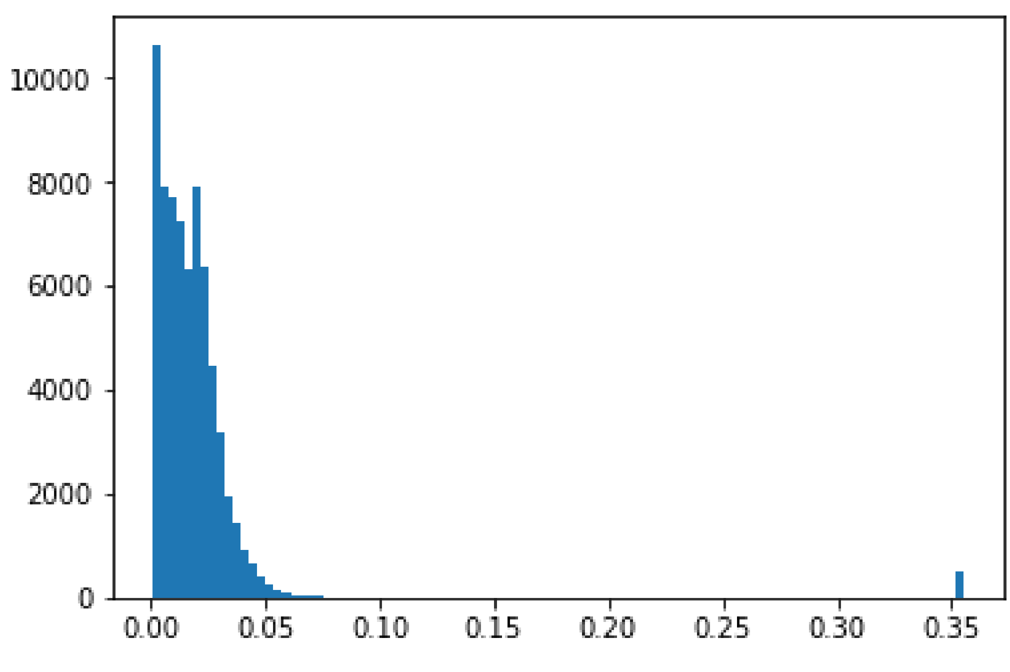
100 steps

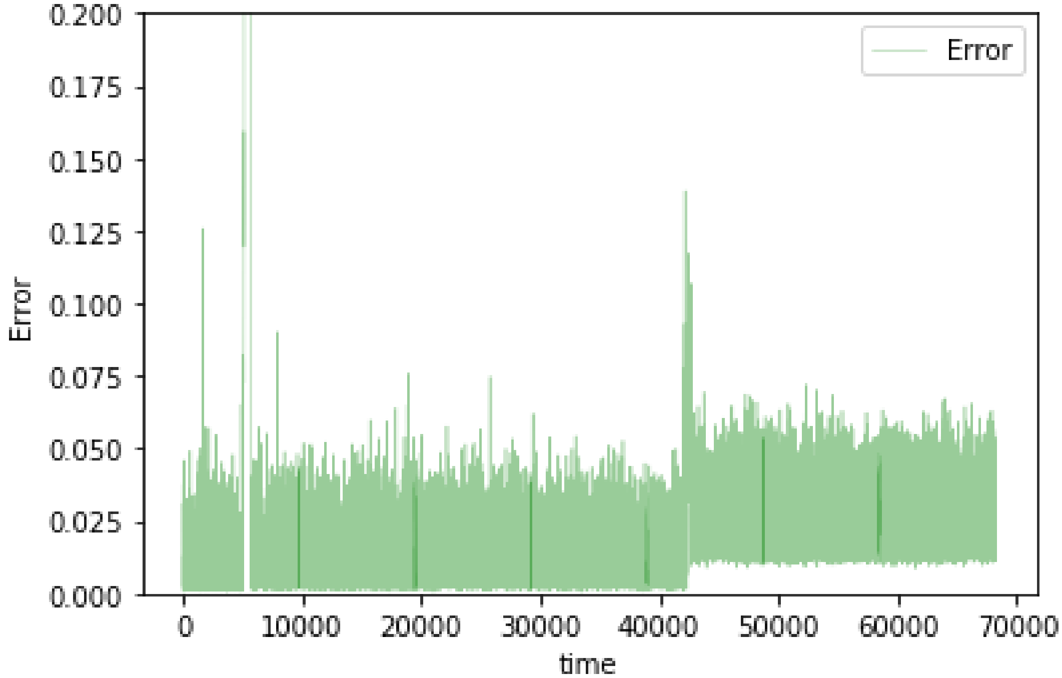
```
In [87]: TIMESTEPS = 100
         DIM = 29
         tgen = flat_generator(X, TIMESTEPS)
         vgen = flat_generator(val_X, TIMESTEPS)
         name = "nn1_100"

In [88]: input_layer = Input(shape=(TIMESTEPS*DIM,))
         hidden = Dense(100,activation='relu')(input_layer)
         output = Dense(DIM, activation='sigmoid')(hidden)

In [89]: model = Model(input_layer, output)
         model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [90]: train(model, tgen, vgen, name=name)
         test(model, name=name, window=TIMESTEPS)
```

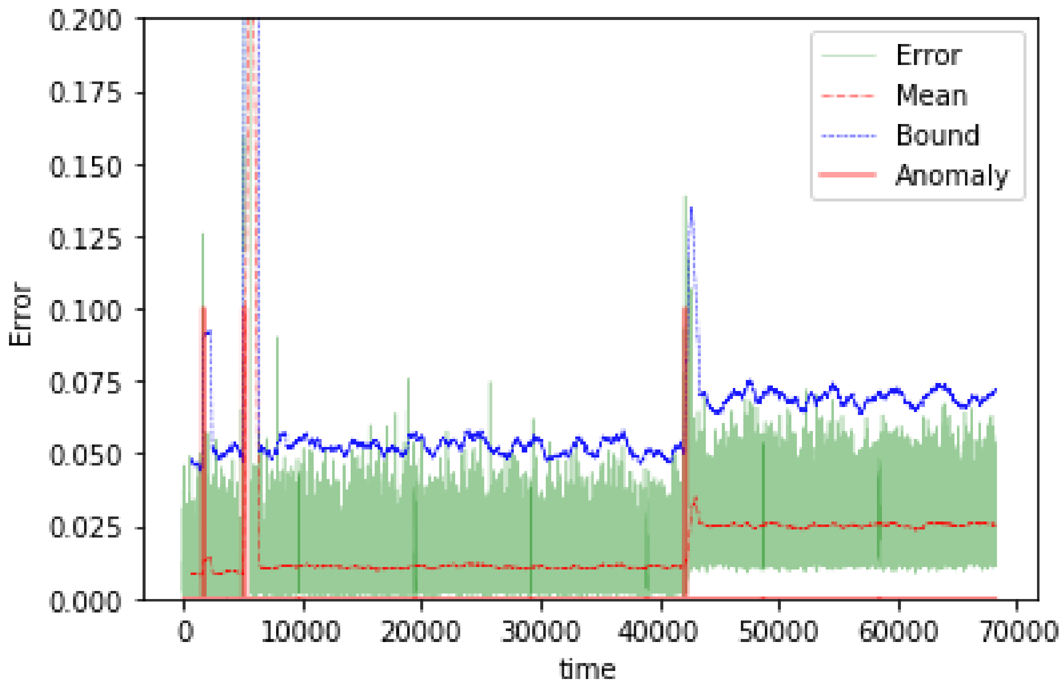


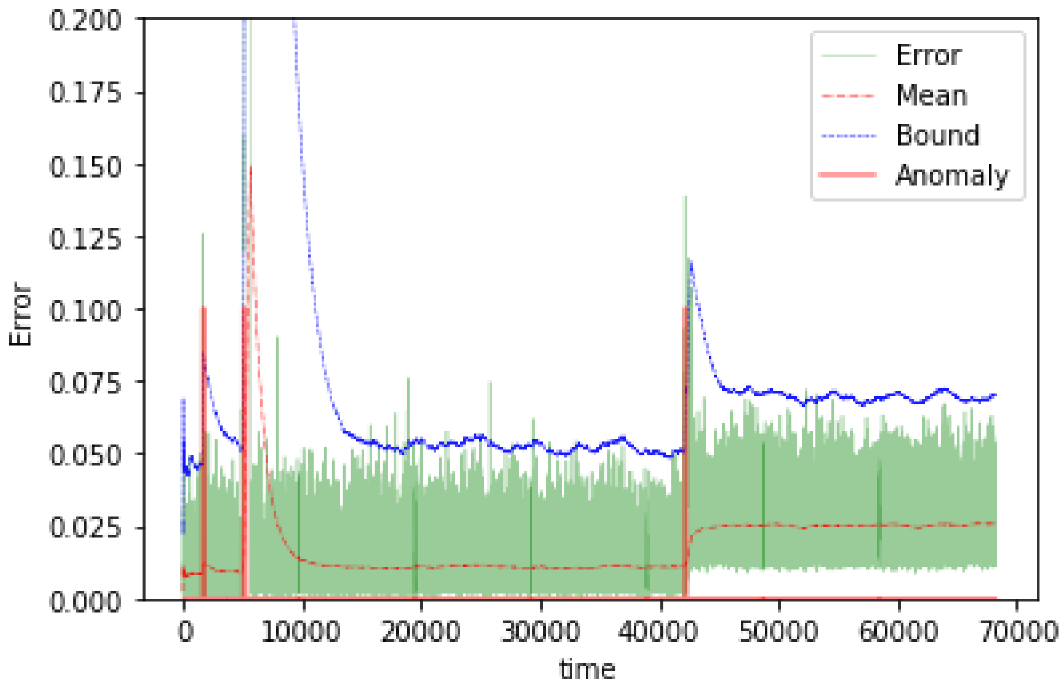

```
Training loss for final epoch is 0.007813127898261883
Validation loss for final epoch is 0.007548550087958575
---------- Beginning tests for nn1_100 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.3
0.2
0.1
0.0
−4
−2
0
2
4

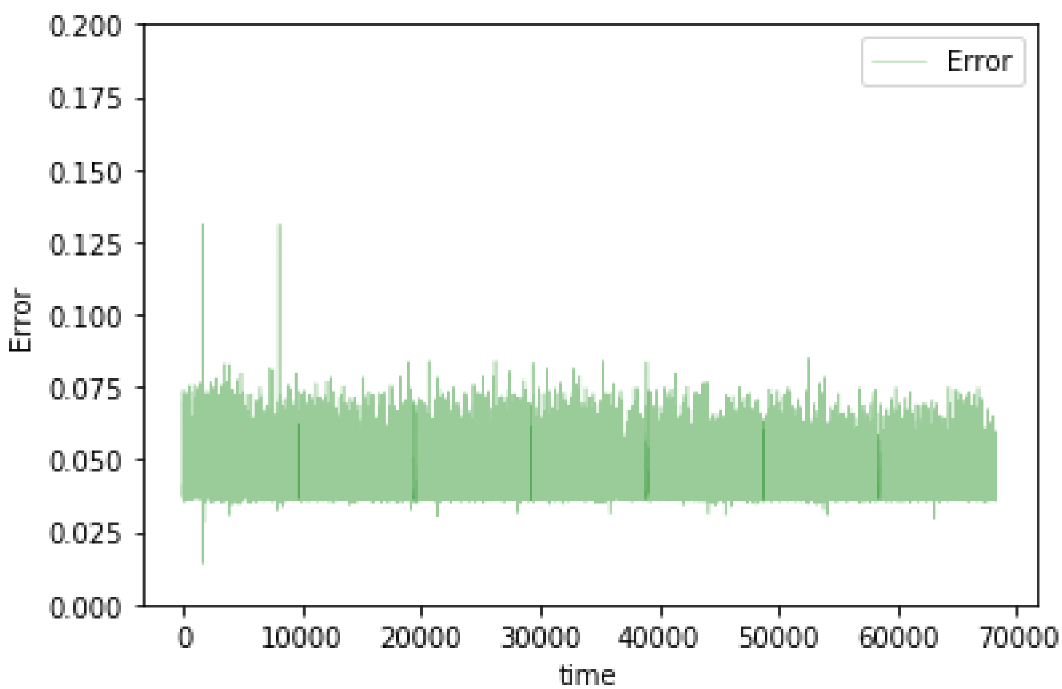

10000
8000
6000
4000
2000
0
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35

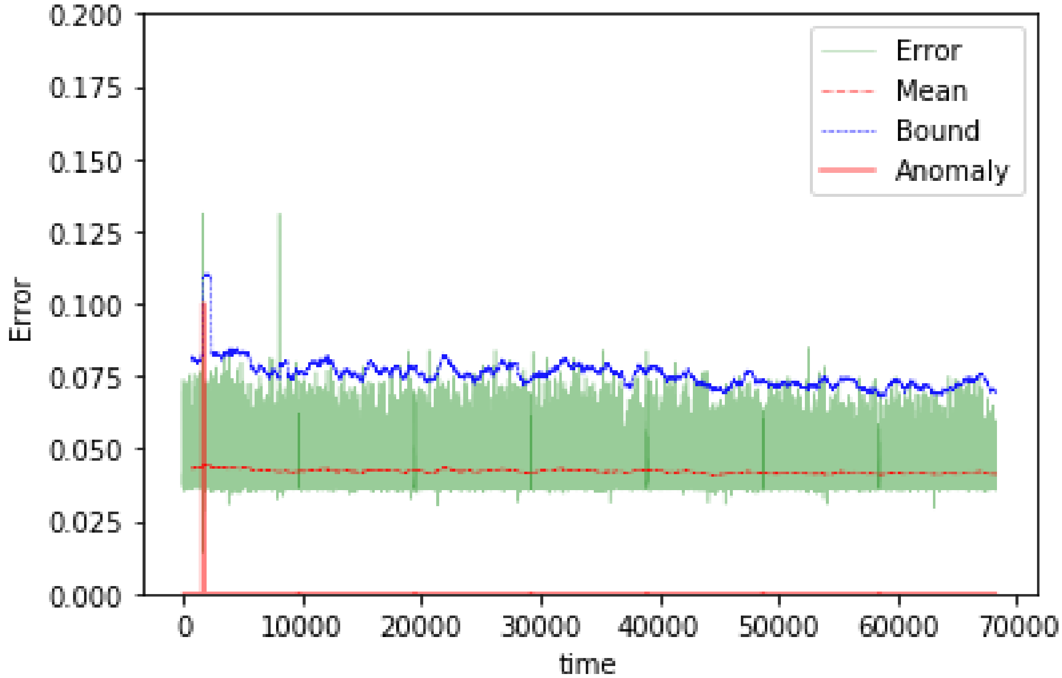

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

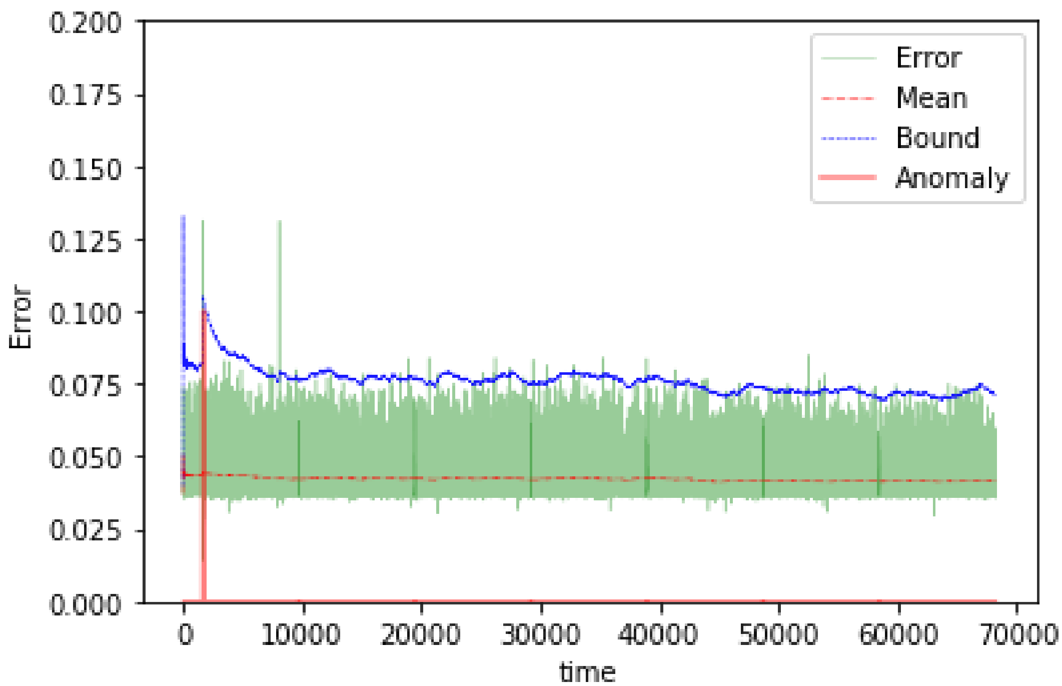



```
The mean error for nn1_100_disk_IO_start_ is 0.01907358164553994 for length 68199
Testing on Avg. load data.
```



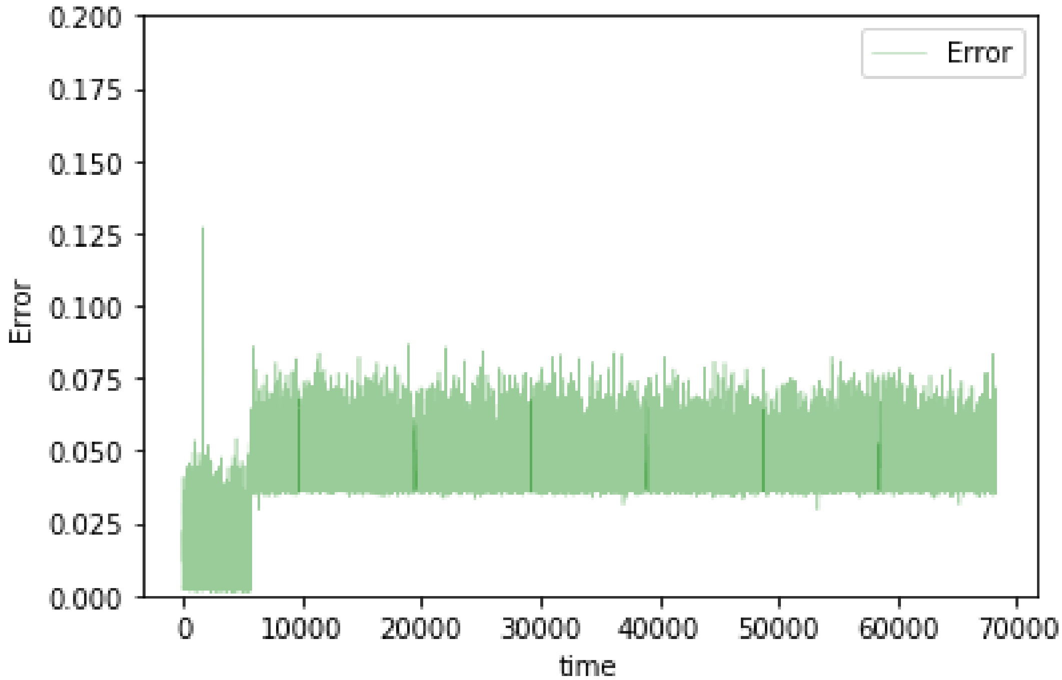


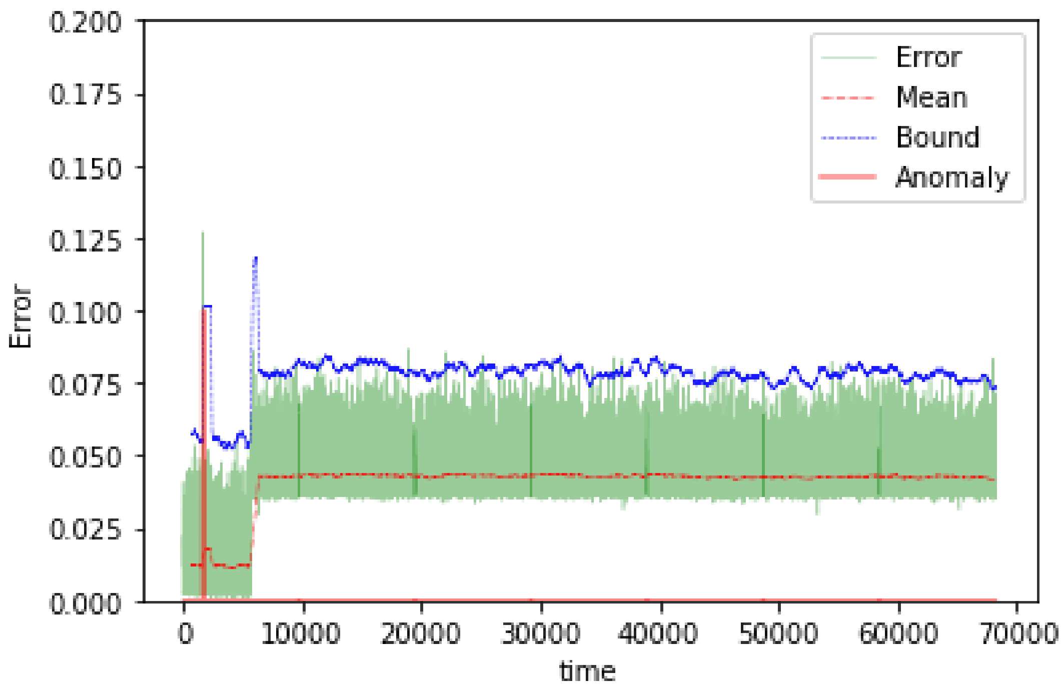
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

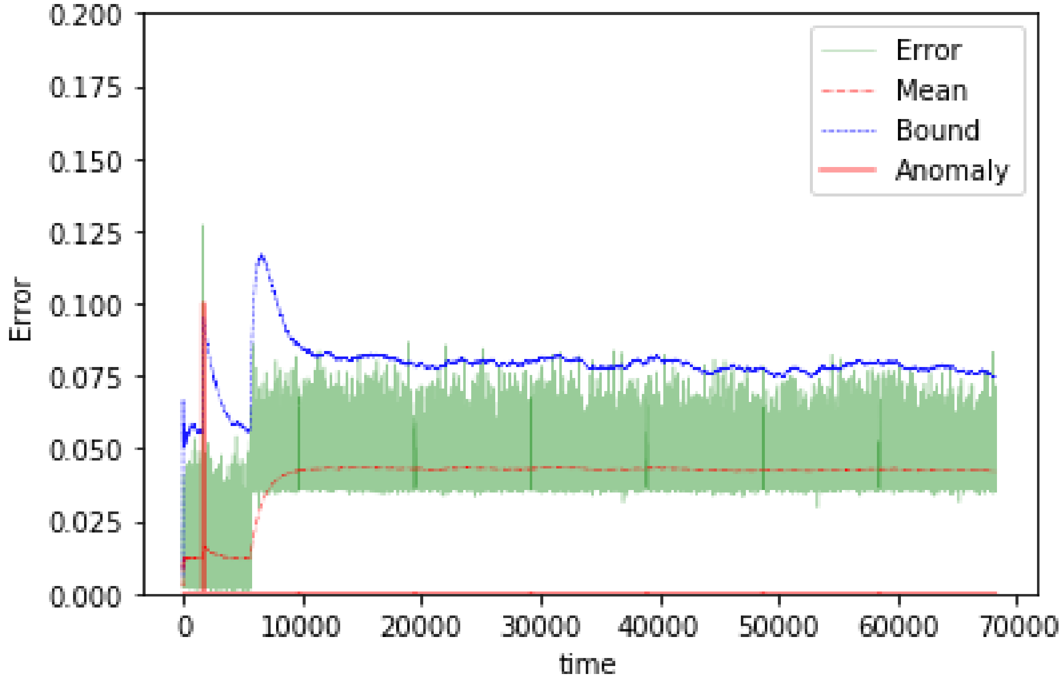


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

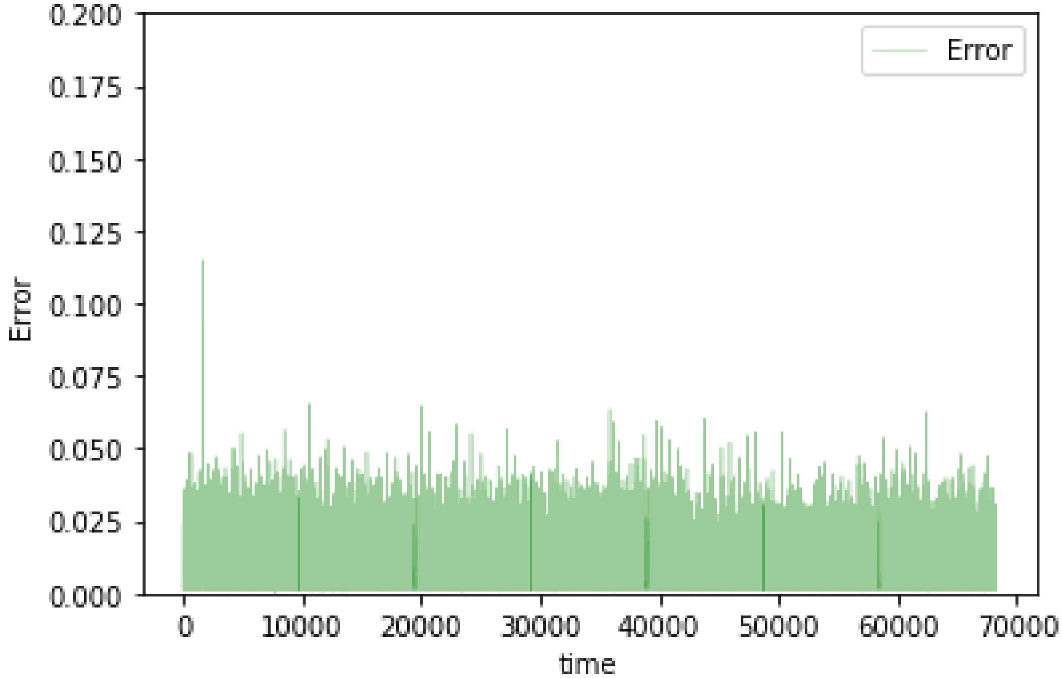
```
The mean error for nn1_100_avg_load_ is 0.042149310222768265 for length 68199
Testing on app change early data.
```



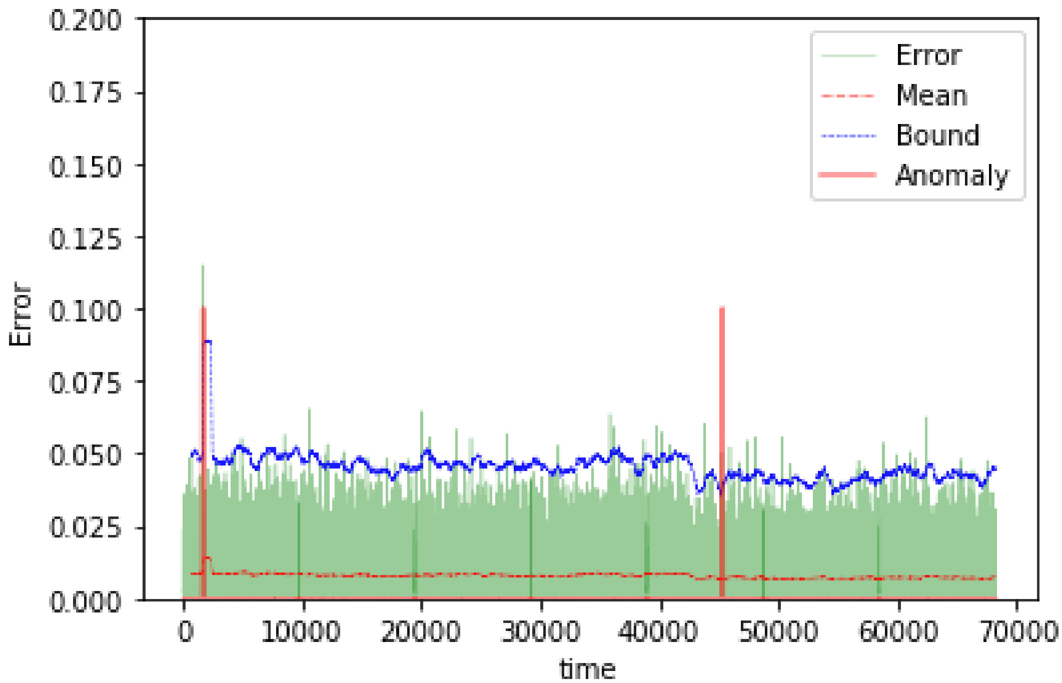




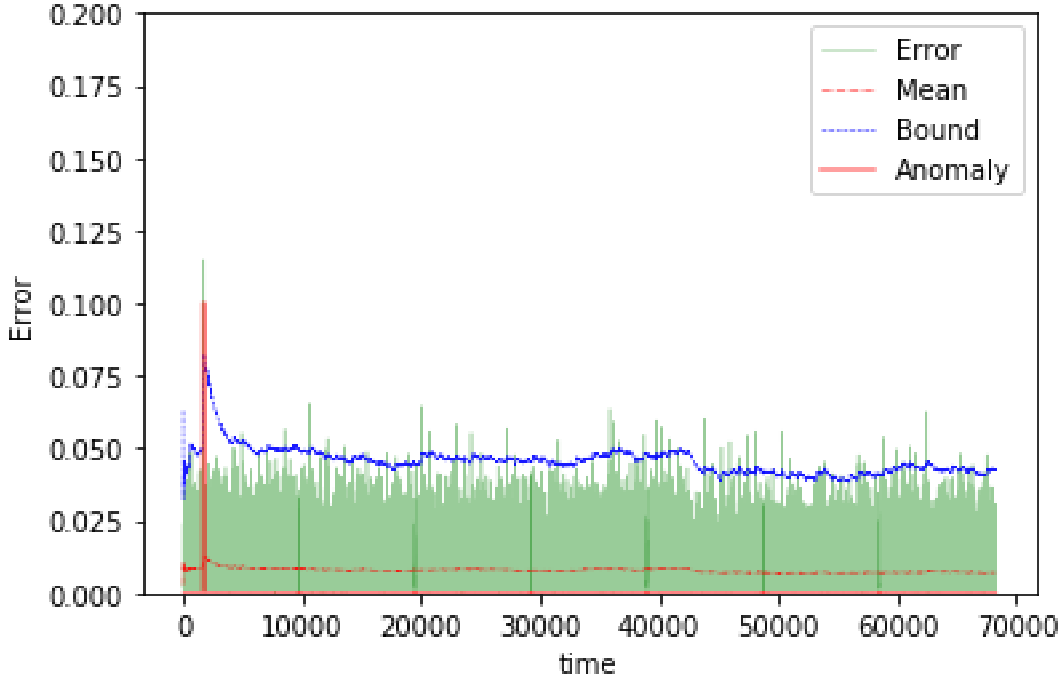

```
The mean error for nn1_100_app_change_early_ is 0.0403459059343831 for length 68199
Testing on Normal data.
```



Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



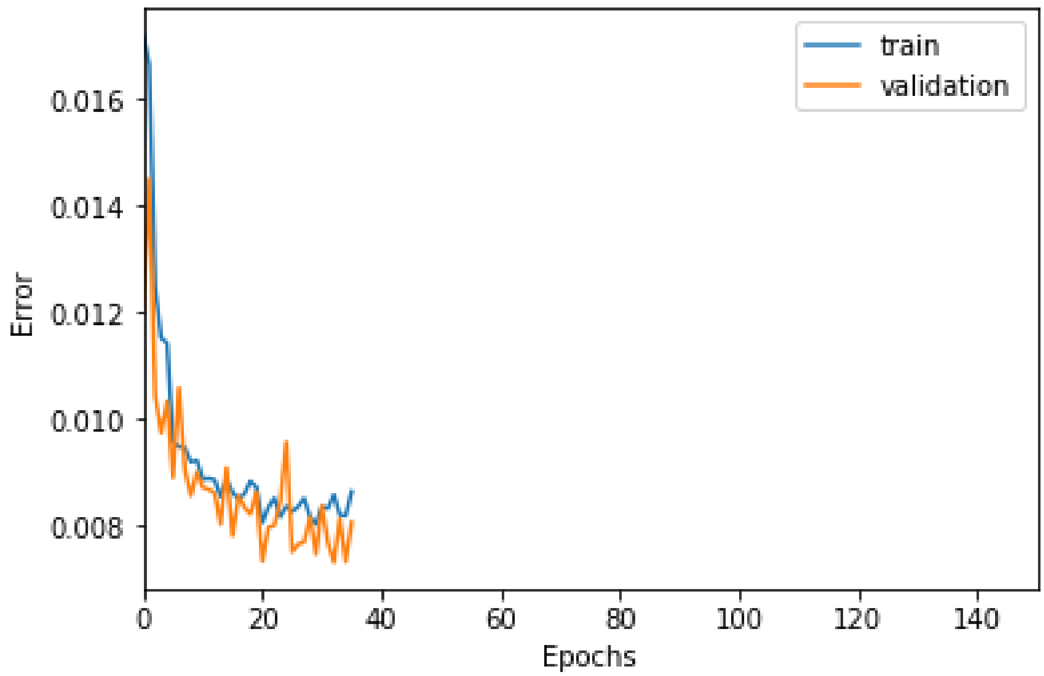
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



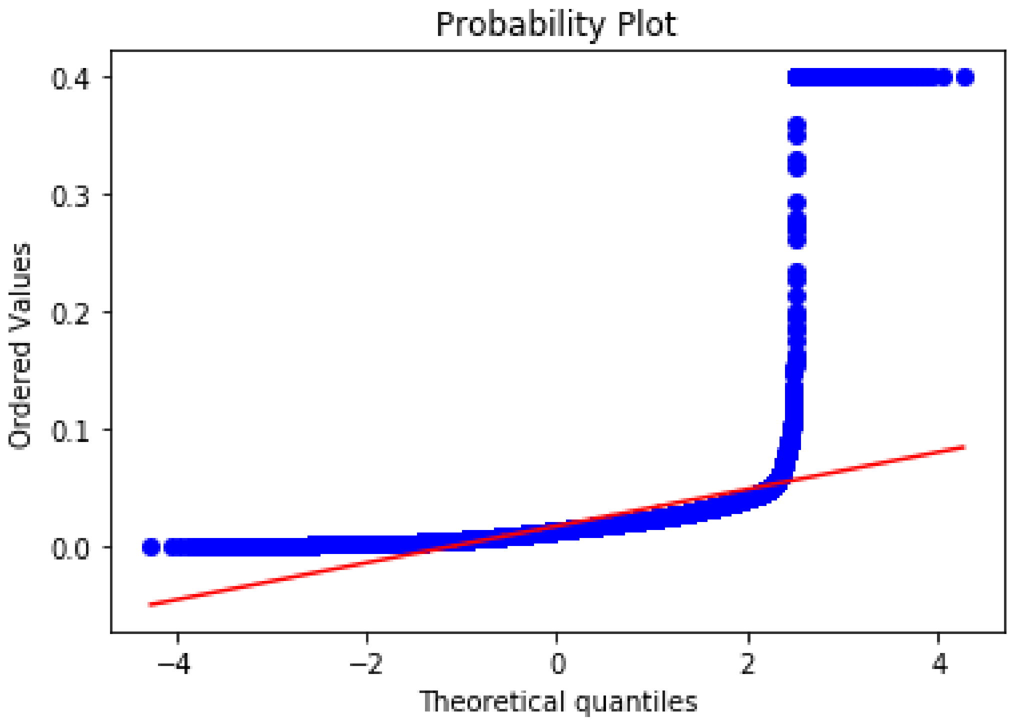

```
The mean error for nn1_100_normal_ is 0.007834870542395685 for length 68199
====================
```

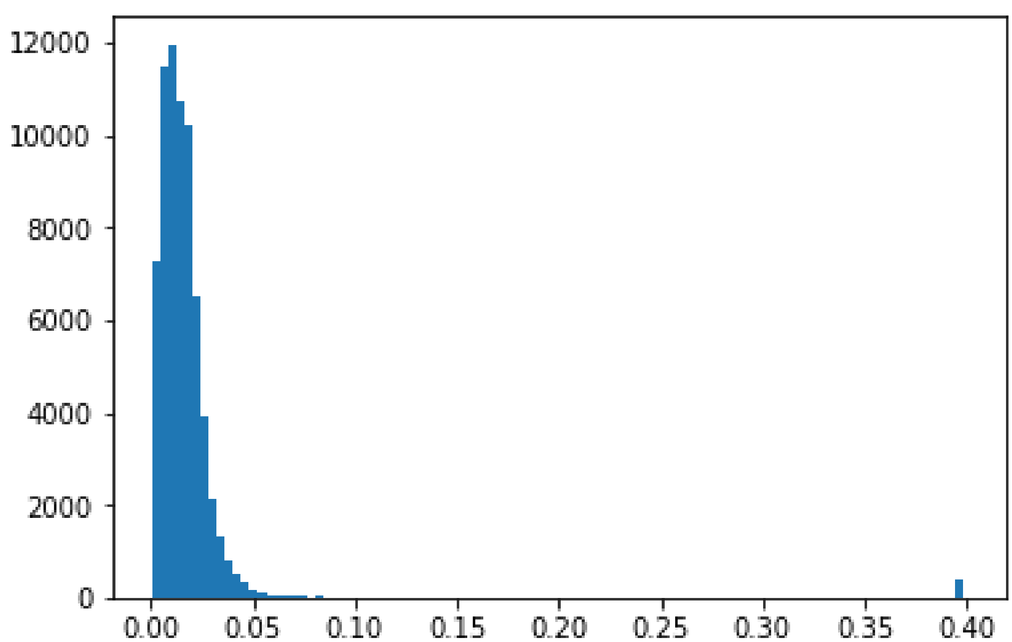
200 steps

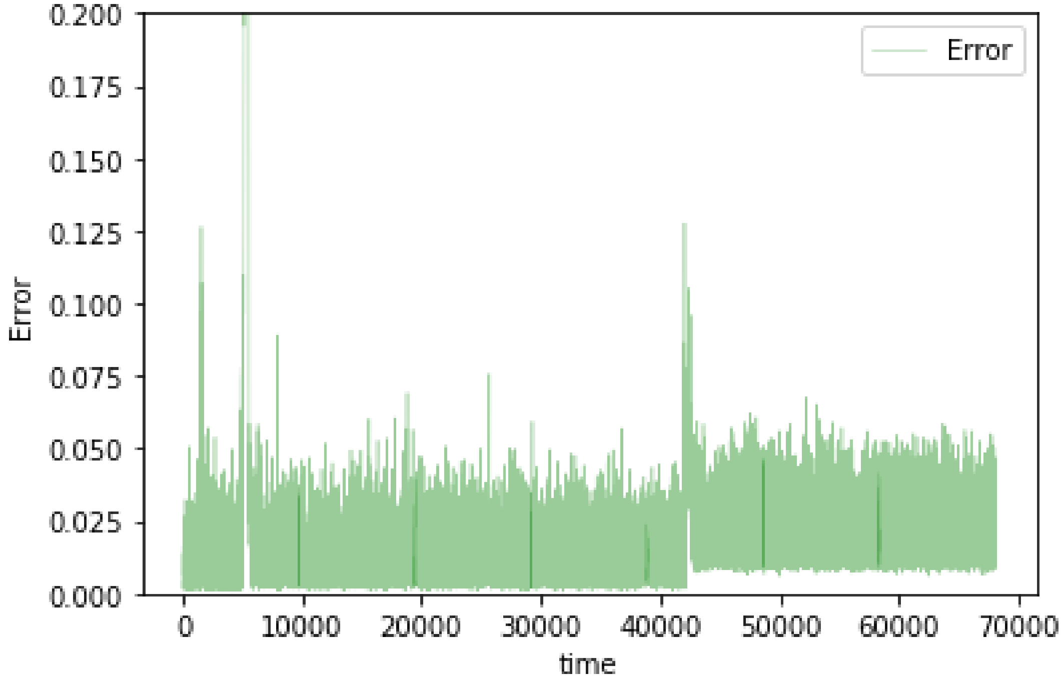
```
In [91]: TIMESTEPS = 200
         DIM = 29
         tgen = flat_generator(X, TIMESTEPS)
         vgen = flat_generator(val_X, TIMESTEPS)
         name = "nn1_200"

In [92]: input_layer = Input(shape=(TIMESTEPS*DIM,))
         hidden = Dense(100,activation='relu')(input_layer)
         output = Dense(DIM, activation='sigmoid')(hidden)

In [93]: model = Model(input_layer, output)
         model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [94]: train(model, tgen, vgen, name=name)
         test(model, name=name, window=TIMESTEPS)
```



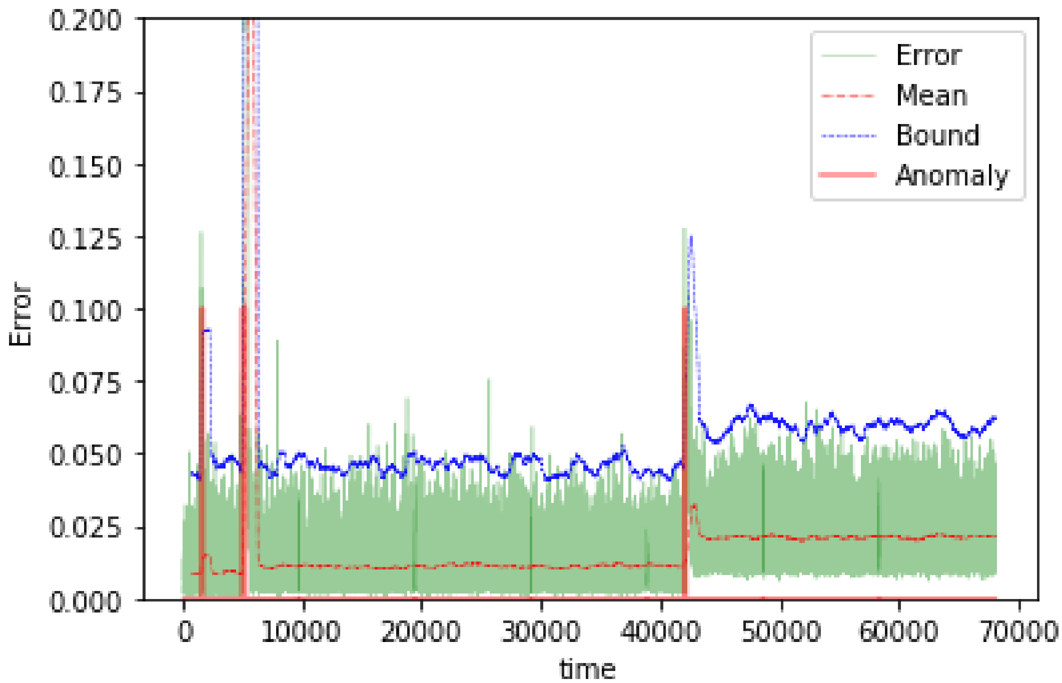

```
Training loss for final epoch is 0.008648808602360078
Validation loss for final epoch is 0.008069756591226905
---------- Beginning tests for nn1_200 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.4
0.3
0.2
0.1
0.0
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

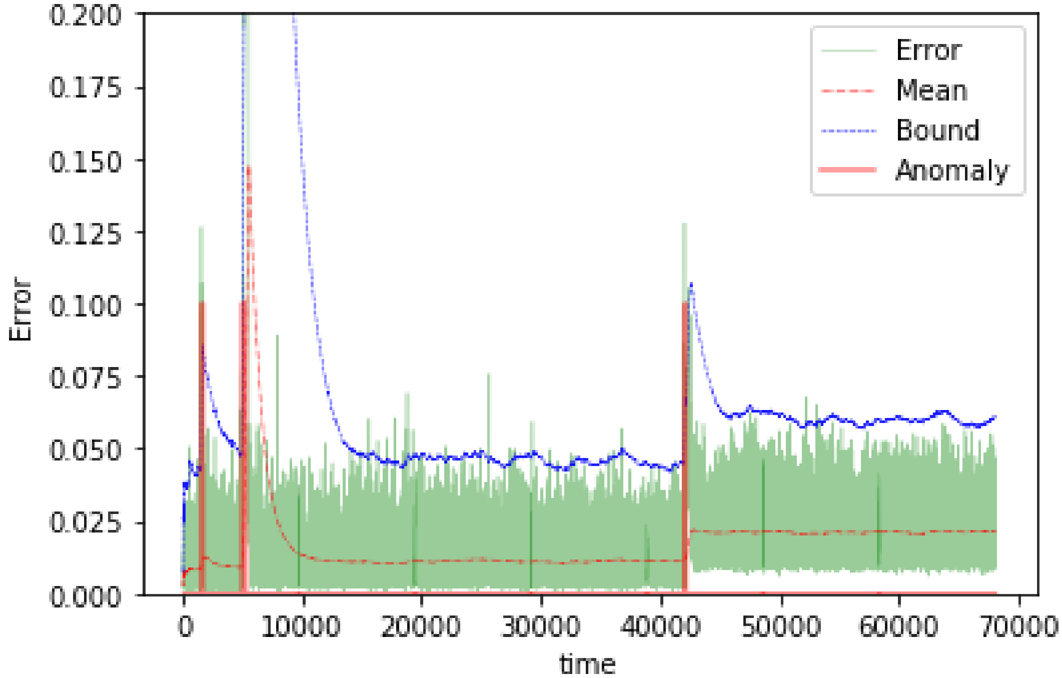

12000
10000
8000
6000
4000
2000
0
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40



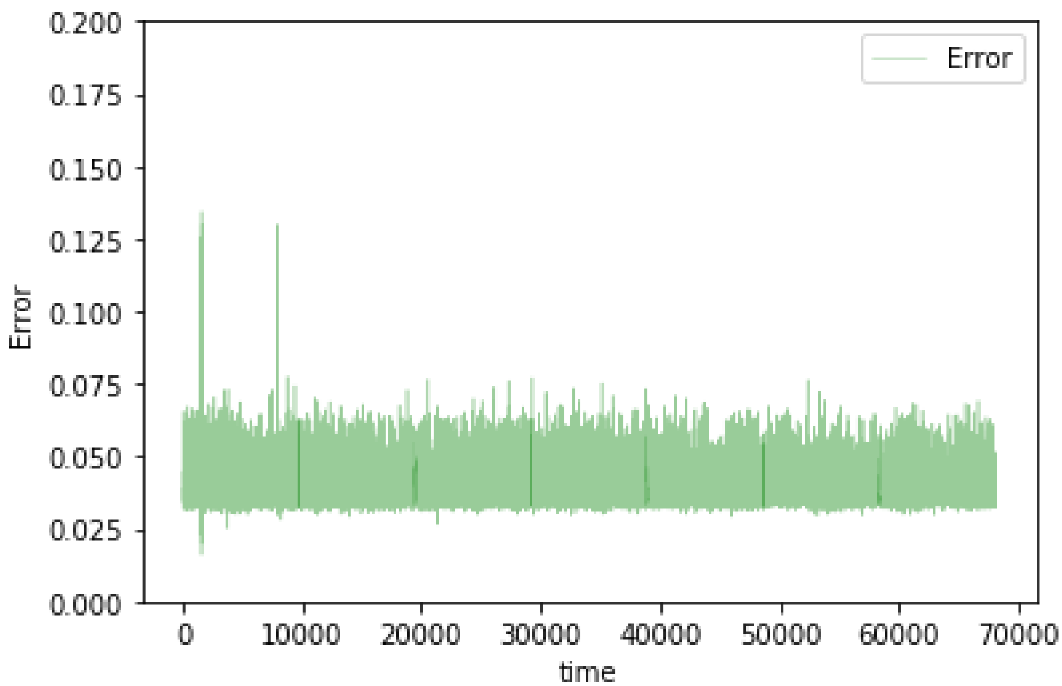
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

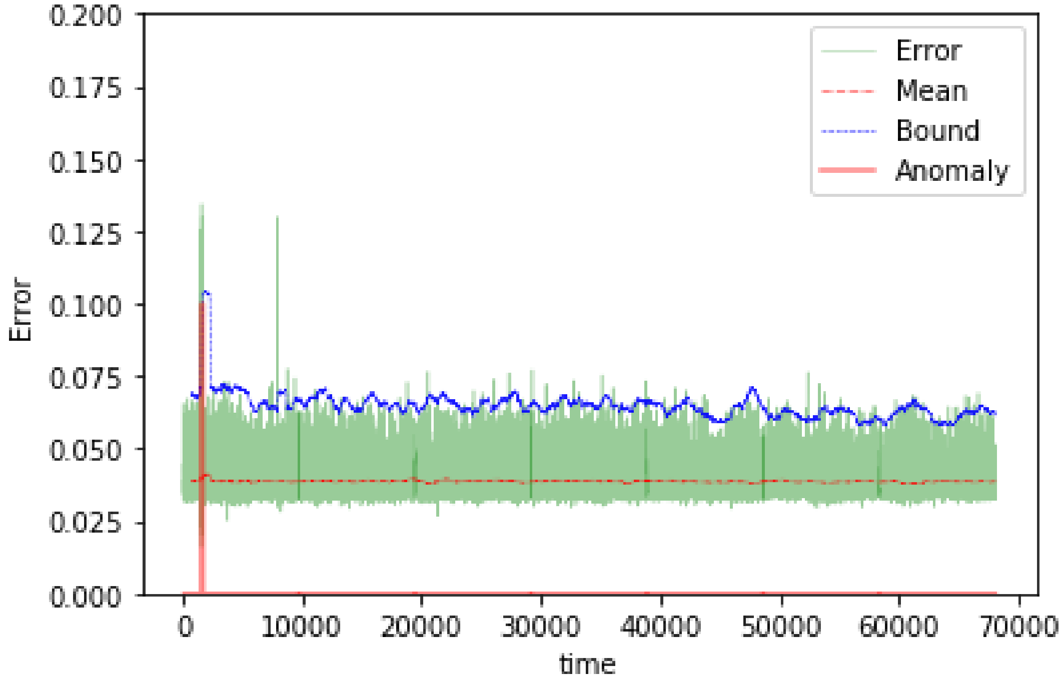


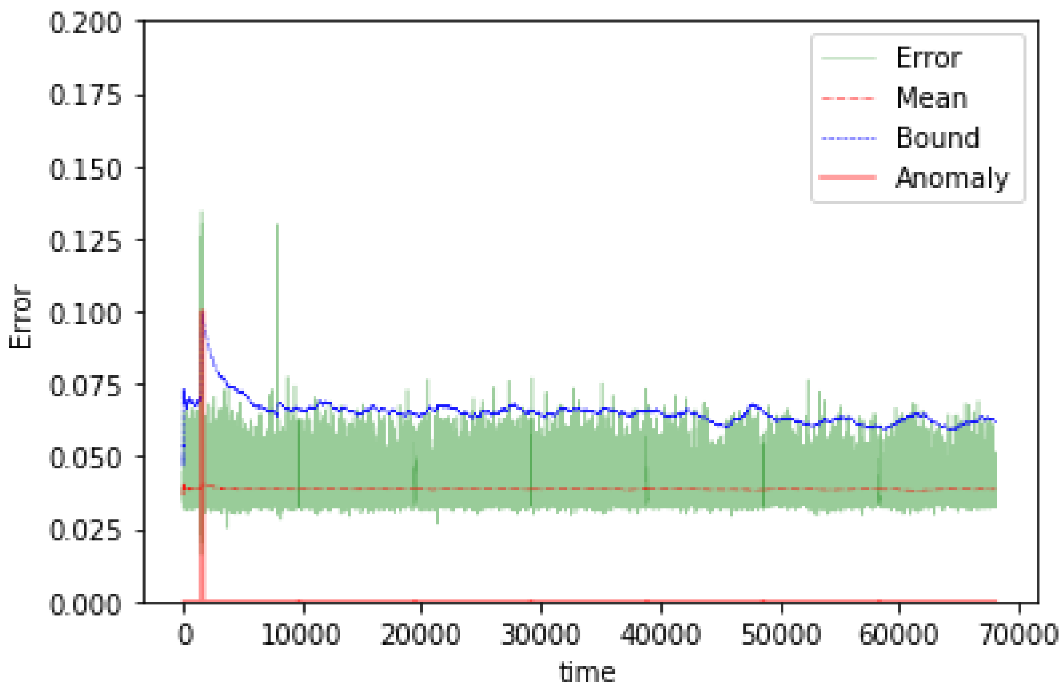
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



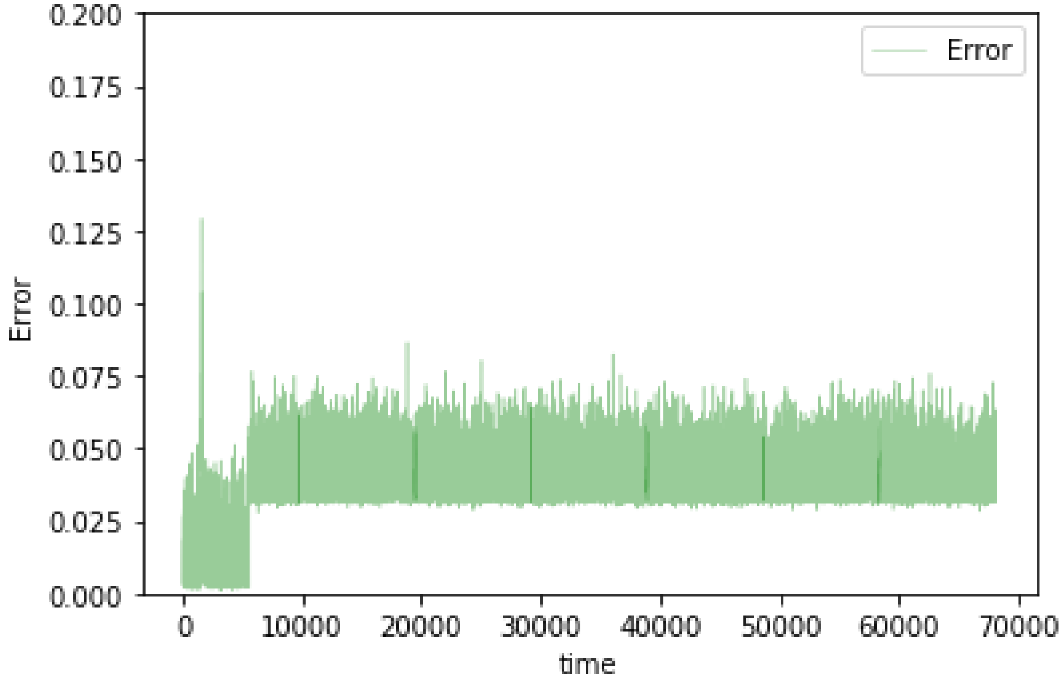

```
The mean error for nn1_200_disk_IO_start_ is 0.017597571291146877 for length 68099
Testing on Avg. load data.
```

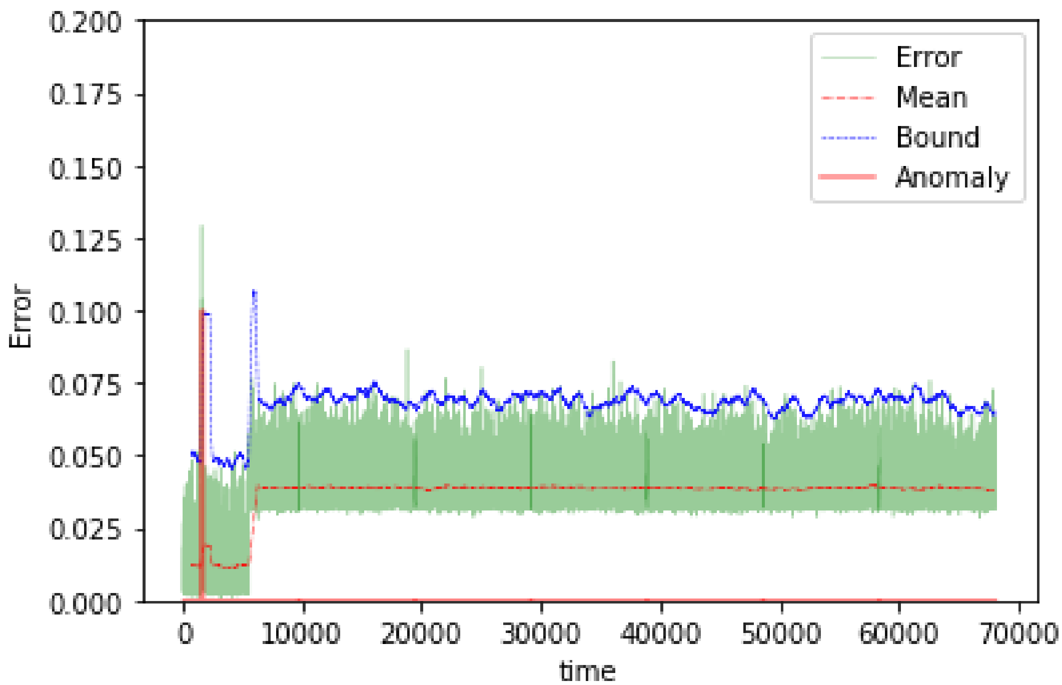


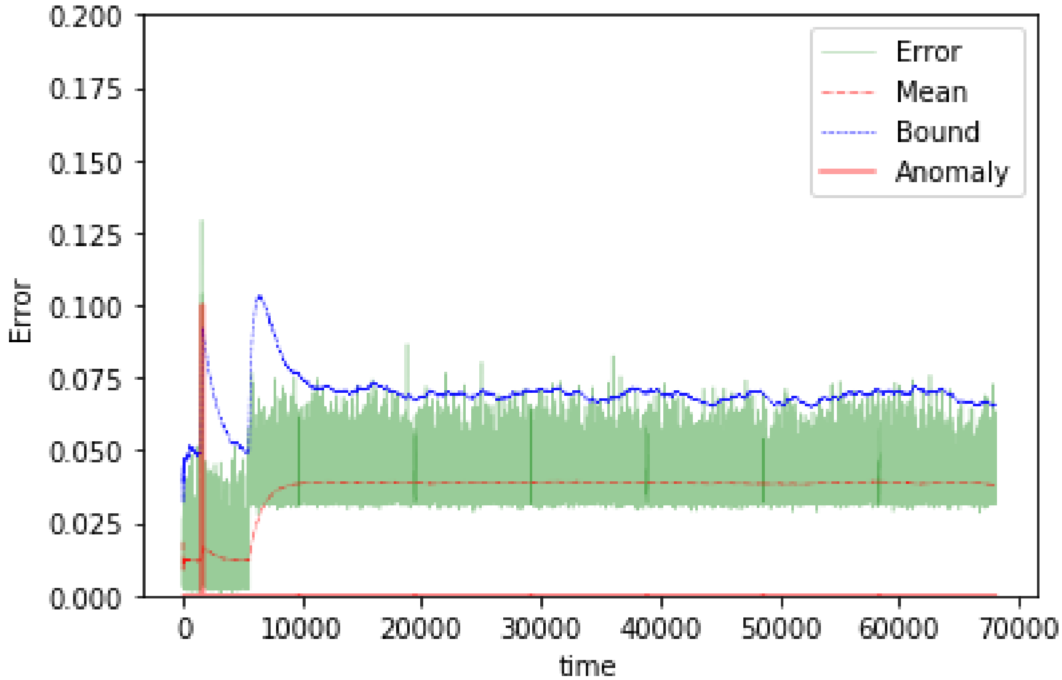

Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000


Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

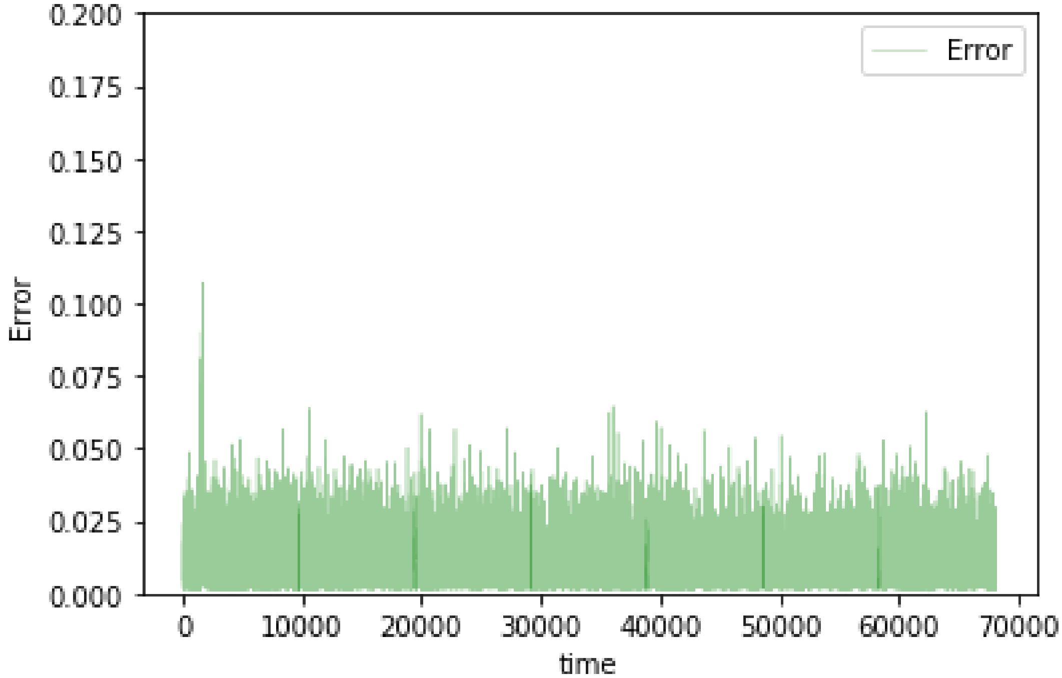
```
The mean error for nn1_200_avg_load_ is 0.03874186886827659 for length 68099
Testing on app change early data.
```

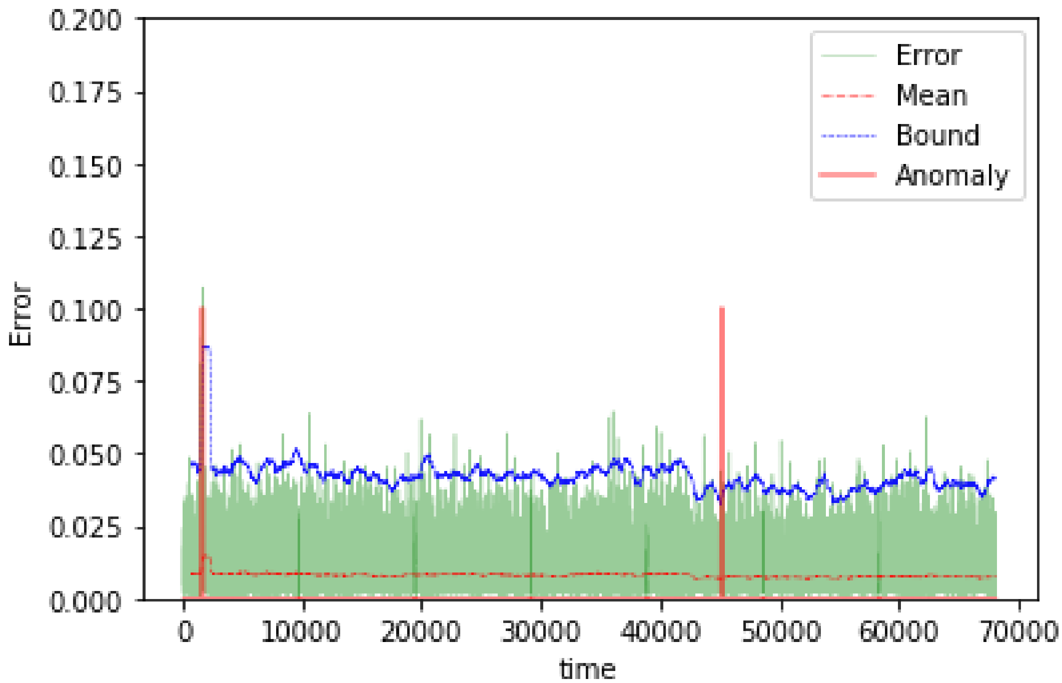


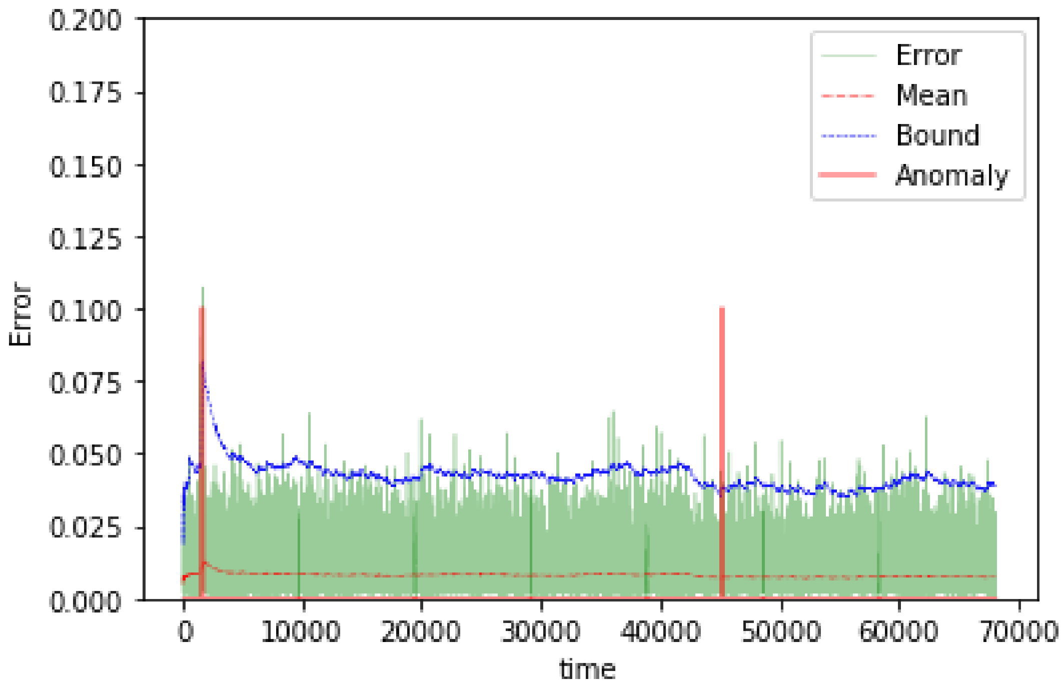





```
The mean error for nn1_200_app_change_early_ is 0.03672398926502097 for length 68099
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



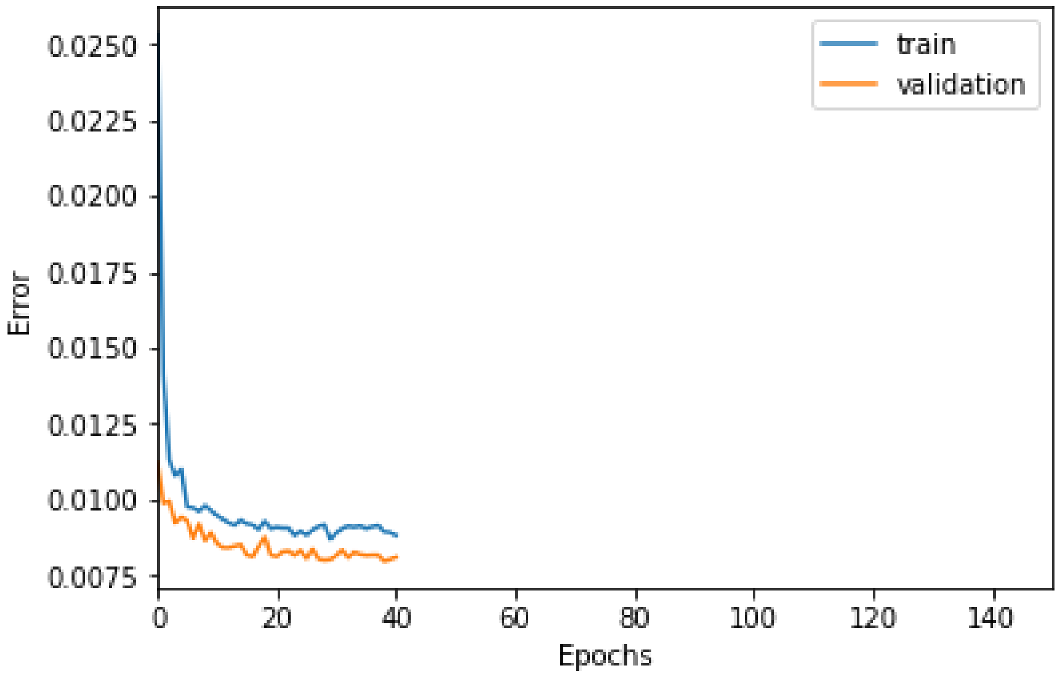

```
The mean error for nn1_200_normal_ is 0.008026084399497023 for length 68099
====================
```

1.11.3 NN with 2 hidden layers

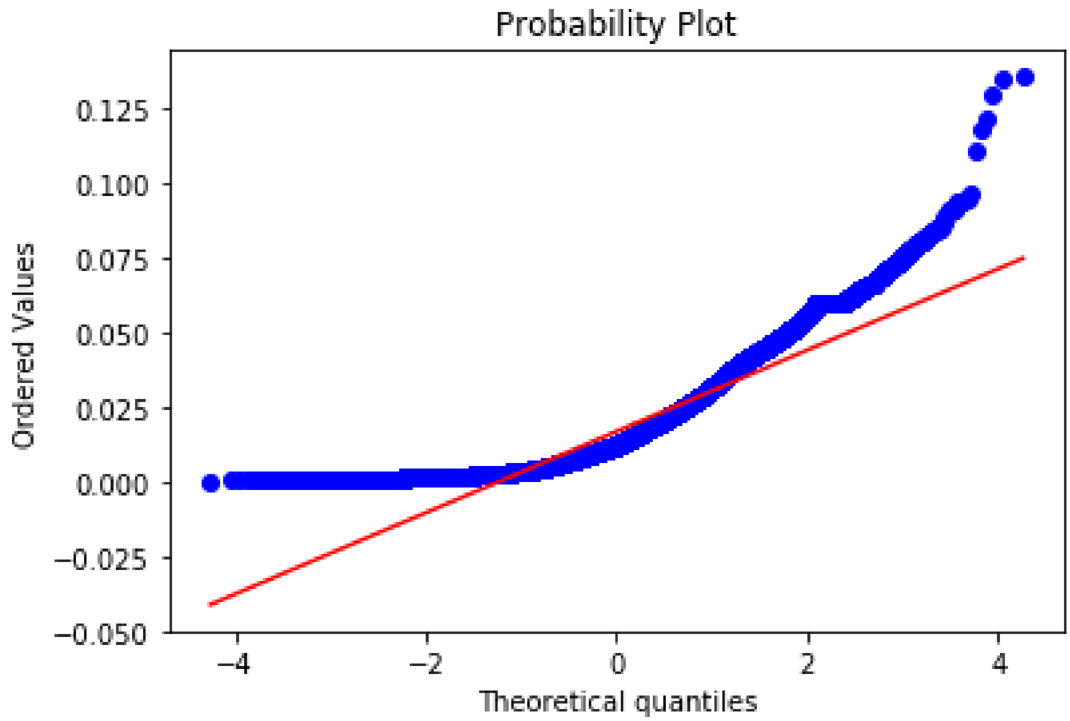
2 steps

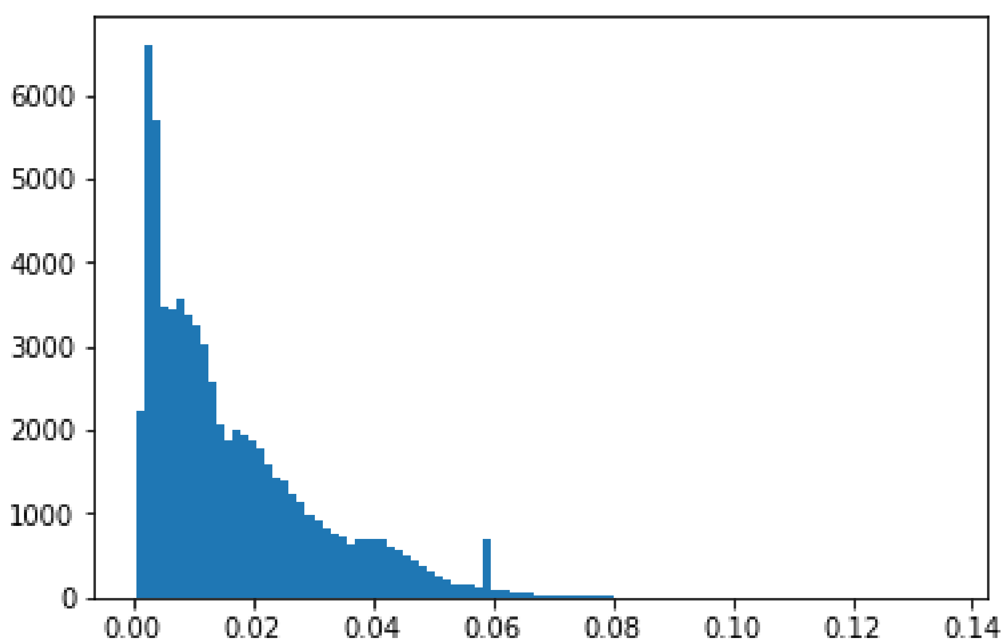
```
In [95]: TIMESTEPS = 2
         DIM = 29
         tgen = flat_generator(X, TIMESTEPS)
         vgen = flat_generator(val_X, TIMESTEPS)
         name = "nn2_2"

In [96]: input_layer = Input(shape=(TIMESTEPS*DIM,))
         hidden = Dense(500, activation='relu')(input_layer)
         hidden = Dense(100, activation='relu')(hidden)
         output = Dense(DIM, activation='sigmoid')(hidden)

In [97]: model = Model(input_layer, output)
         model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])
```

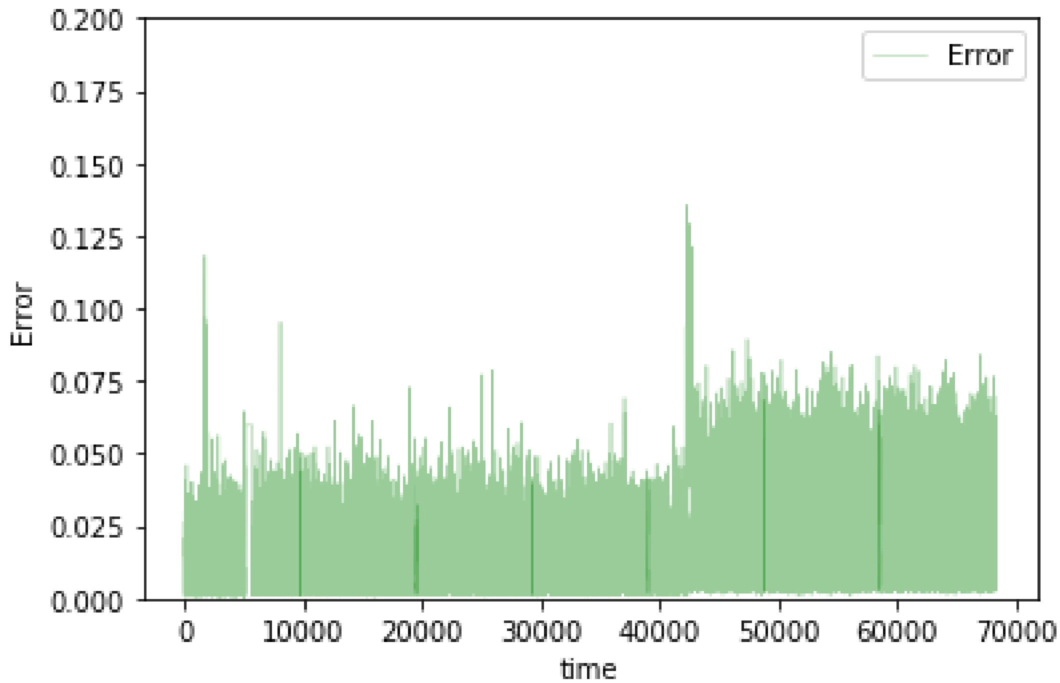
```
In [98]: train(model, tgen, vgen, name=name)
         test(model, name=name, window=TIMESTEPS)
```

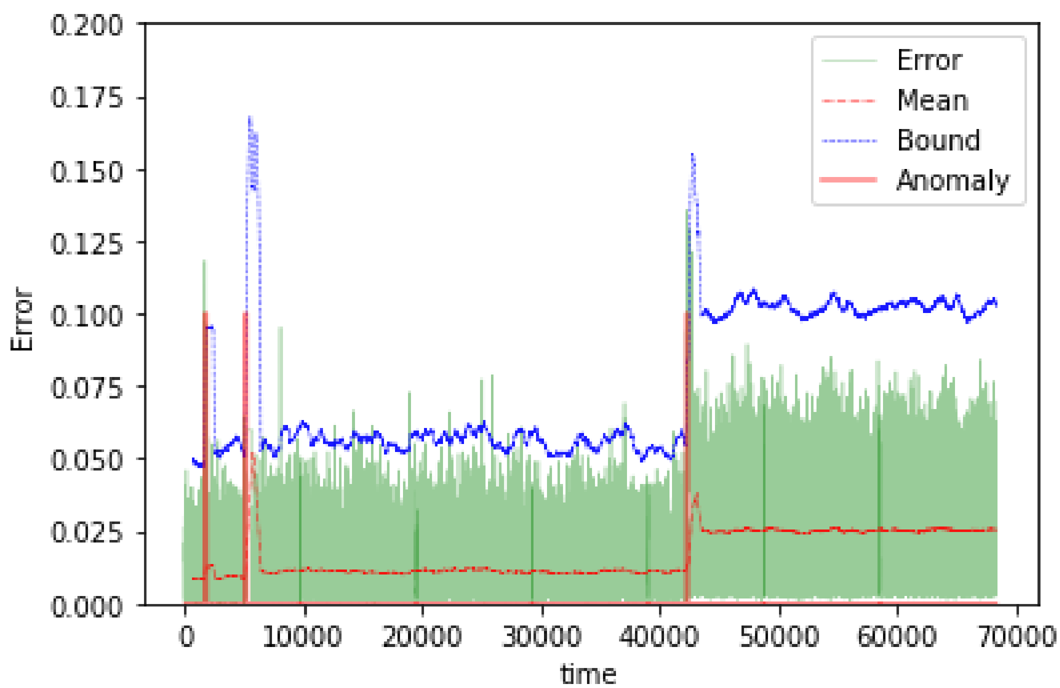


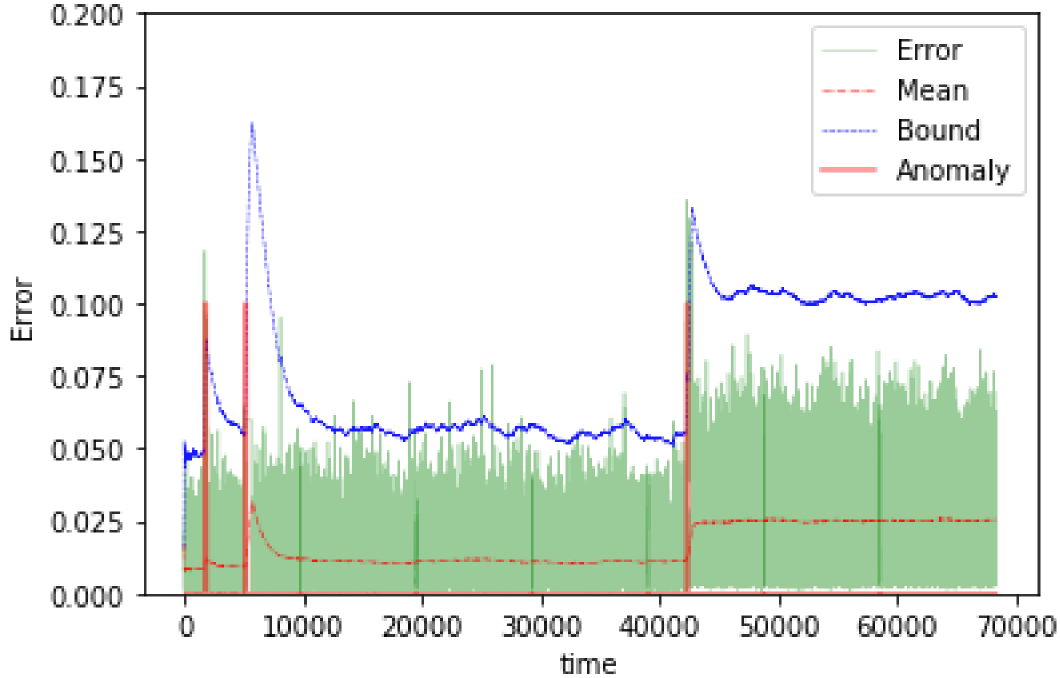

```
Training loss for final epoch is 0.008825131132500247
Validation loss for final epoch is 0.008101381783606485
---------- Beginning tests for nn2_2 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

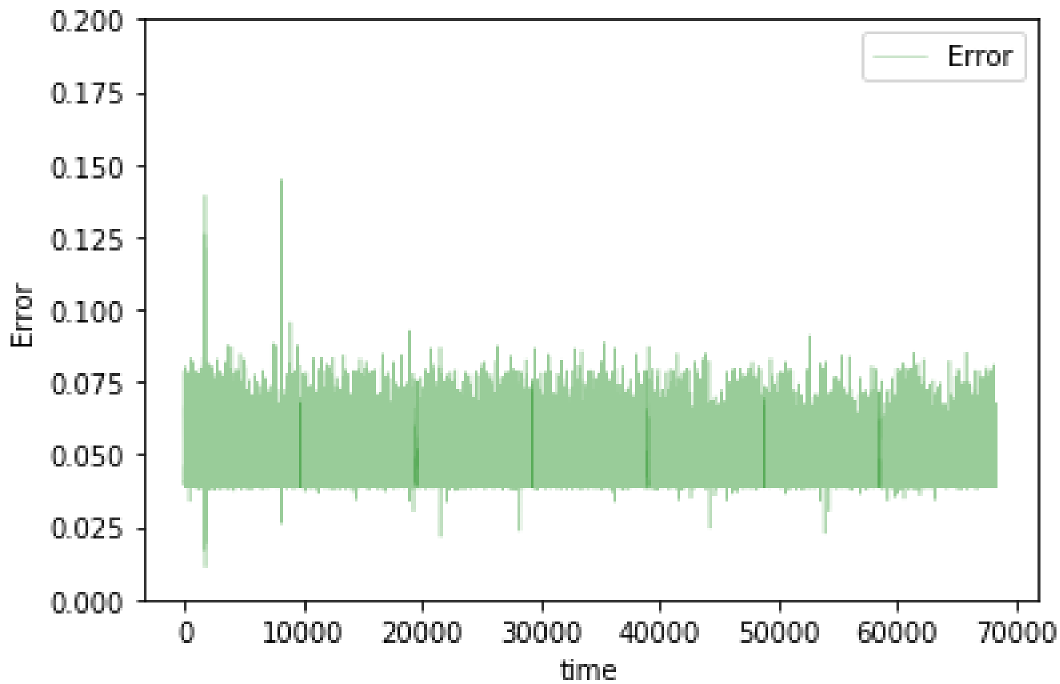

6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

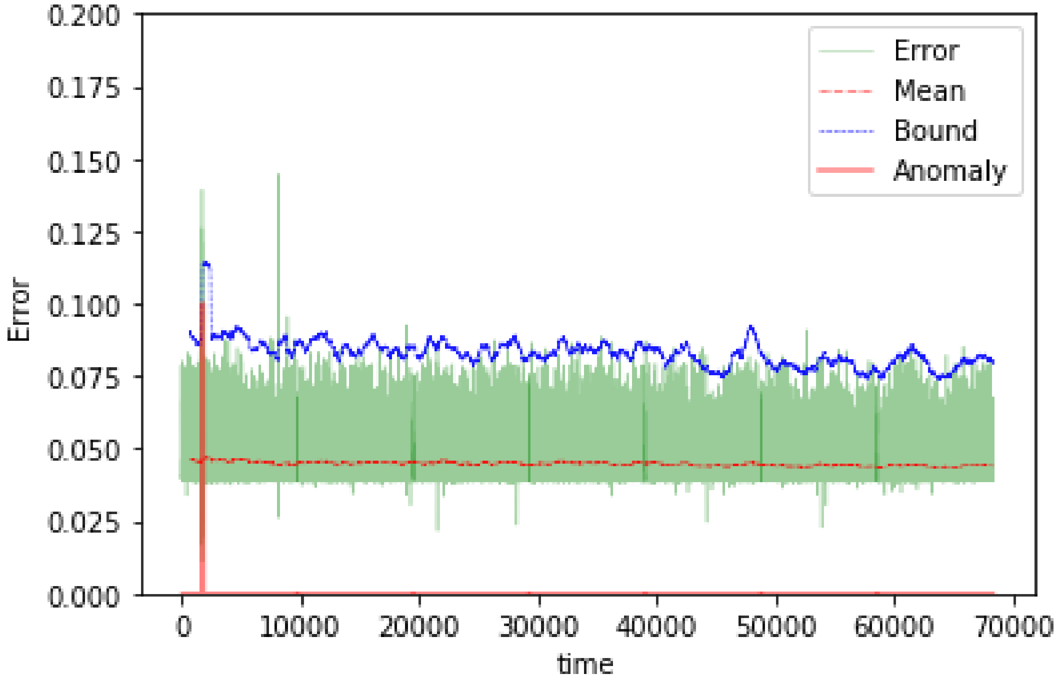

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

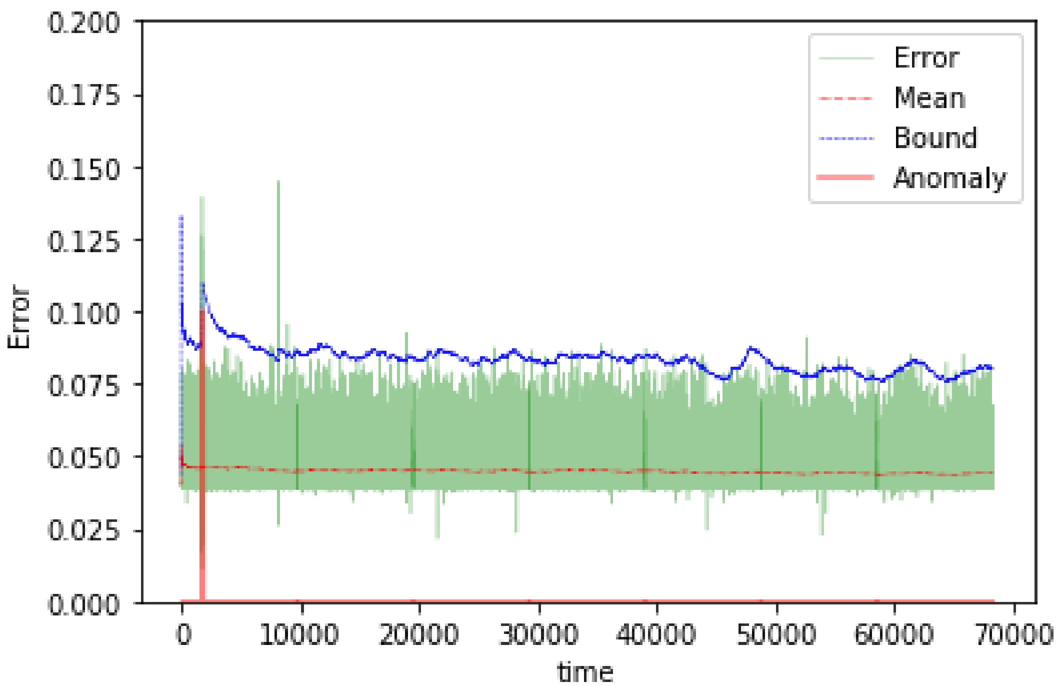

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



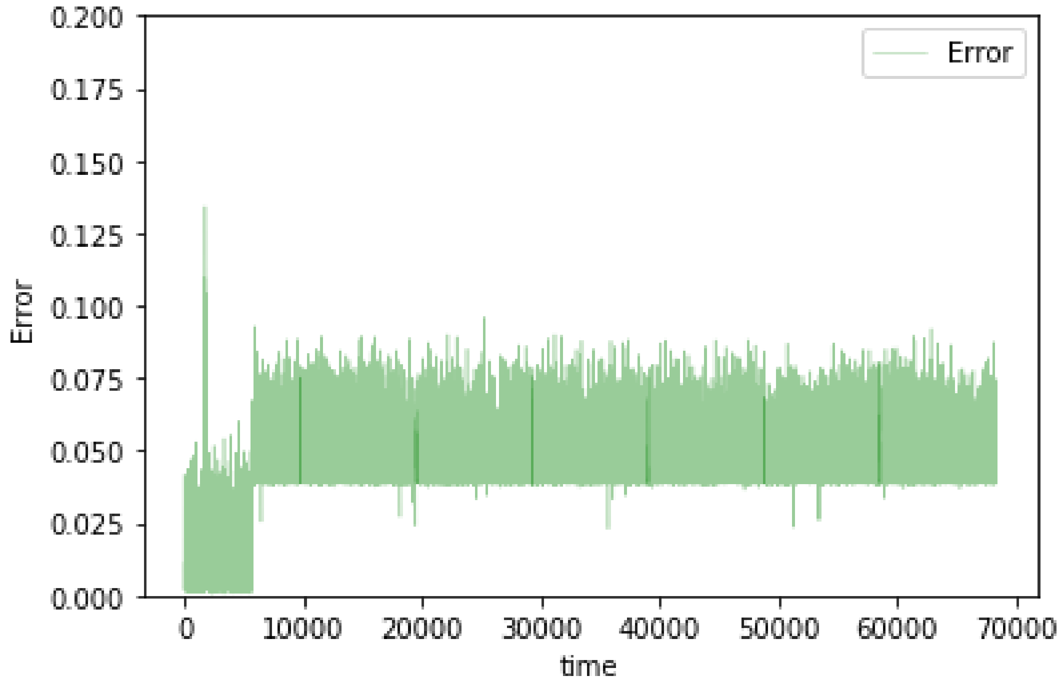

```
The mean error for nn2_2_disk_IO_start_ is 0.016882113911824564 for length 68297
Testing on Avg. load data.
```

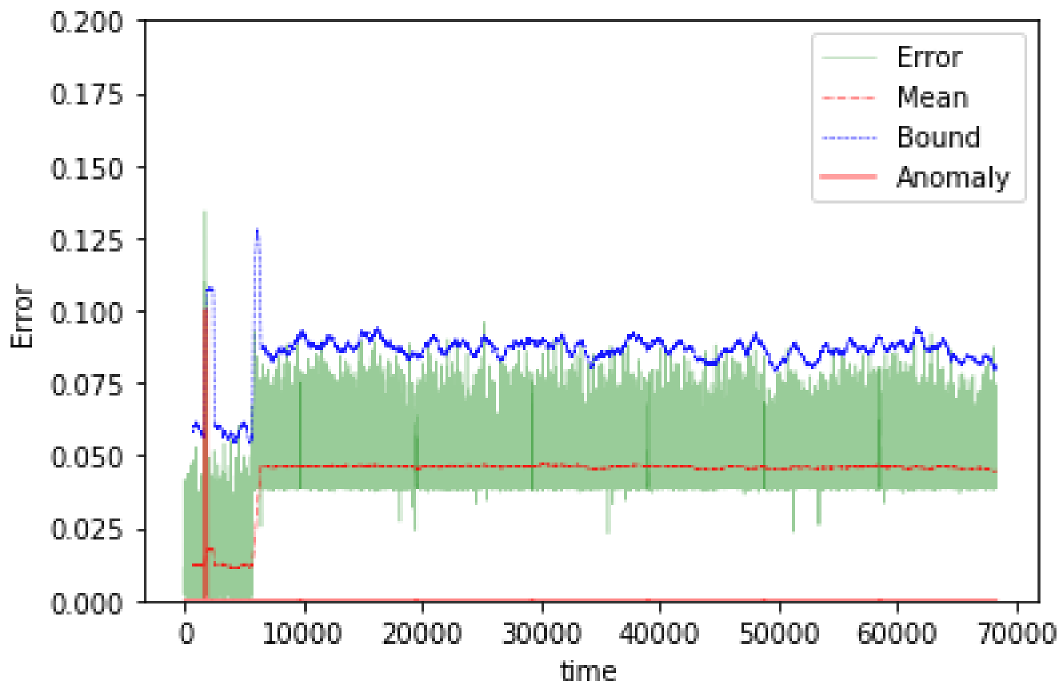


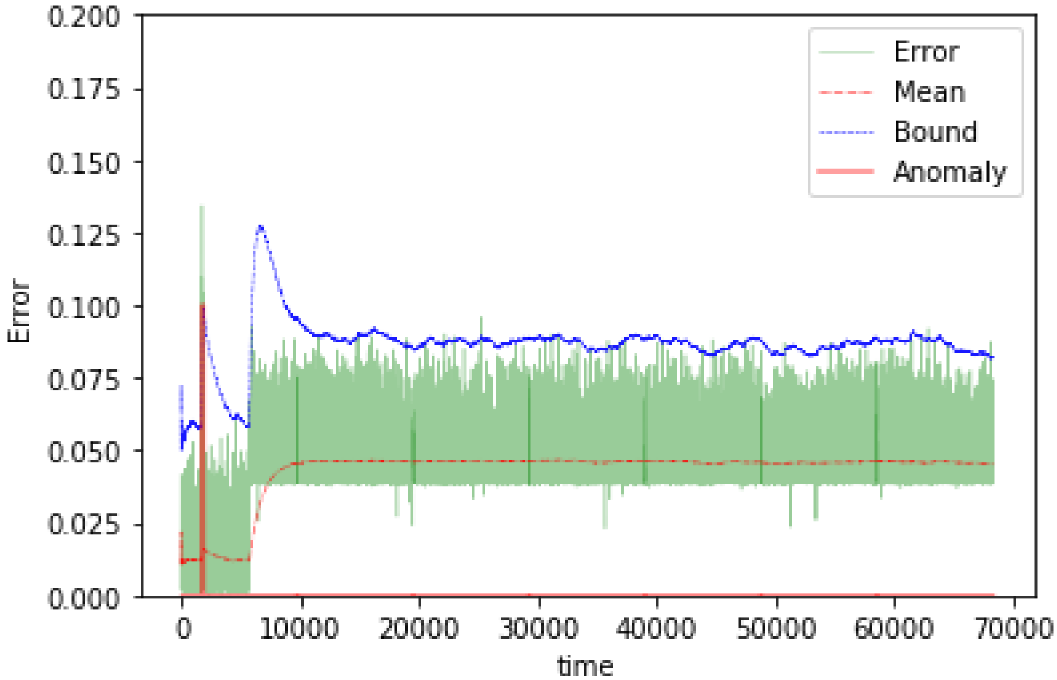

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

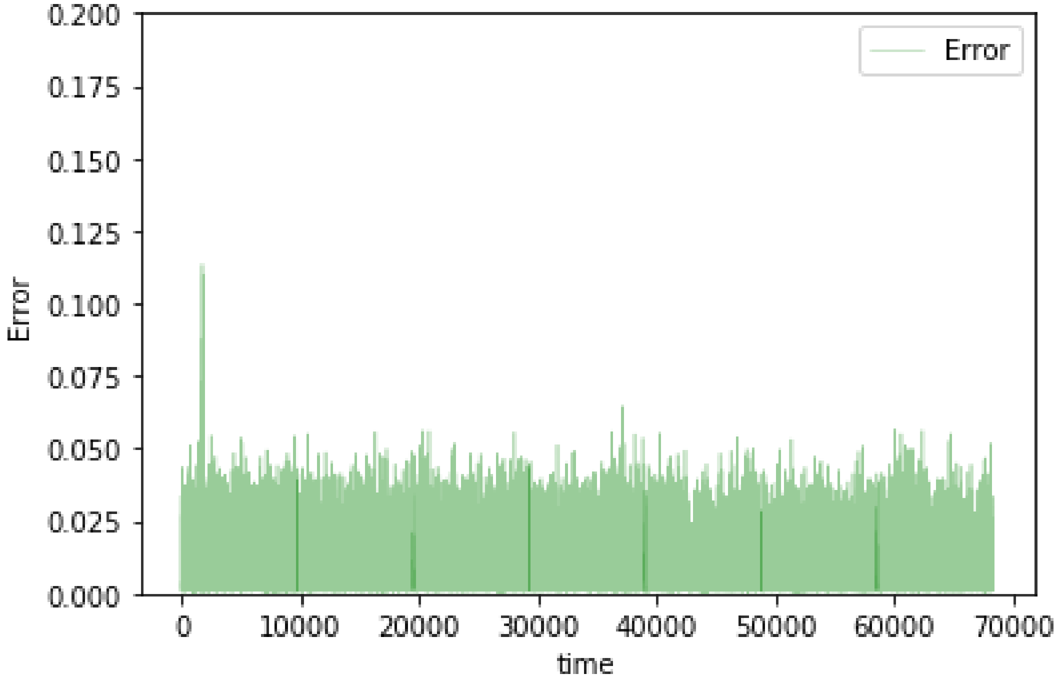
```
The mean error for nn2_2_avg_load_ is 0.044798596910774 for length 68297
Testing on app change early data.
```

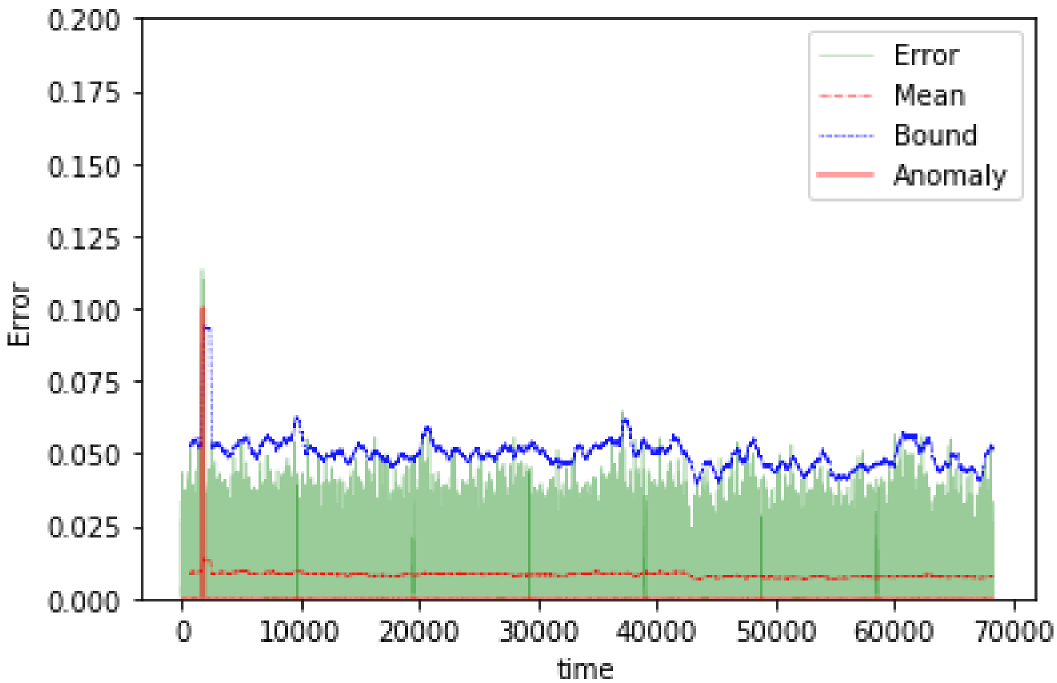


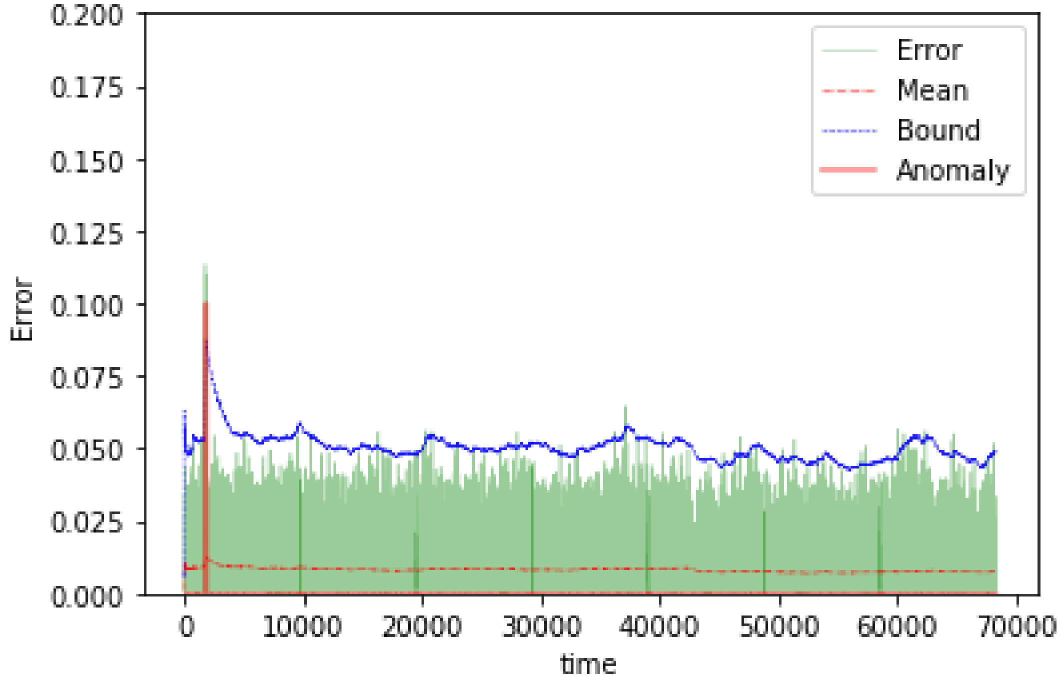





```
The mean error for nn2_2_app_change_early_ is 0.04326271375892231 for length 68297
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

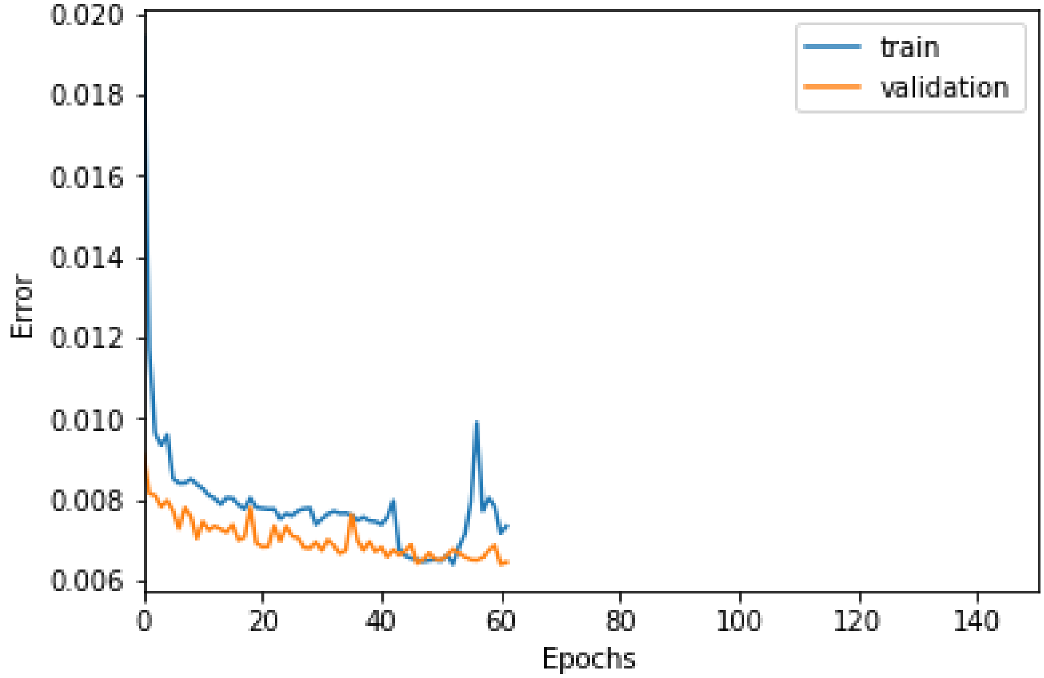

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



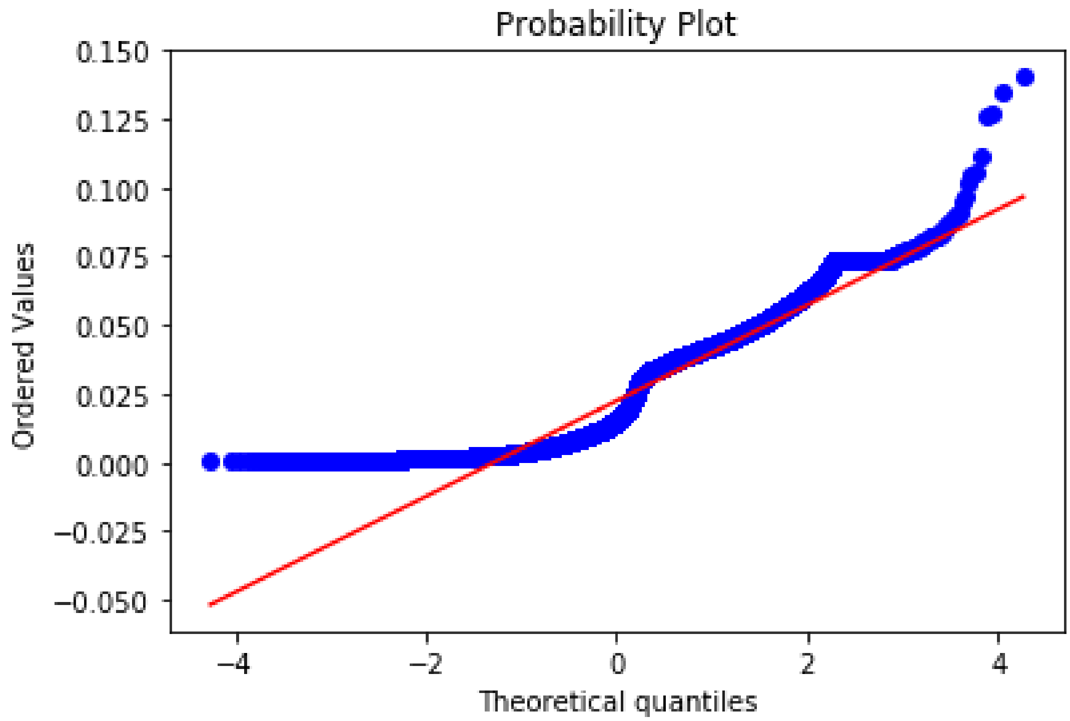

```
The mean error for nn2_2_normal_ is 0.008148489950116408 for length 68297
====================
```

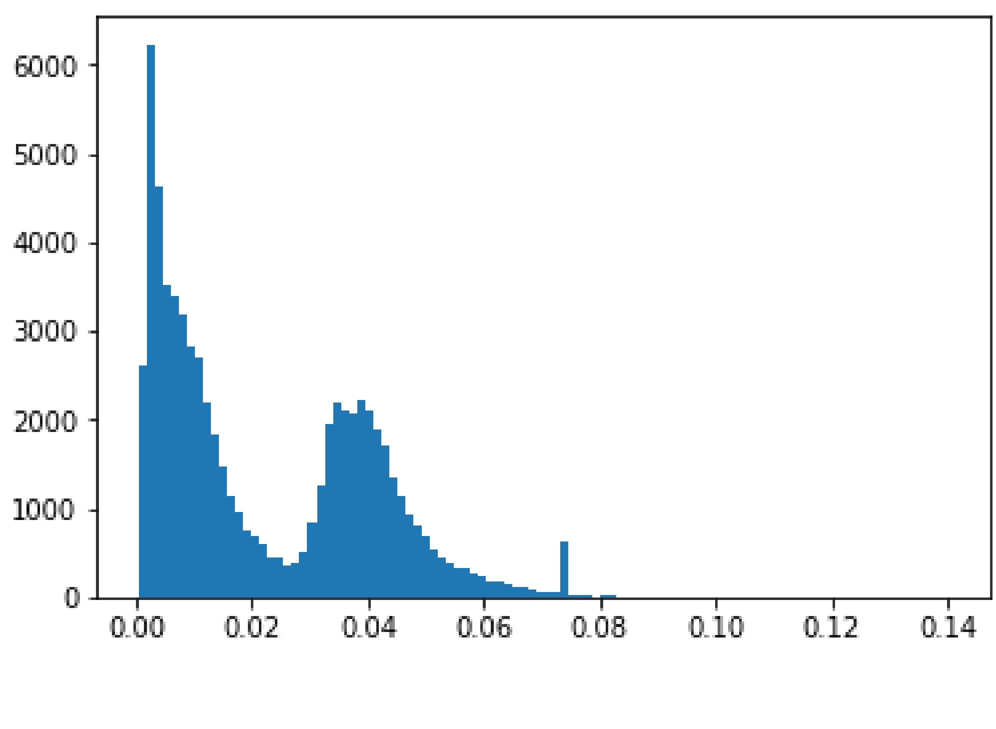
5 steps

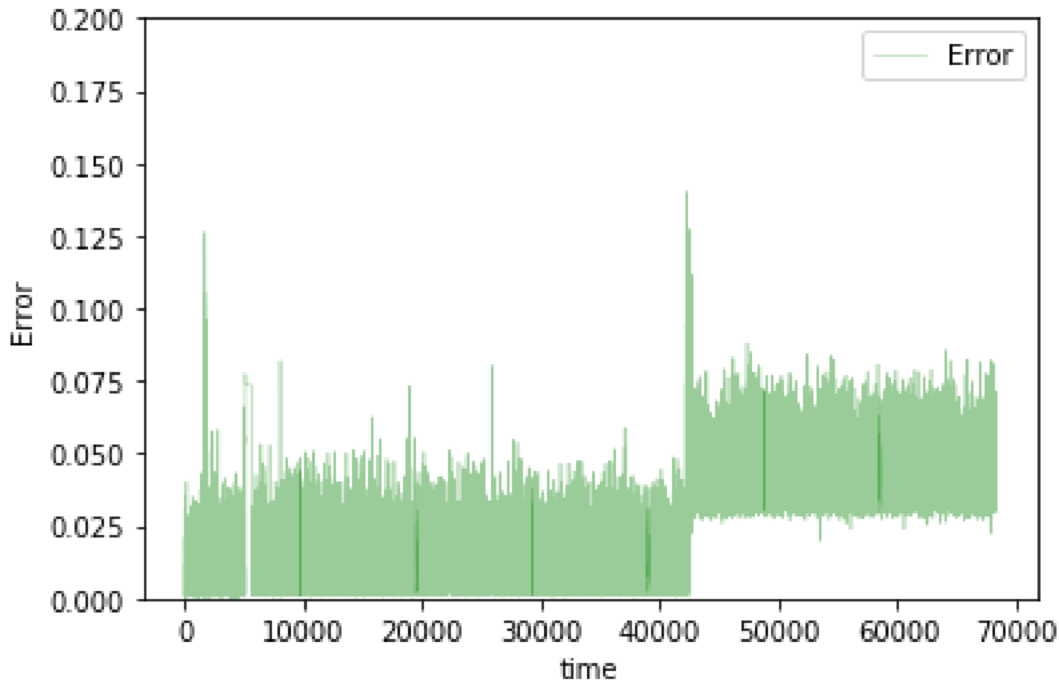
```
In [99]: TIMESTEPS = 5
         DIM = 29
         tgen = flat_generator(X, TIMESTEPS)
         vgen = flat_generator(val_X, TIMESTEPS)
         name = "nn2_5"

In [100]: input_layer = Input(shape=(TIMESTEPS*DIM,))
          hidden = Dense(500, activation='relu')(input_layer)
          hidden = Dense(100, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [101]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [102]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```

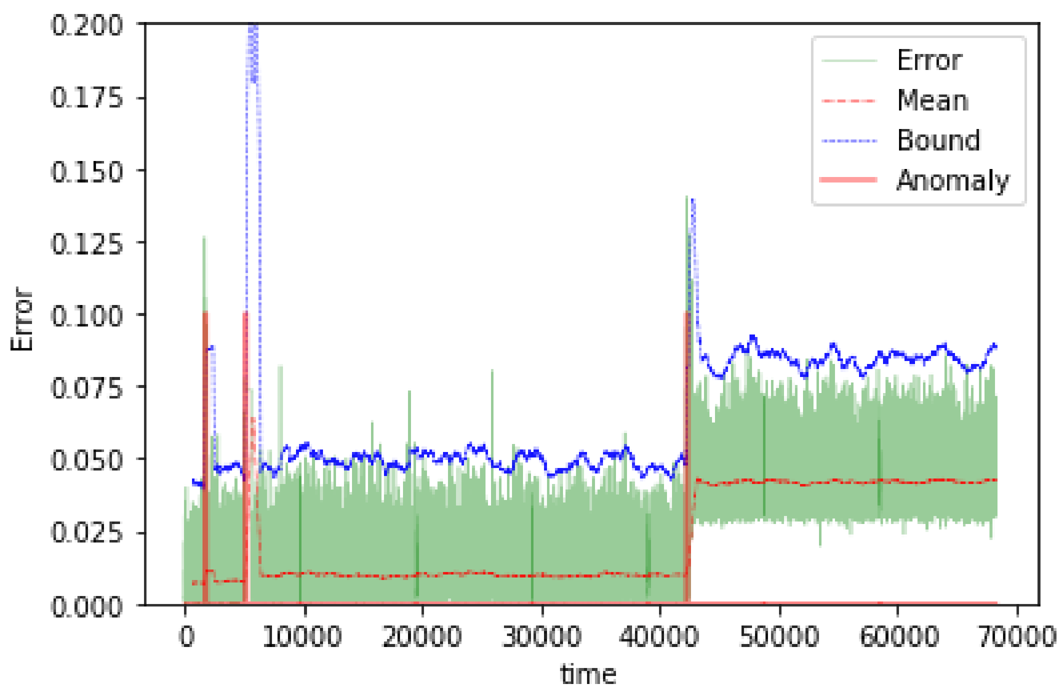


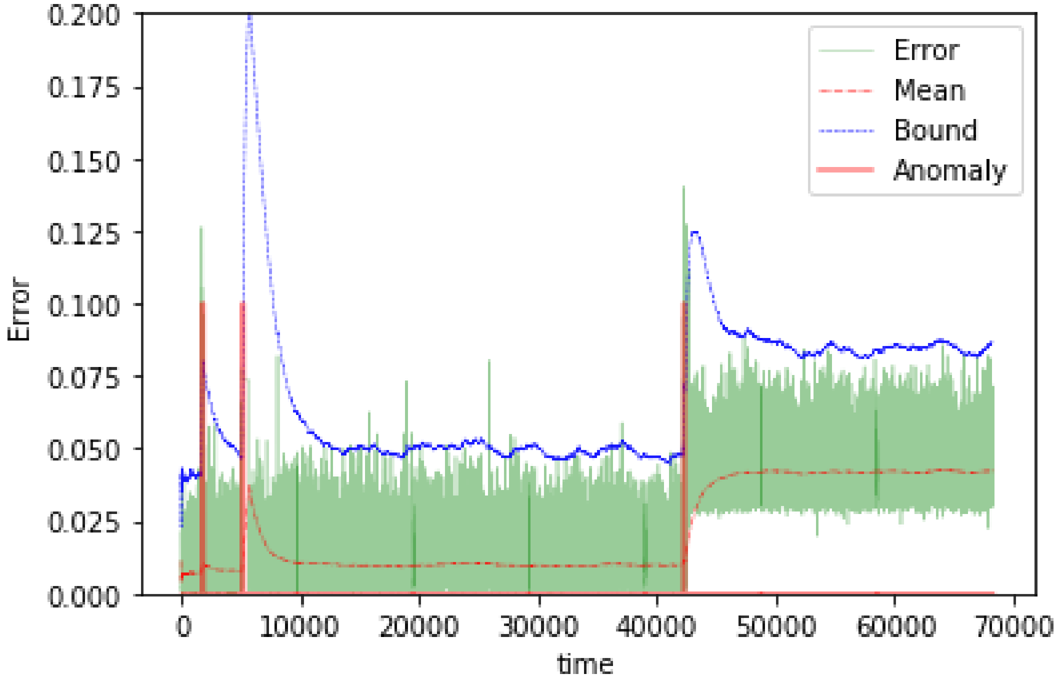

```
Training loss for final epoch is 0.007326699576806277
Validation loss for final epoch is 0.006439157382585109
---------- Beginning tests for nn2_5 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−4
−2
0
2
4

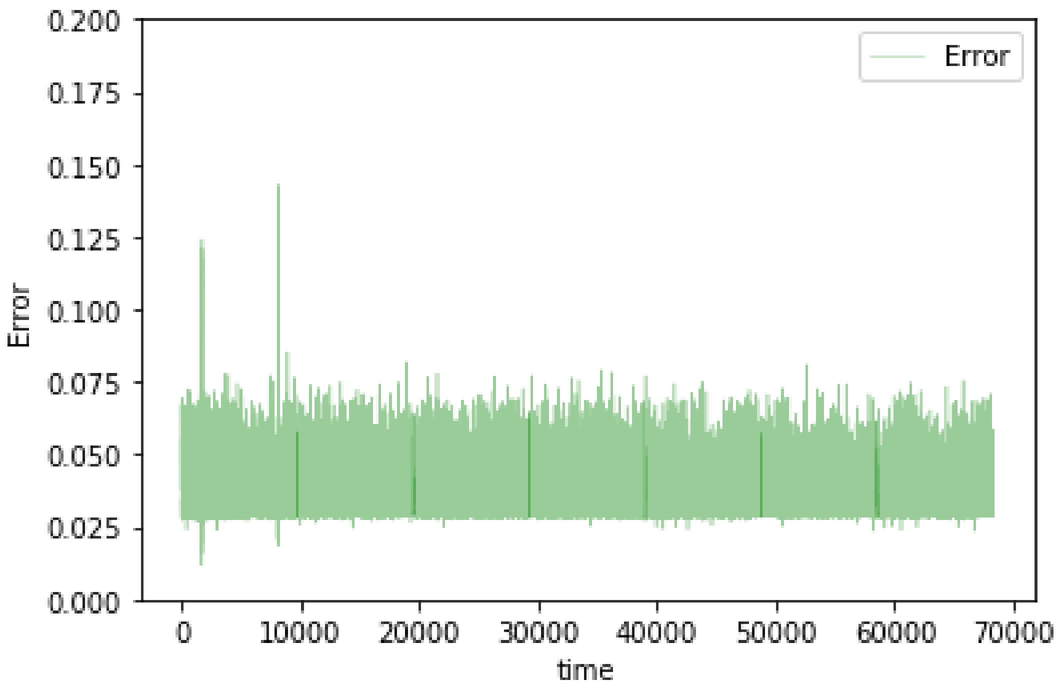

6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

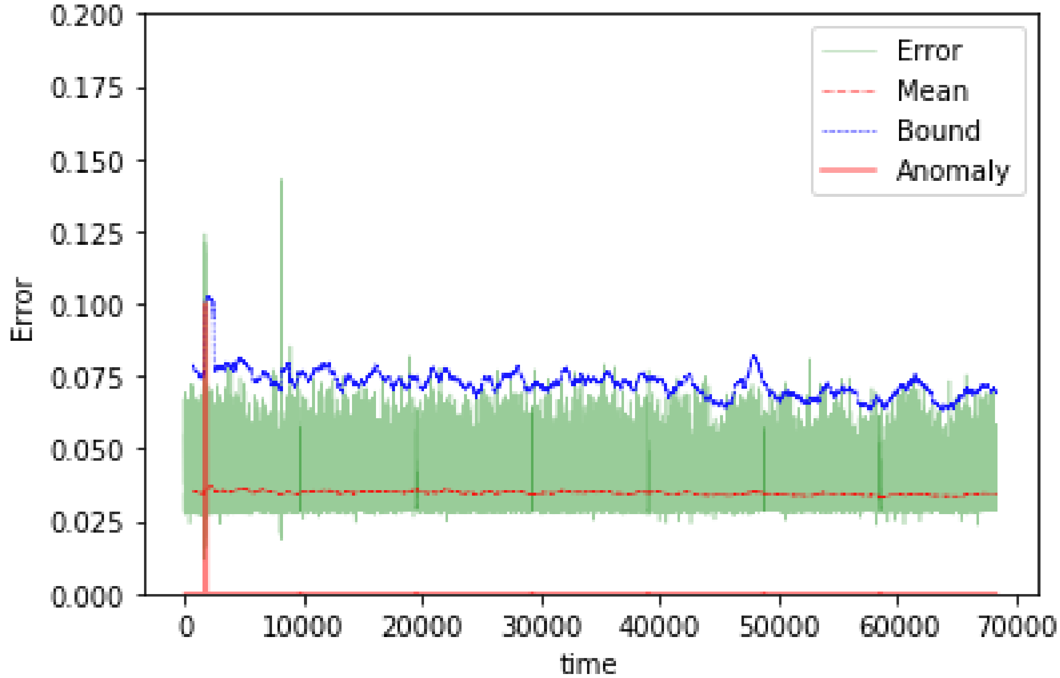

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

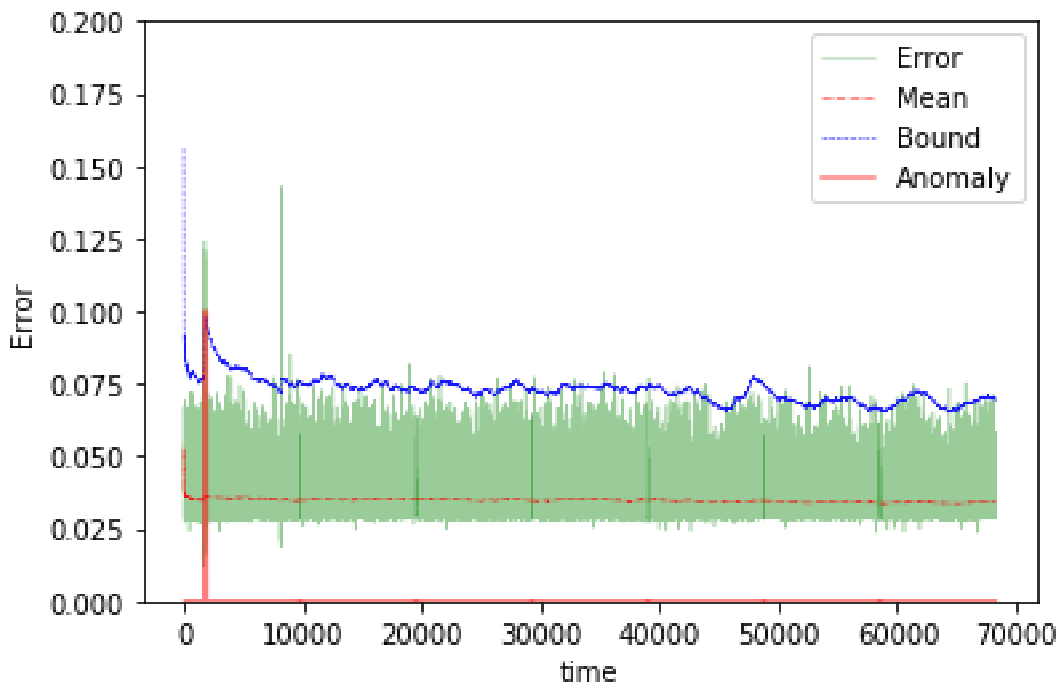

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



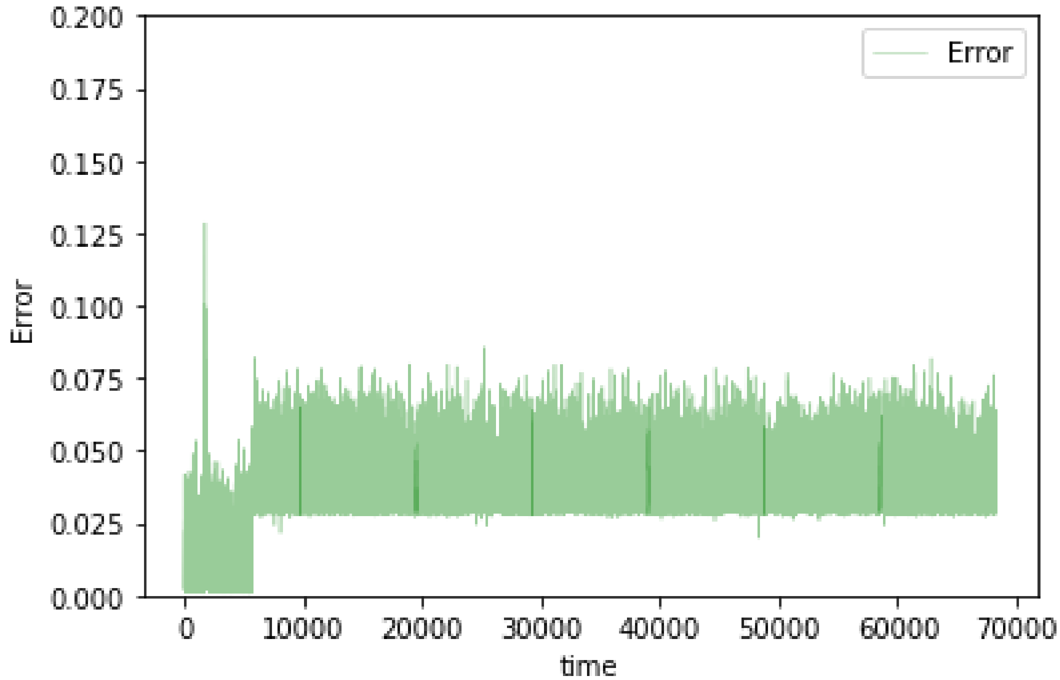

```
The mean error for nn2_5_disk_IO_start_ is 0.022448879488858434 for length 68294
Testing on Avg. load data.
```

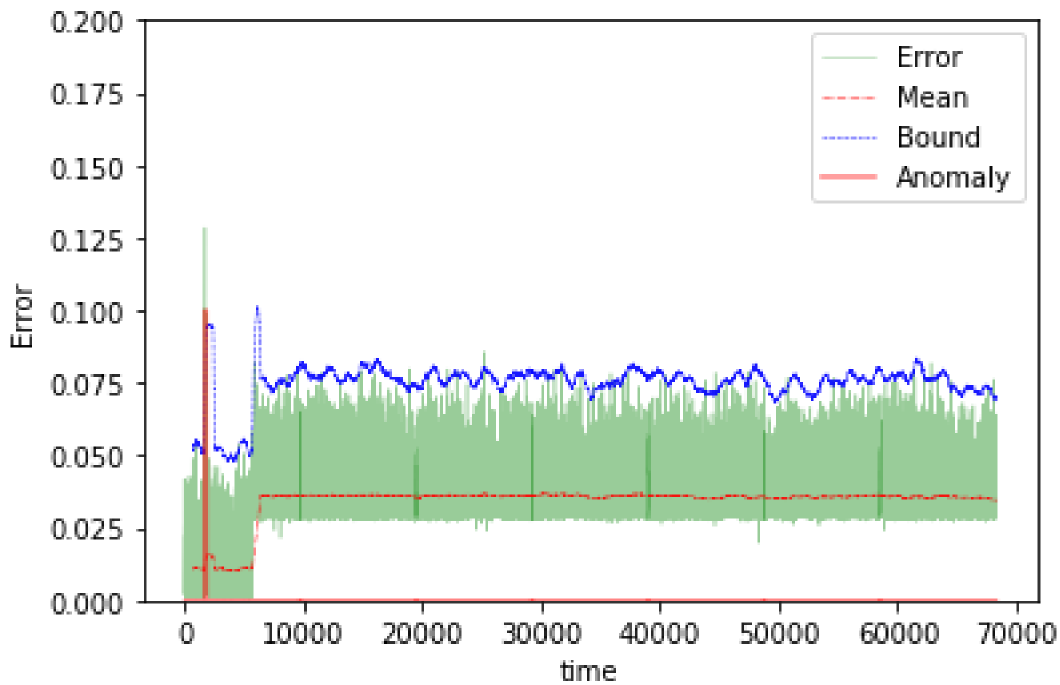


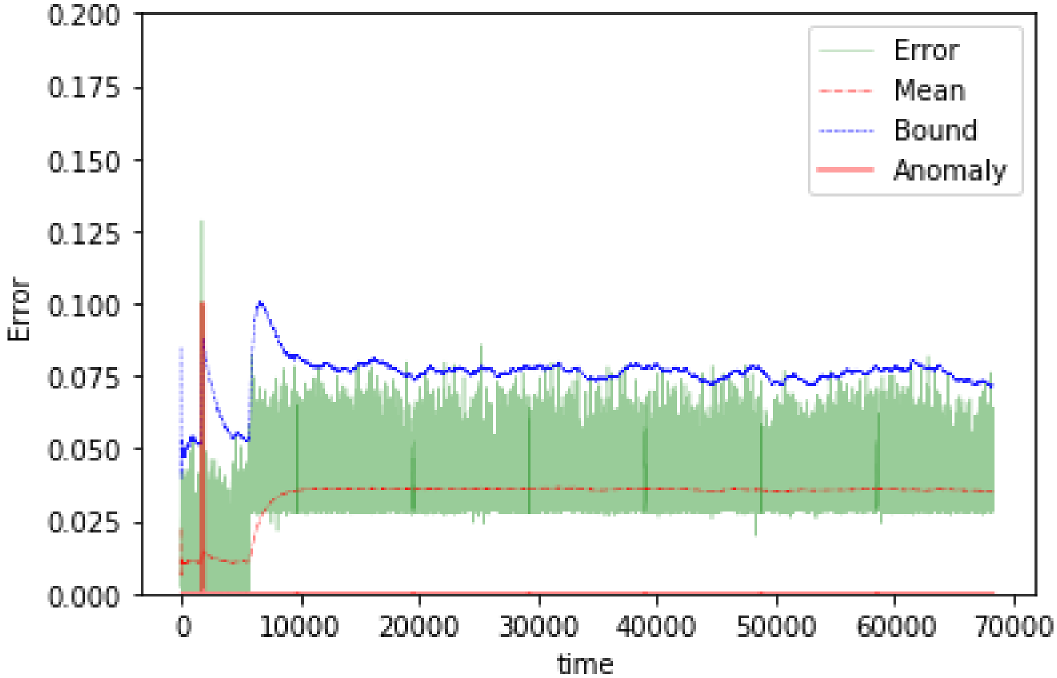

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

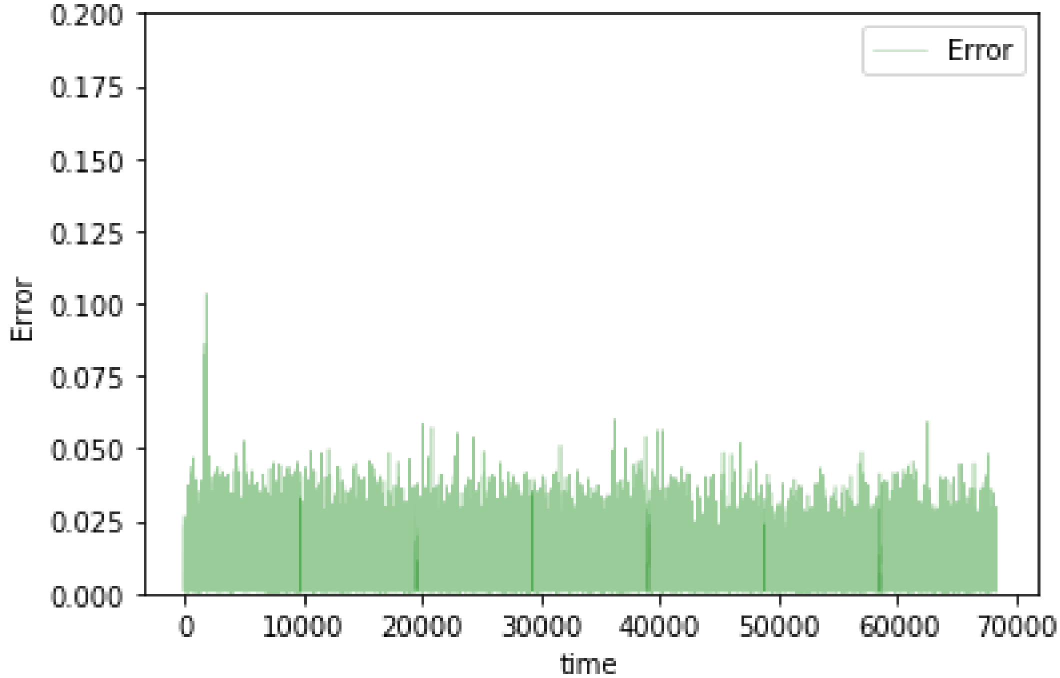
```
The mean error for nn2_5_avg_load_ is 0.03472324219922851 for length 68294
Testing on app change early data.
```

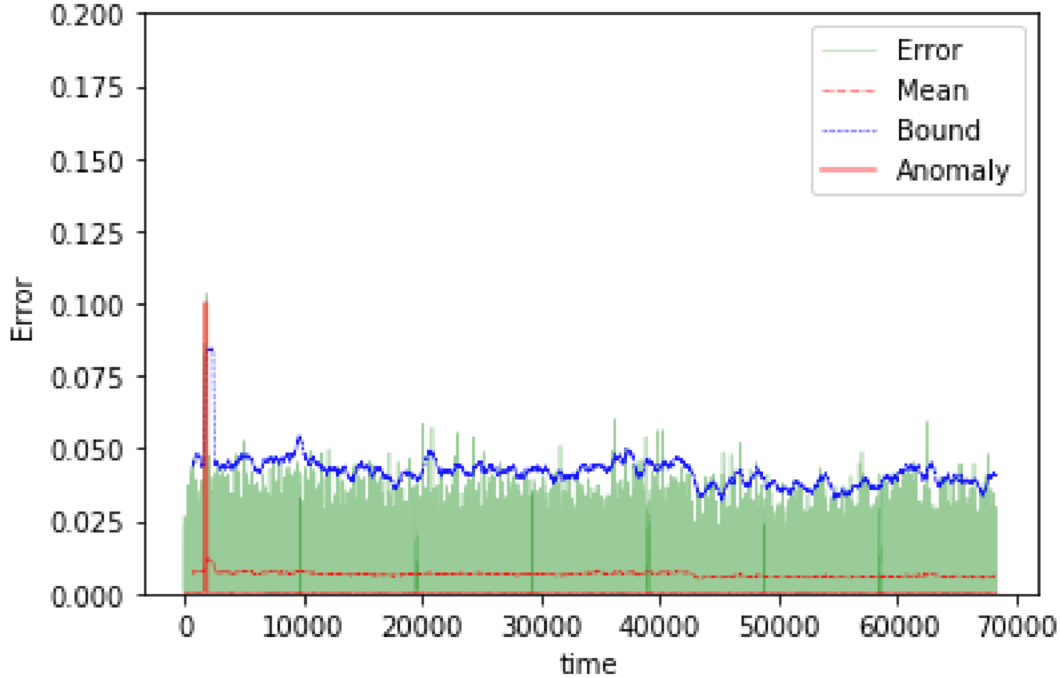


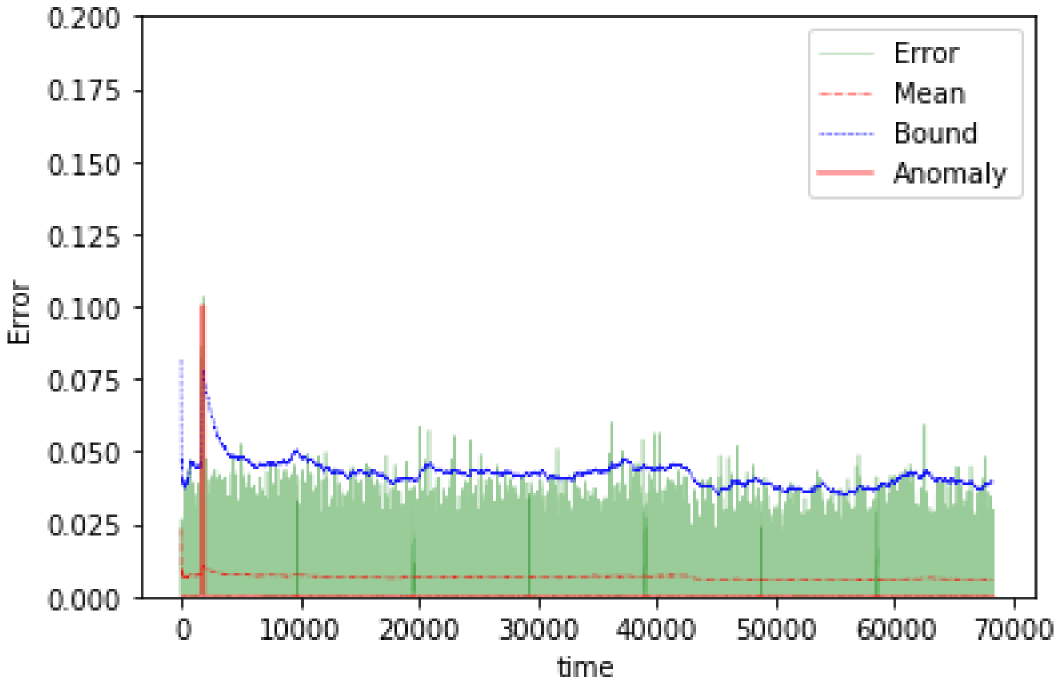





```
The mean error for nn2_5_app_change_early_ is 0.033884485405774246 for length 68294
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

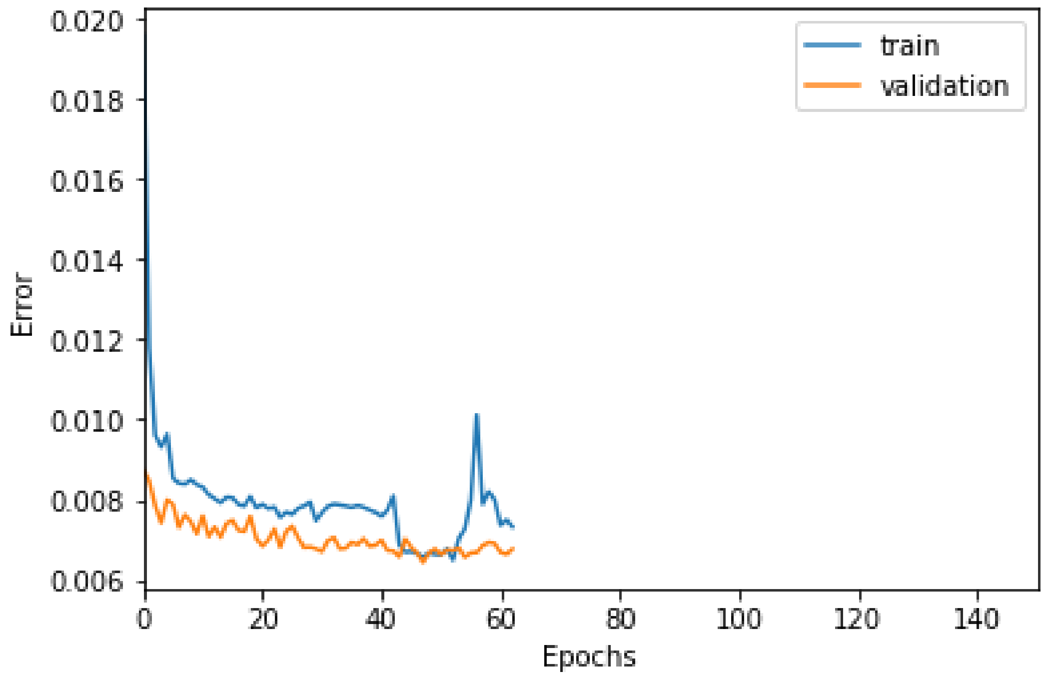

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



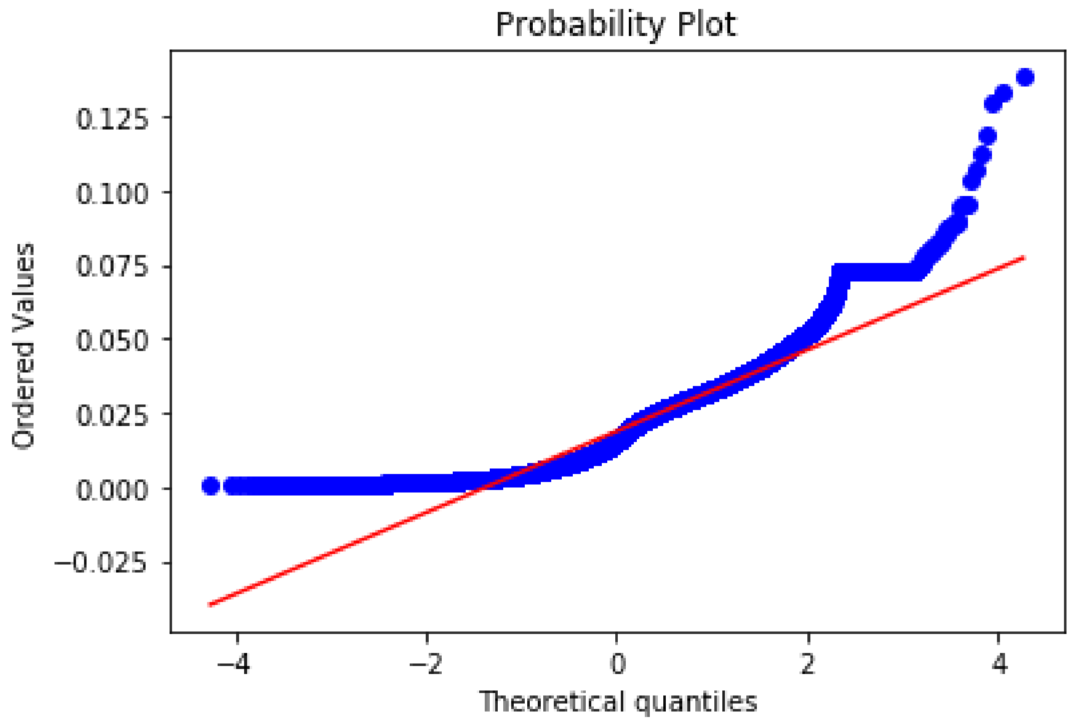

```
The mean error for nn2_5_normal_ is 0.006574076098585556 for length 68294
====================
```

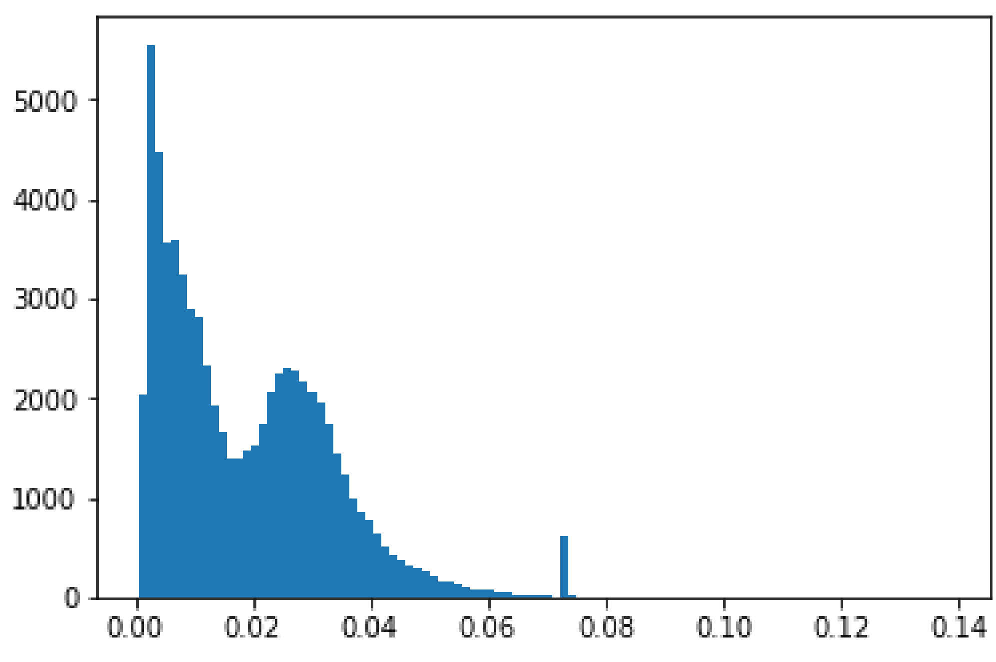
10 steps

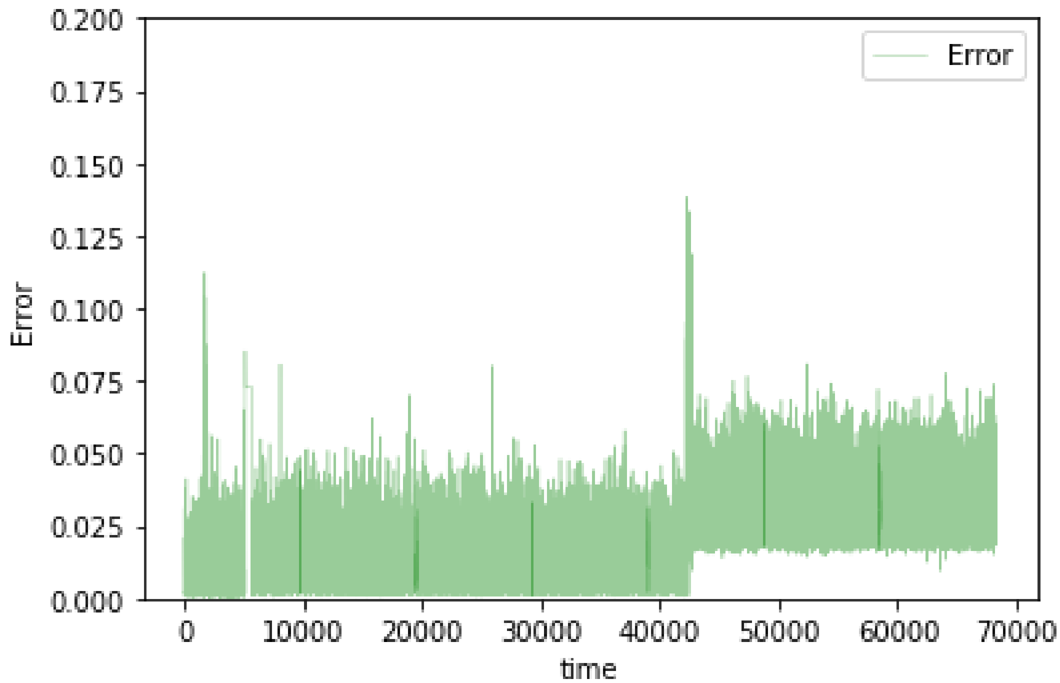
```
In [103]: TIMESTEPS = 10
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS)
          vgen = flat_generator(val_X, TIMESTEPS)
          name = "nn2_10"

In [104]: input_layer = Input(shape=(TIMESTEPS*DIM,))
          hidden = Dense(500, activation='relu')(input_layer)
          hidden = Dense(100, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [105]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [106]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```

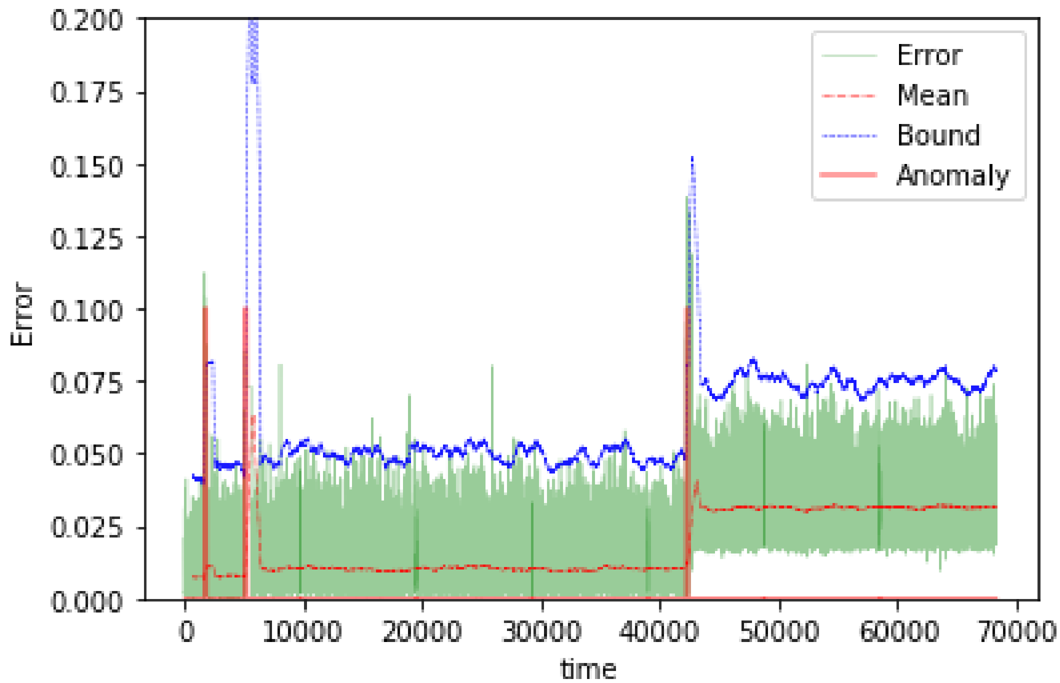


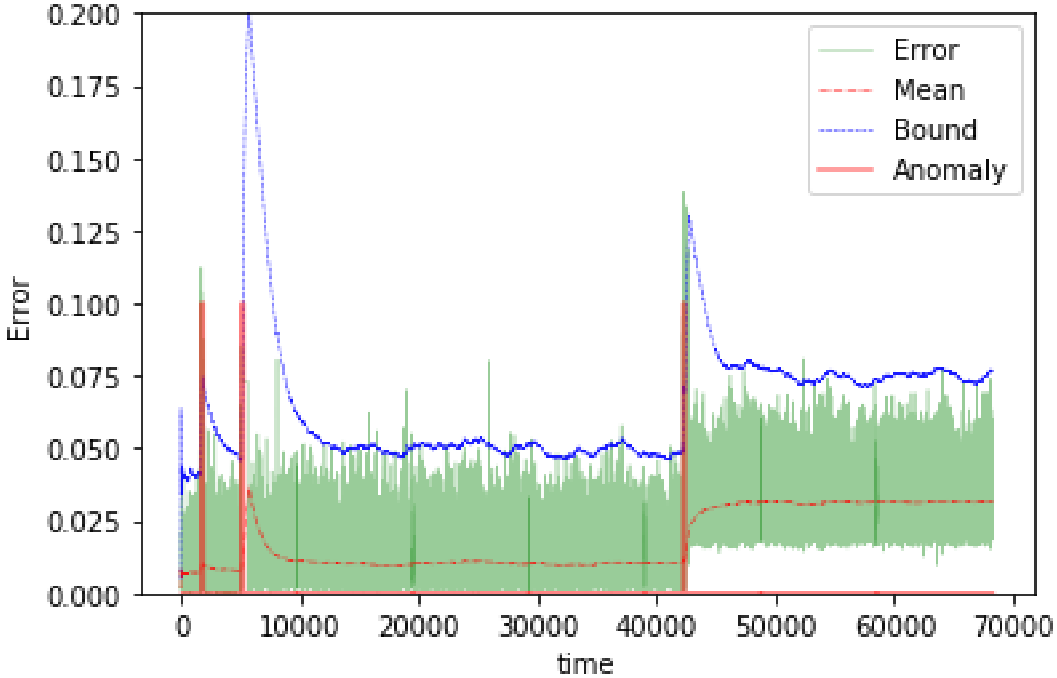

```
Training loss for final epoch is 0.007328209287719801
Validation loss for final epoch is 0.006775599585031159
---------- Beginning tests for nn2_10 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−4
−2
0
2
4

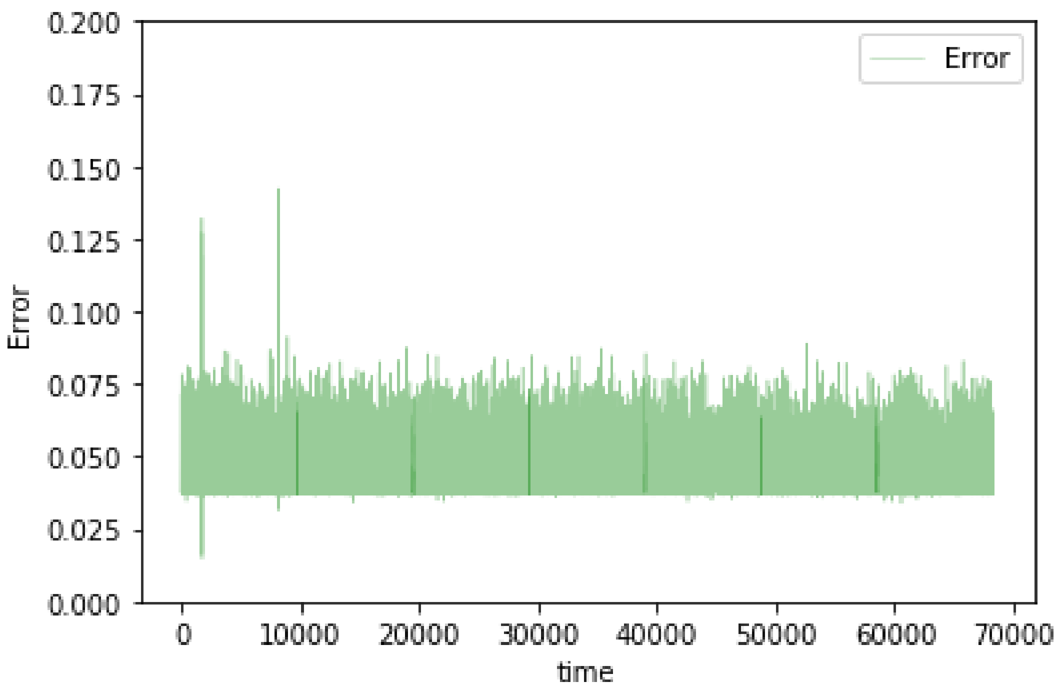

5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

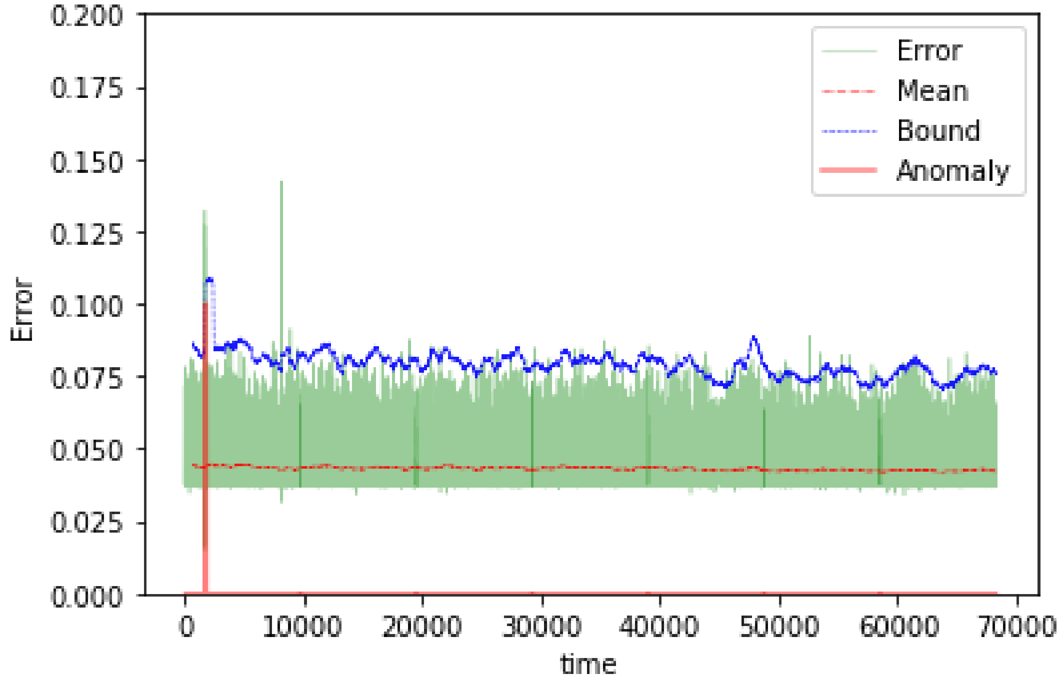

Error
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

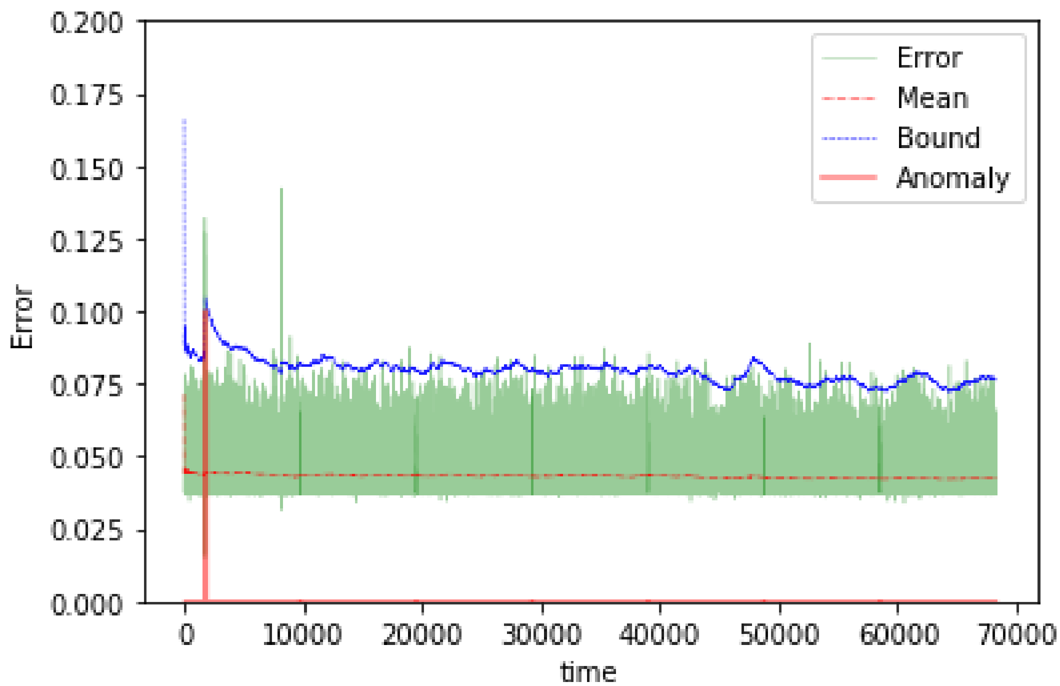



```
The mean error for nn2_10_disk_IO_start_ is 0.01873871565588481 for length 68289
Testing on Avg. load data.
```



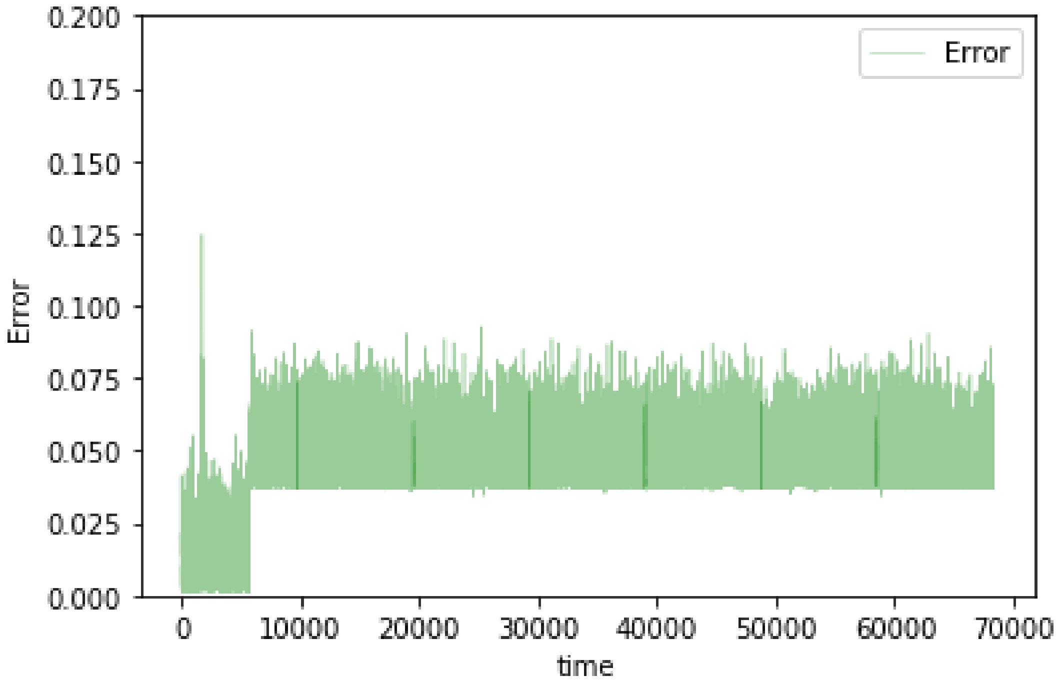


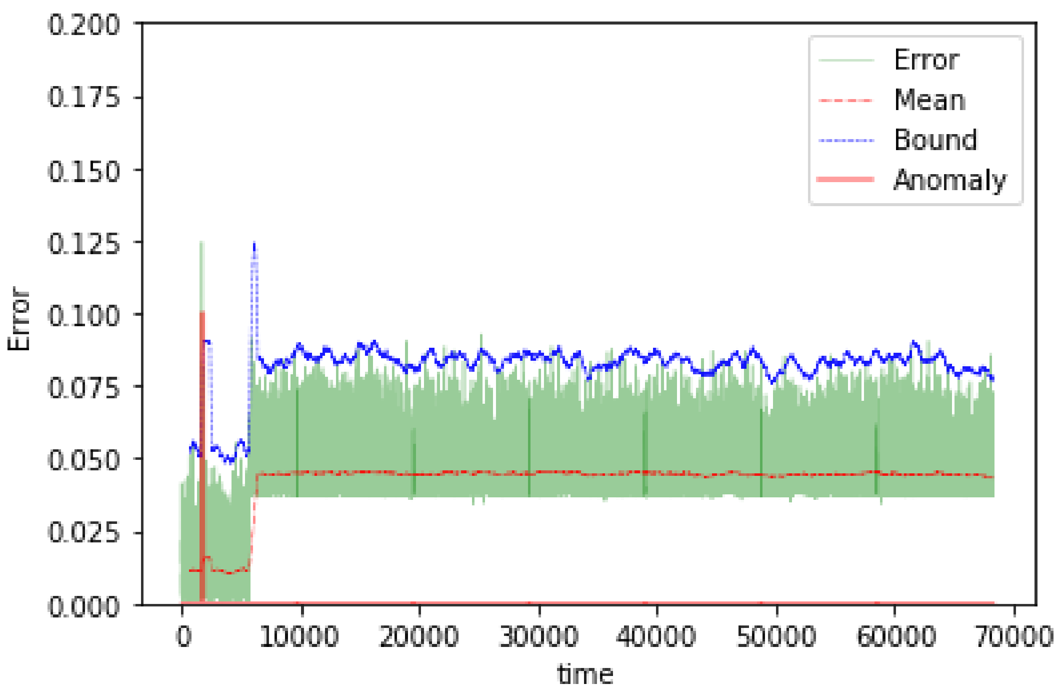
Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

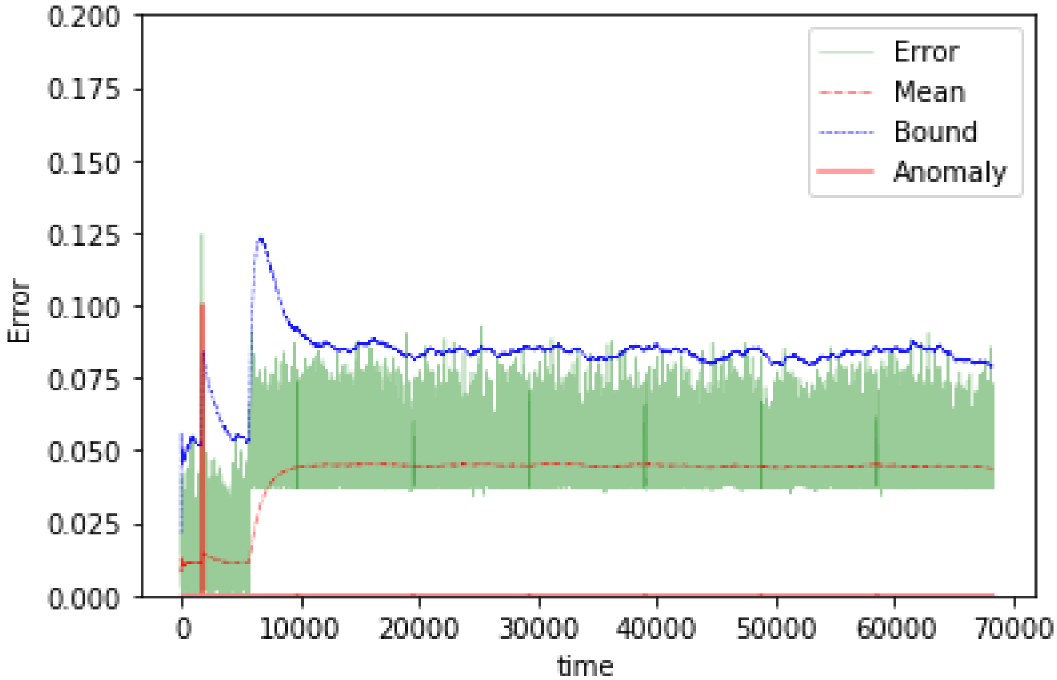


Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

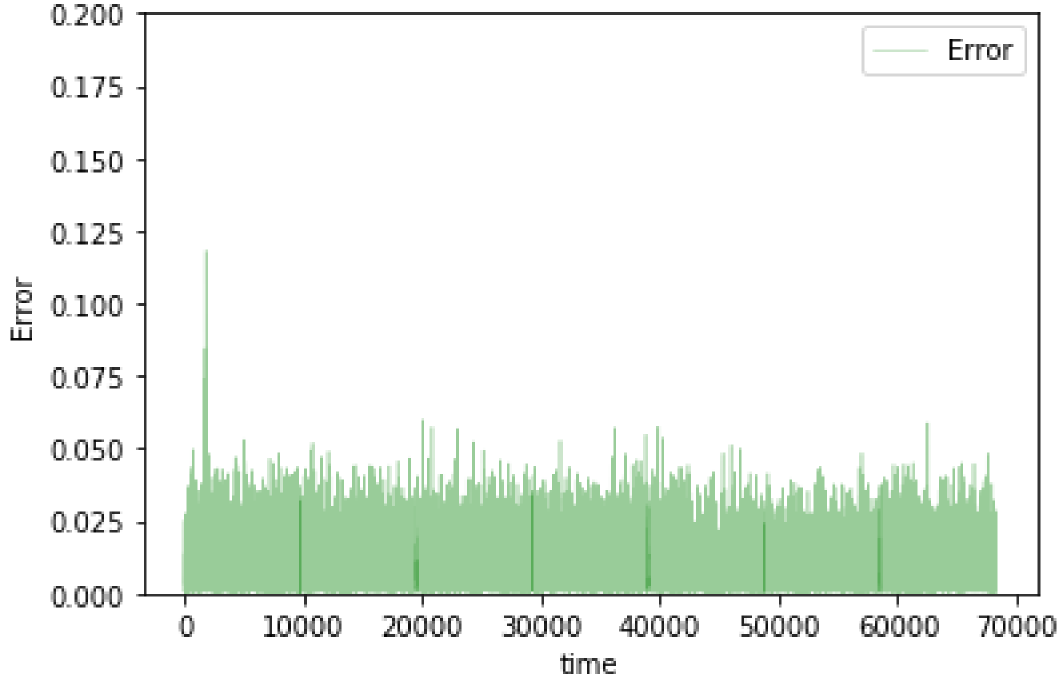
```
The mean error for nn2_10_avg_load_ is 0.04312592327915144 for length 68289
Testing on app change early data.
```

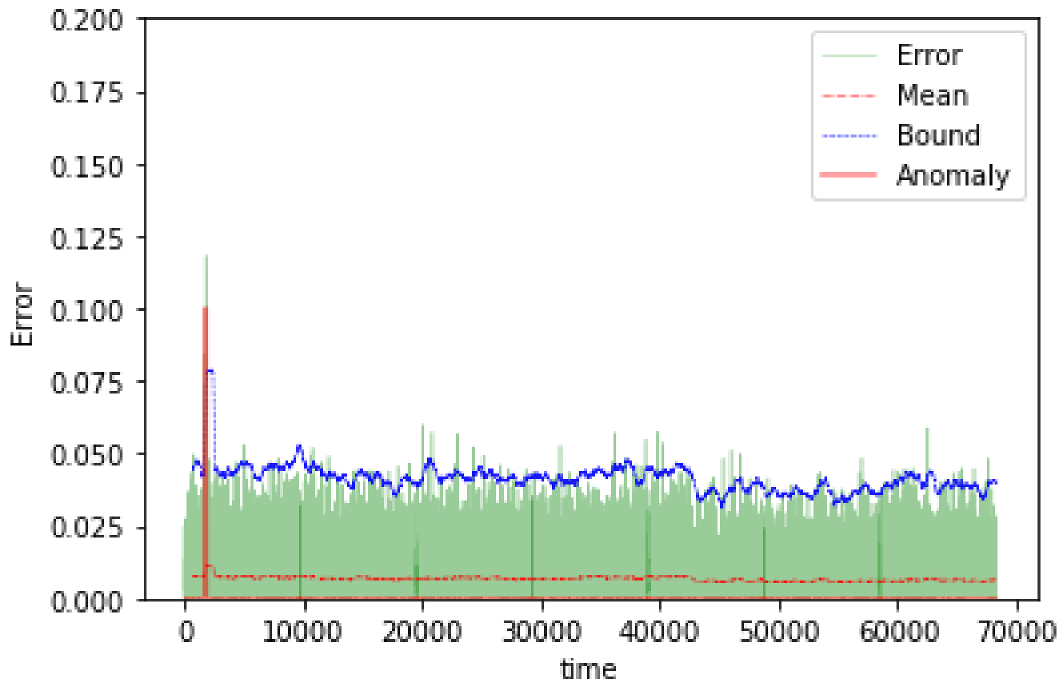


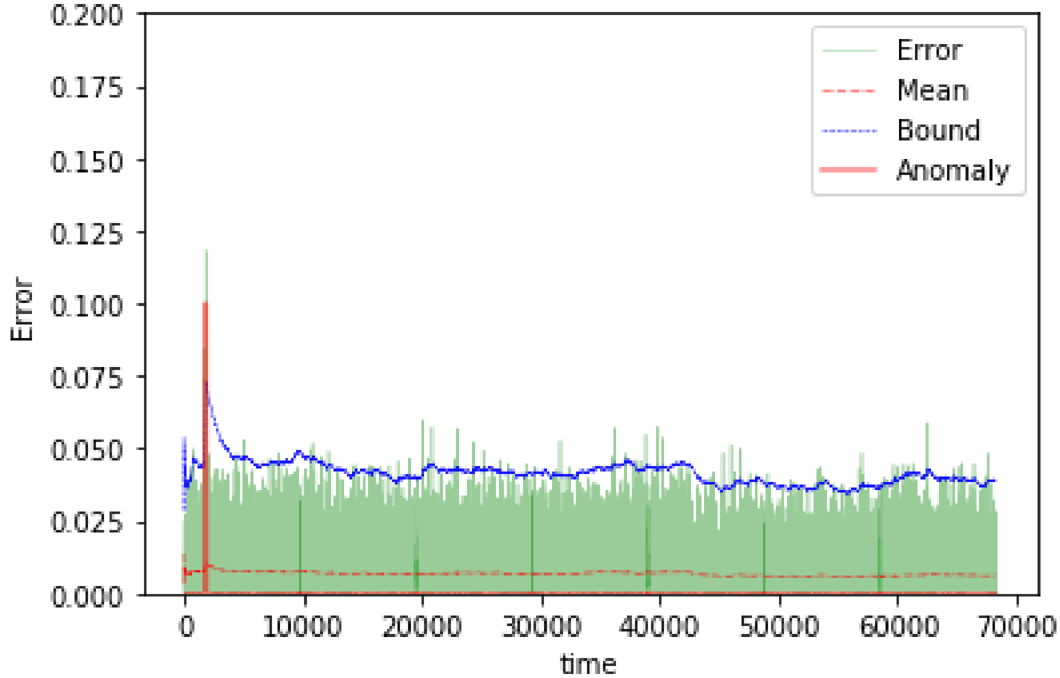





```
The mean error for nn2_10_app_change_early_ is 0.041910586998065734 for length 68289
Testing on Normal data.
```


Error
Error
time

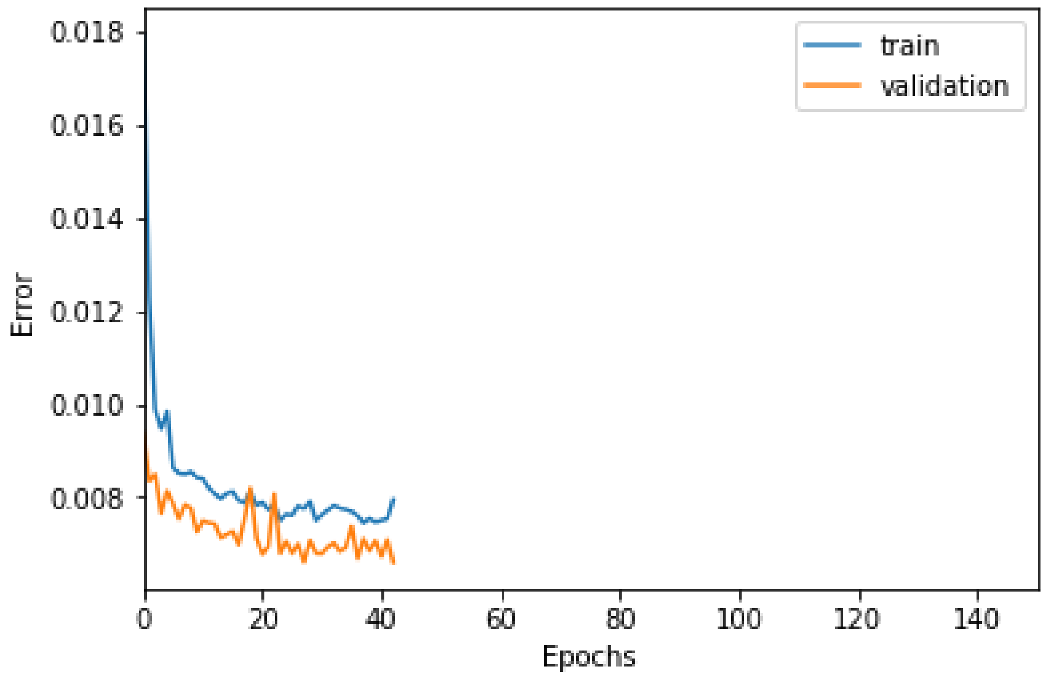

Error
Mean
Bound
Anomaly
Error
time



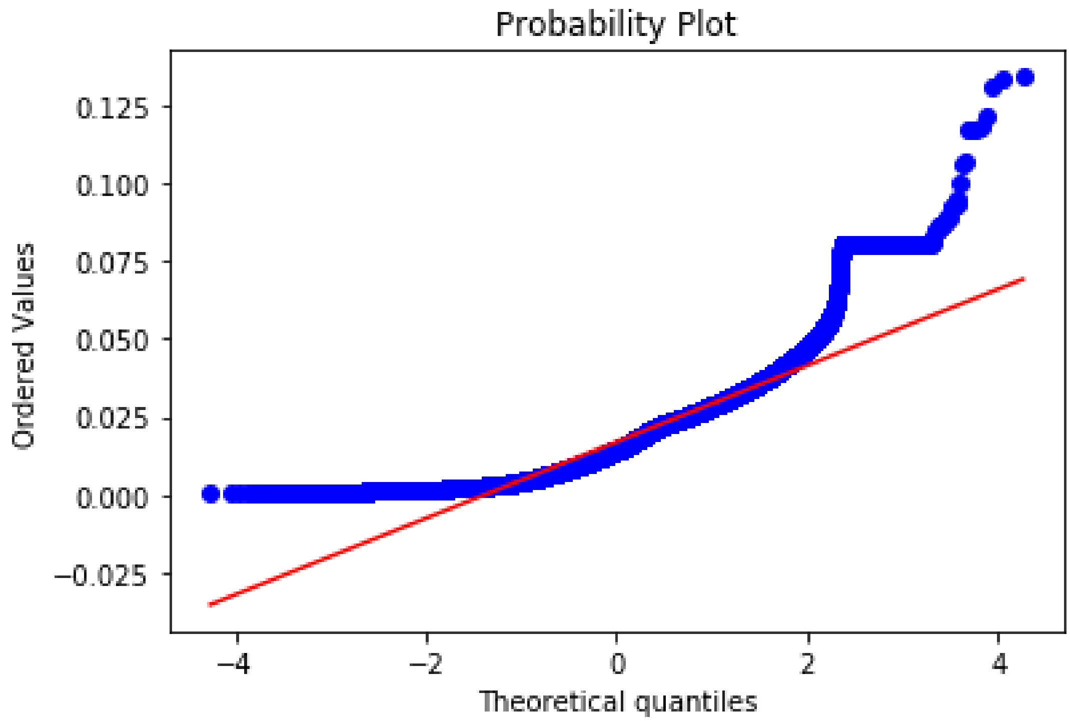

```
The mean error for nn2_10_normal_ is 0.006778446547927589 for length 68289
====================
```

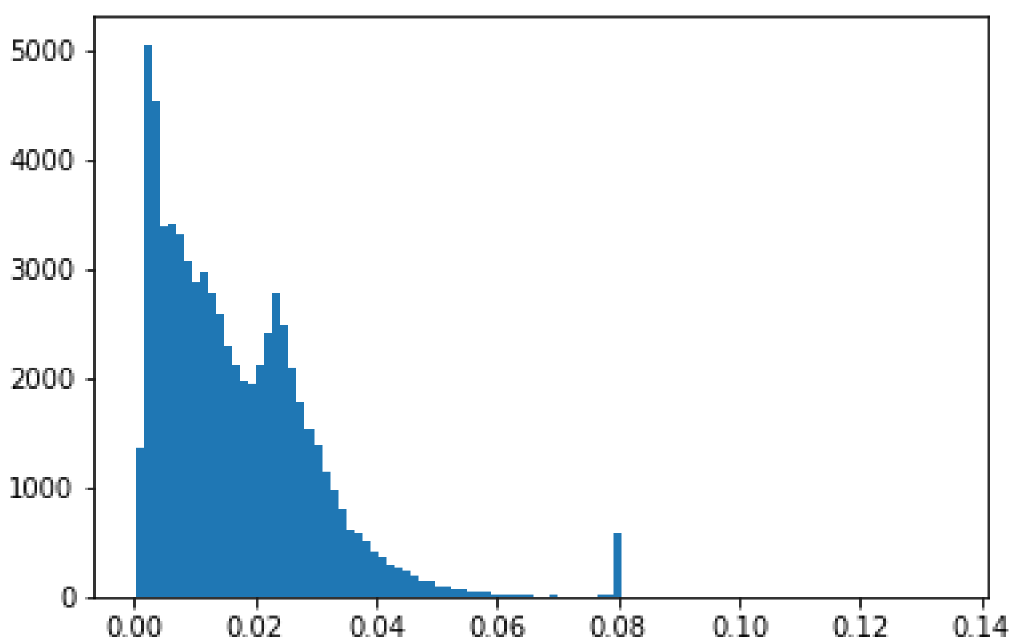
20 steps

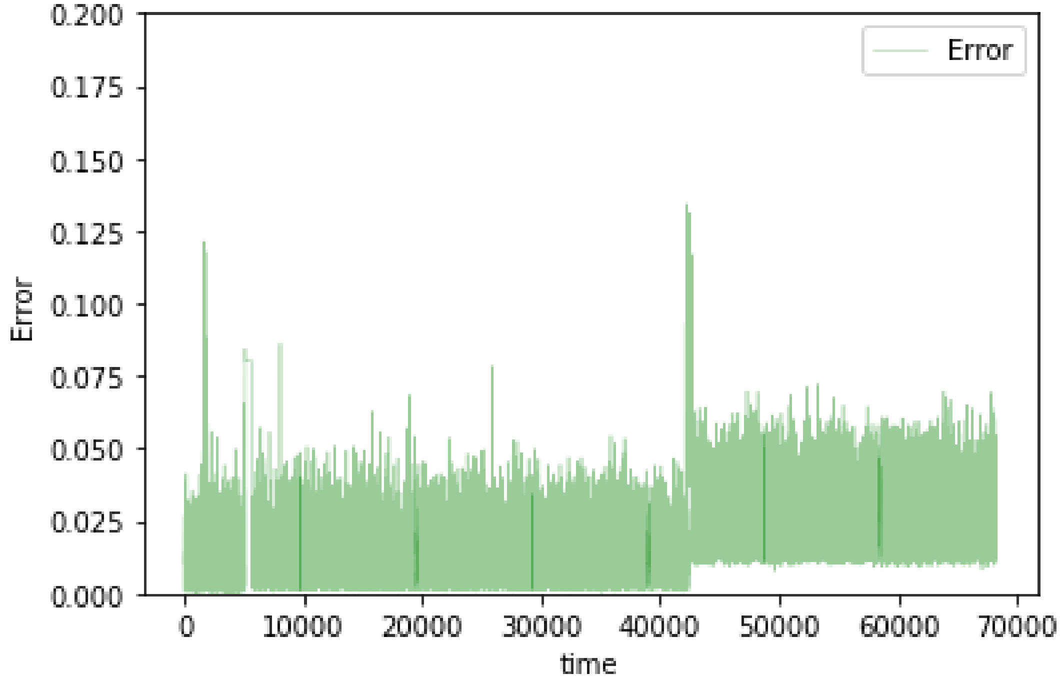
```
In [107]: TIMESTEPS = 20
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS)
          vgen = flat_generator(val_X, TIMESTEPS)
          name = "nn2_20"

In [108]: input_layer = Input(shape=(TIMESTEPS*DIM,))
          hidden = Dense(500, activation='relu')(input_layer)
          hidden = Dense(100, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [109]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [110]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```



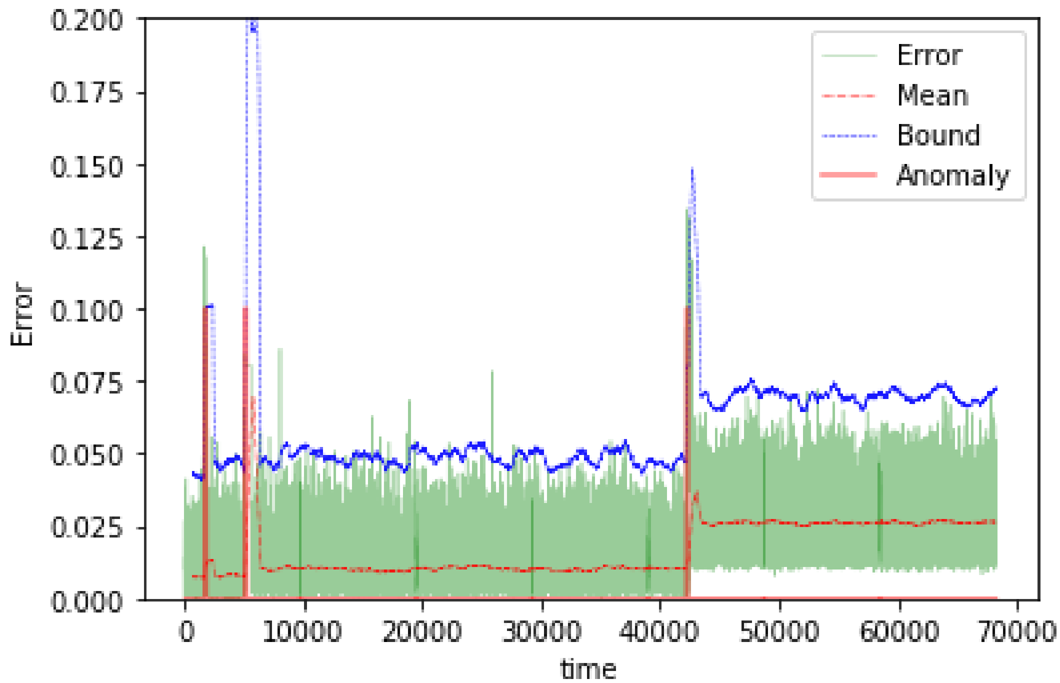

```
Training loss for final epoch is 0.007947208319325
Validation loss for final epoch is 0.006600935542257502
---------- Beginning tests for nn2_20 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

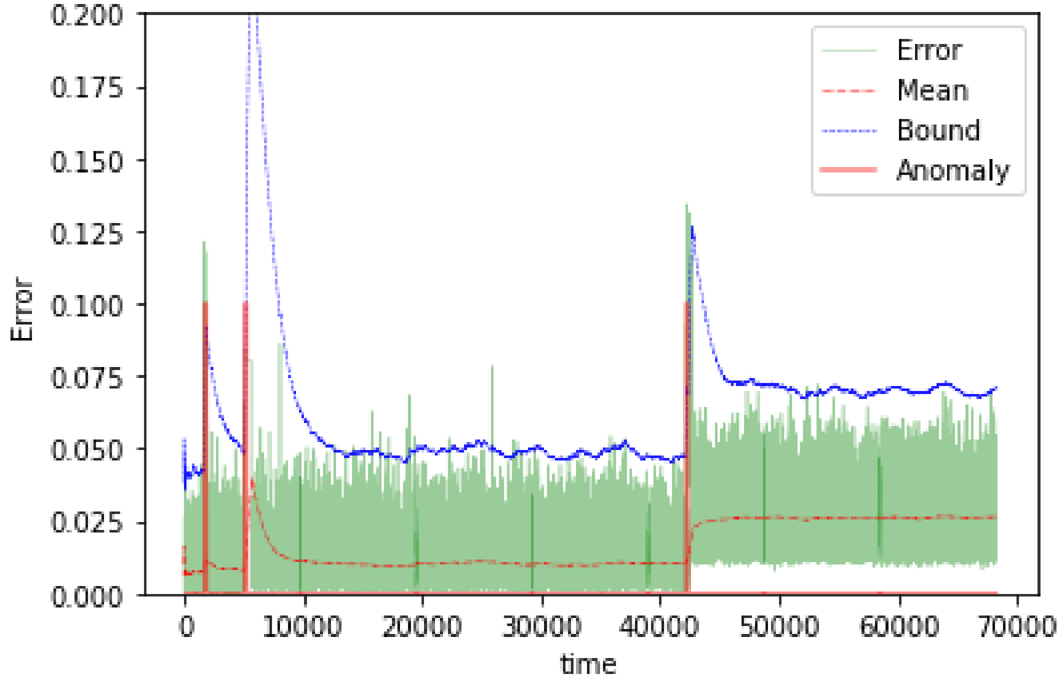

5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14



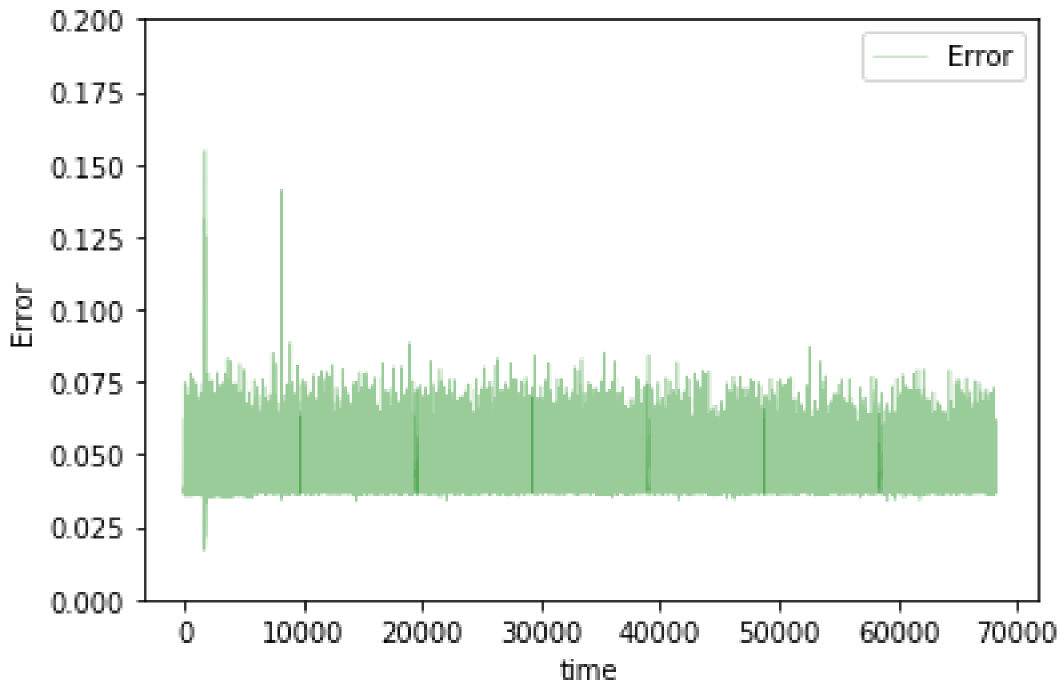
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

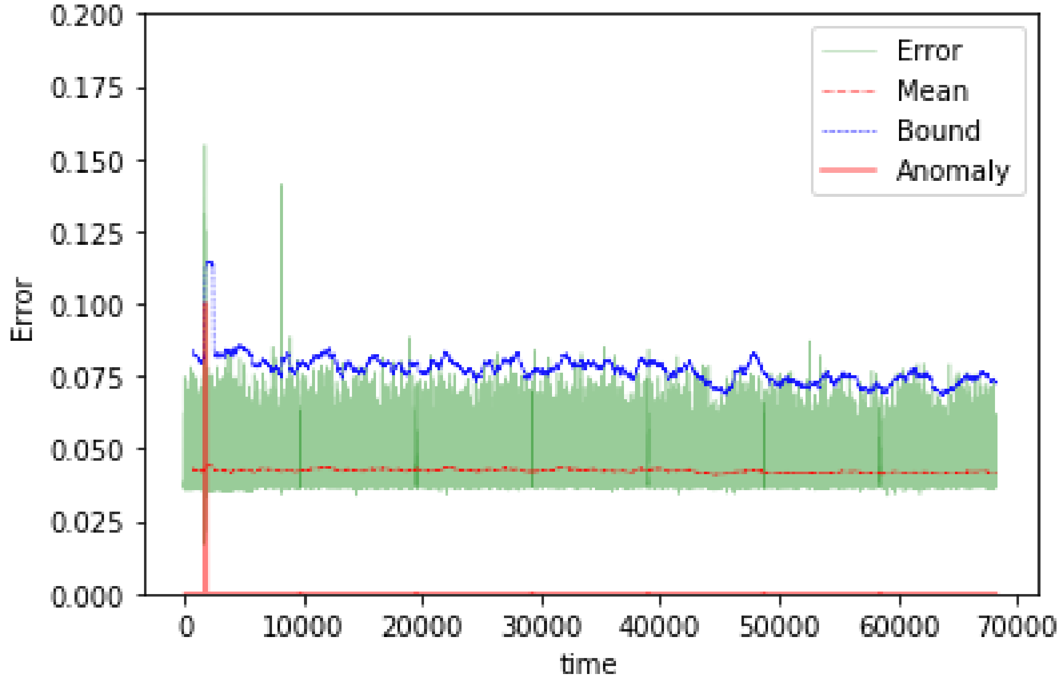


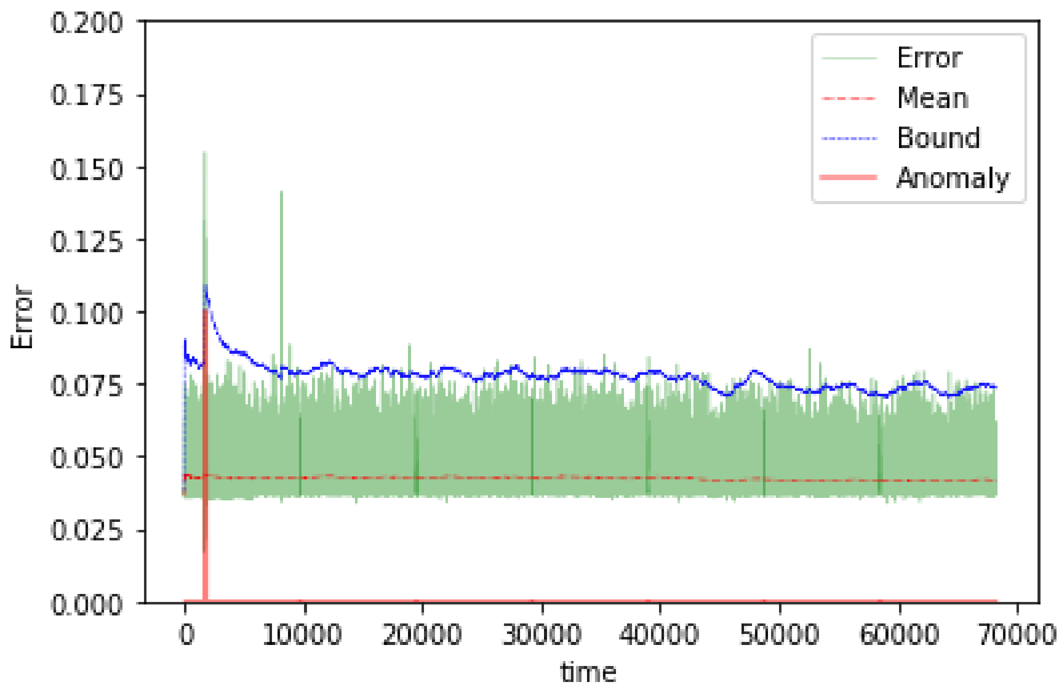
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



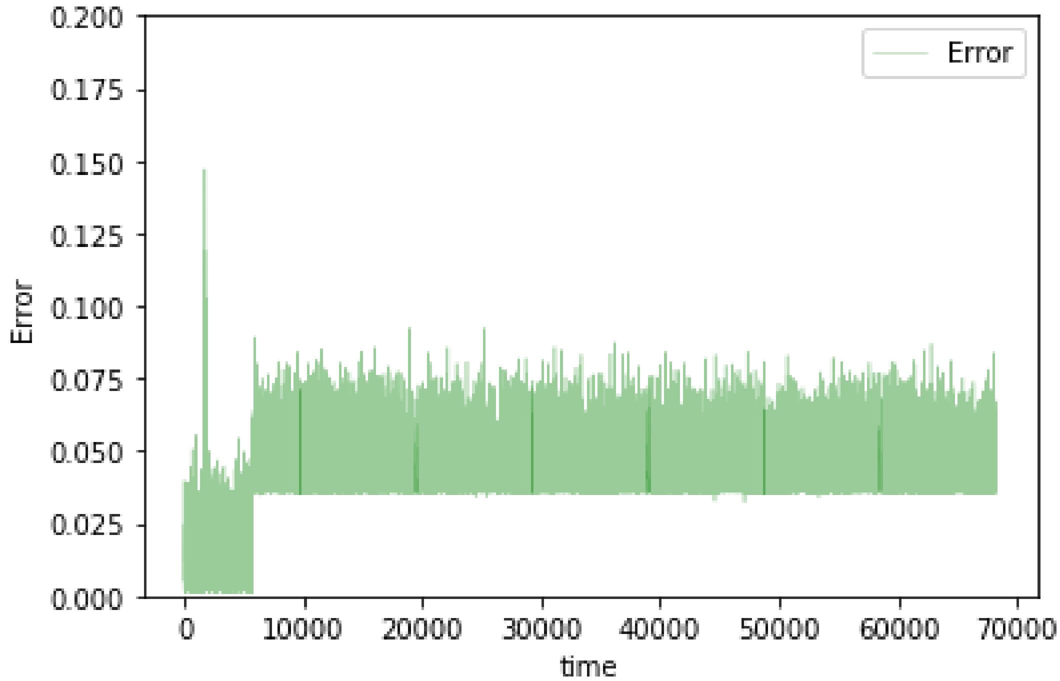

```
The mean error for nn2_20_disk_IO_start_ is 0.01689149934008681 for length 68279
Testing on Avg. load data.
```

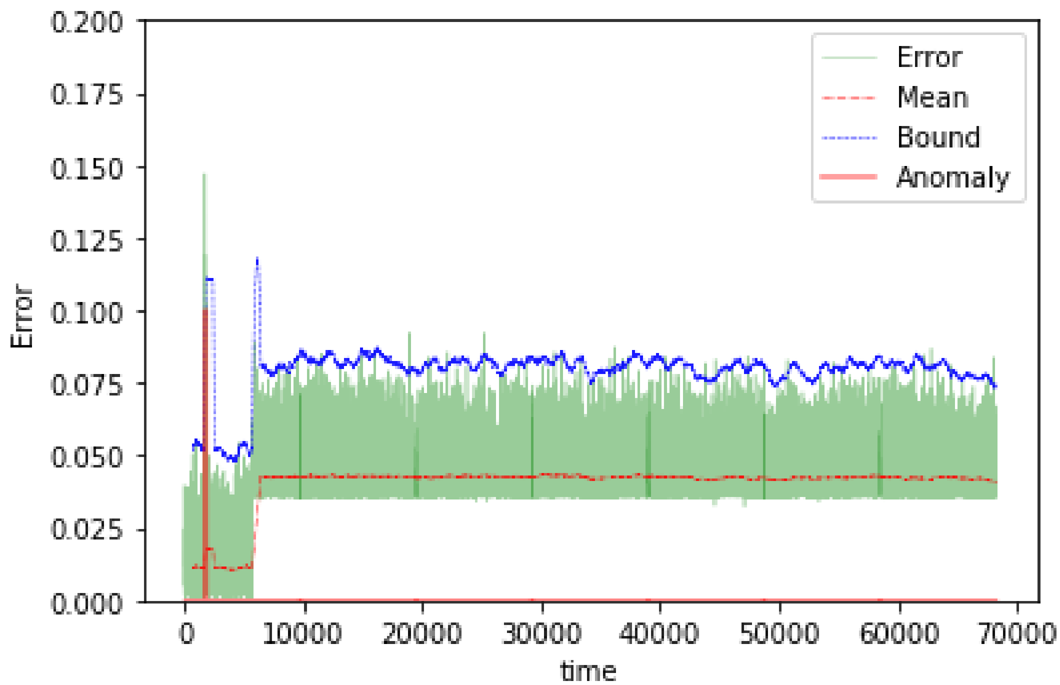


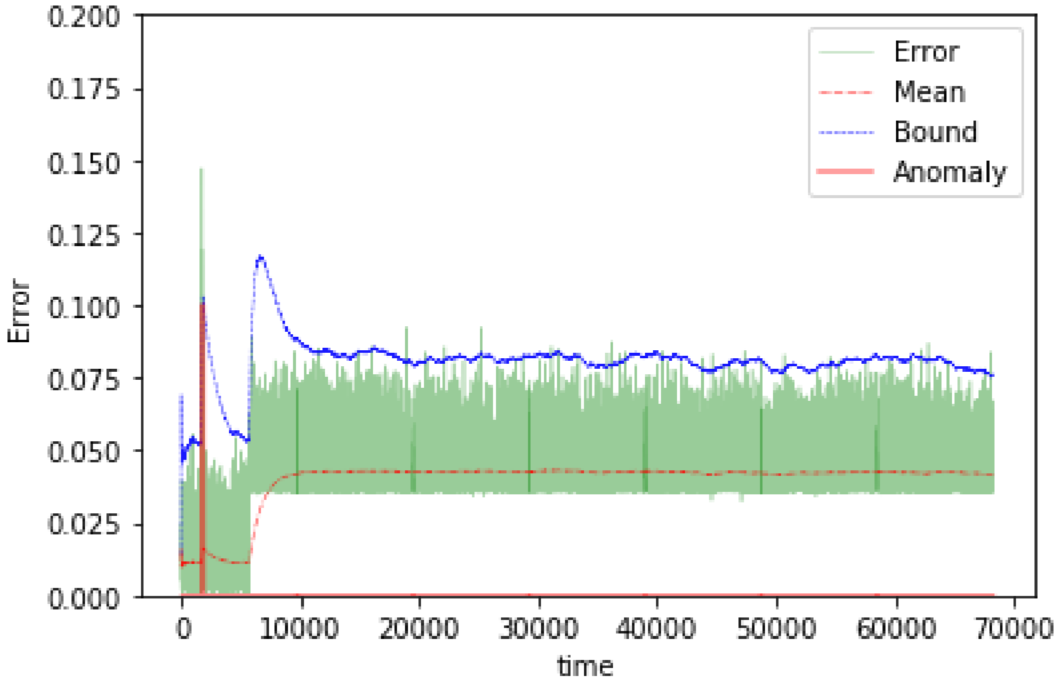

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

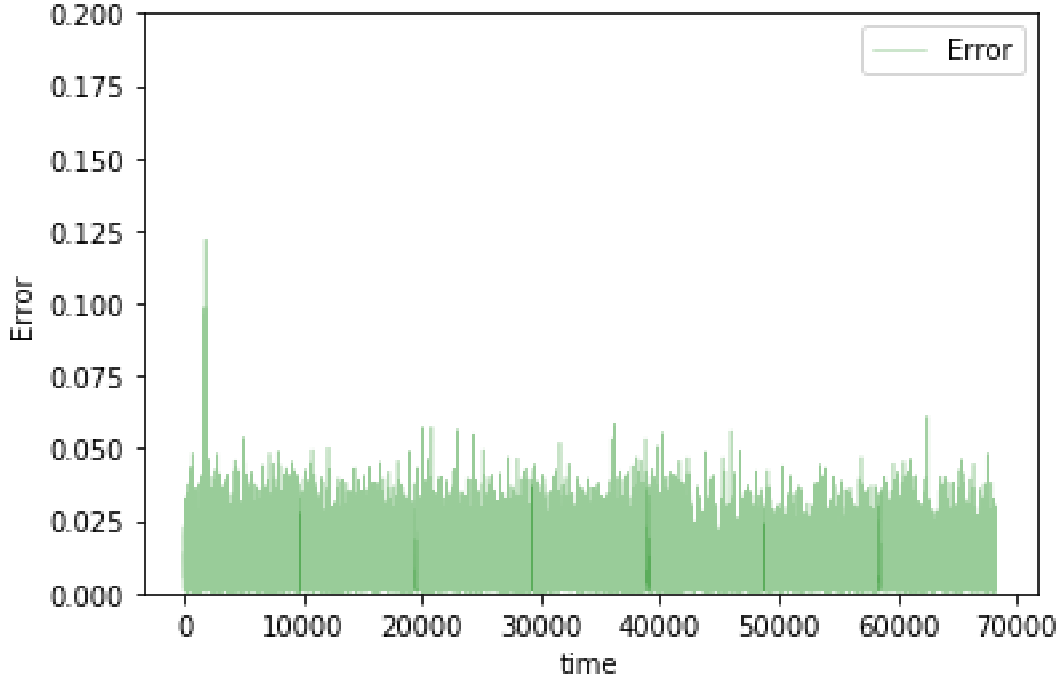
```
The mean error for nn2_20_avg_load_ is 0.04235663090951715 for length 68279
Testing on app change early data.
```



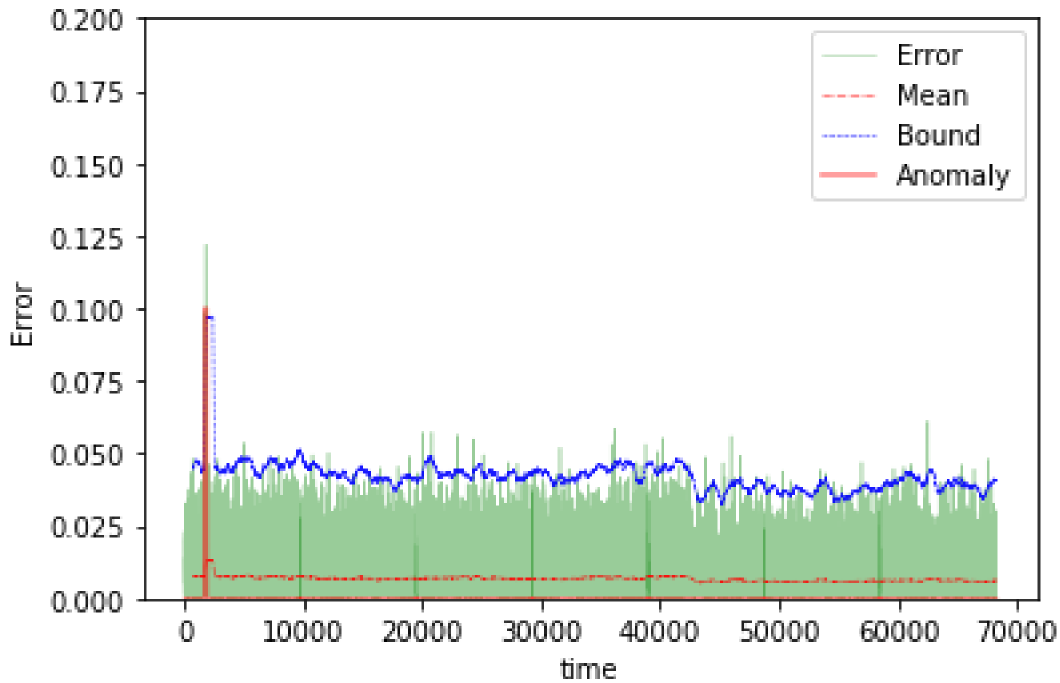




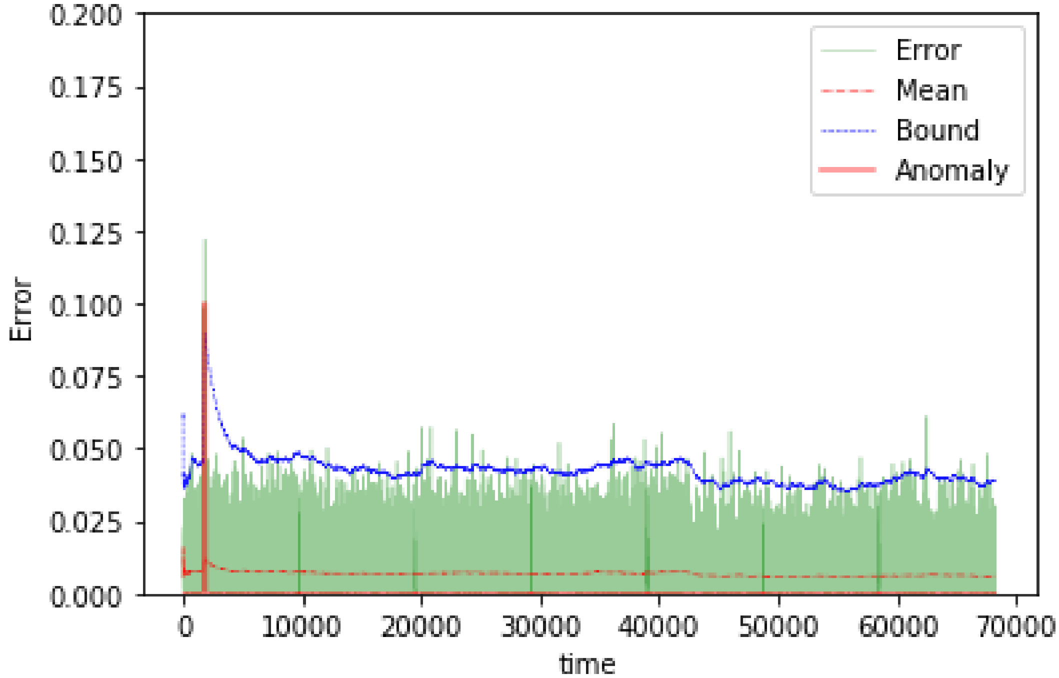

```
The mean error for nn2_20_app_change_early_ is 0.03994599118033188 for length 68279
Testing on Normal data.
```



0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
Error
0
10000
20000
30000
40000
50000
60000
70000
time



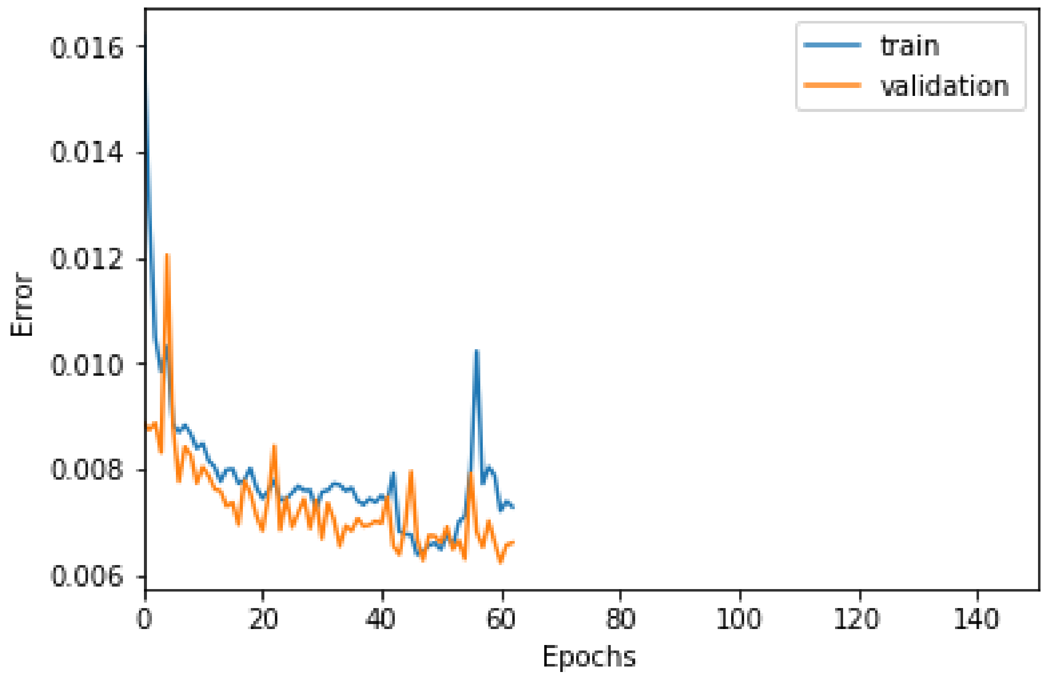
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
Error
Mean
Bound
Anomaly
0
10000
20000
30000
40000
50000
60000
70000
time



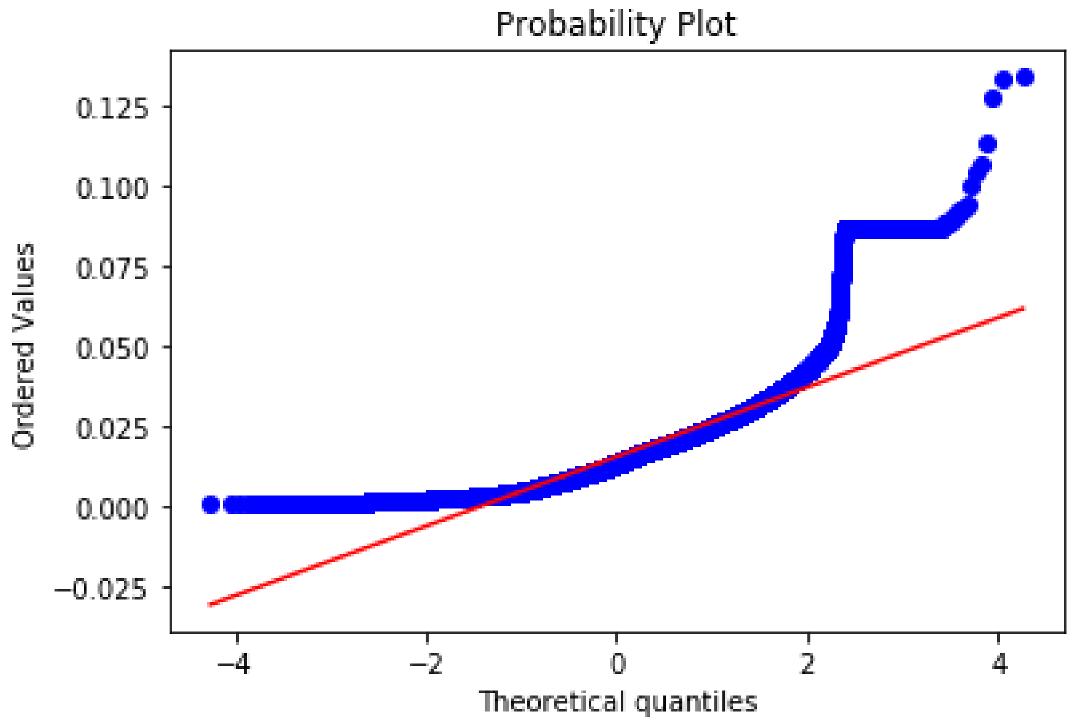

```
The mean error for nn2_20_normal_ is 0.006776399928838544 for length 68279
====================
```

50 steps

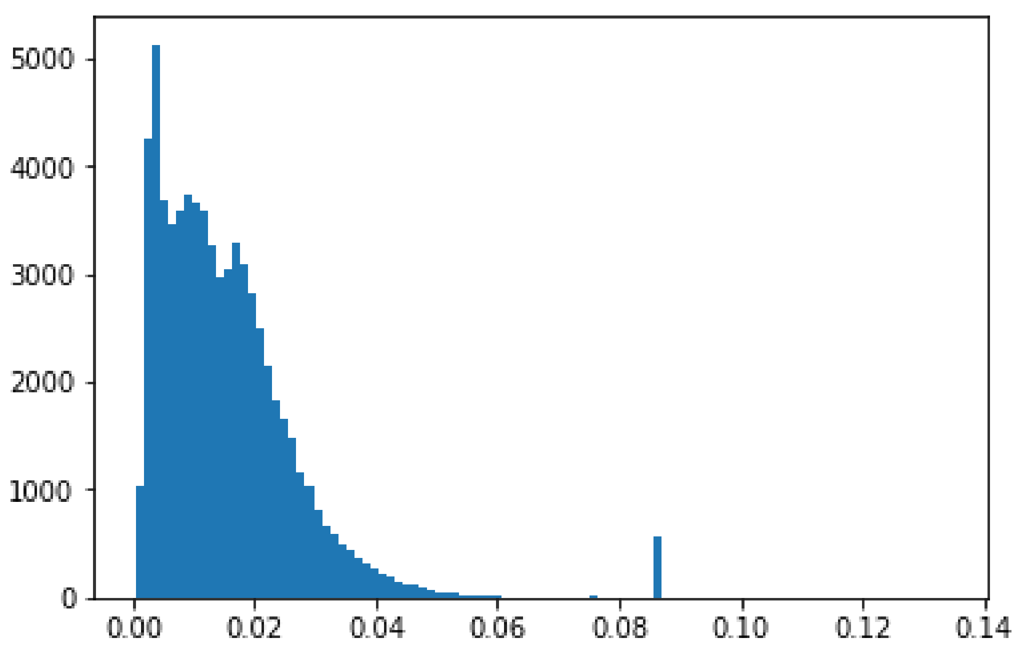
```
In [111]: TIMESTEPS = 50
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS)
          vgen = flat_generator(val_X, TIMESTEPS)
          name = "nn2_50"

In [112]: input_layer = Input(shape=(TIMESTEPS*DIM,))
          hidden = Dense(500, activation='relu')(input_layer)
          hidden = Dense(100, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [113]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [114]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```



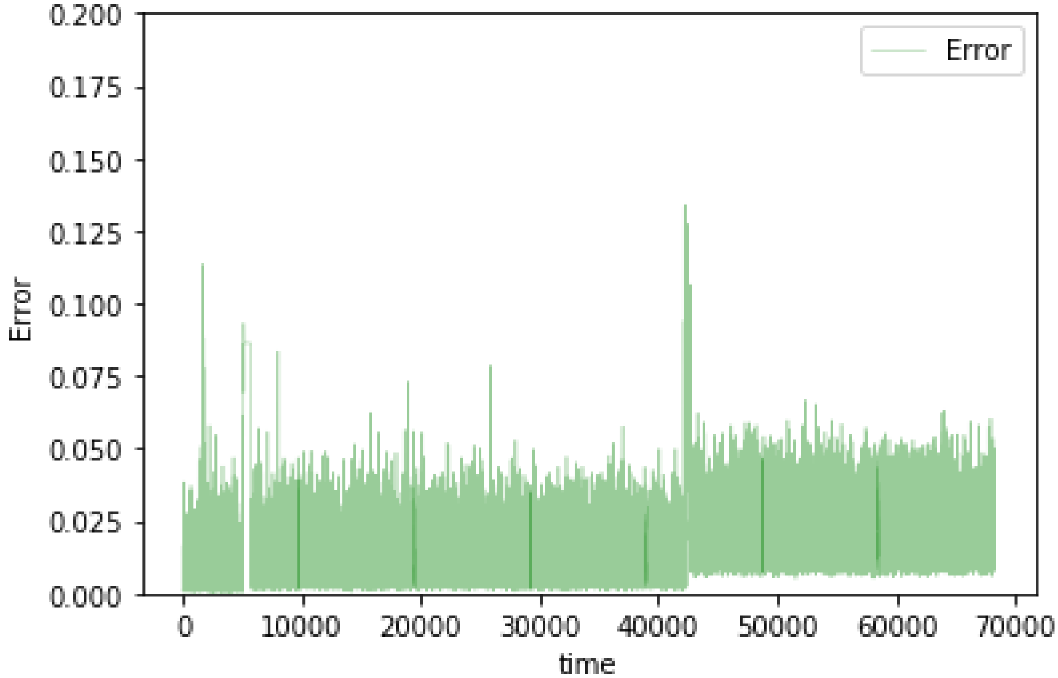

```
Training loss for final epoch is 0.0072879627015208824
Validation loss for final epoch is 0.006609717860235832
---------- Beginning tests for nn2_50 ----------
Testing on Disk IO begin data.
```

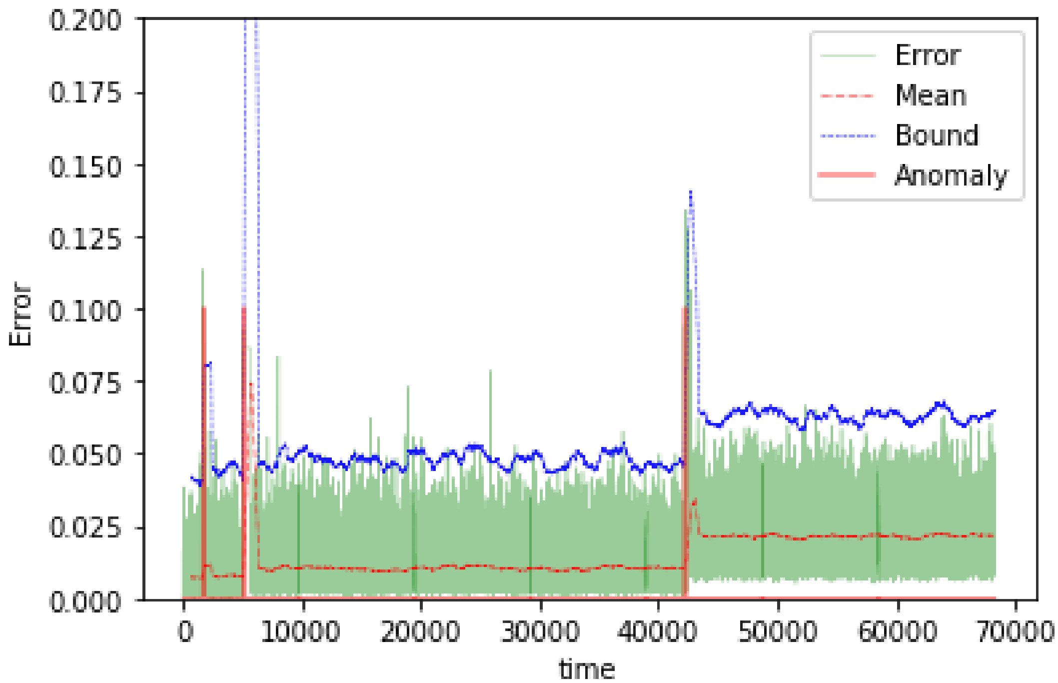


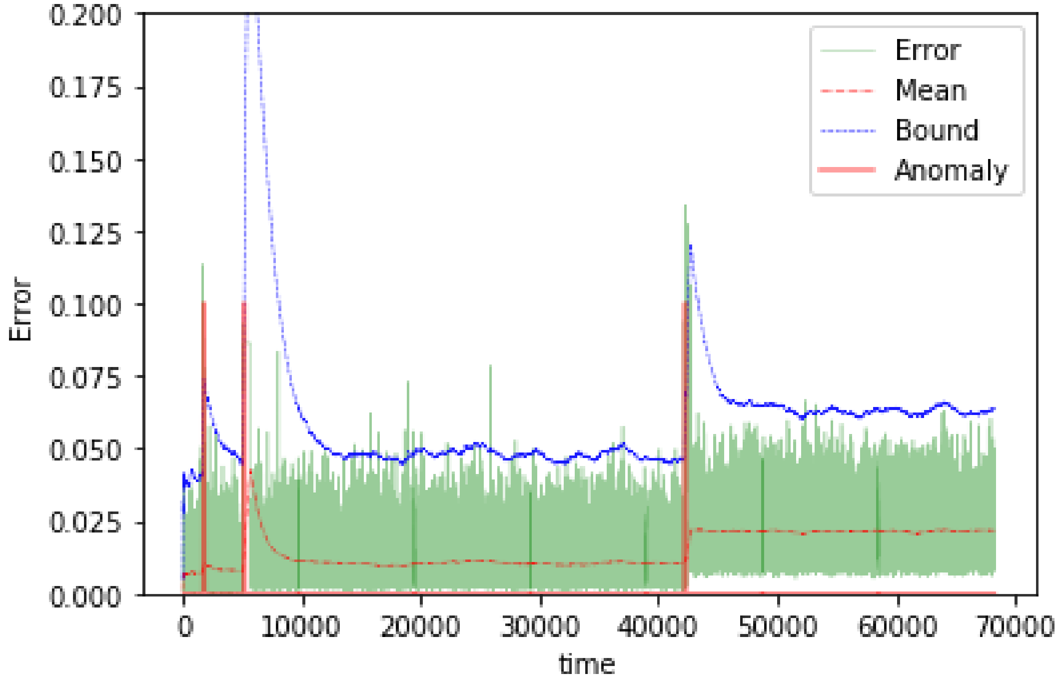
Probability Plot
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
Ordered Values
−4
−2
0
2
4
Theoretical quantiles



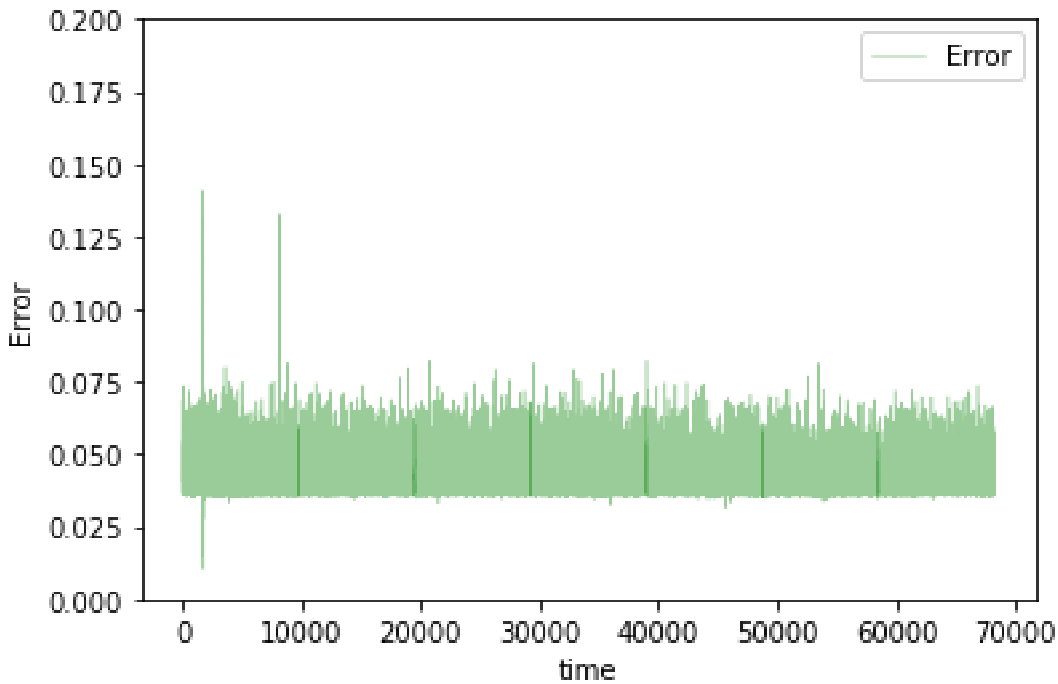
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

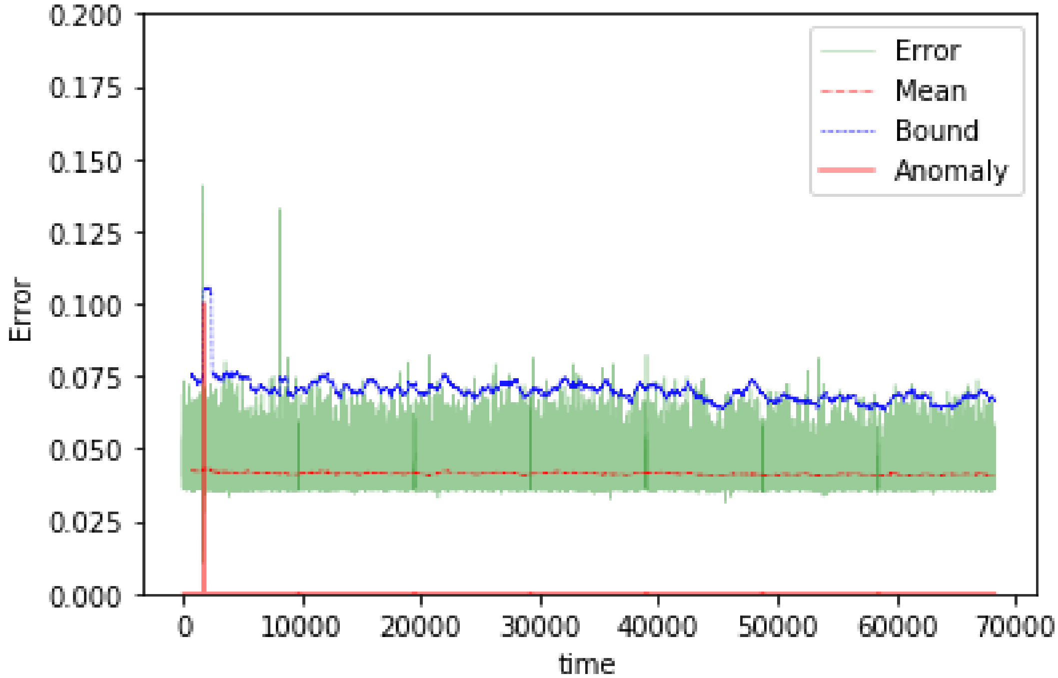

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error

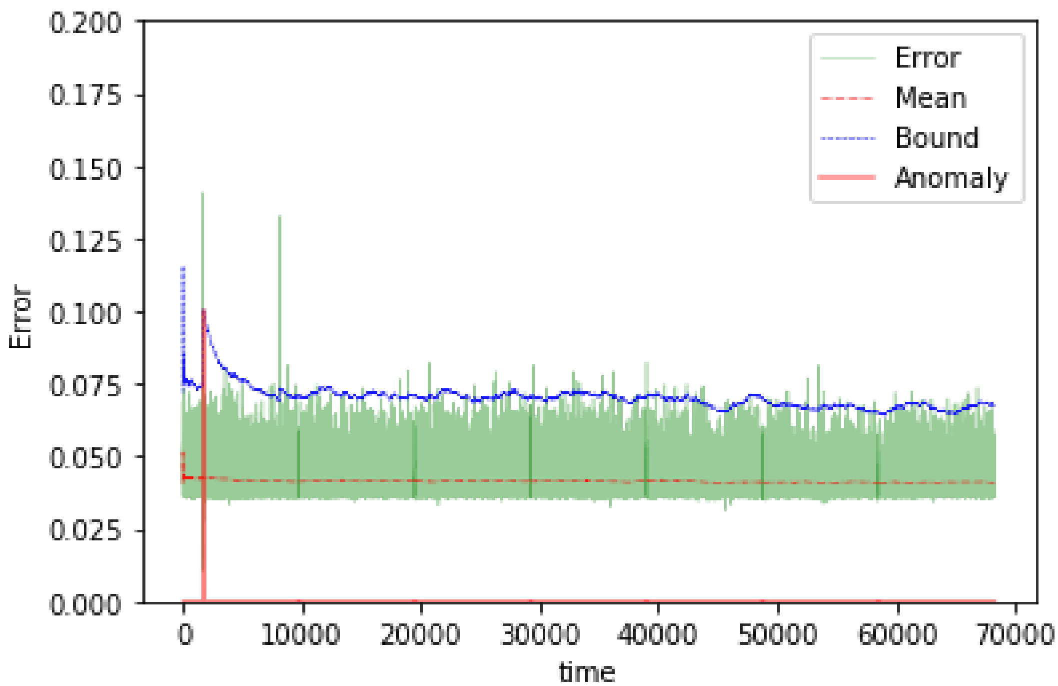

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly



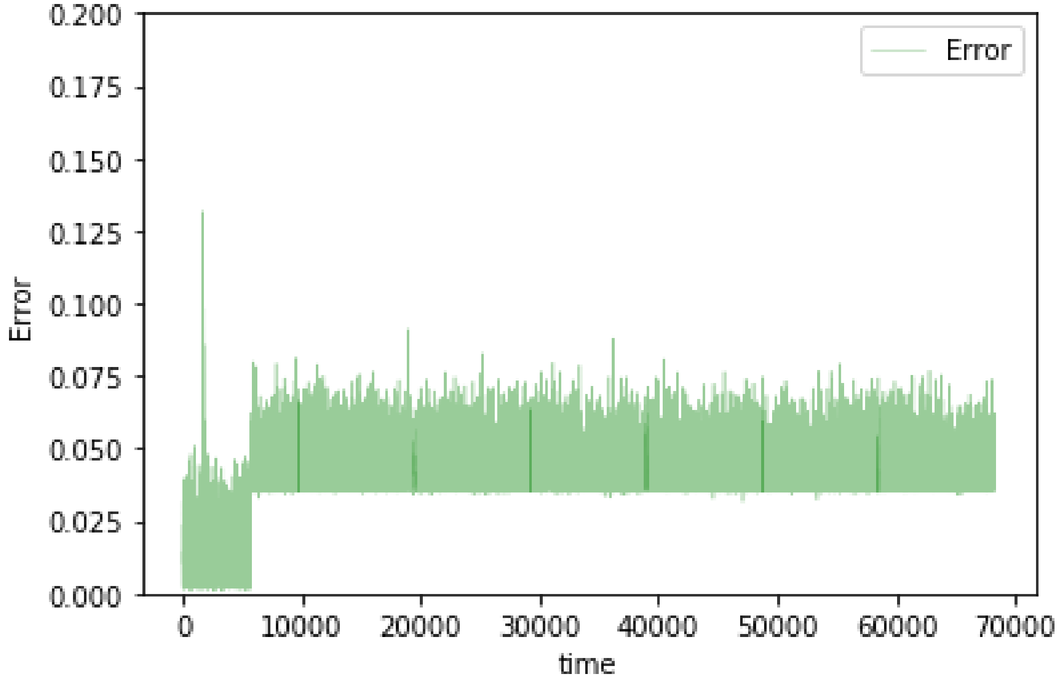

```
The mean error for nn2_50_disk_IO_start_ is 0.015279135303309738 for length 68249
Testing on Avg. load data.
```

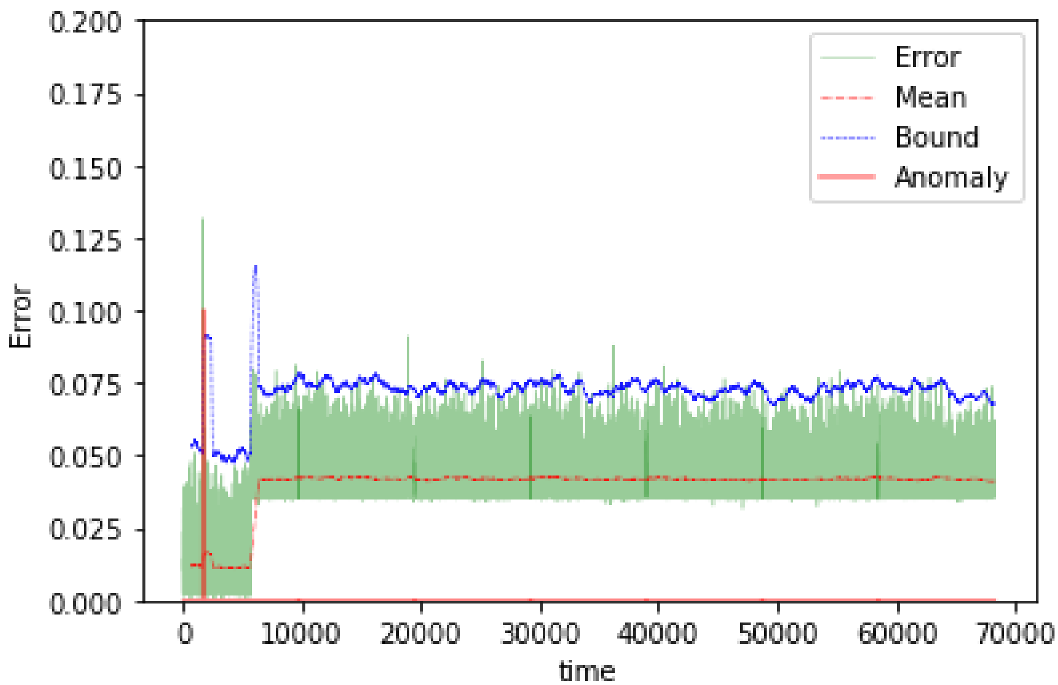


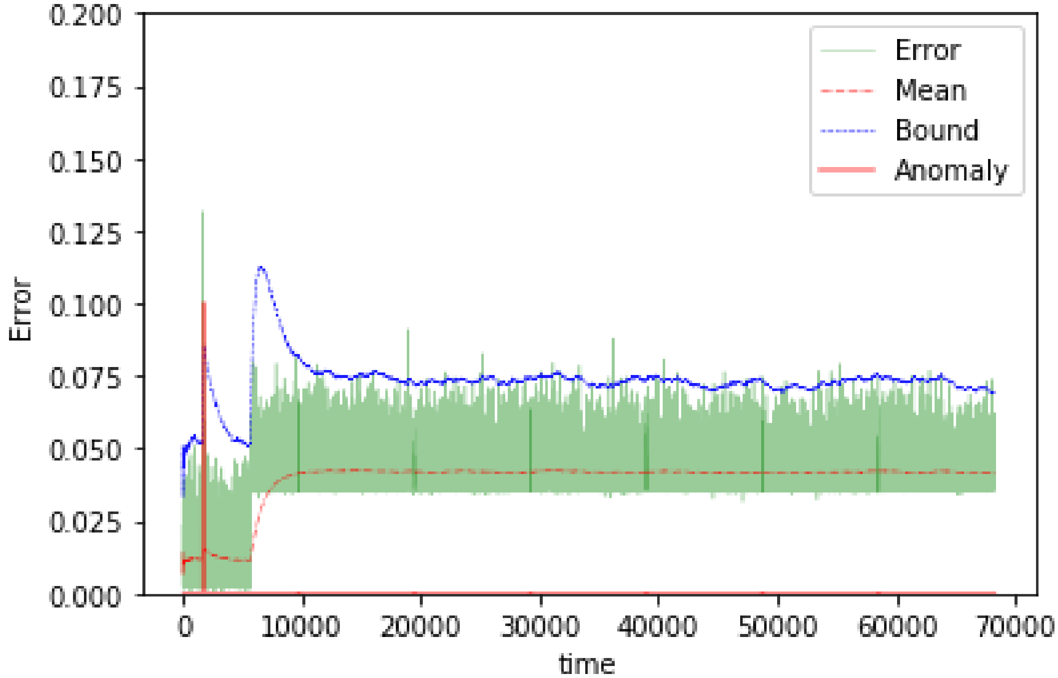

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

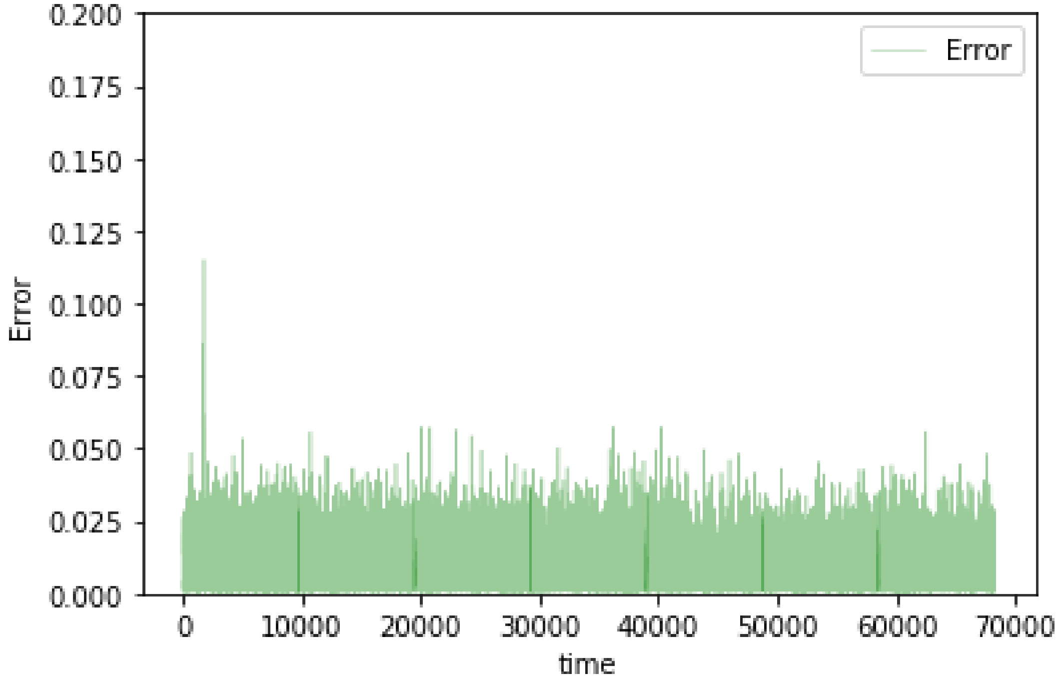
```
The mean error for nn2_50_avg_load_ is 0.041404957178482305 for length 68249
Testing on app change early data.
```

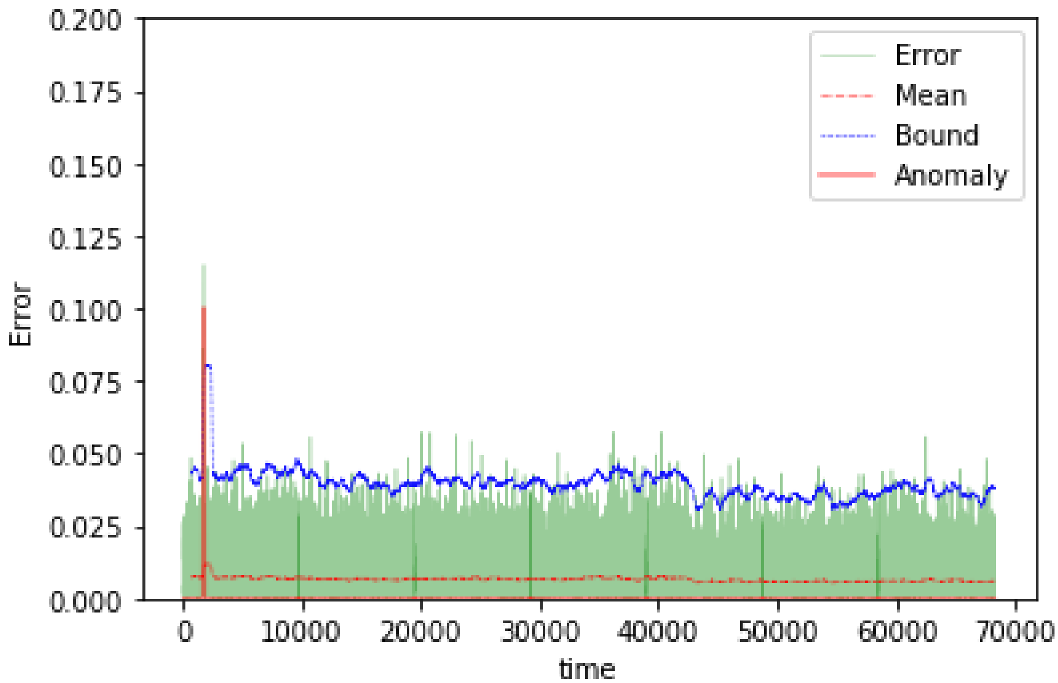


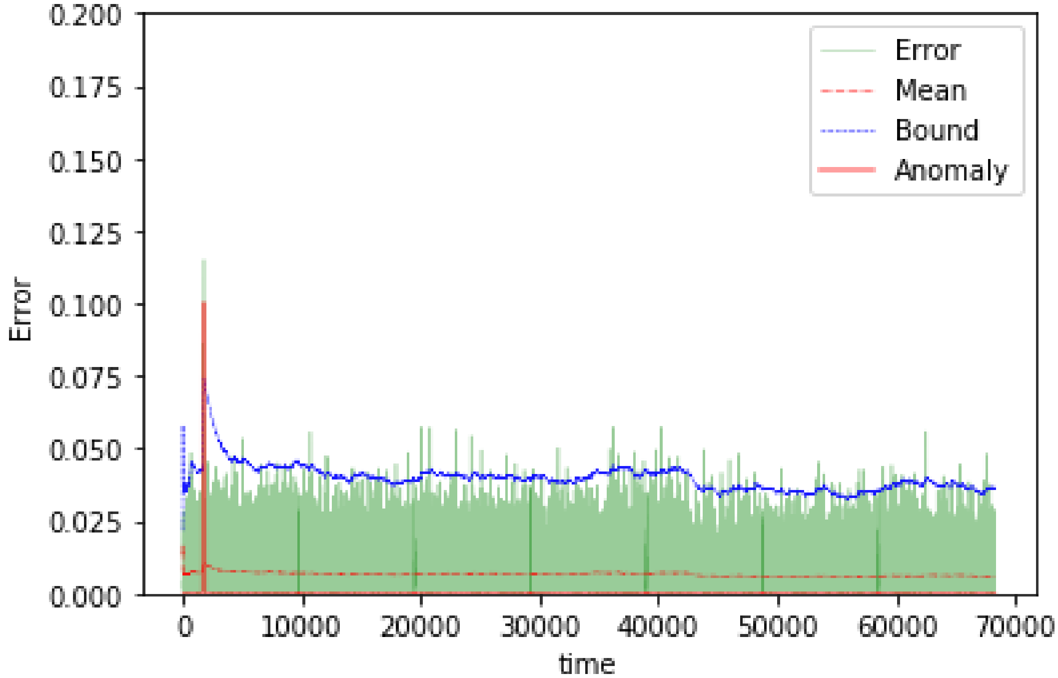





```
The mean error for nn2_50_app_change_early_ is 0.0394648724008489 for length 68249
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

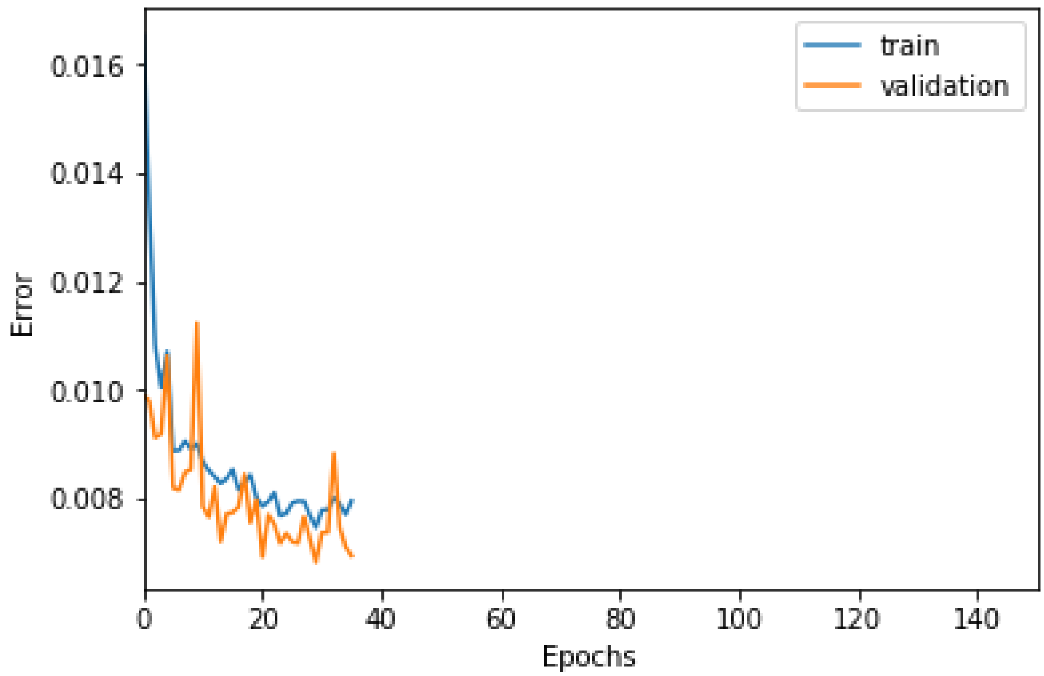

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



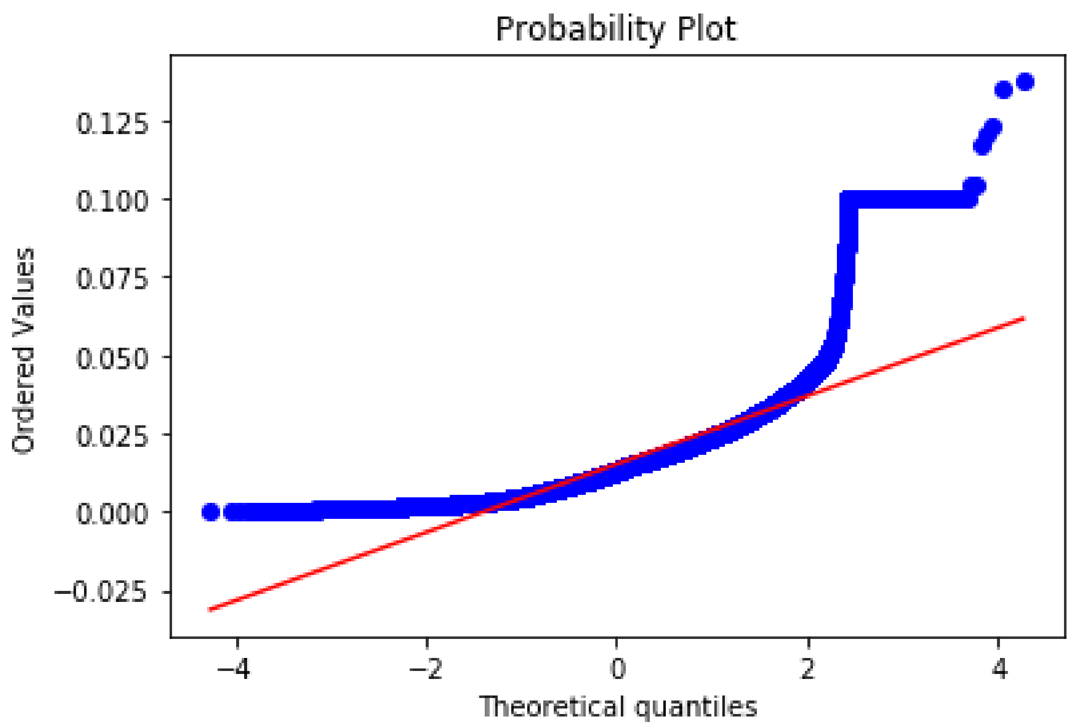

```
The mean error for nn2_50_normal_ is 0.006607745739538172 for length 68249
====================
```

100 steps

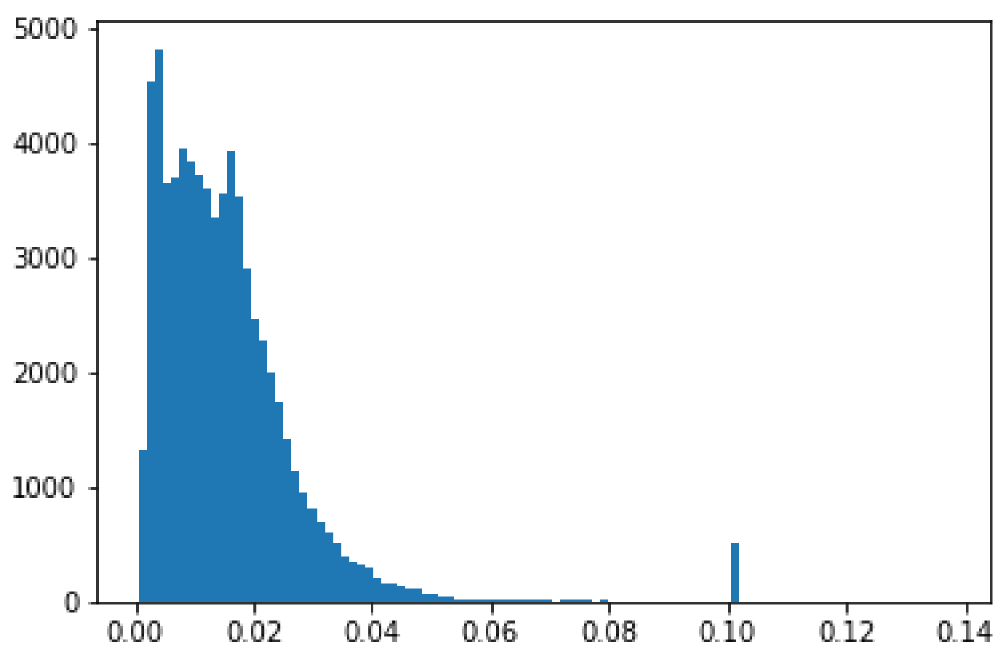
```
In [115]: TIMESTEPS = 100
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS)
          vgen = flat_generator(val_X, TIMESTEPS)
          name = "nn2_100"

In [116]: input_layer = Input(shape=(TIMESTEPS*DIM,))
          hidden = Dense(500, activation='relu')(input_layer)
          hidden = Dense(100, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [117]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [118]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```



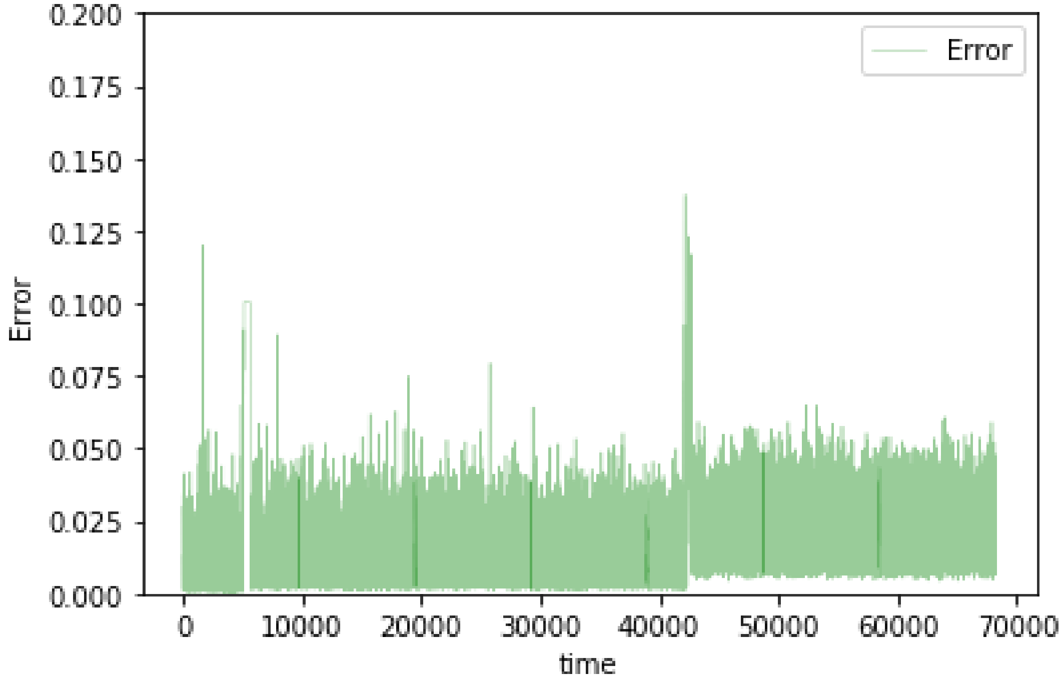

```
Training loss for final epoch is 0.007953368138056248
Validation loss for final epoch is 0.006940100140171126
---------- Beginning tests for nn2_100 ----------
Testing on Disk IO begin data.
```

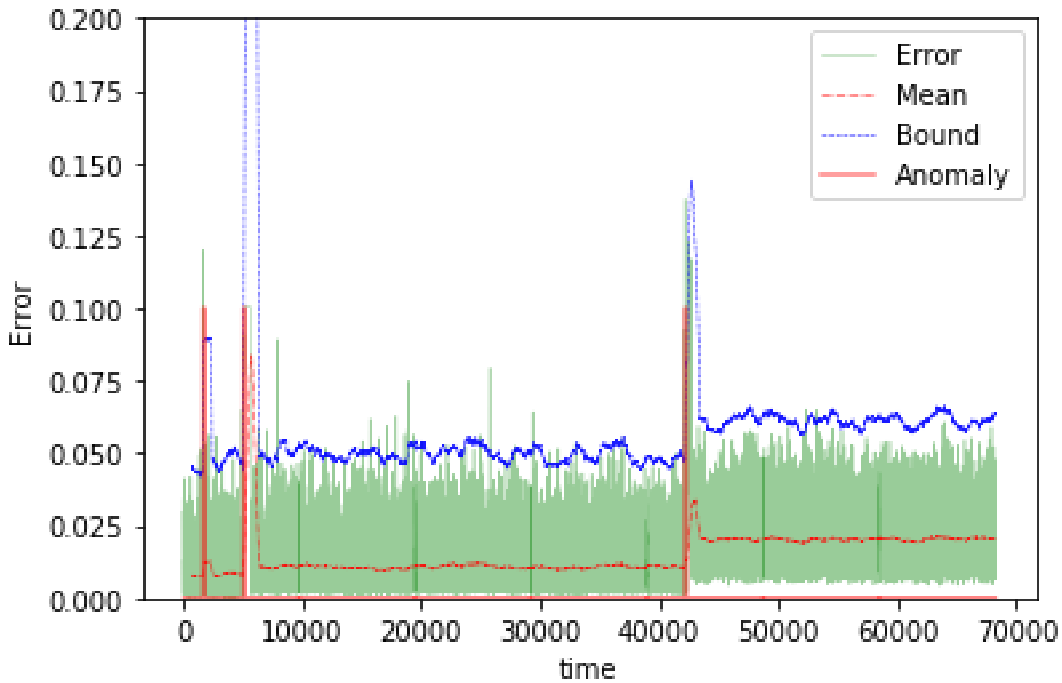


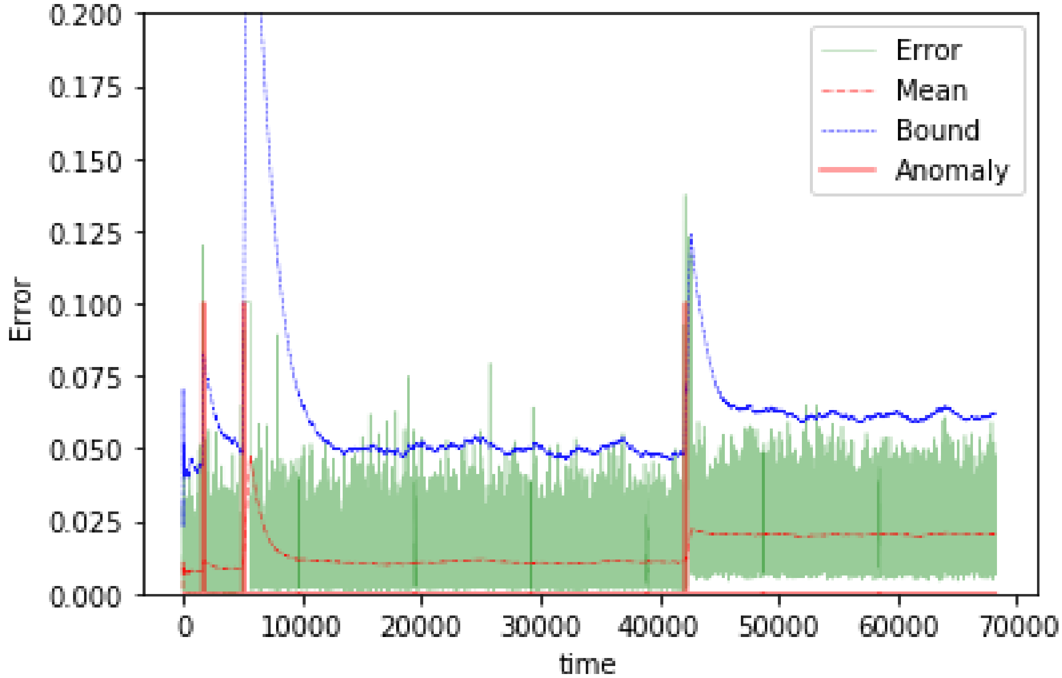
Probability Plot
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
Ordered Values
−4
−2
0
2
4
Theoretical quantiles



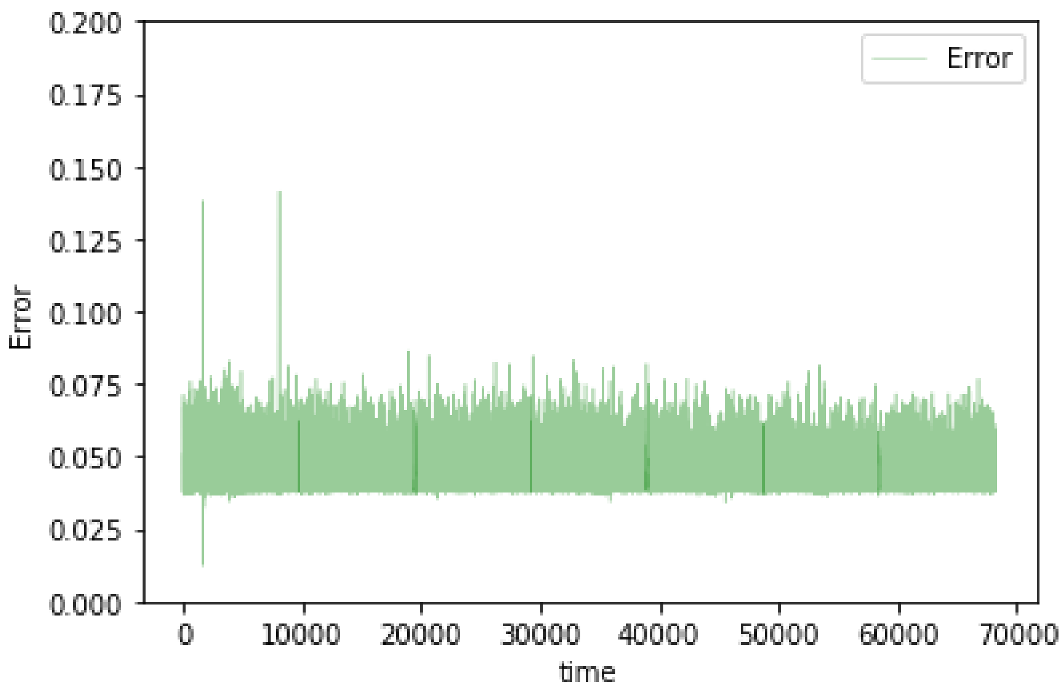
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

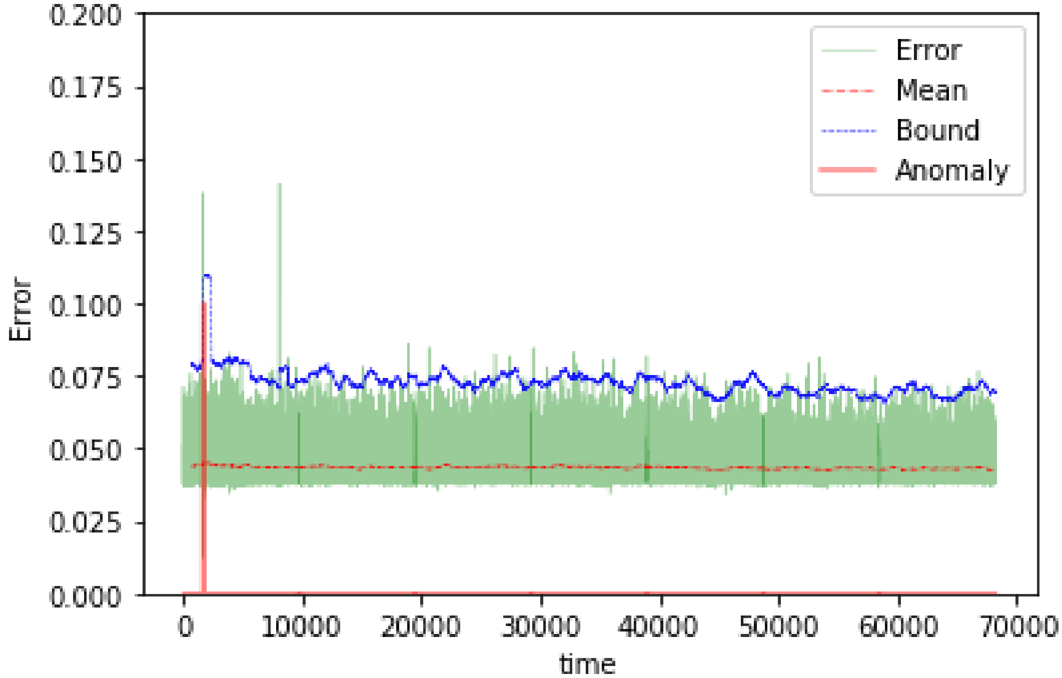

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

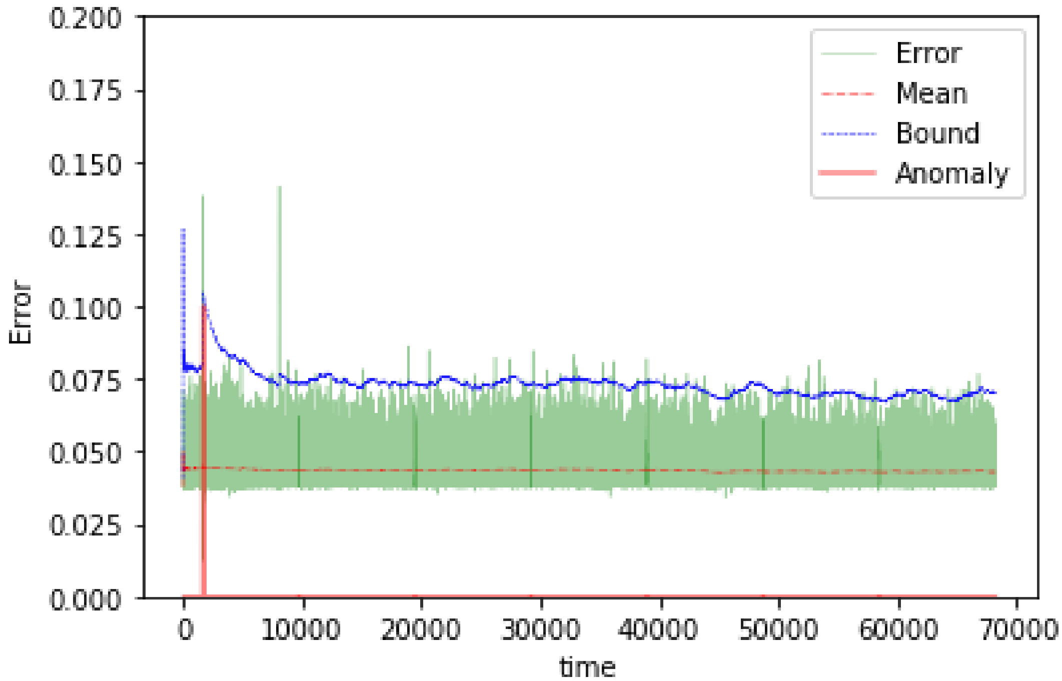

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



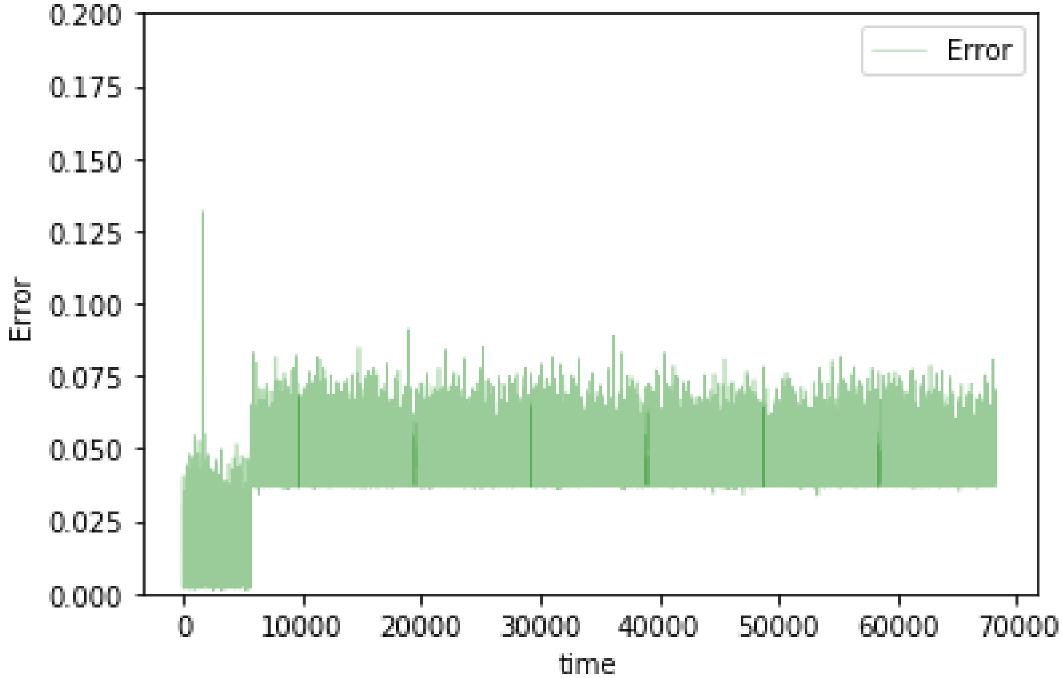

```
The mean error for nn2_100_disk_IO_start_ is 0.015187225828721703 for length 68199
Testing on Avg. load data.
```

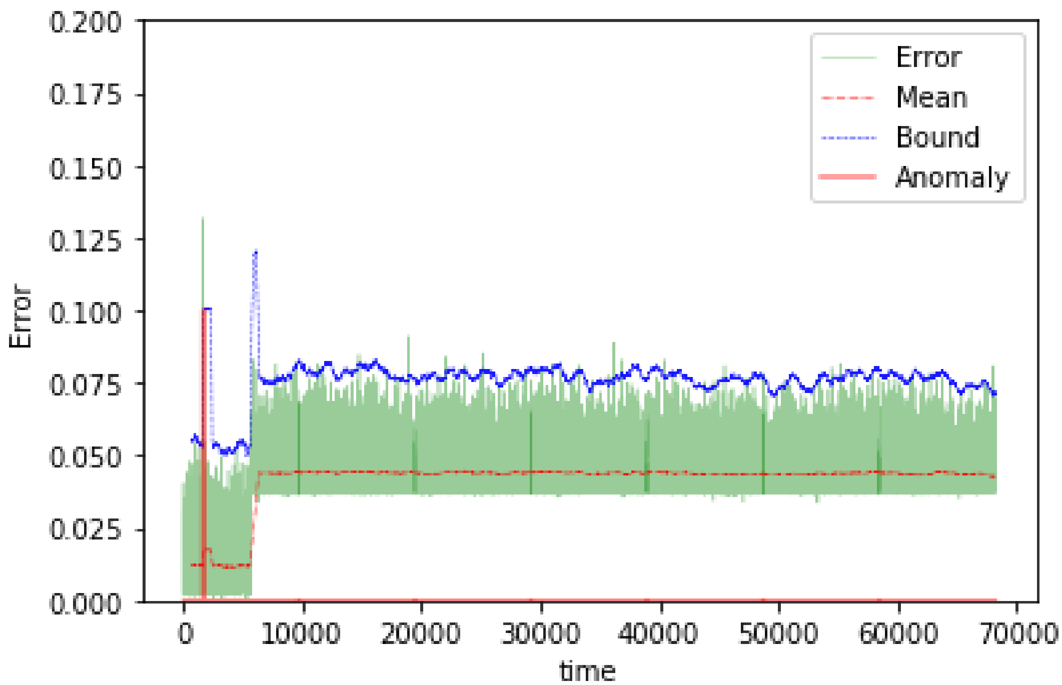


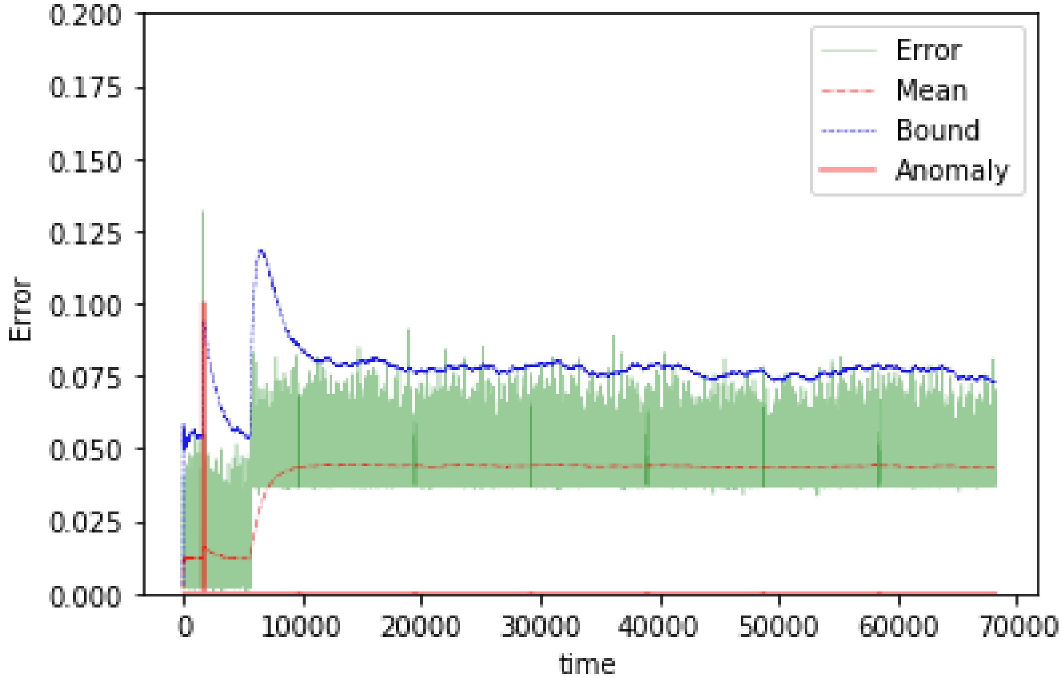

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

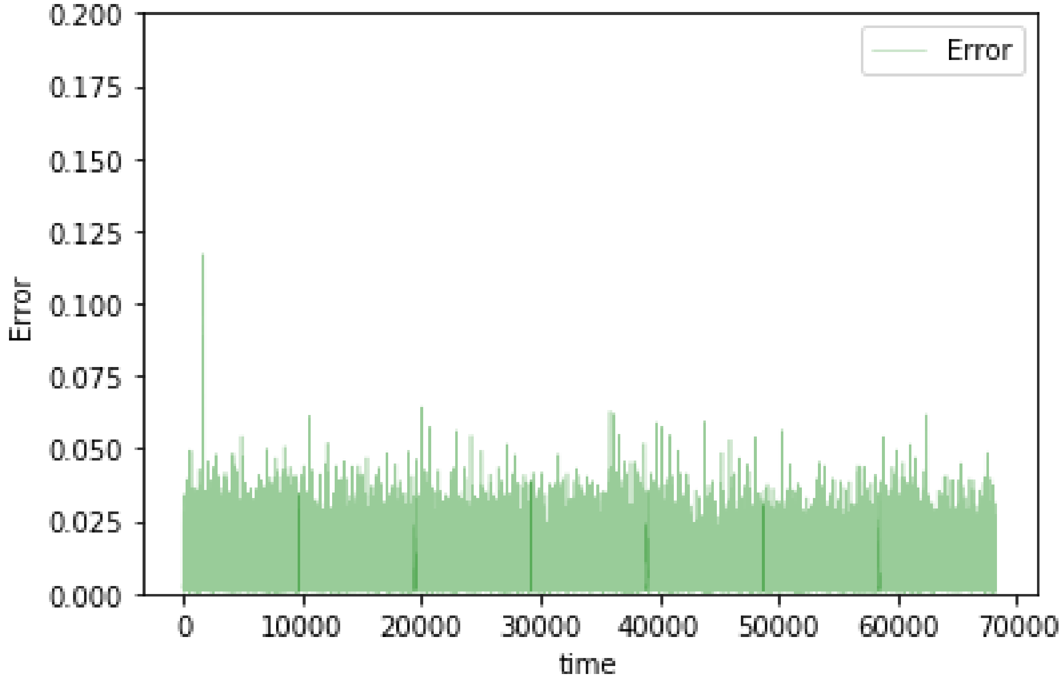
```
The mean error for nn2_100_avg_load_ is 0.0434462998213212 for length 68199
Testing on app change early data.
```

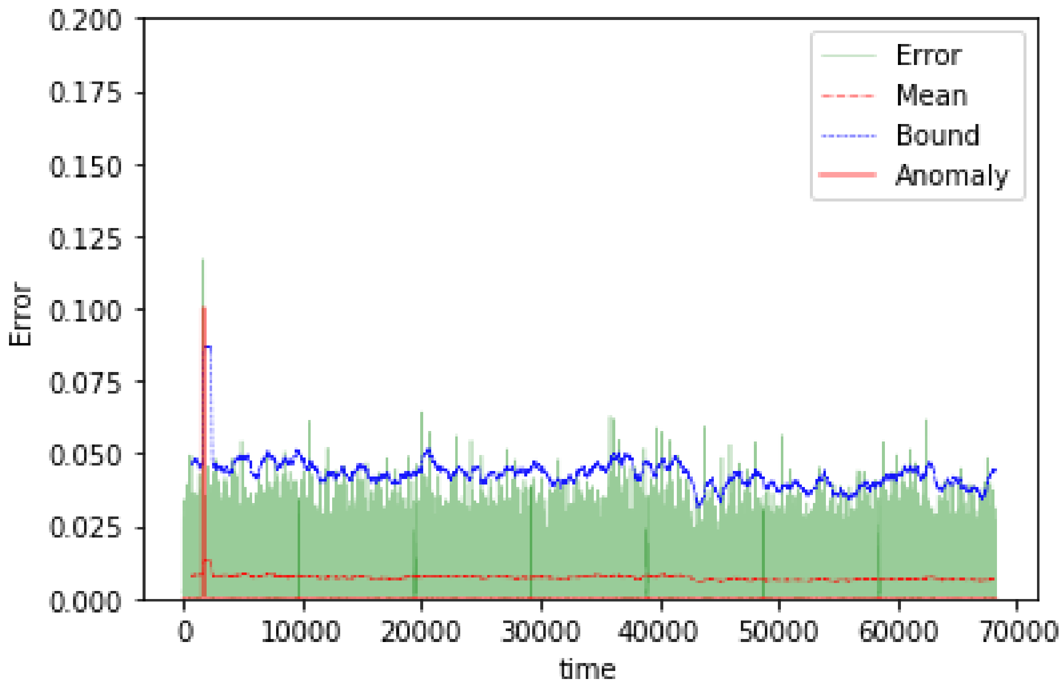


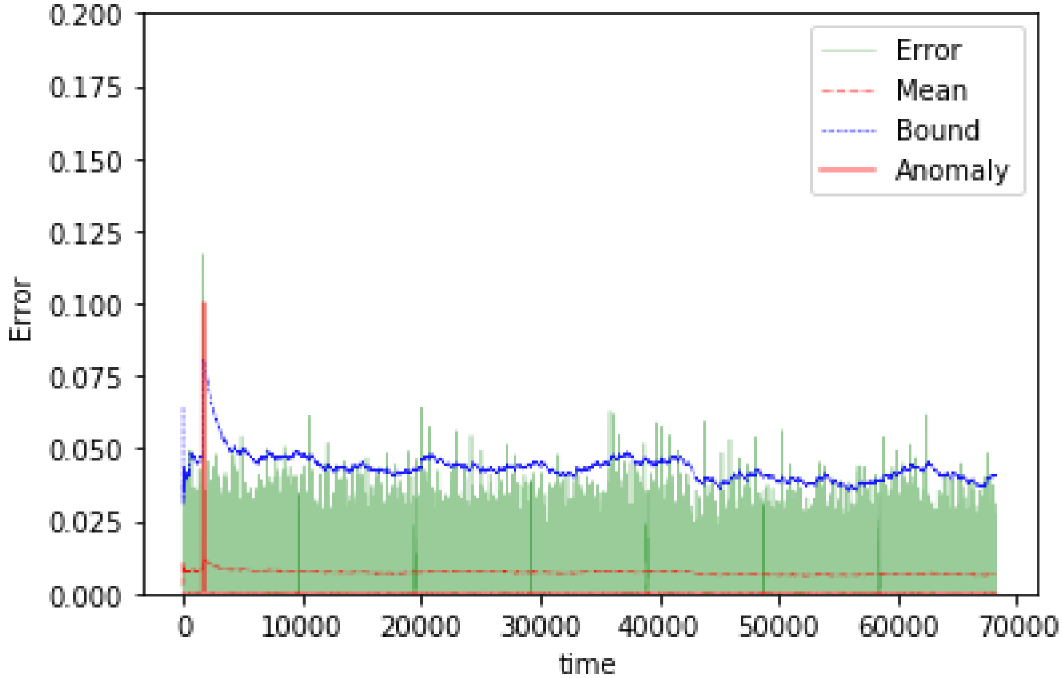




The mean error for nn2_100_app_change_early_ is 0.041334993924880936 for length 68199
Testing on Normal data.


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

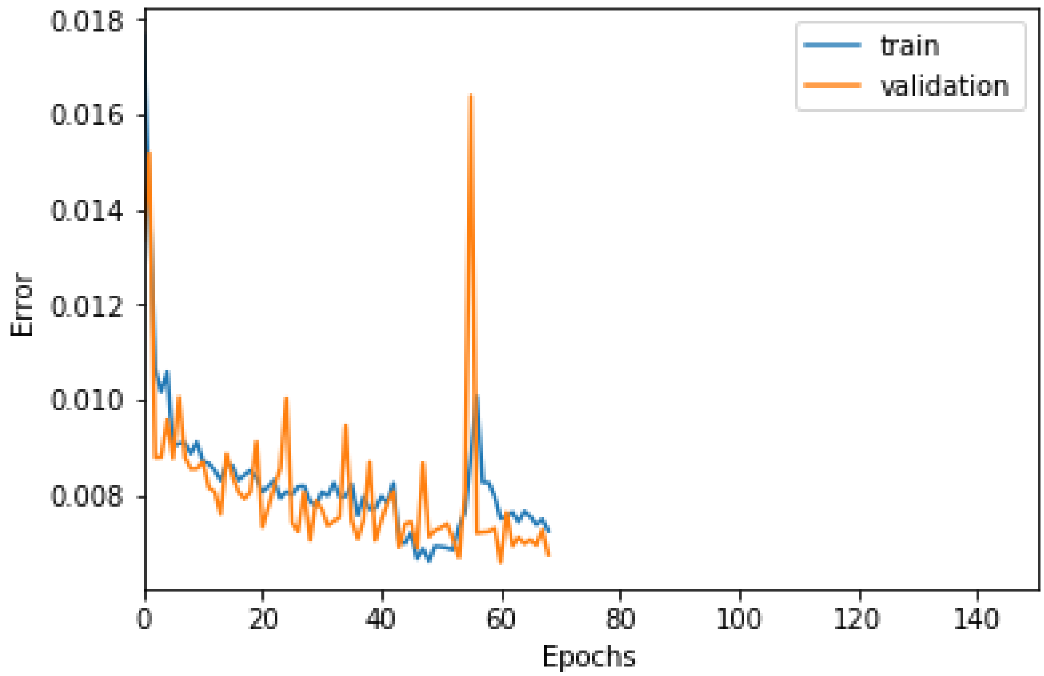

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



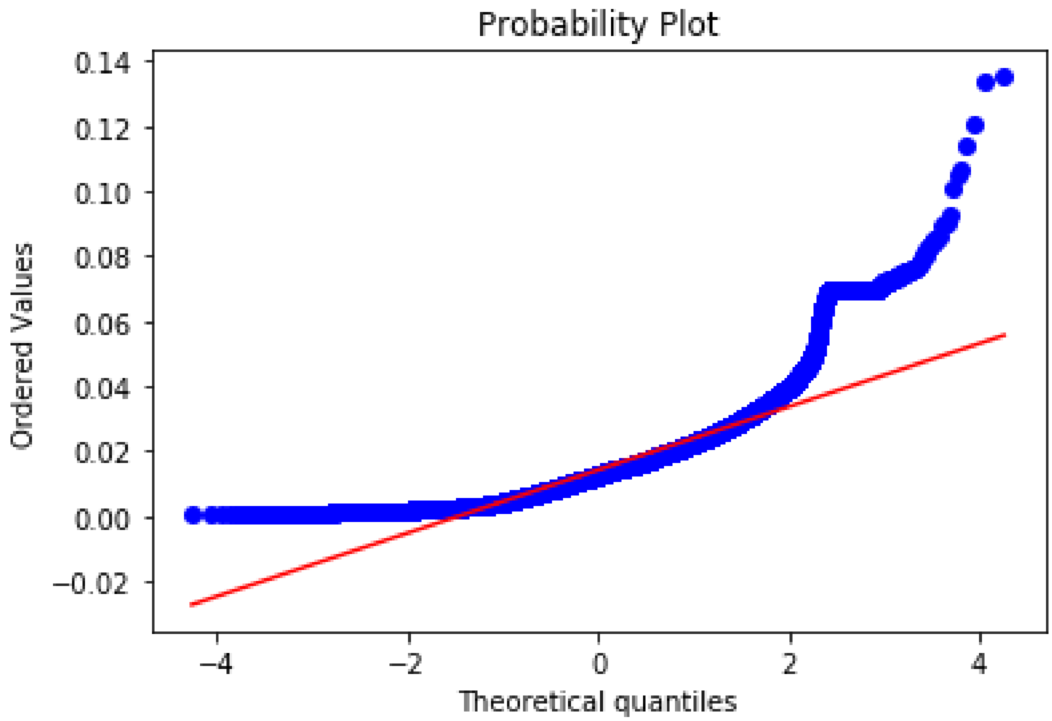

```
The mean error for nn2_100_normal_ is 0.007197523728946972 for length 68199
====================
```

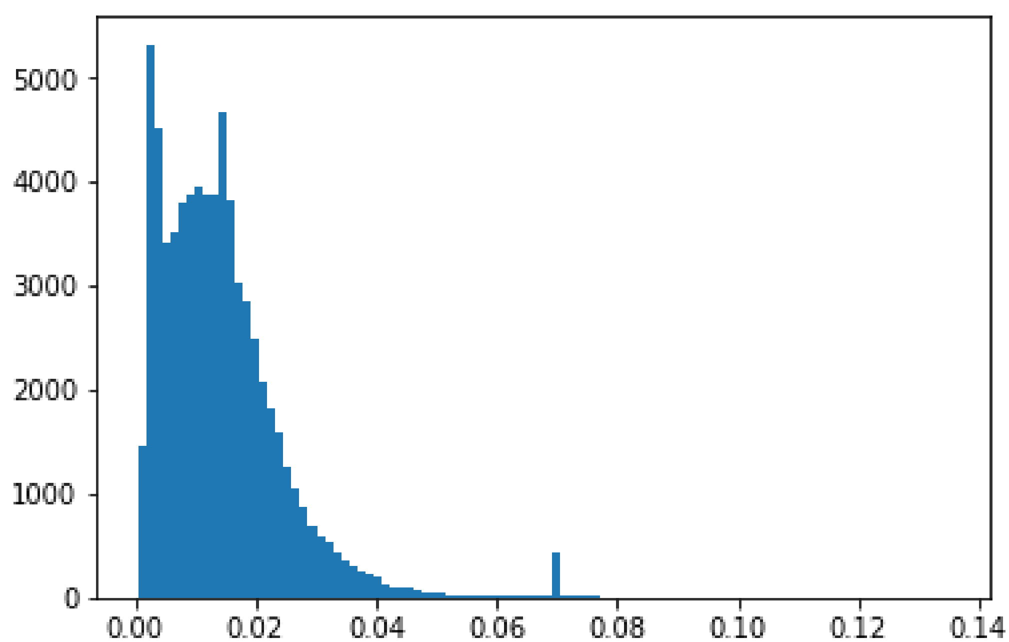
200 steps

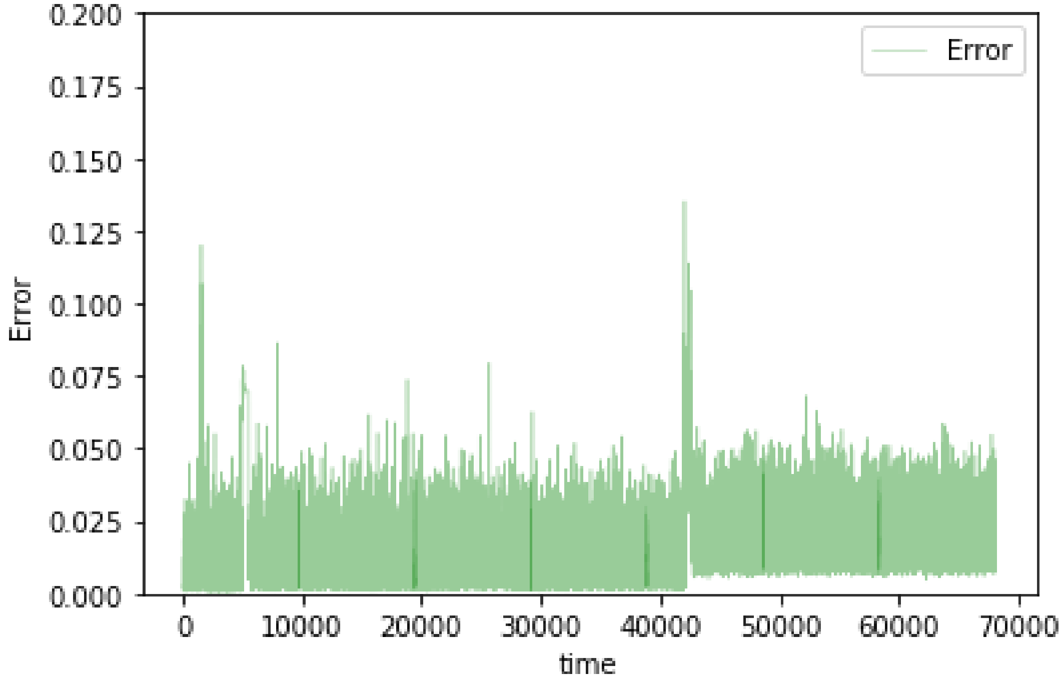
```
In [119]: TIMESTEPS = 200
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS)
          vgen = flat_generator(val_X, TIMESTEPS)
          name = "nn2_200"

In [120]: input_layer = Input(shape=(TIMESTEPS*DIM,))
          hidden = Dense(500, activation='relu')(input_layer)
          hidden = Dense(100, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [121]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [122]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```

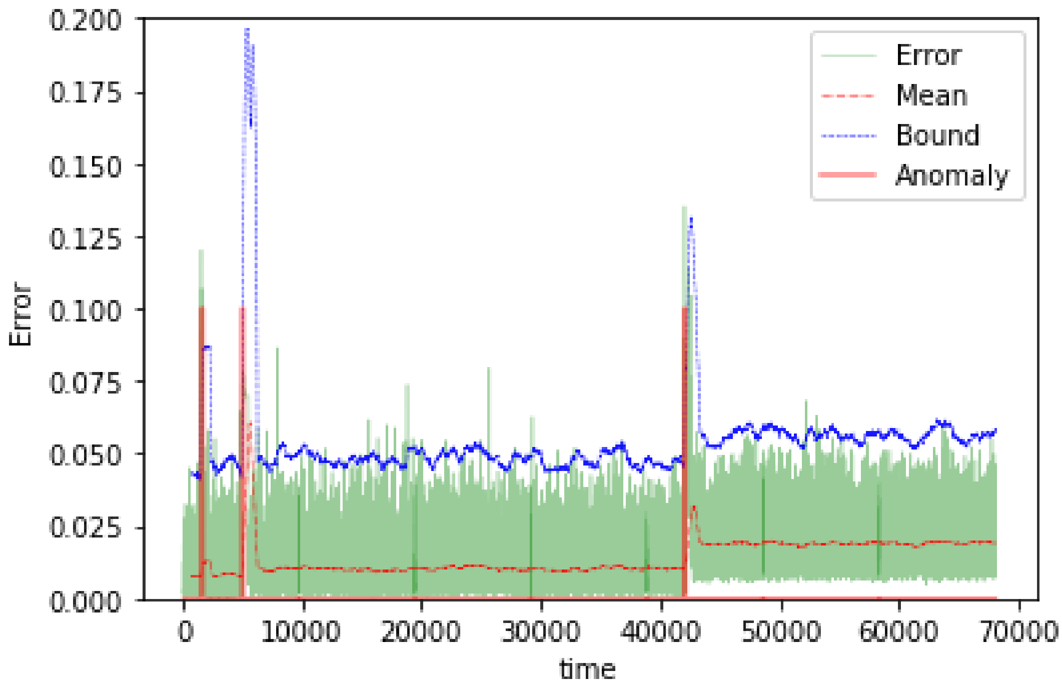


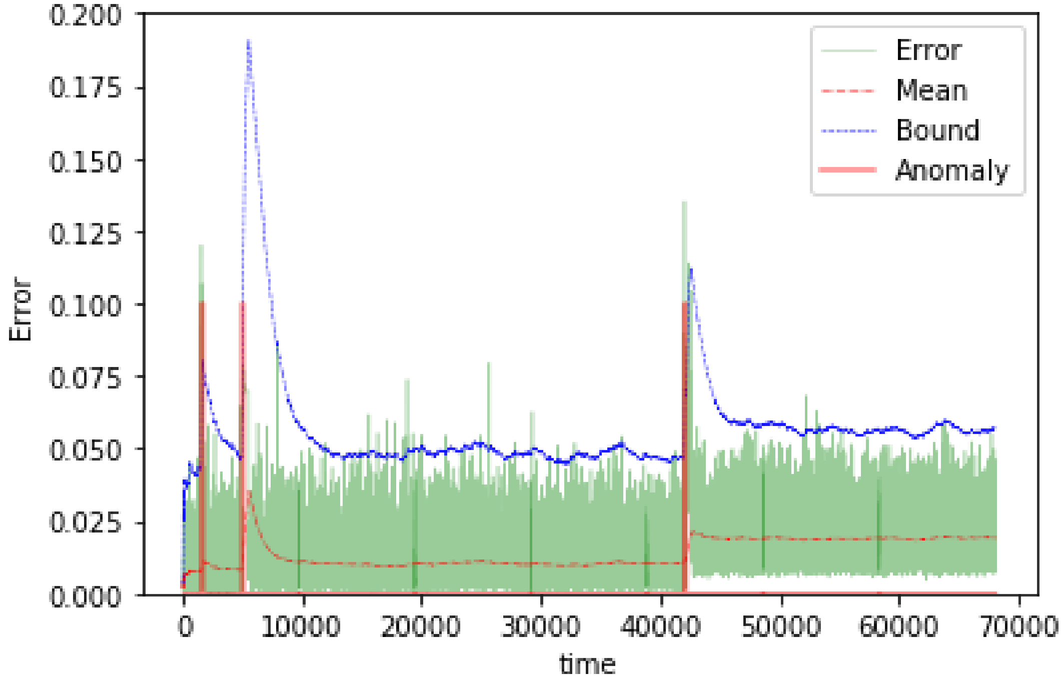

```
Training loss for final epoch is 0.007250135332928039
Validation loss for final epoch is 0.006746463899617083
---------- Beginning tests for nn2_200 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
−0.02
−4
−2
0
2
4

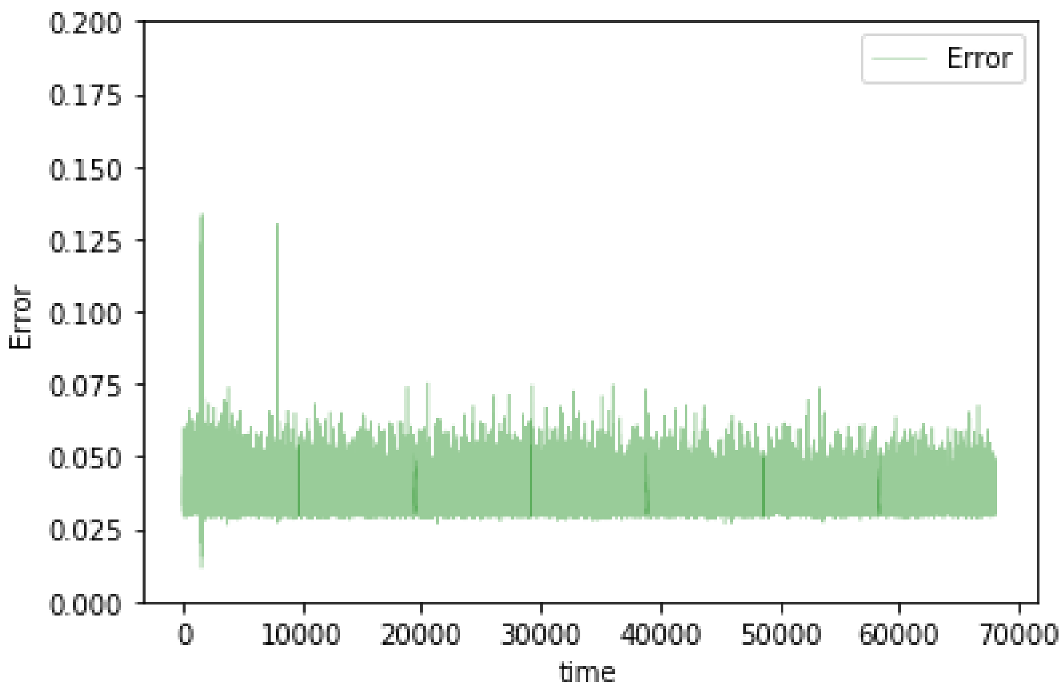

5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

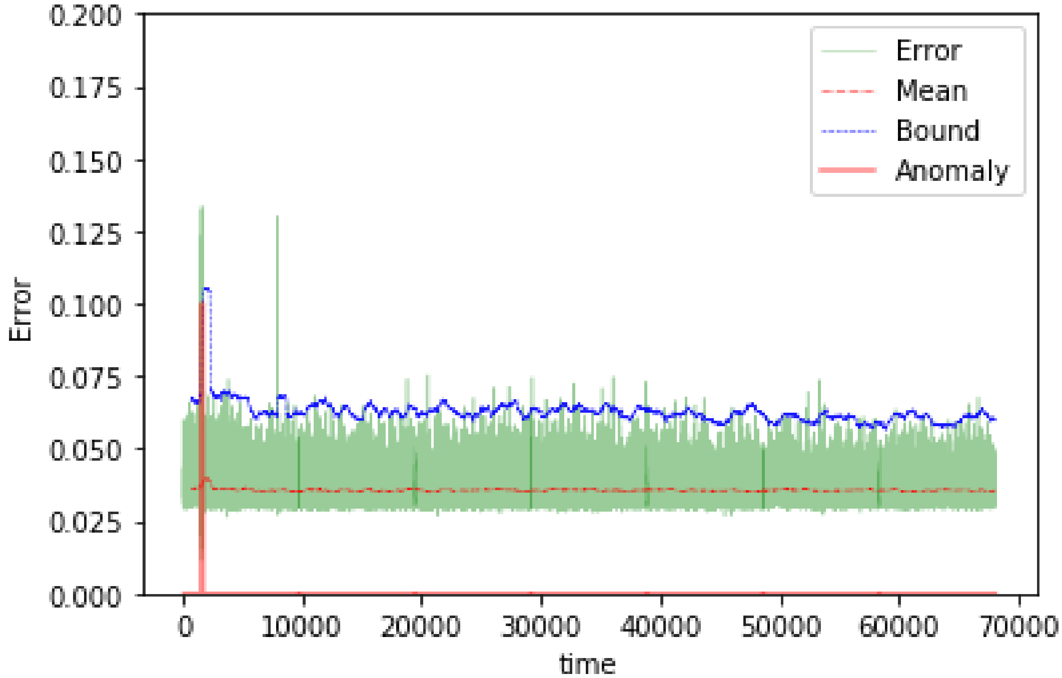

Error
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000


Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

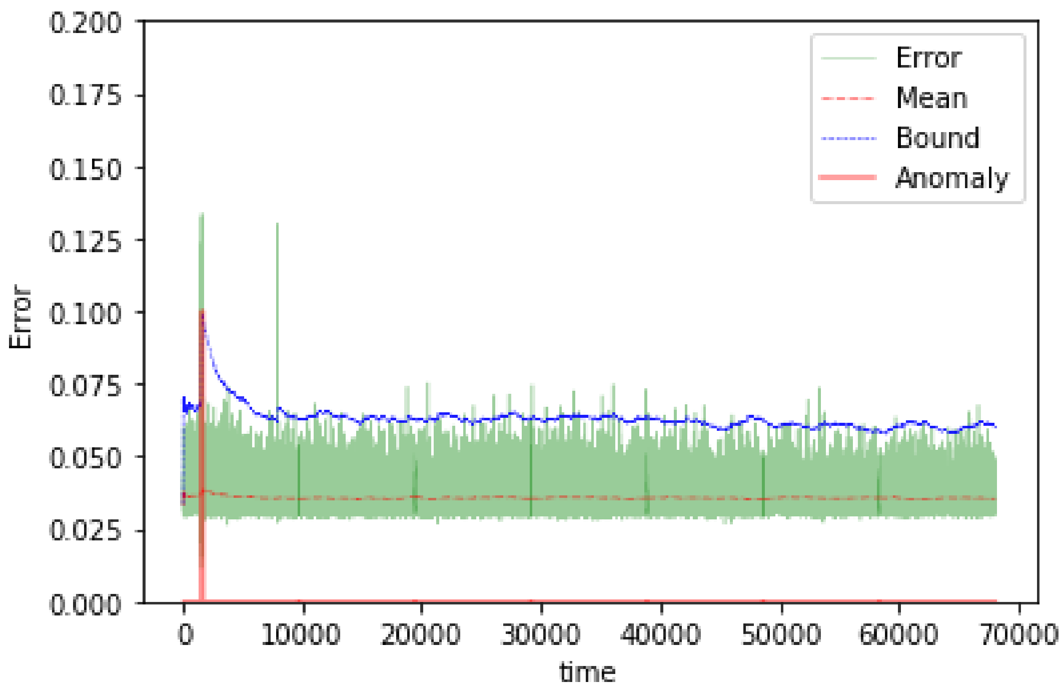



```
The mean error for nn2_200_disk_IO_start_ is 0.014146107455659388 for length 68099
Testing on Avg. load data.
```



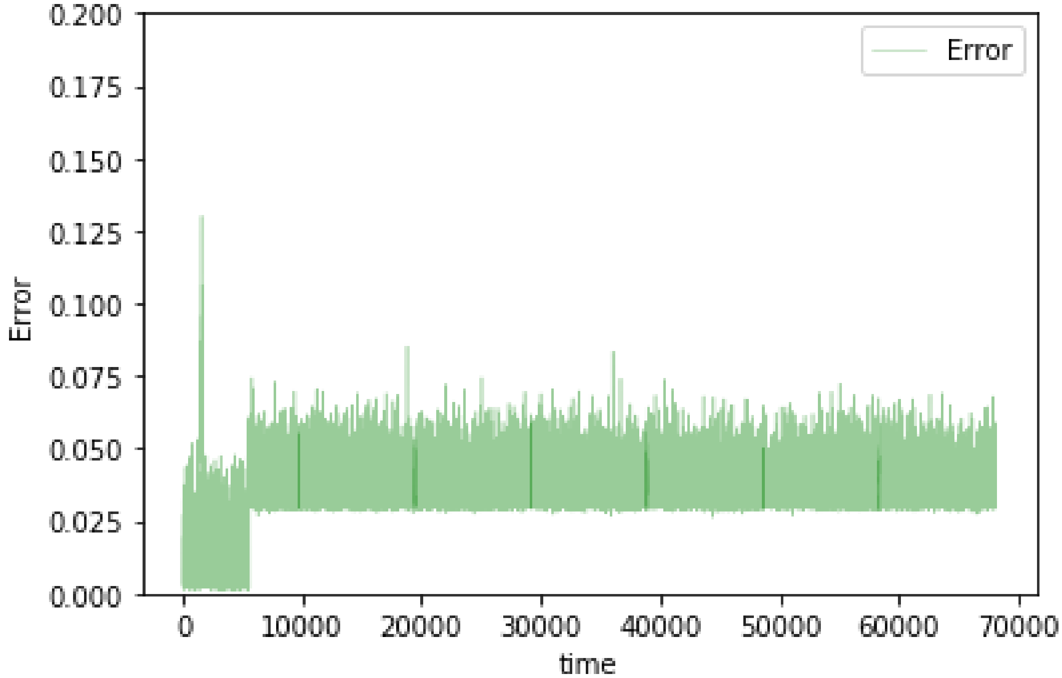


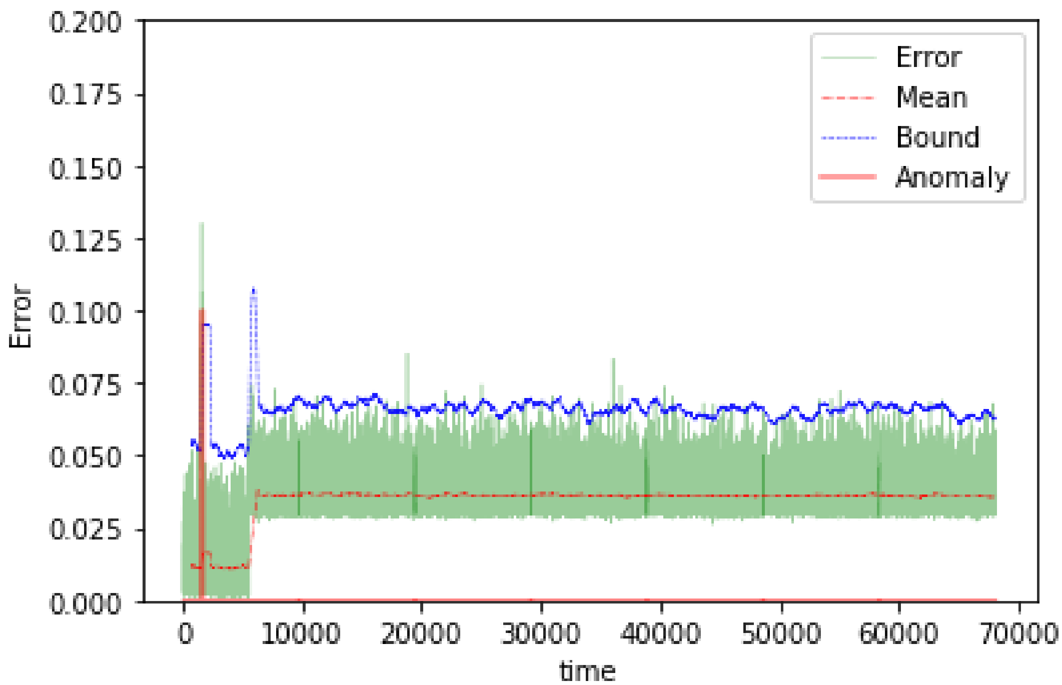
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

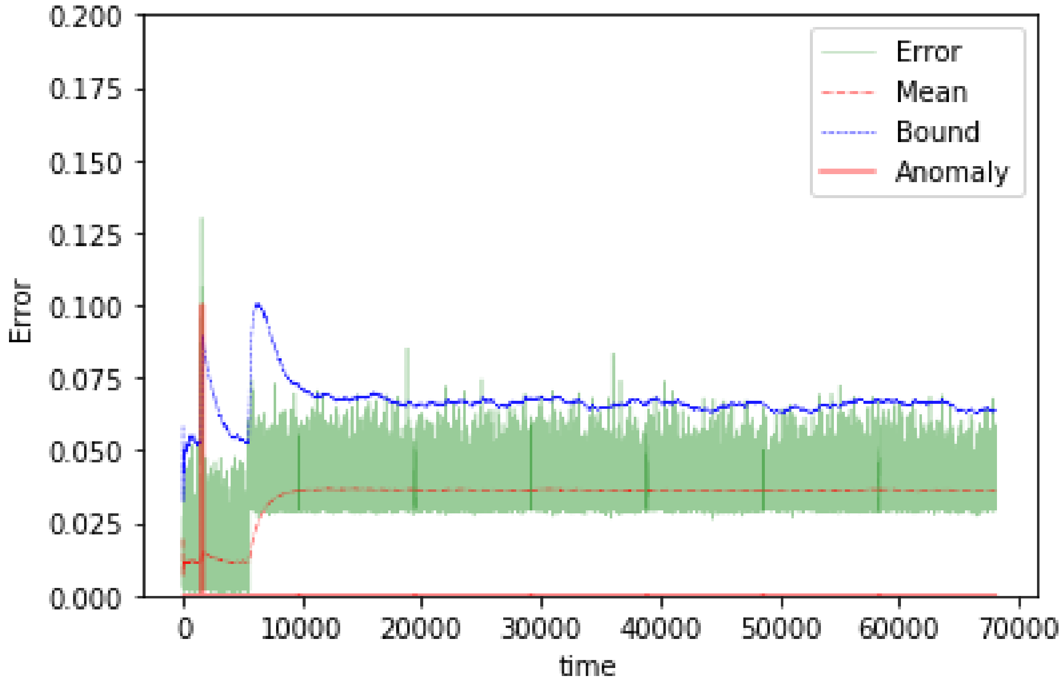


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

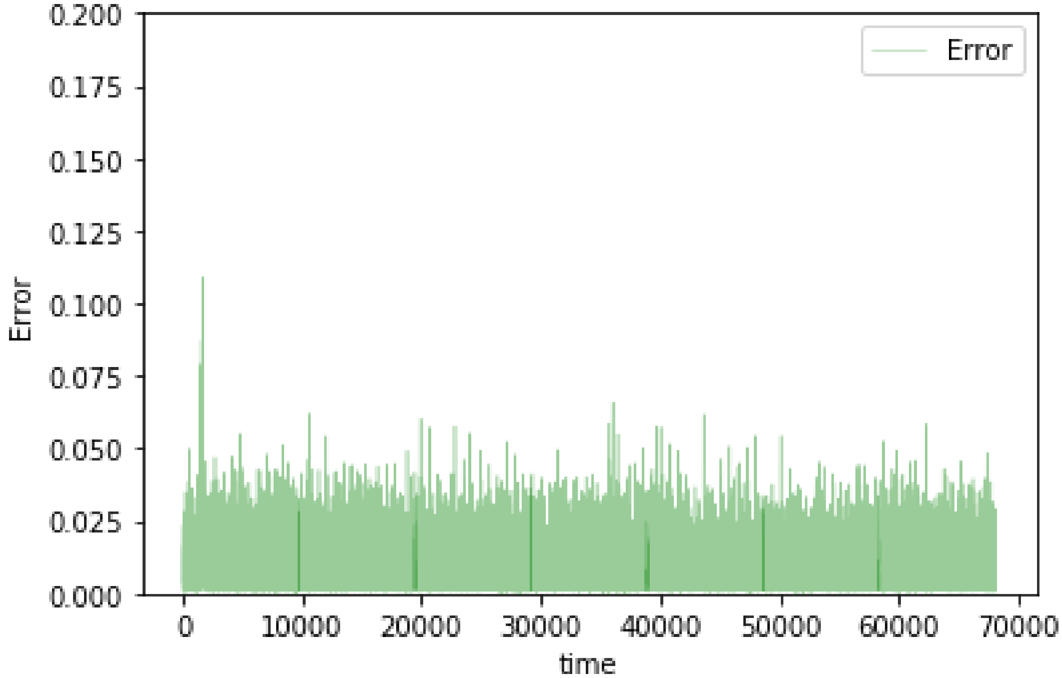
```
The mean error for nn2_200_avg_load_ is 0.03571459114132466 for length 68099
Testing on app change early data.
```



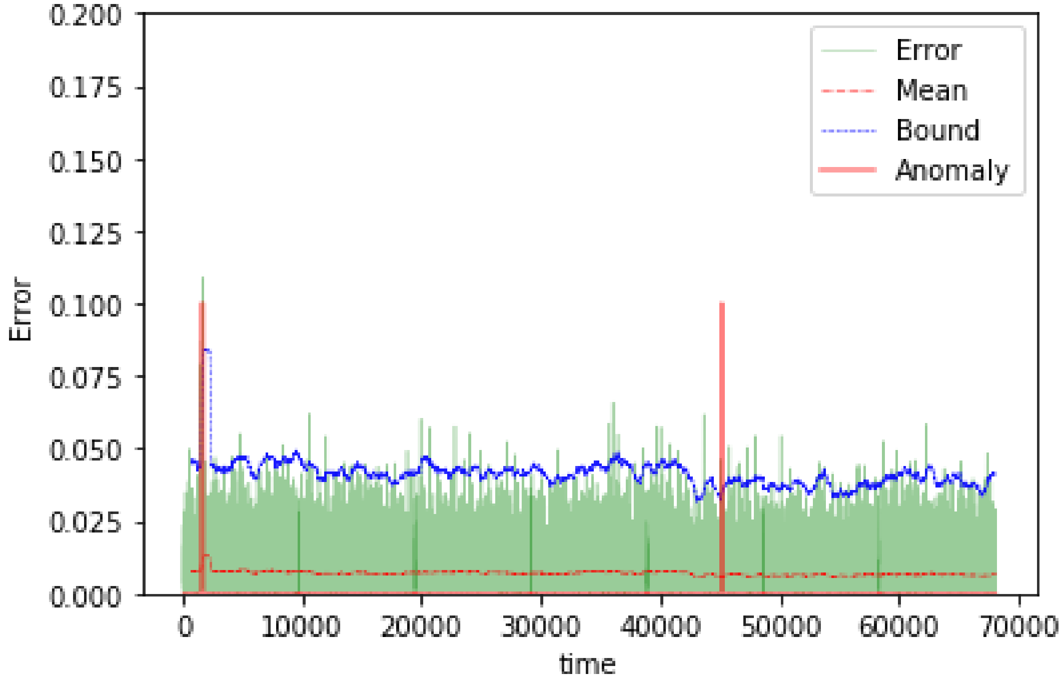




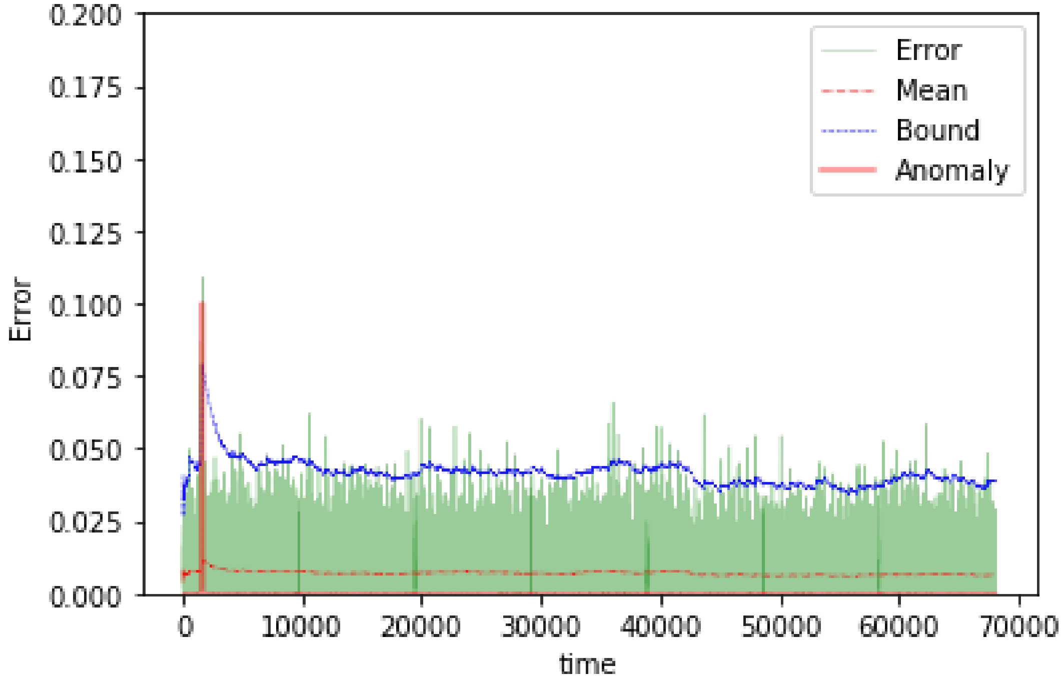

```
The mean error for nn2_200_app_change_early_ is 0.03434637744902744 for length 68099
Testing on Normal data.
```



Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



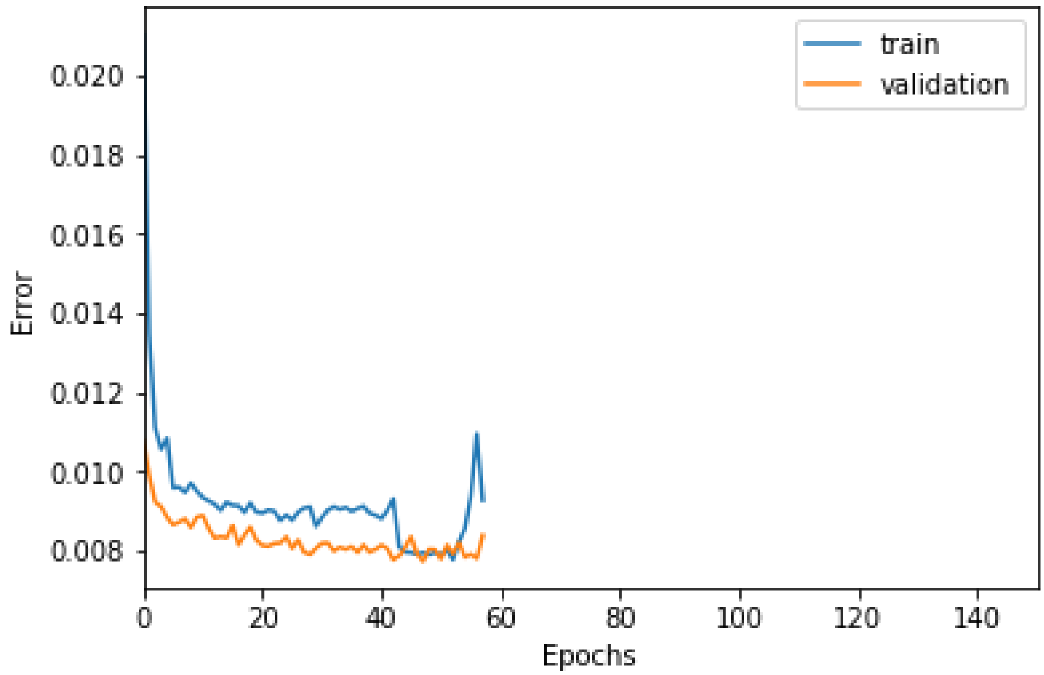

```
The mean error for nn2_200_normal_ is 0.006974438326160408 for length 68099
====================
```

1.11.4 NN with 3 hidden layers

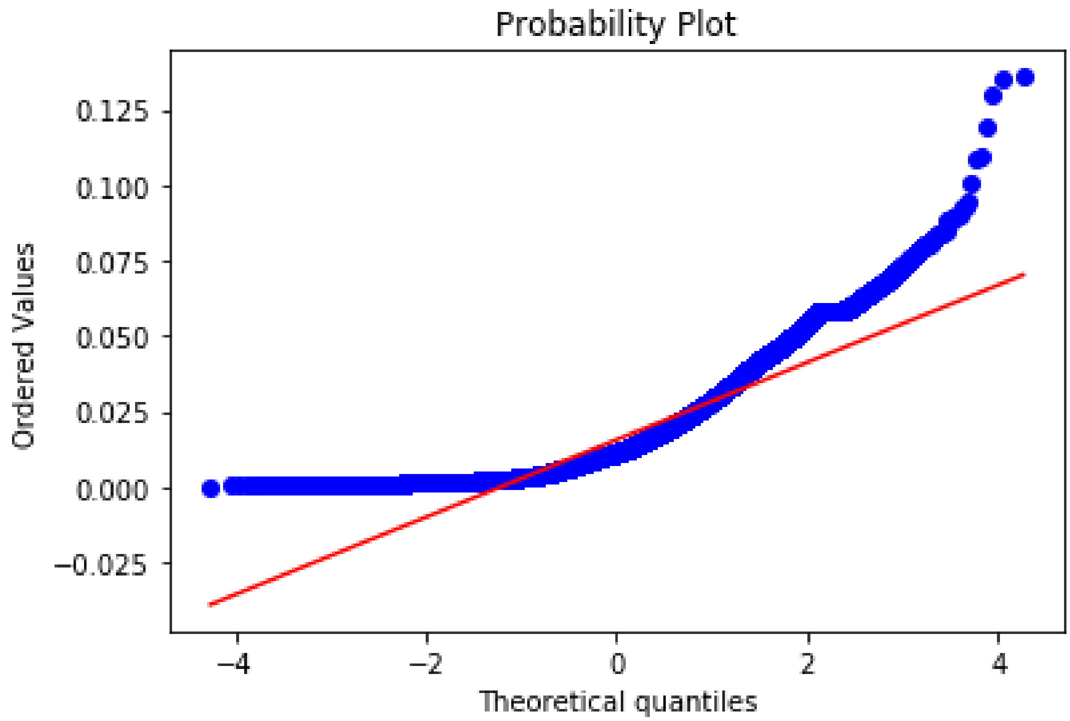
2 steps

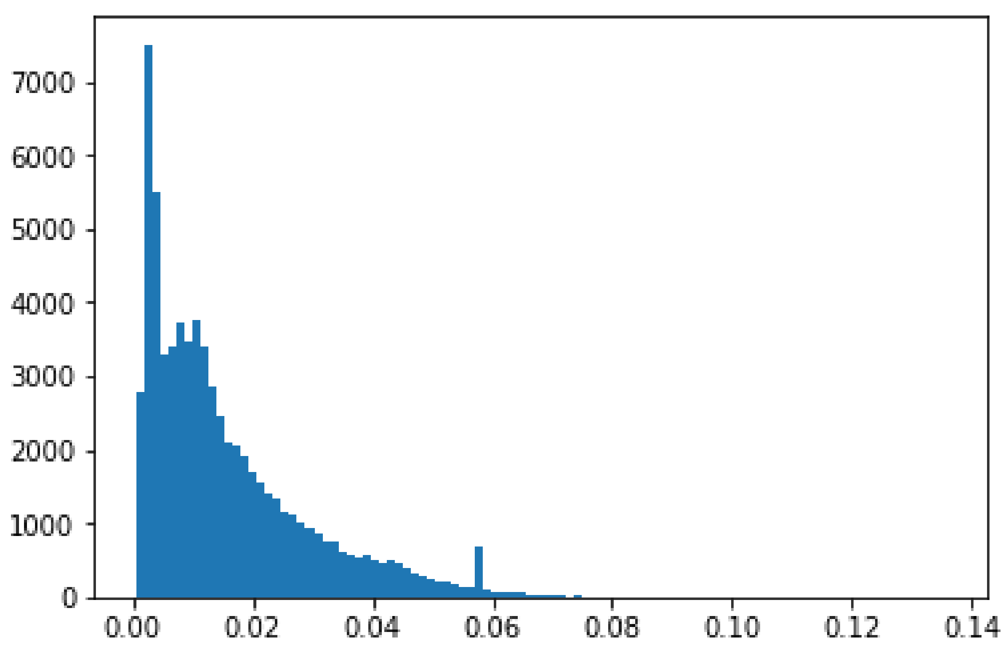
```
In [123]: TIMESTEPS = 2
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS)
          vgen = flat_generator(val_X, TIMESTEPS)
          name = "nn3_2"

In [124]: input_layer = Input(shape=(TIMESTEPS*DIM,))
          hidden = Dense(1000, activation='relu')(input_layer)
          hidden = Dense(500, activation='relu')(hidden)
          hidden = Dense(100, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [125]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])
```

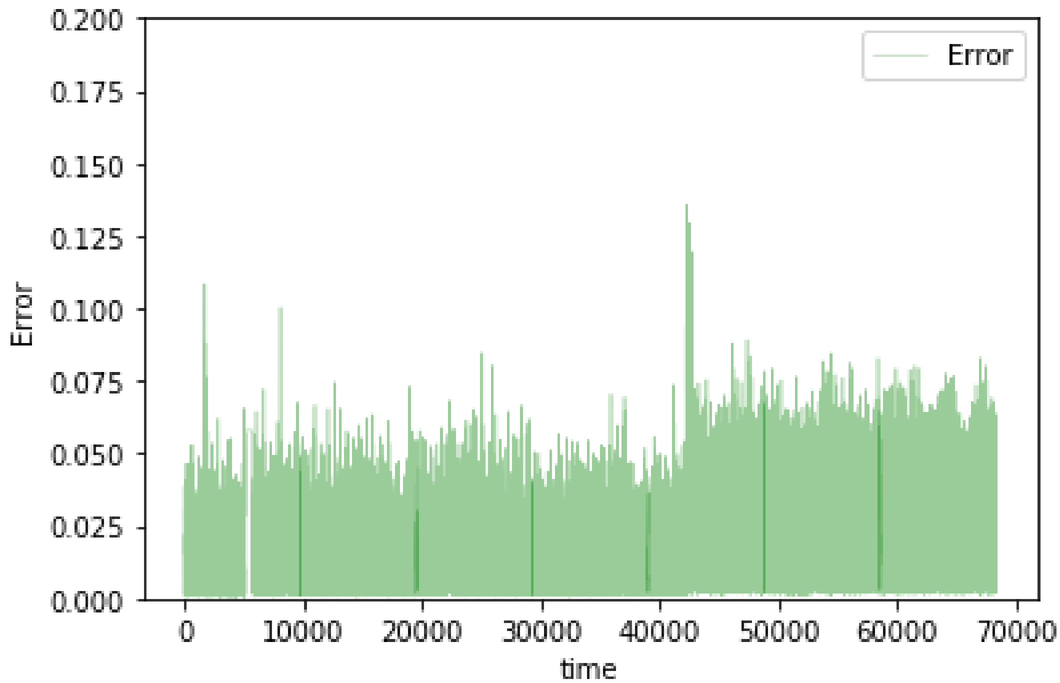
In [126]:
```
train(model, tgen, vgen, name=name)
test(model, name=name, window=TIMESTEPS)
```

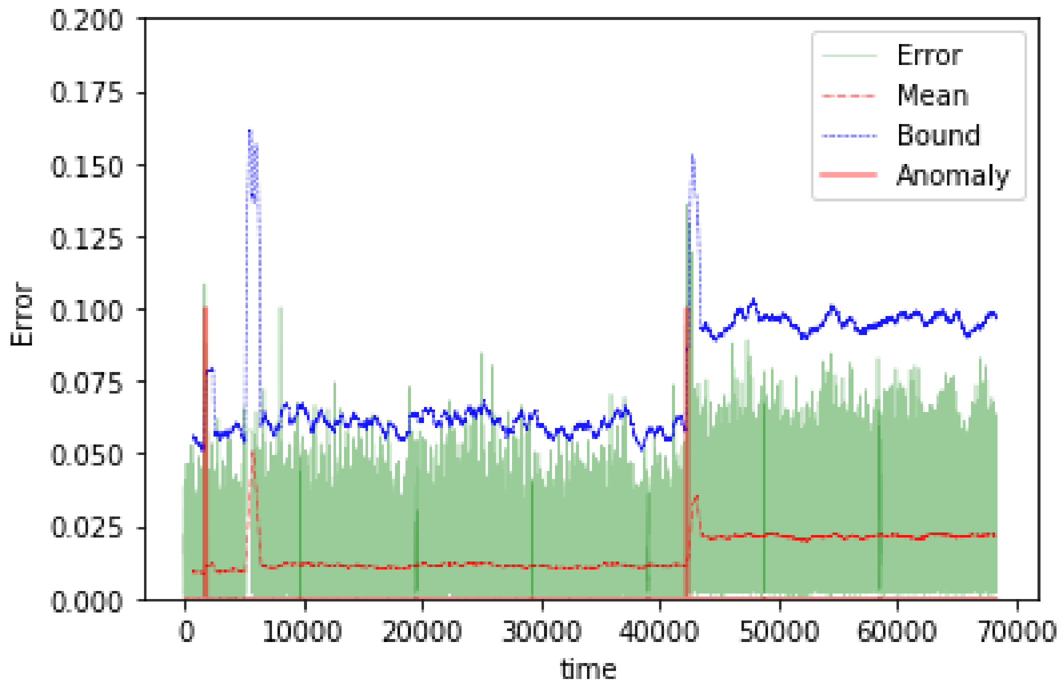


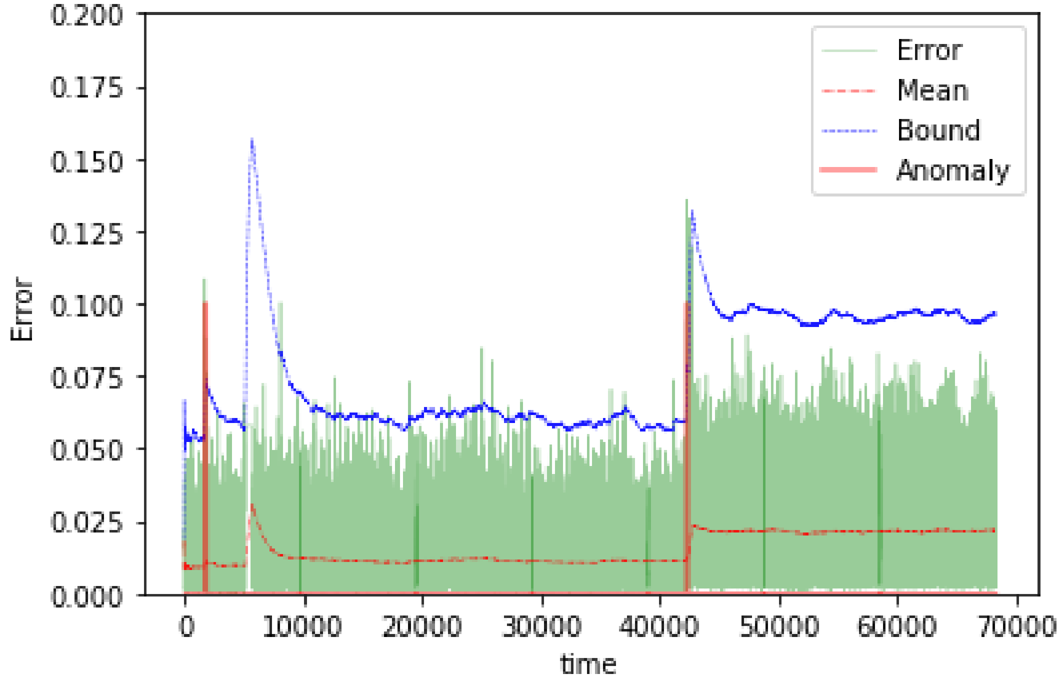

```
Training loss for final epoch is 0.009267749716877006
Validation loss for final epoch is 0.00838193044939544
---------- Beginning tests for nn3_2 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

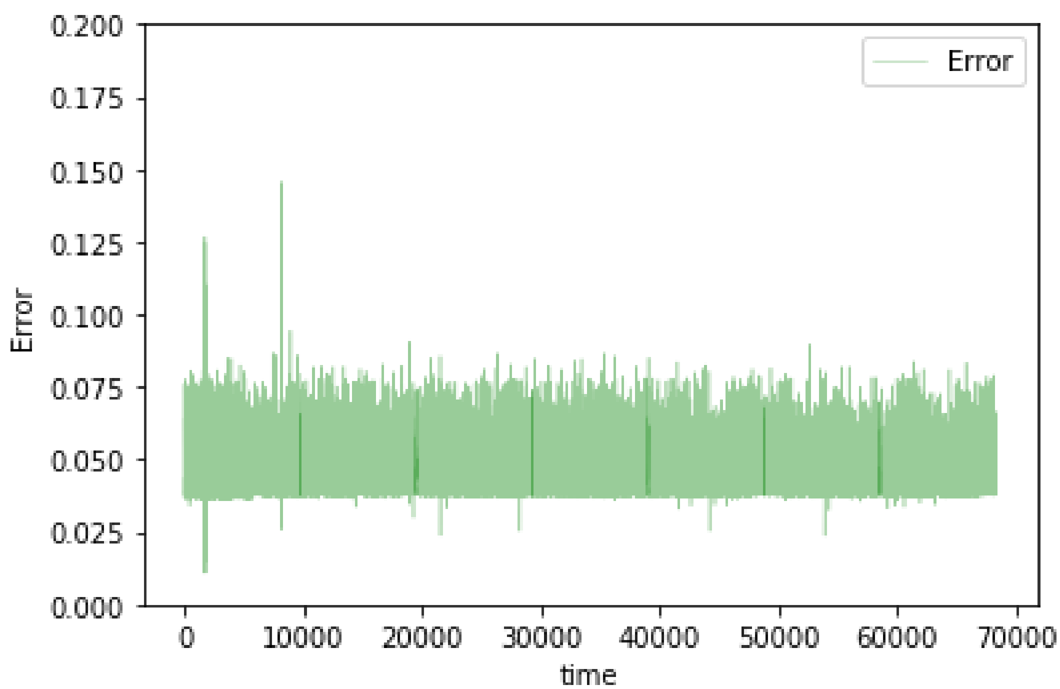

7000
6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

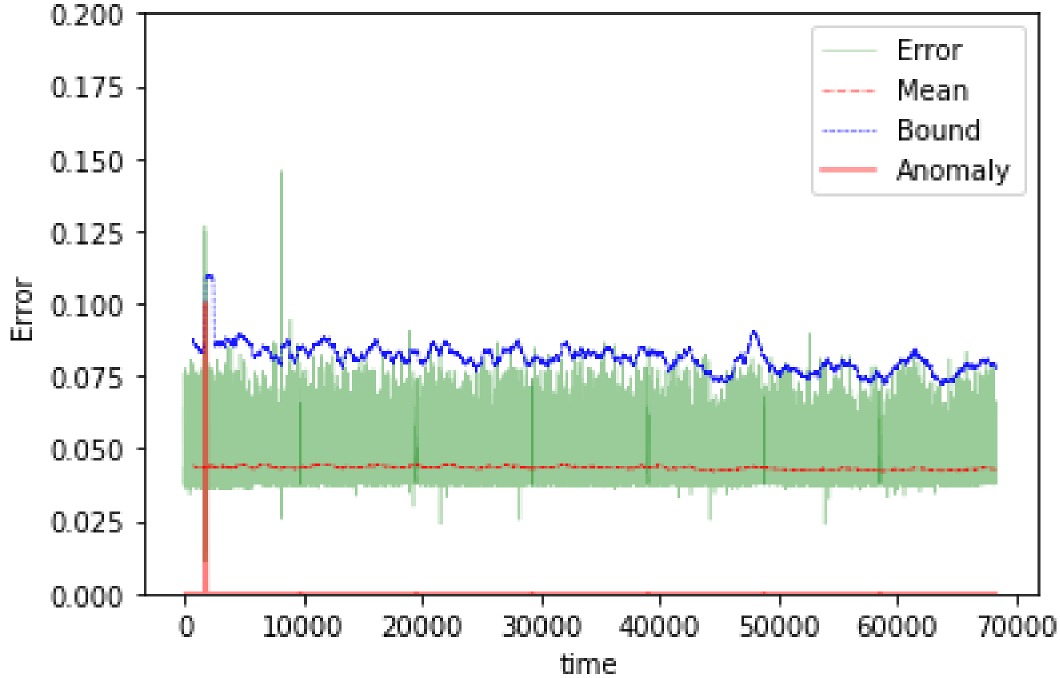

Error
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000


Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

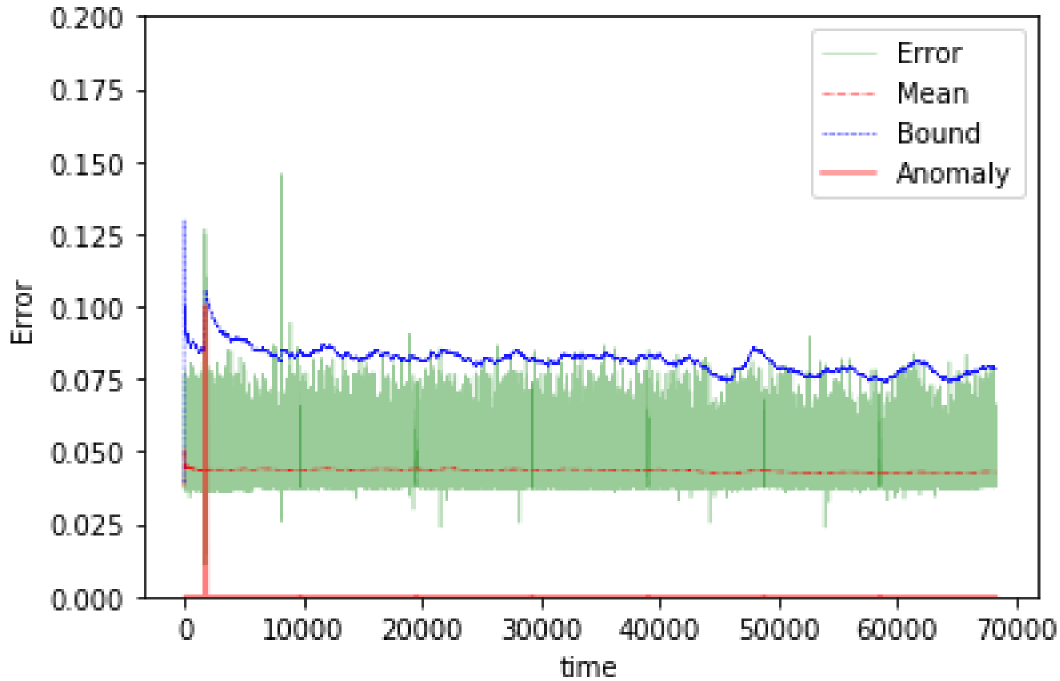



```
The mean error for nn3_2_disk_IO_start_ is 0.01562098514226377 for length 68297
Testing on Avg. load data.
```



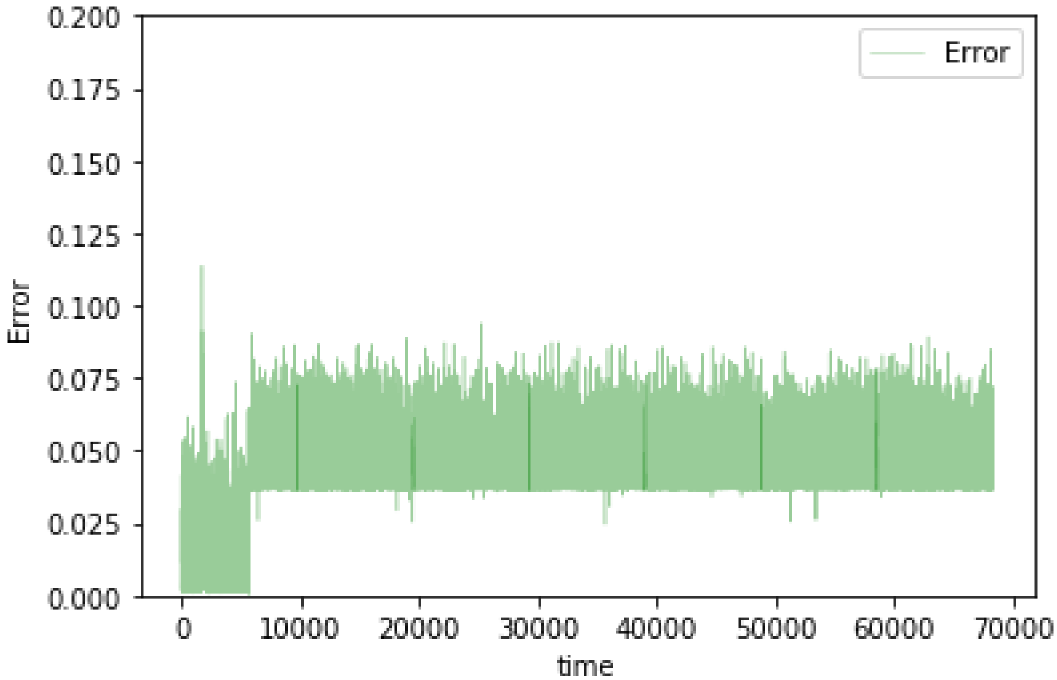


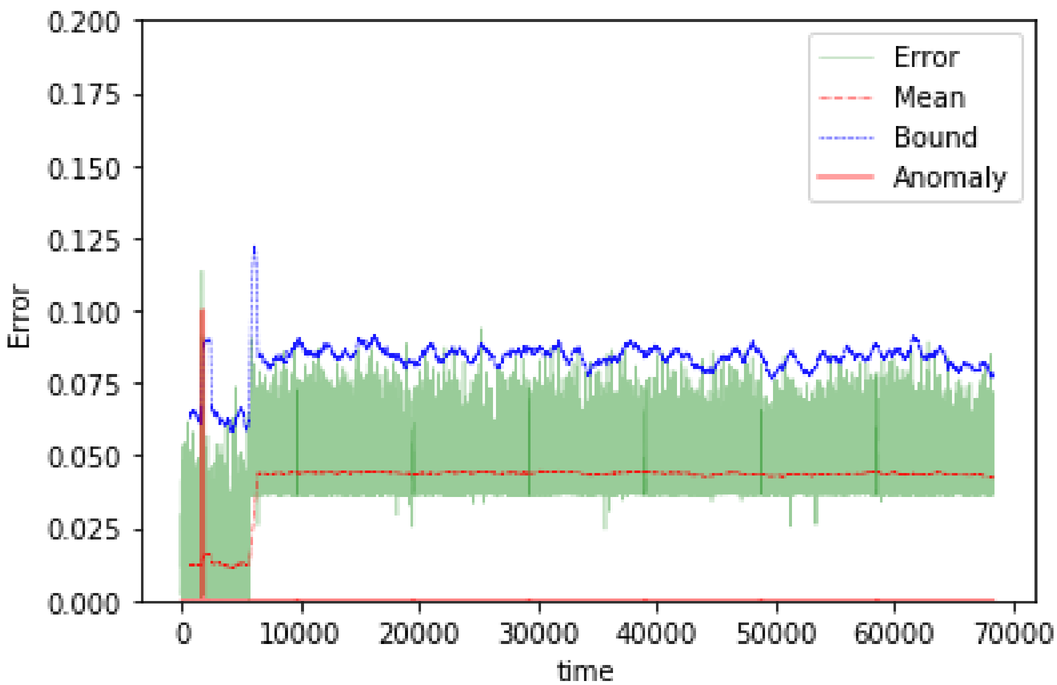
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

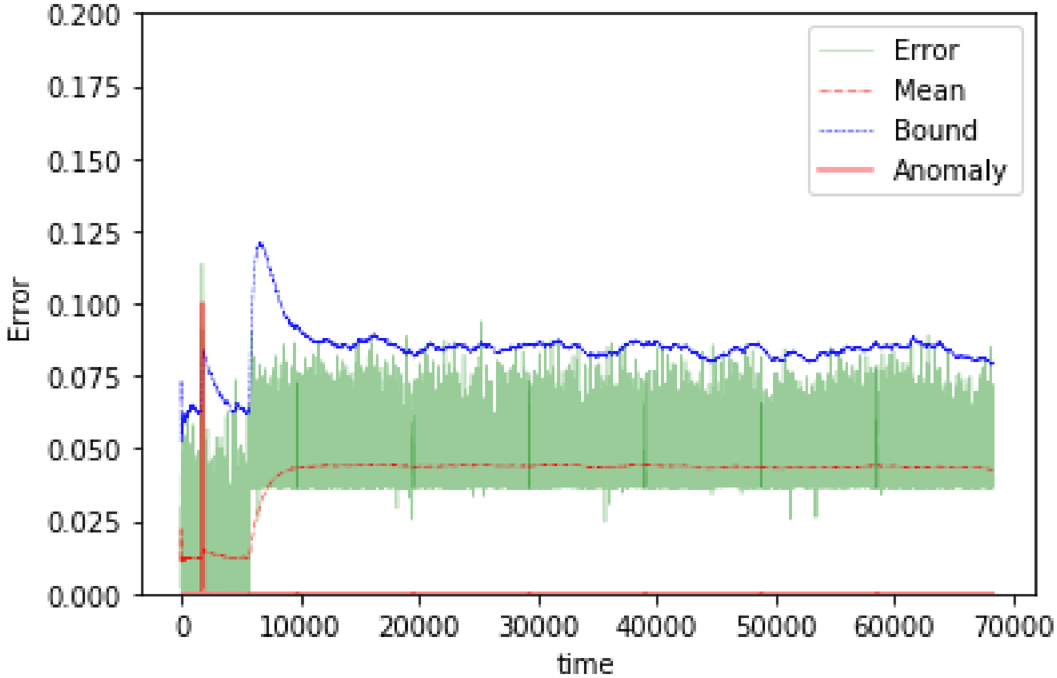


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

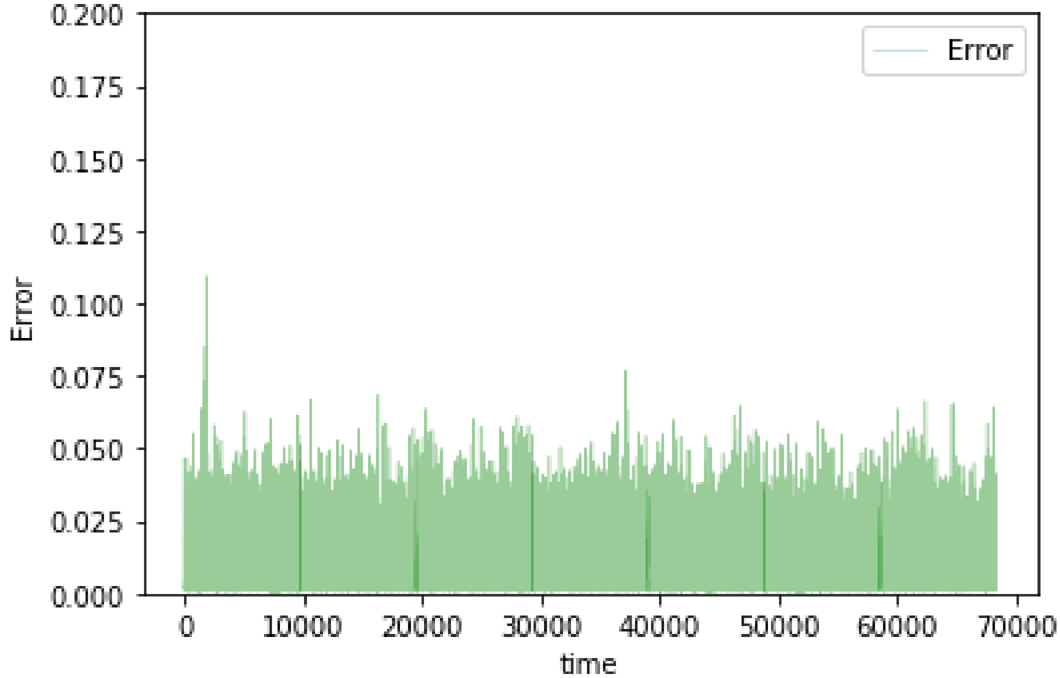
```
The mean error for nn3_2_avg_load_ is 0.04335760921088666 for length 68297
Testing on app change early data.
```

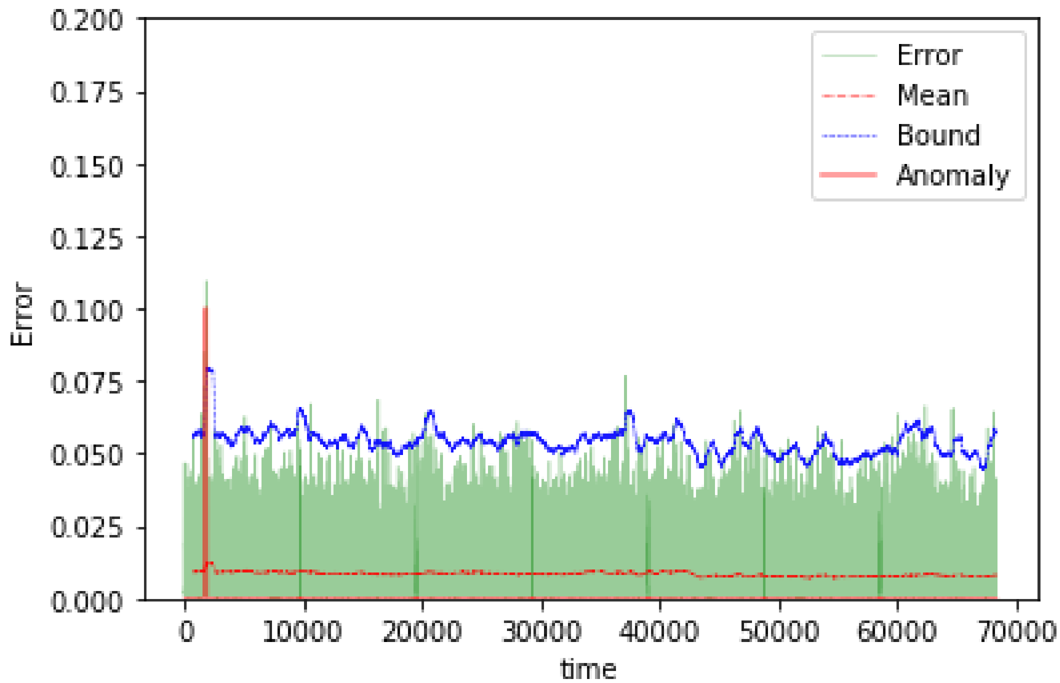


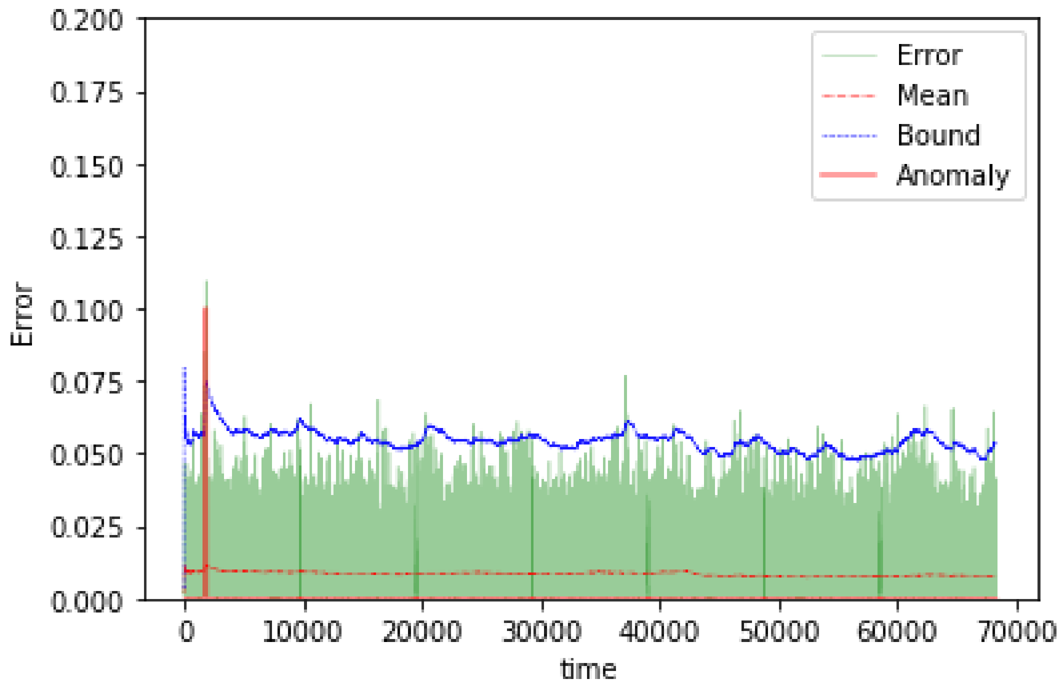





```
The mean error for nn3_2_app_change_early_ is 0.04120824543491214 for length 68297
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

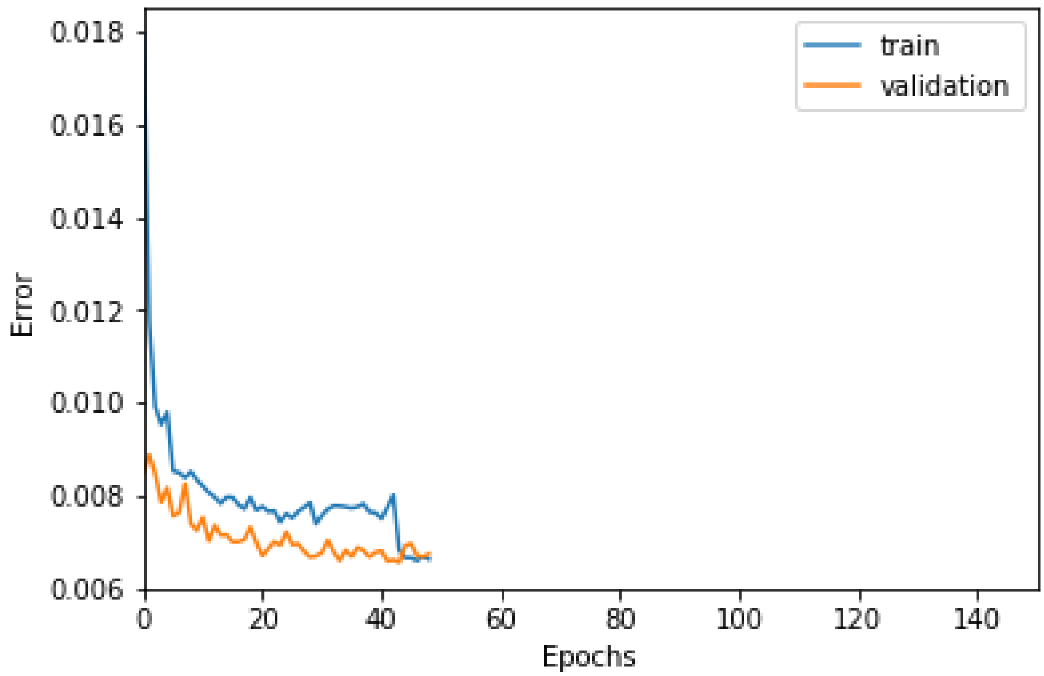

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



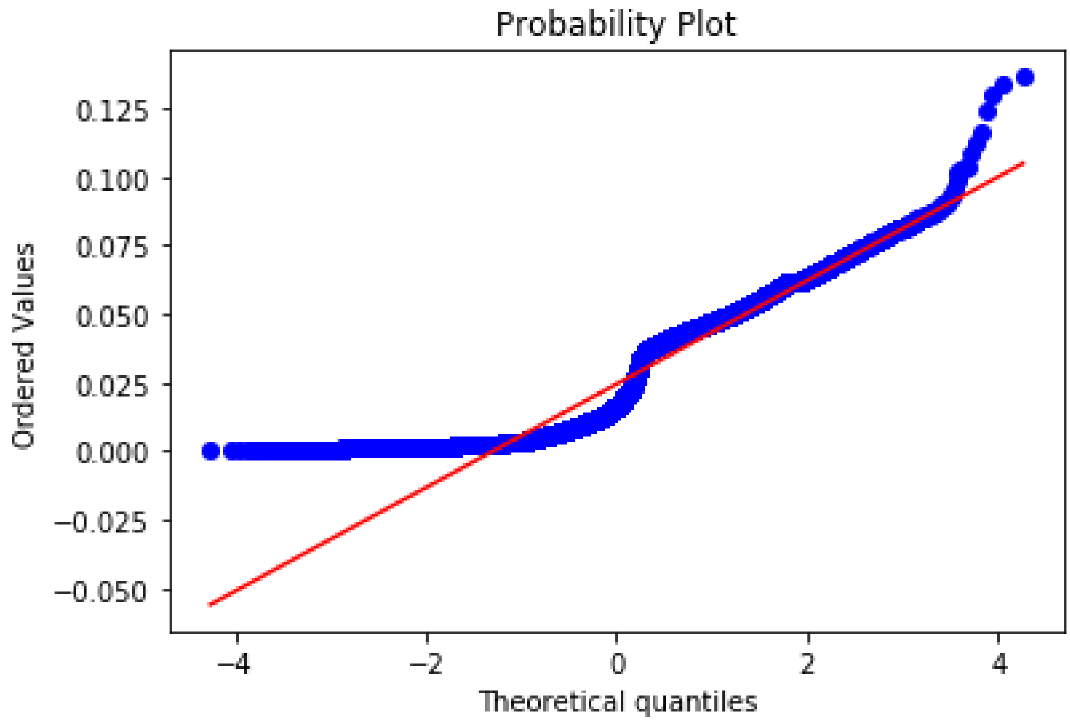

```
The mean error for nn3_2_normal_ is 0.008463966538516903 for length 68297
====================
```

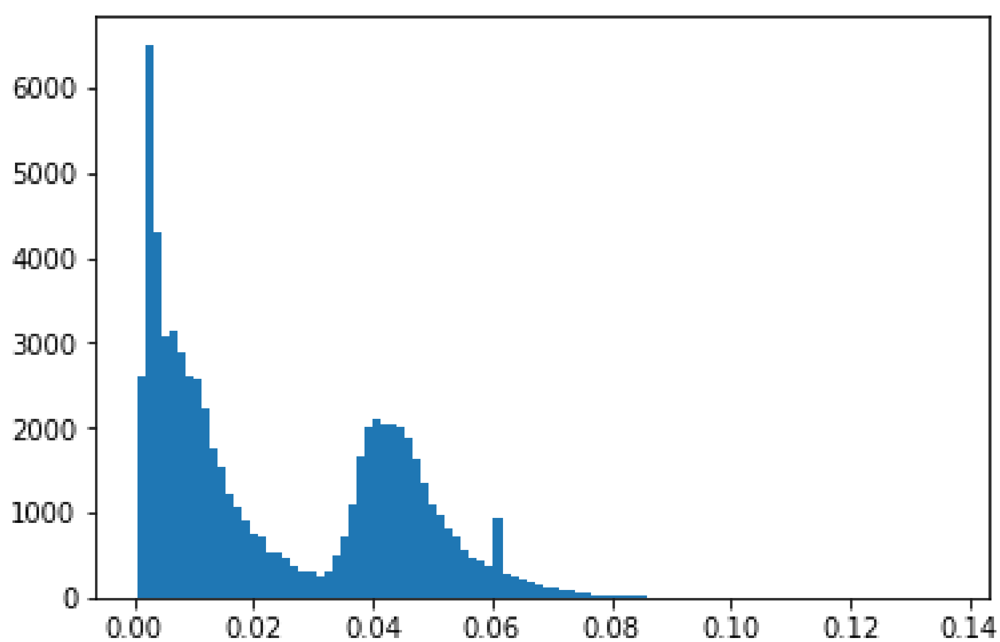
5 steps

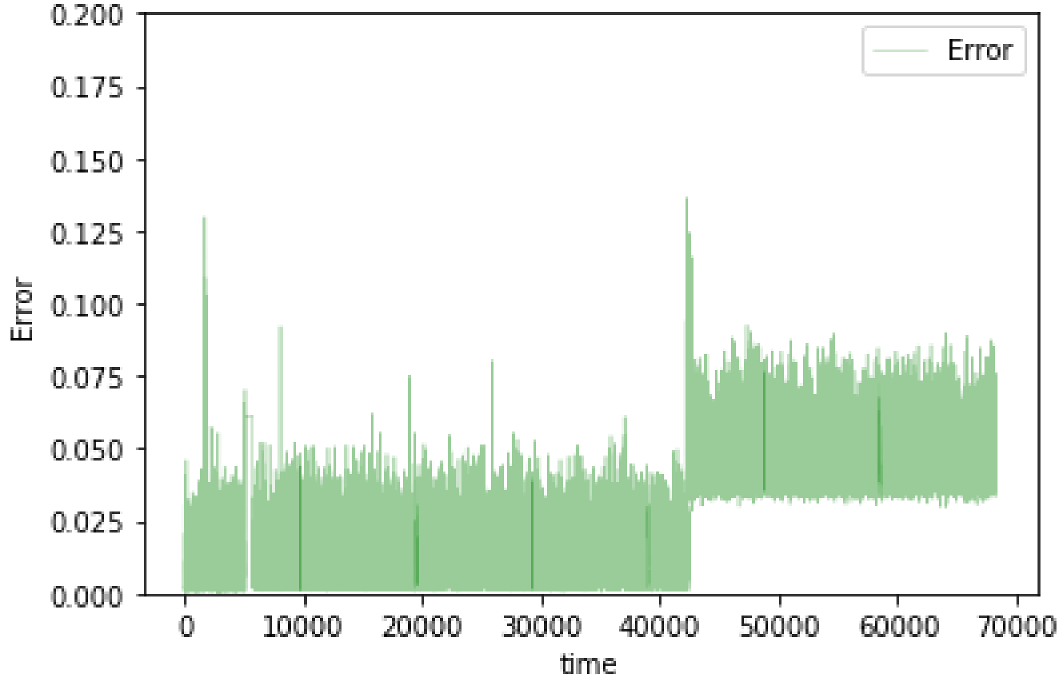
```
In [127]: TIMESTEPS = 5
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS)
          vgen = flat_generator(val_X, TIMESTEPS)
          name = "nn3_5"

In [128]: input_layer = Input(shape=(TIMESTEPS*DIM,))
          hidden = Dense(1000, activation='relu')(input_layer)
          hidden = Dense(500, activation='relu')(hidden)
          hidden = Dense(100, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [129]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [130]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```

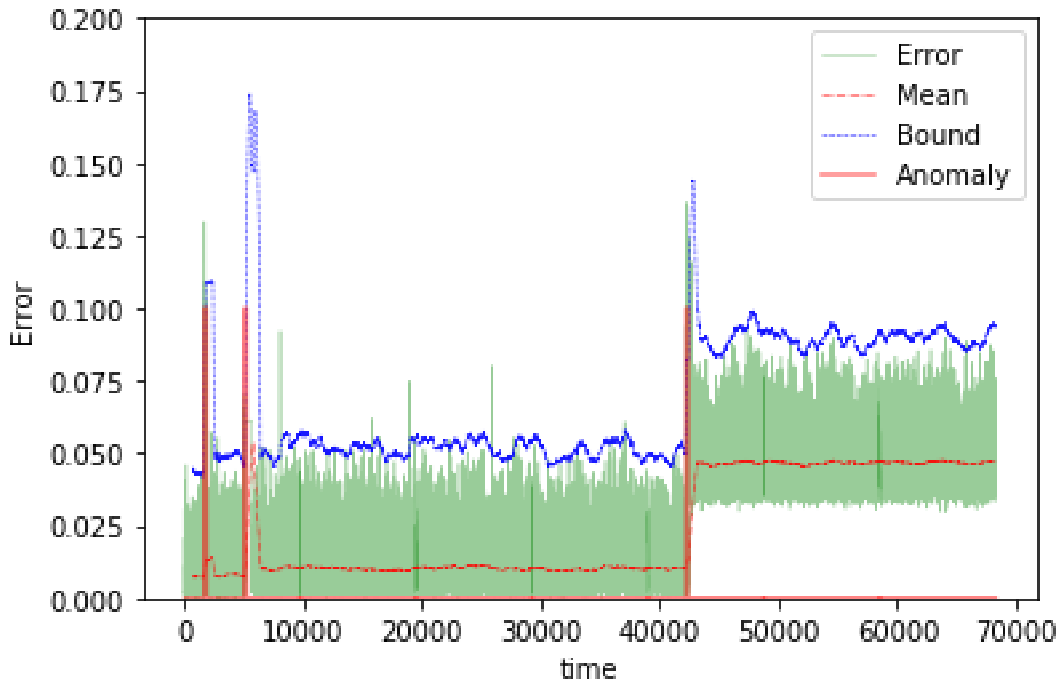


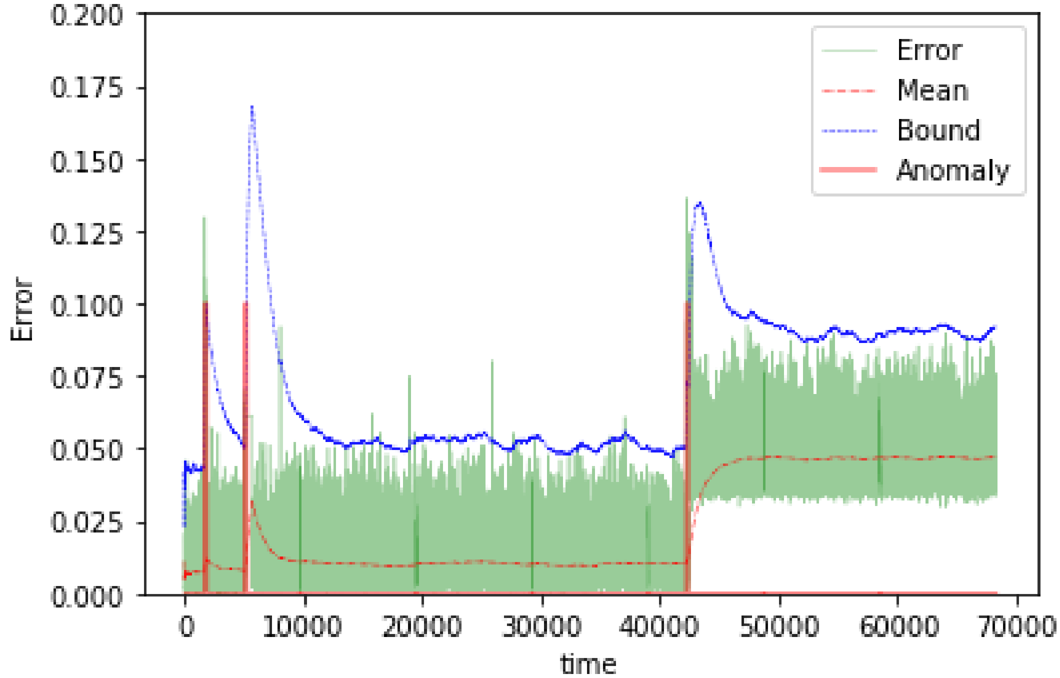

```
Training loss for final epoch is 0.006621745623706374
Validation loss for final epoch is 0.006735541237168945
---------- Beginning tests for nn3_5 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−4
−2
0
2
4

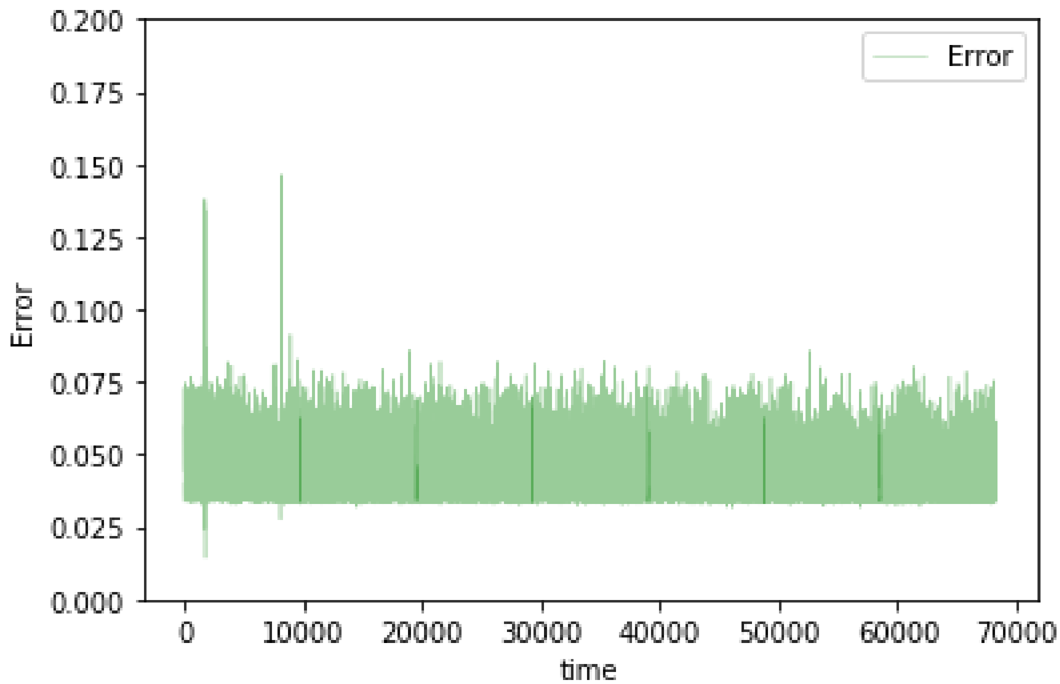

6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

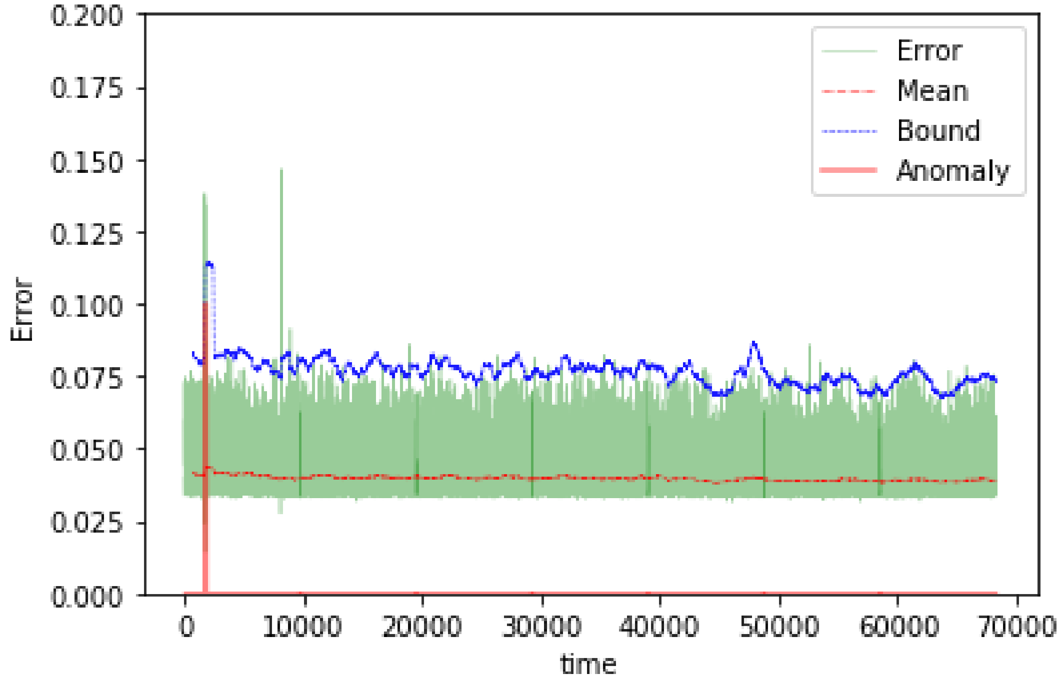

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

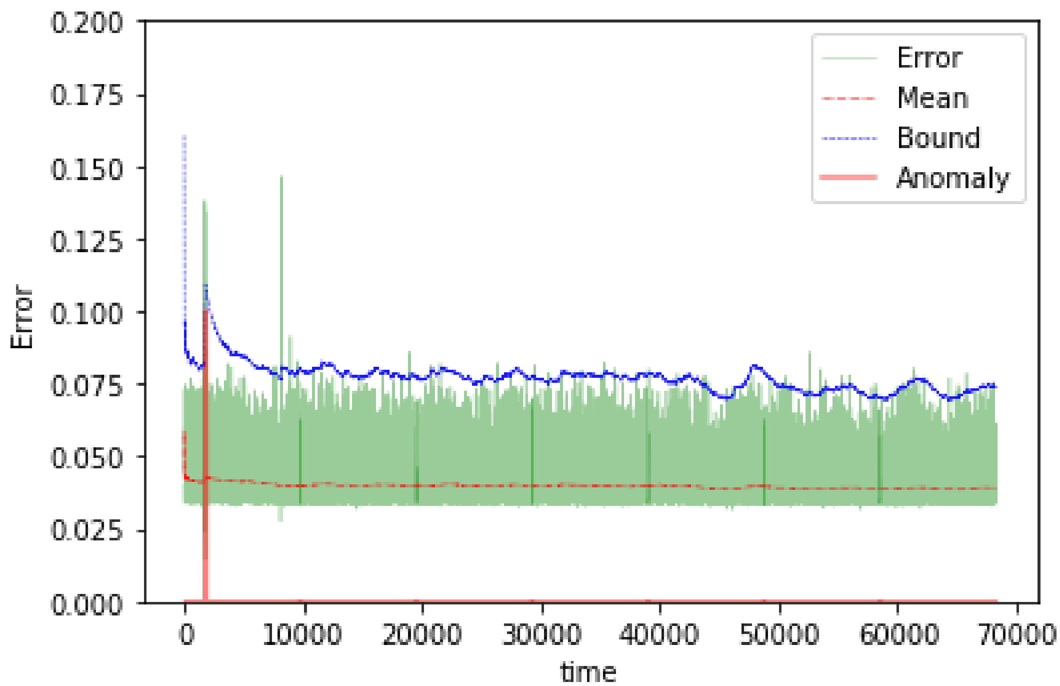

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



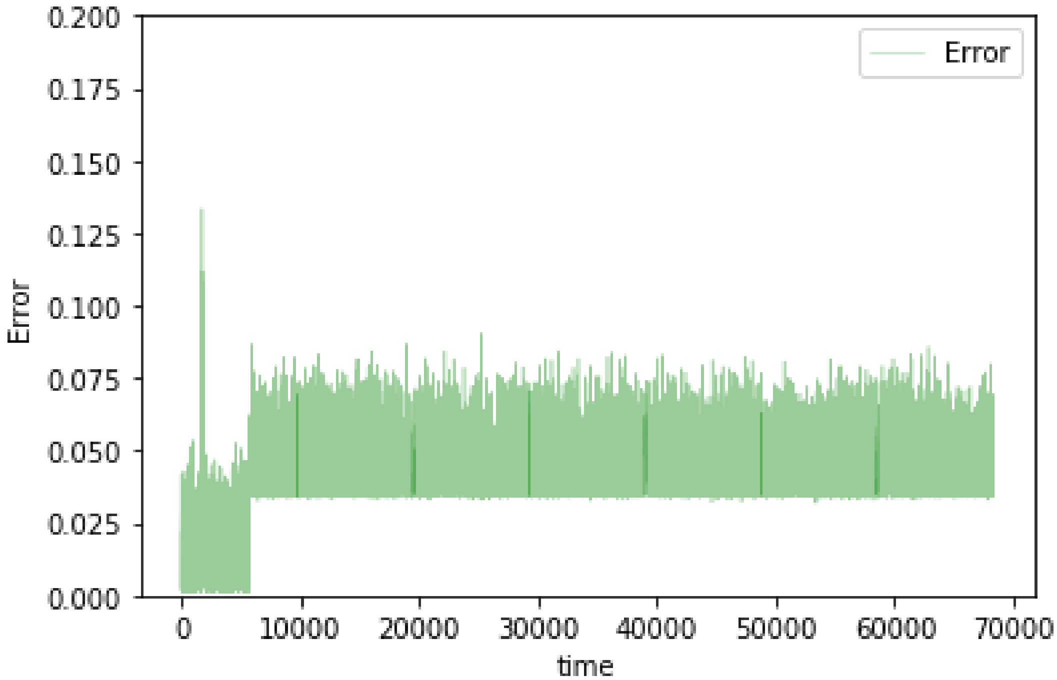

```
The mean error for nn3_5_disk_IO_start_ is 0.024383452637757548 for length 68294
Testing on Avg. load data.
```

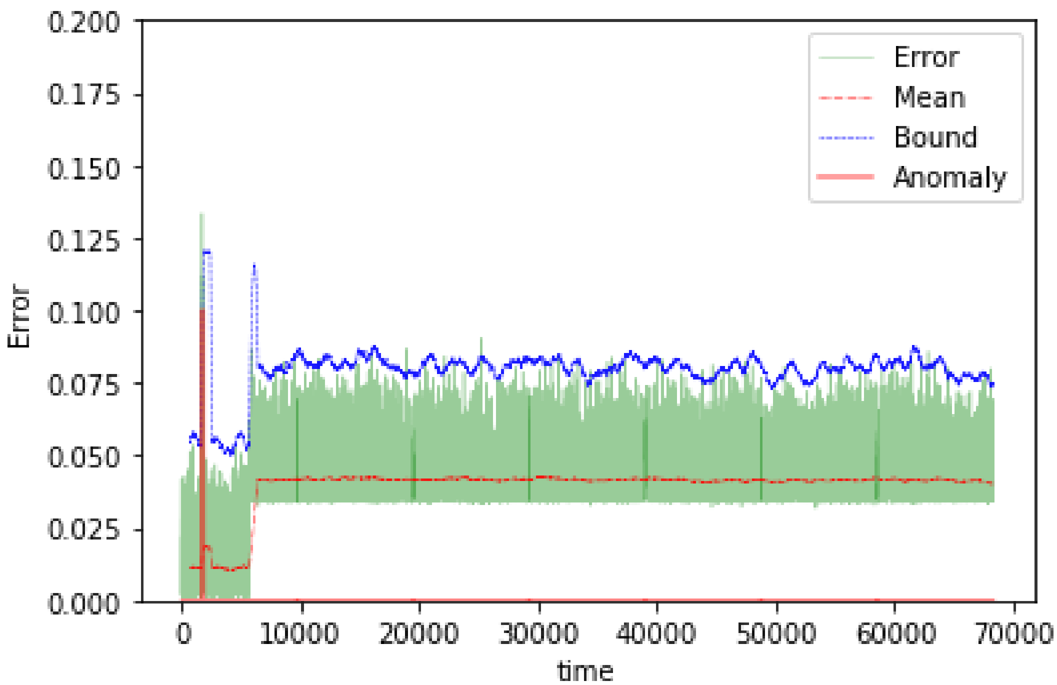


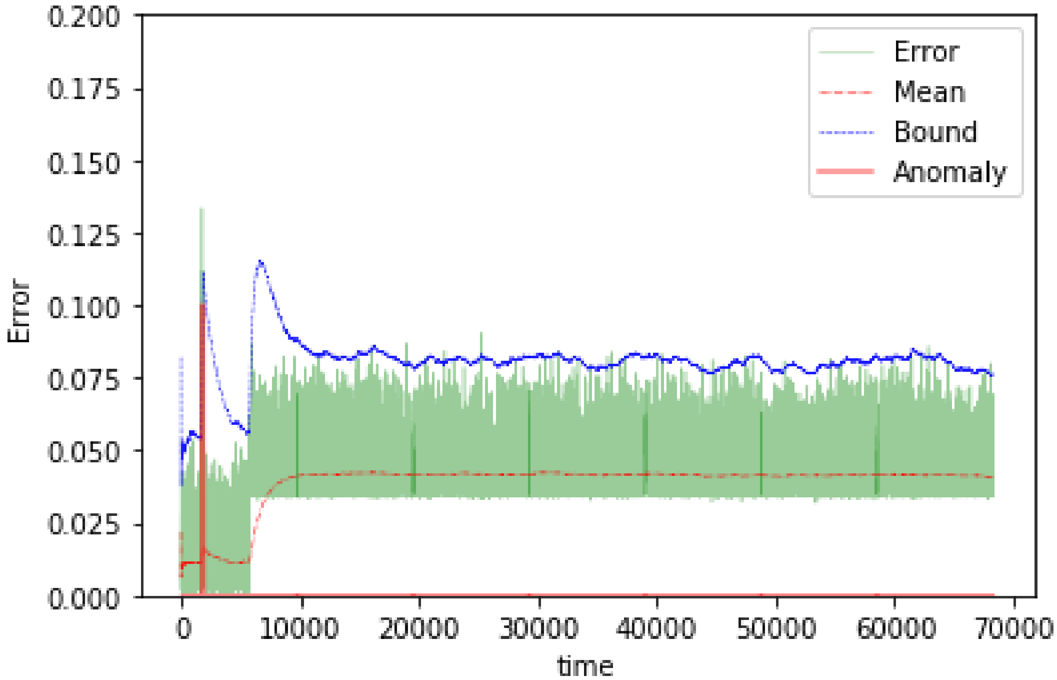

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

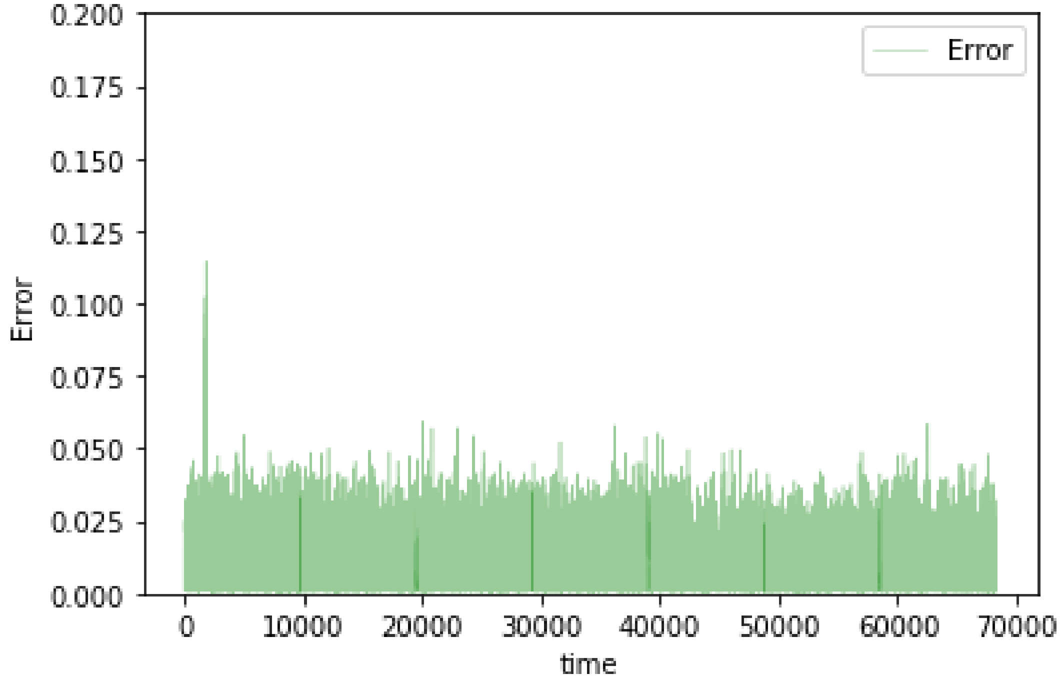
```
The mean error for nn3_5_avg_load_ is 0.039759870271003556 for length 68294
Testing on app change early data.
```

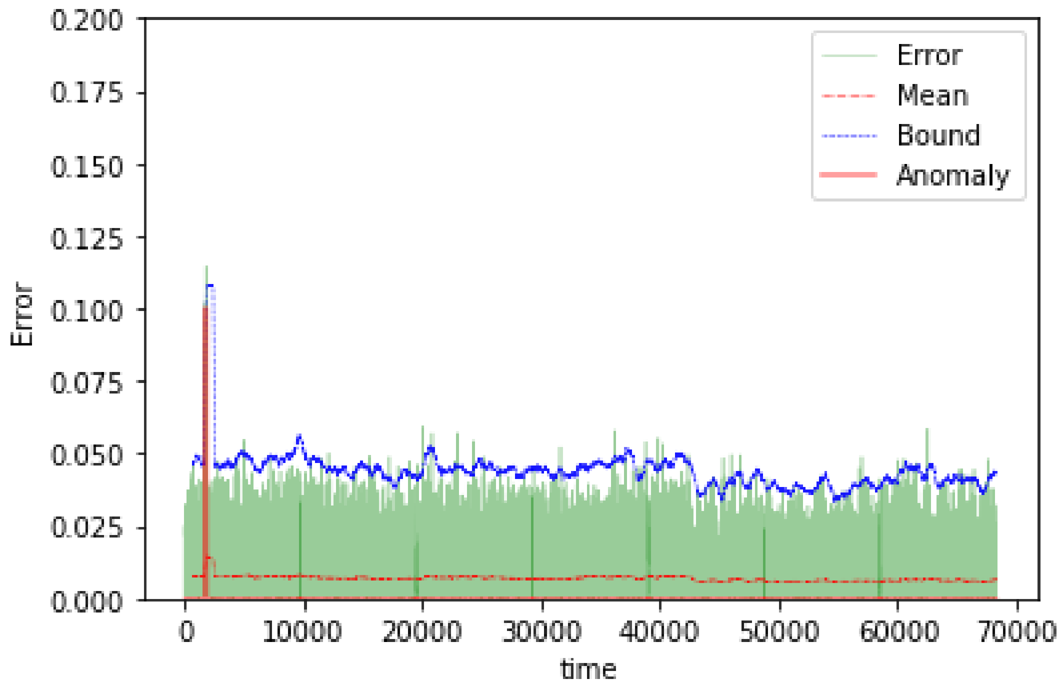


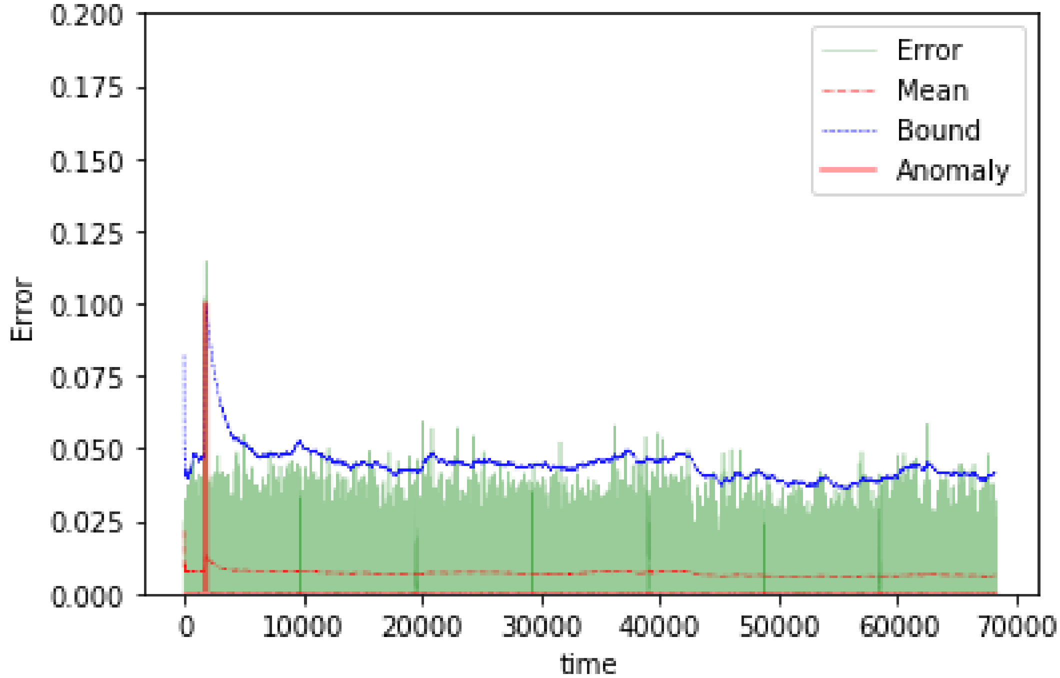




The mean error for nn3_5_app_change_early_ is 0.03916664550189386 for length 68294
Testing on Normal data.


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

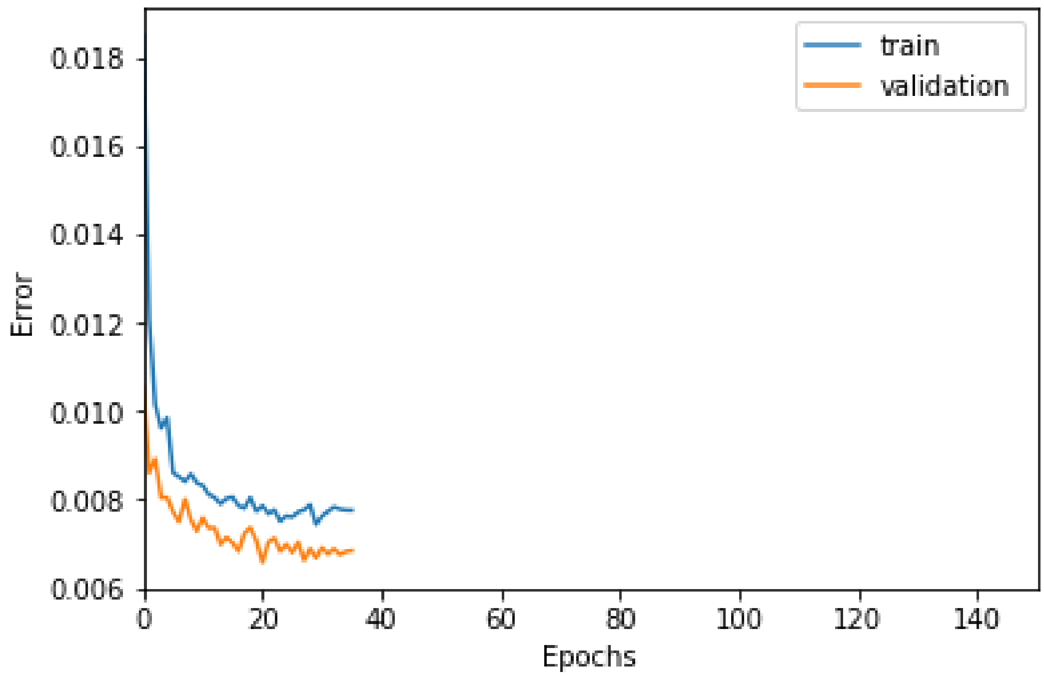

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



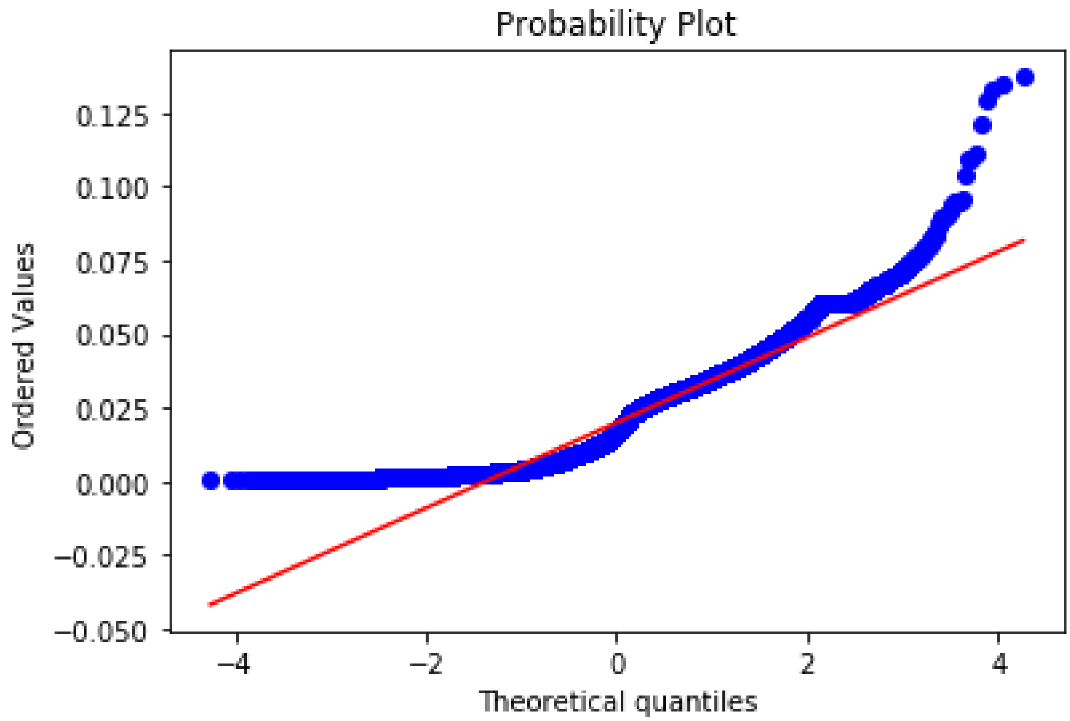

```
The mean error for nn3_5_normal_ is 0.006896238125829558 for length 68294
====================
```

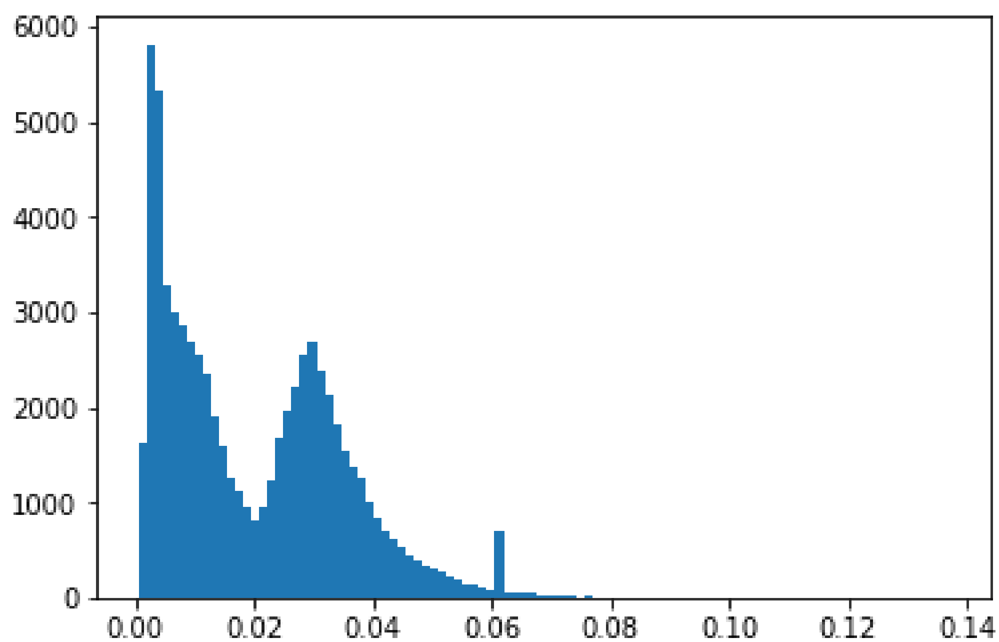
10 steps

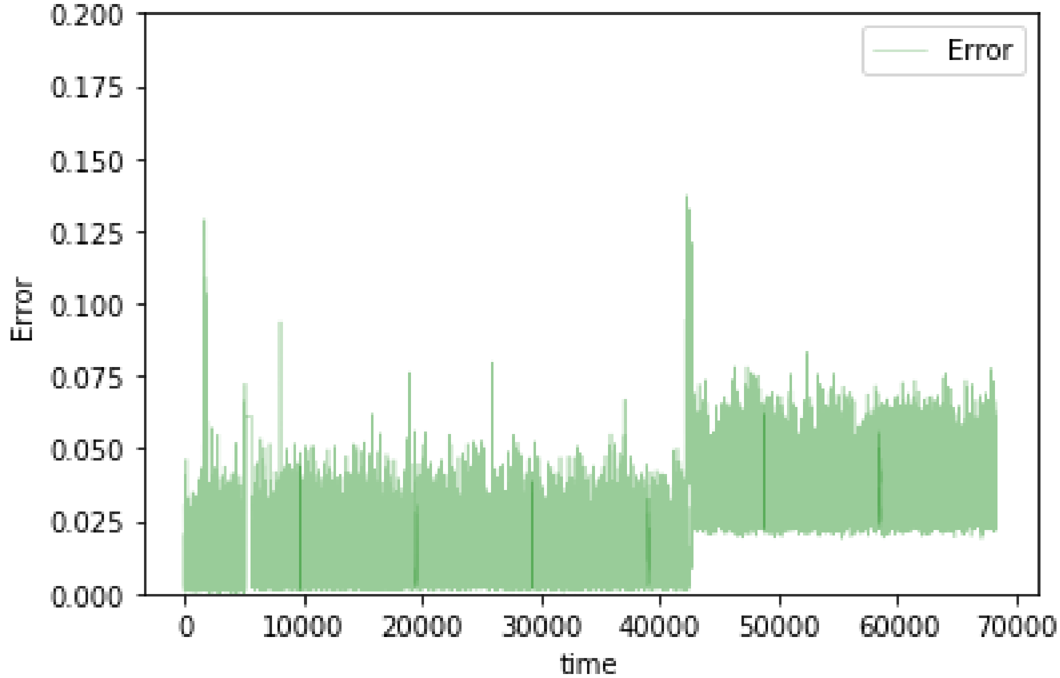
```
In [131]: TIMESTEPS = 10
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS)
          vgen = flat_generator(val_X, TIMESTEPS)
          name = "nn3_10"

In [132]: input_layer = Input(shape=(TIMESTEPS*DIM,))
          hidden = Dense(1000, activation='relu')(input_layer)
          hidden = Dense(500, activation='relu')(hidden)
          hidden = Dense(100, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [133]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [134]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```



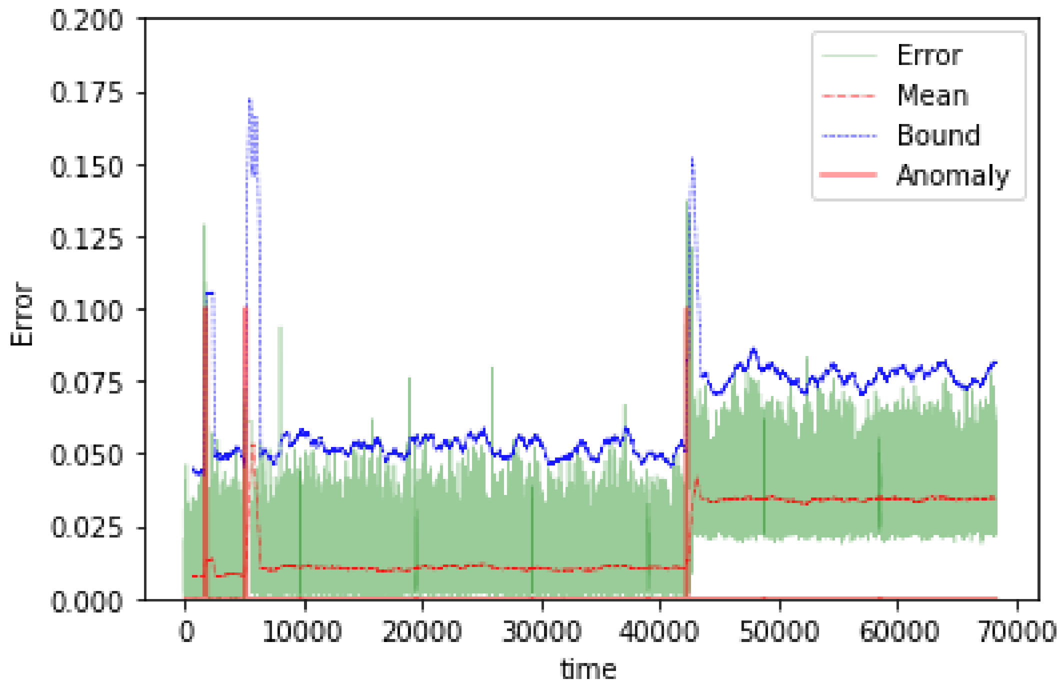

```
Training loss for final epoch is 0.007761832438642159
Validation loss for final epoch is 0.006848905569117051
---------- Beginning tests for nn3_10 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

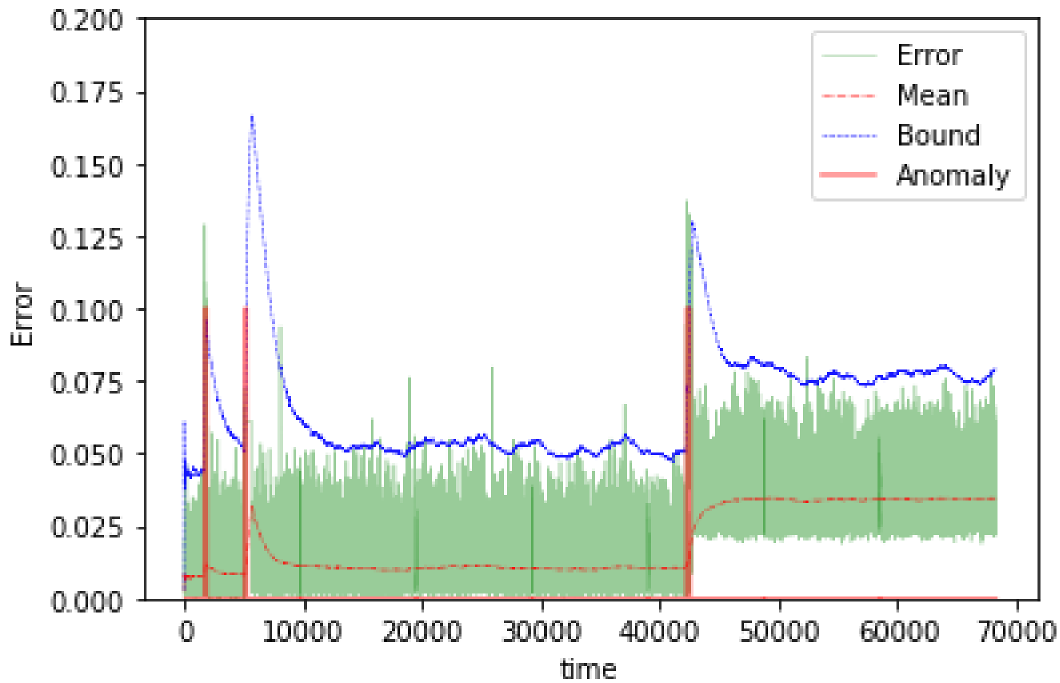

6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14



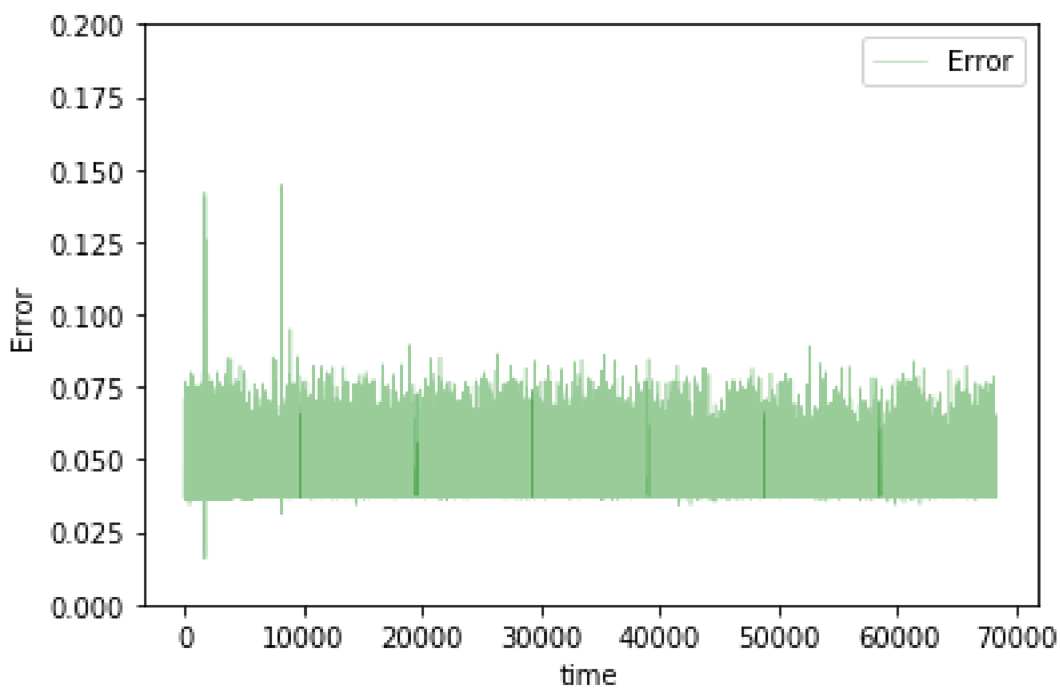
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

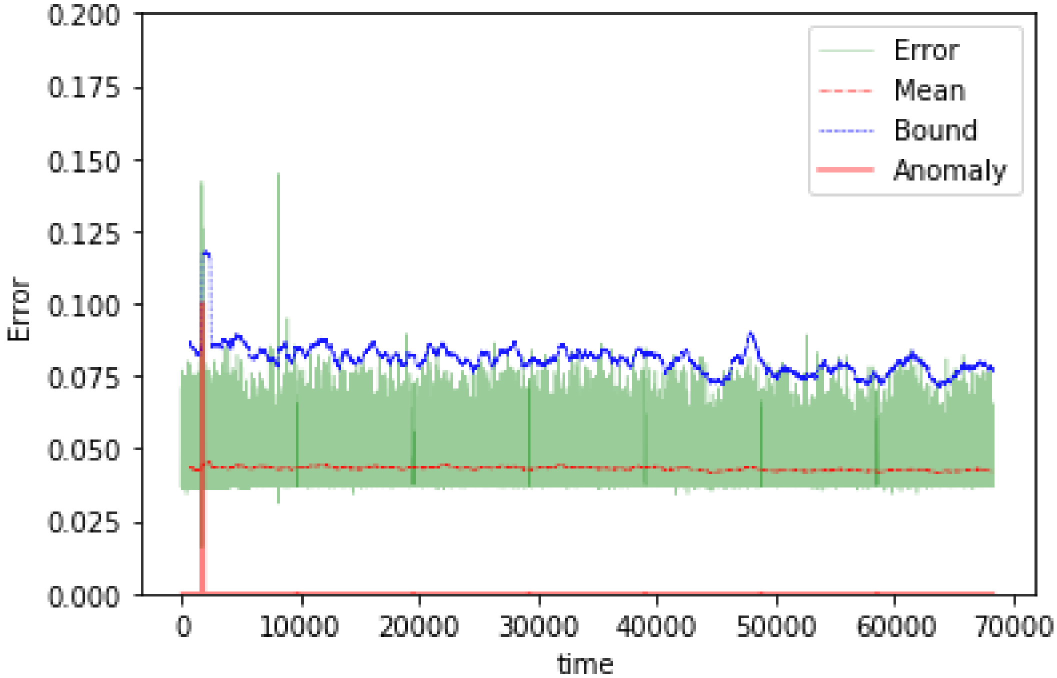


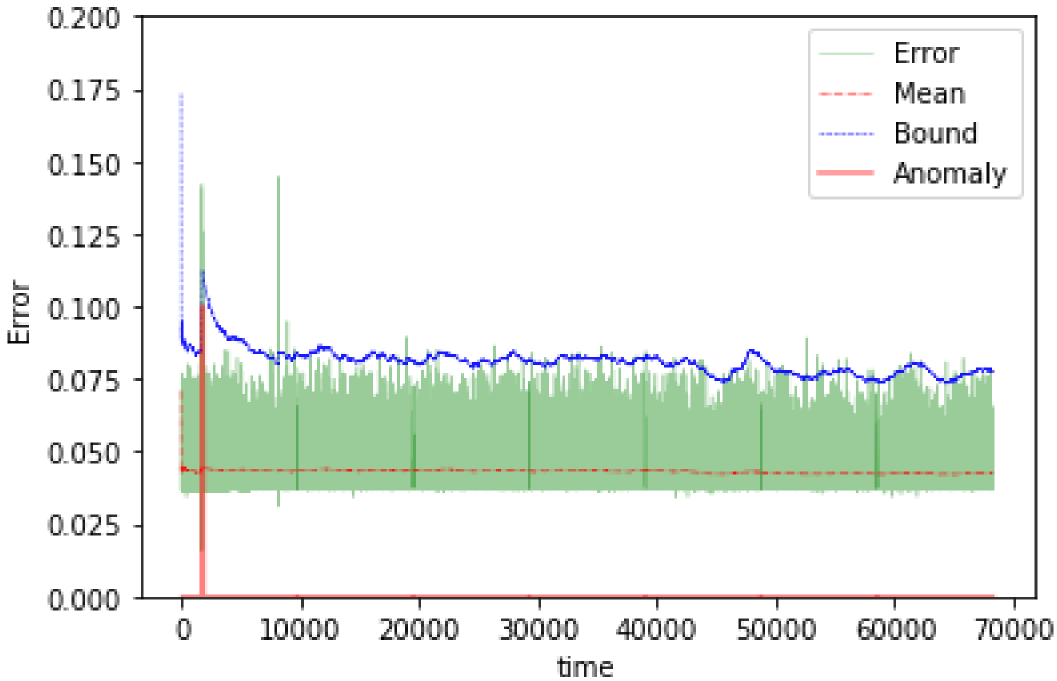
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



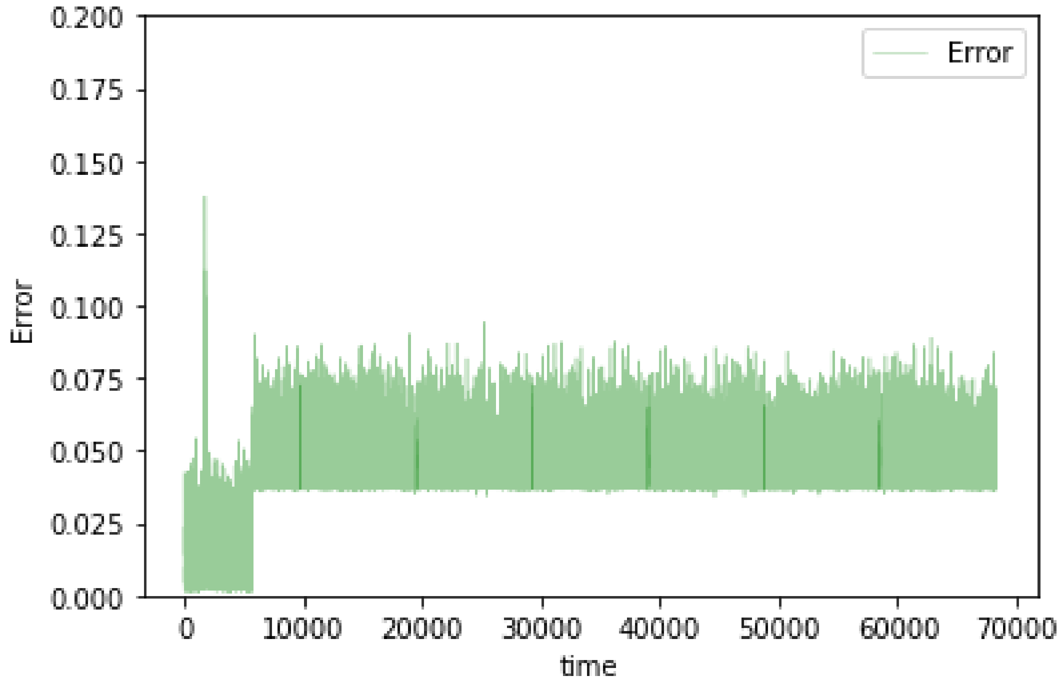

```
The mean error for nn3_10_disk_IO_start_ is 0.019927473123956468 for length 68289
Testing on Avg. load data.
```

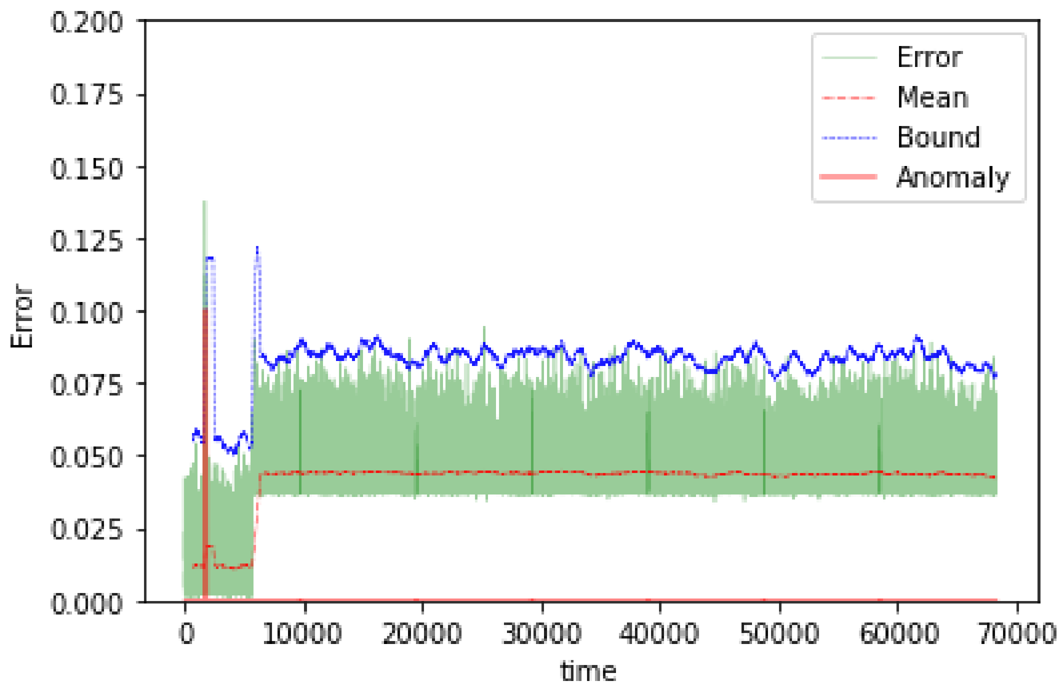


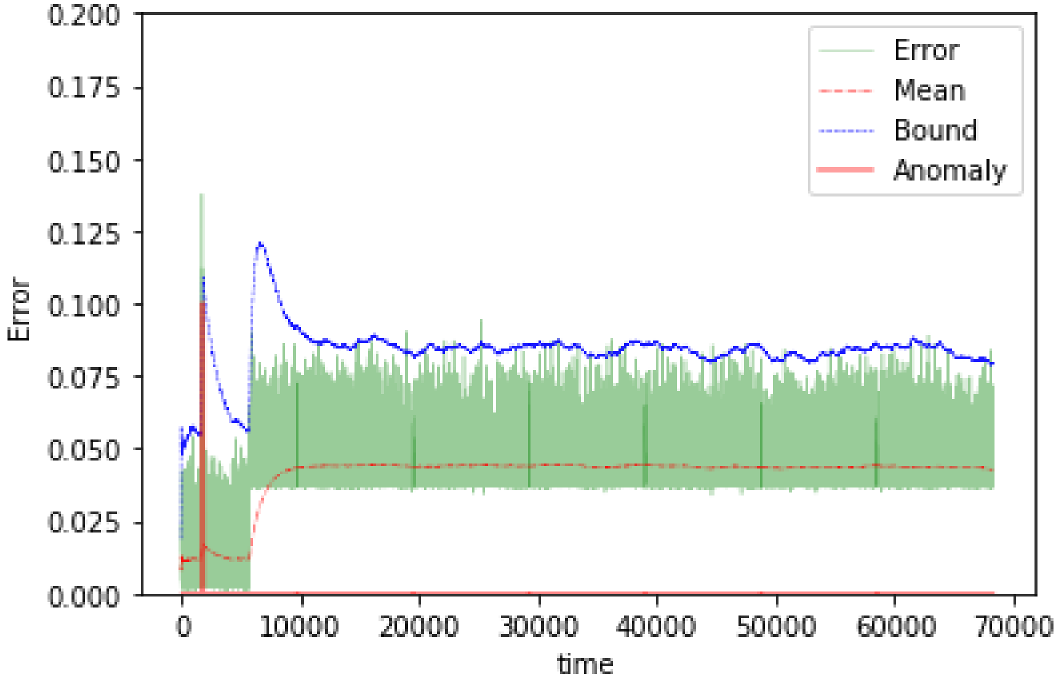

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

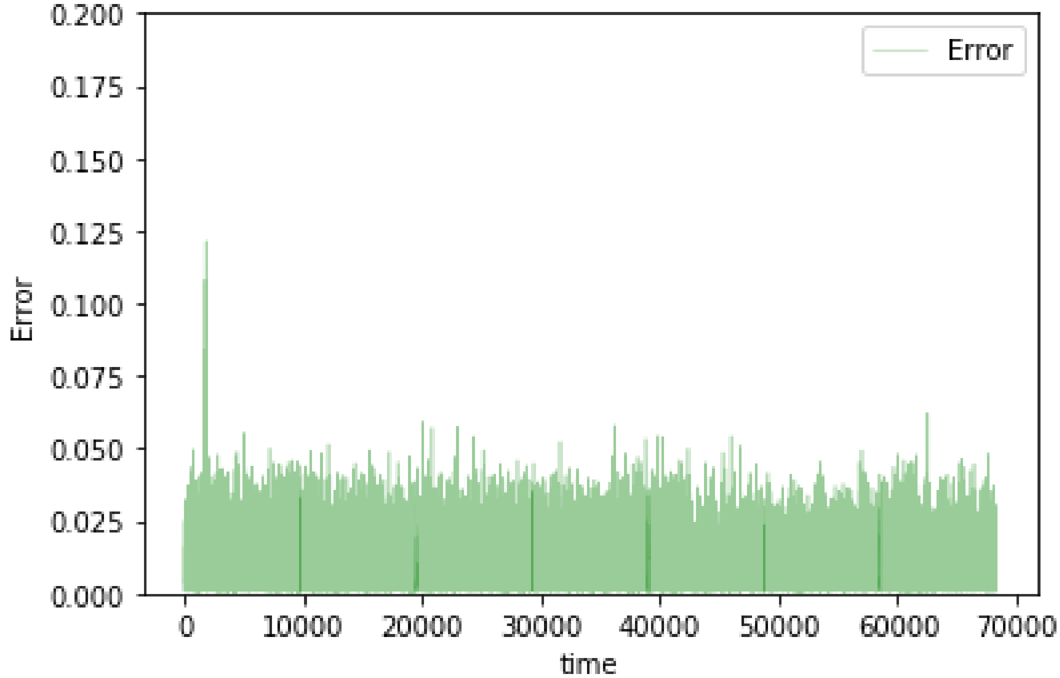
```
The mean error for nn3_10_avg_load_ is 0.043089312279137905 for length 68289
Testing on app change early data.
```

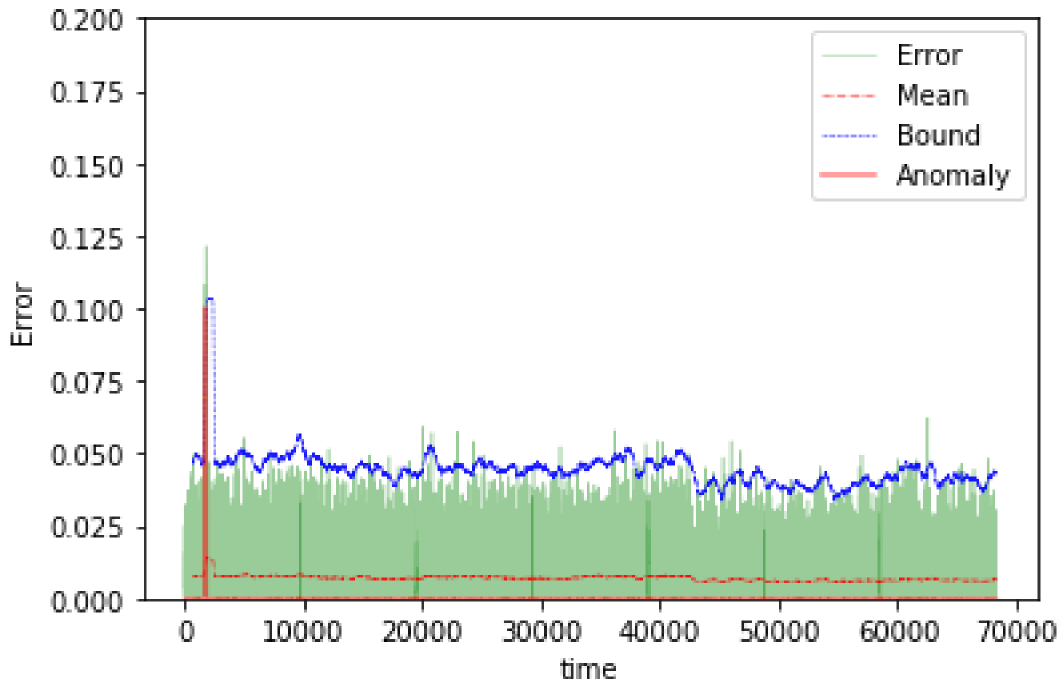


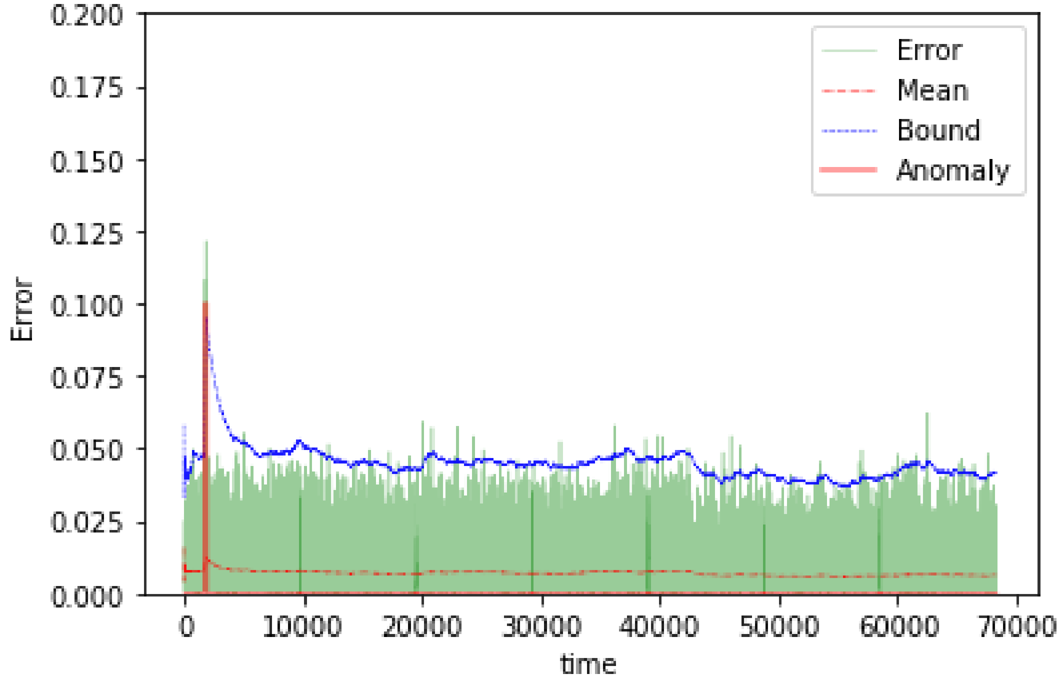





```
The mean error for nn3_10_app_change_early_ is 0.04118084863338845 for length 68289
Testing on Normal data.
```


Error
Error
time


Error
Mean
Bound
Anomaly
Error
time




```
The mean error for nn3_10_normal_ is 0.007001696697292422 for length 68289
====================
```

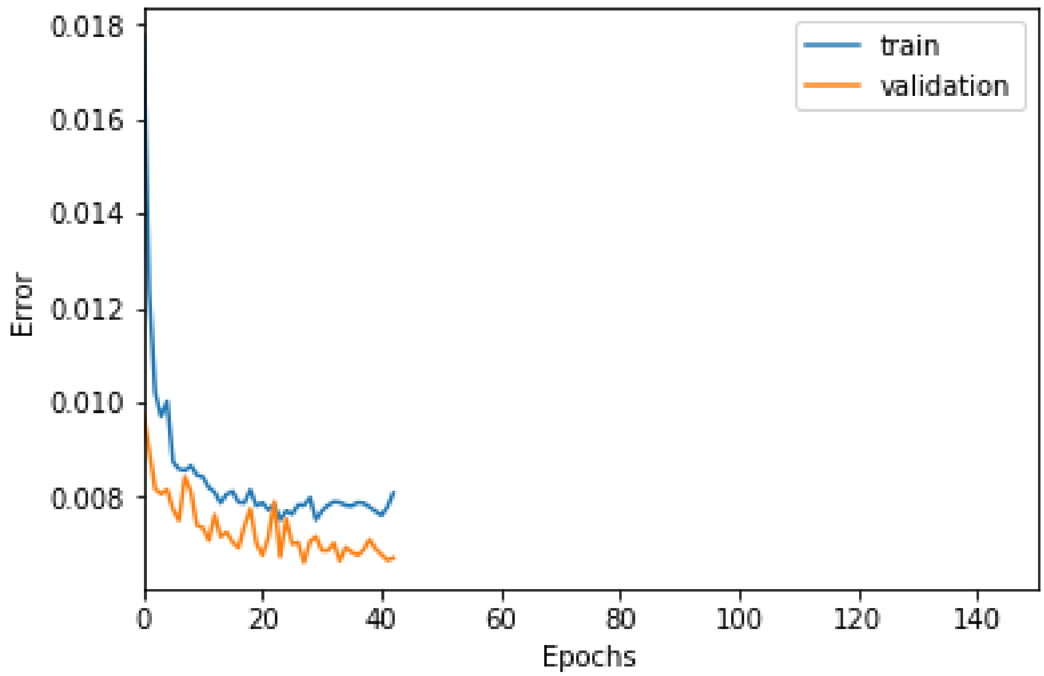
20 steps

```
In [135]: TIMESTEPS = 20
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS)
          vgen = flat_generator(val_X, TIMESTEPS)
          name = "nn3_20"
```

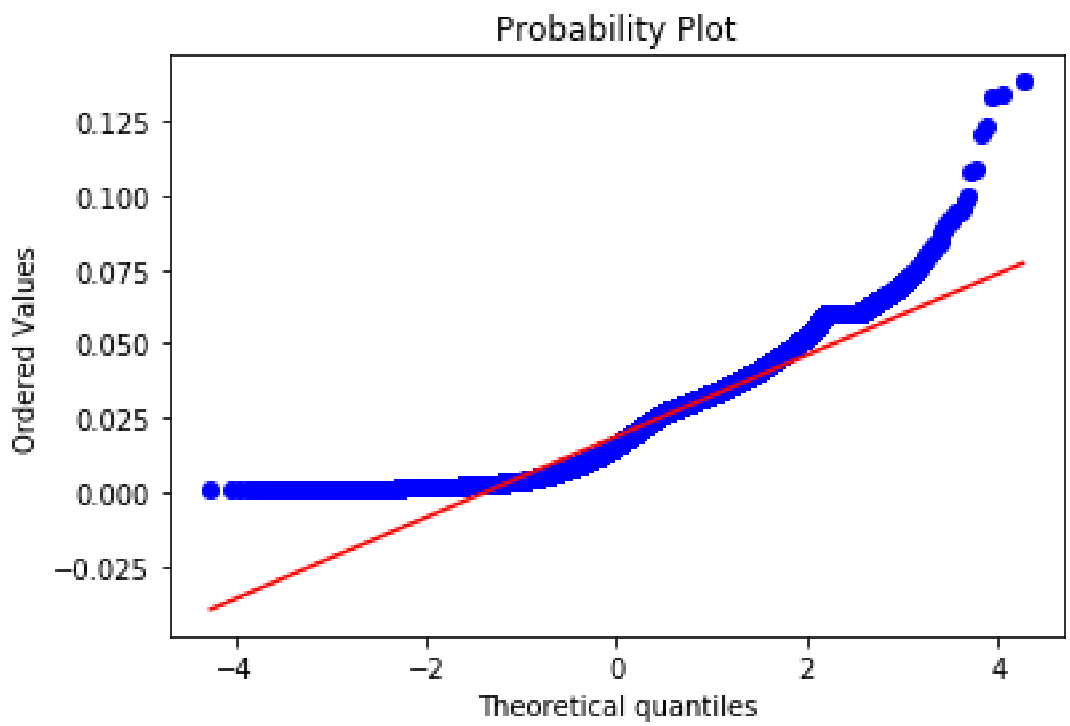
```
In [136]: input_layer = Input(shape=(TIMESTEPS*DIM,))
          hidden = Dense(1000, activation='relu')(input_layer)
          hidden = Dense(500, activation='relu')(hidden)
          hidden = Dense(100, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)
```

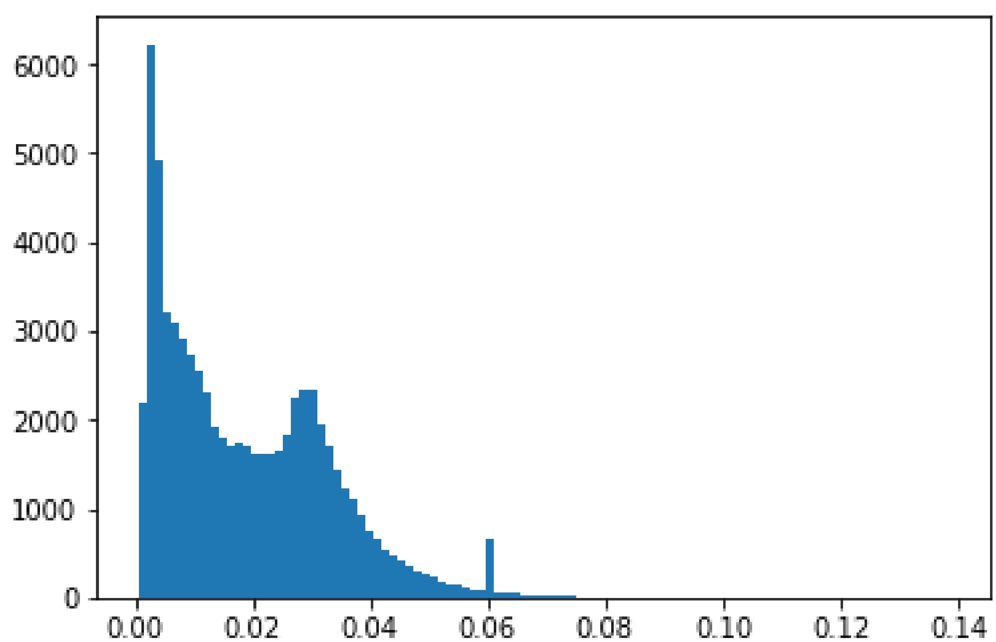
```
In [137]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])
```

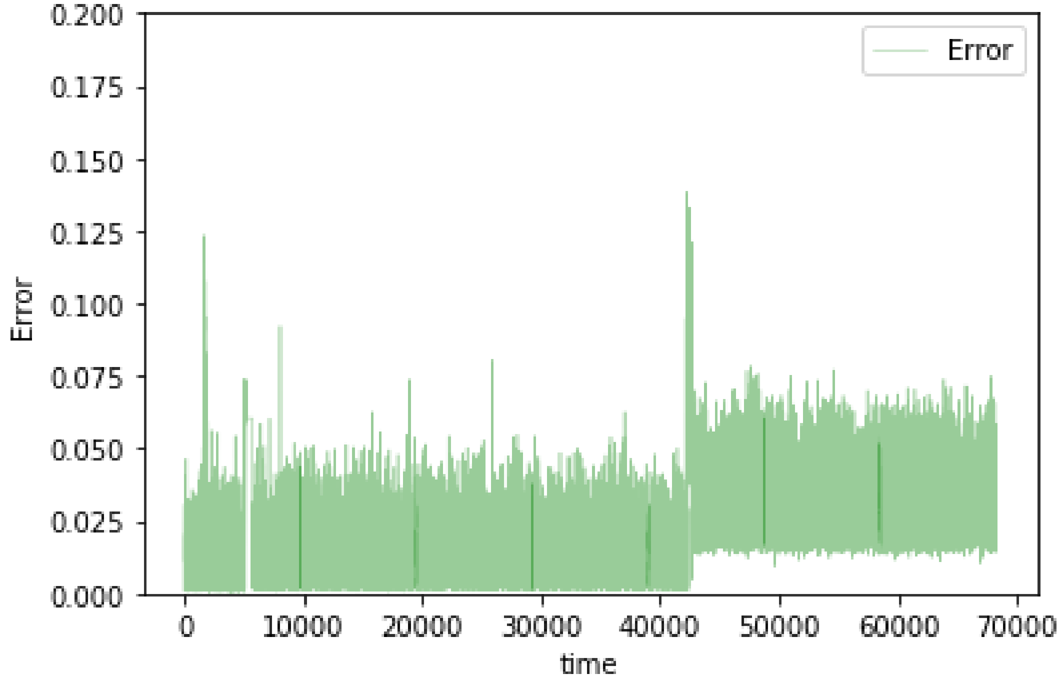
```
In [138]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```



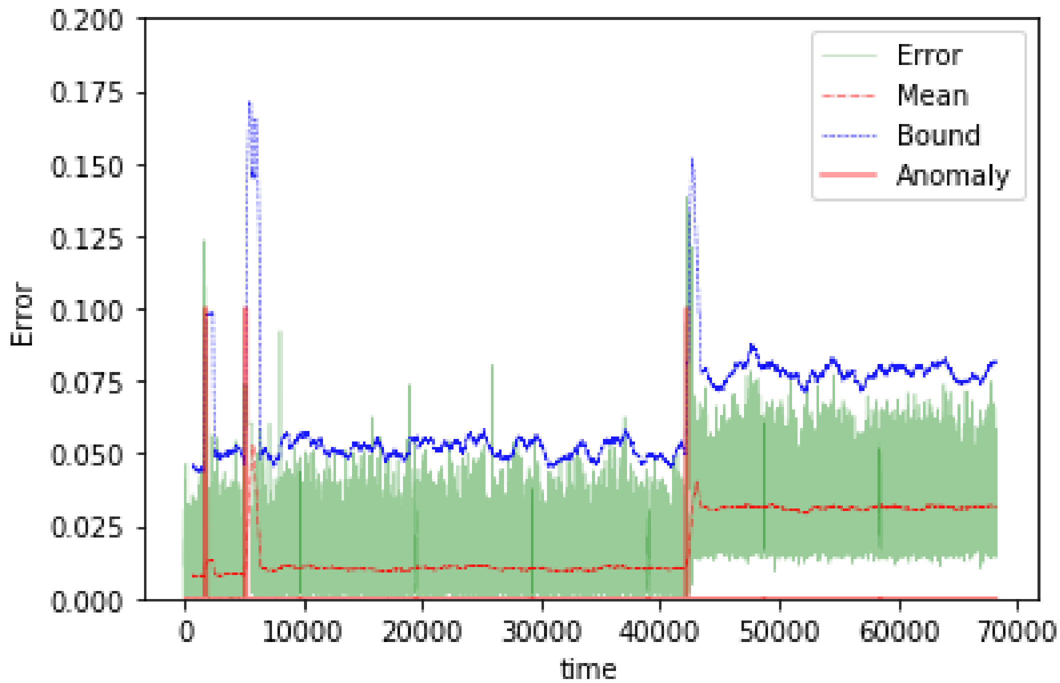

```
Training loss for final epoch is 0.008078020080691204
Validation loss for final epoch is 0.006703652210999281
---------- Beginning tests for nn3_20 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

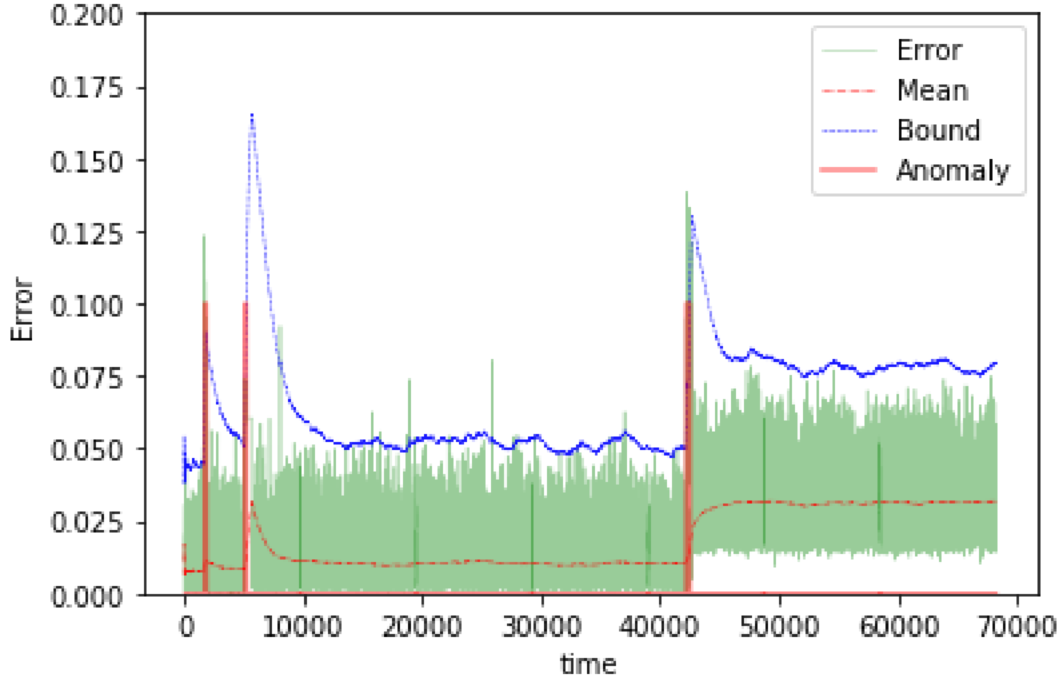

6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14



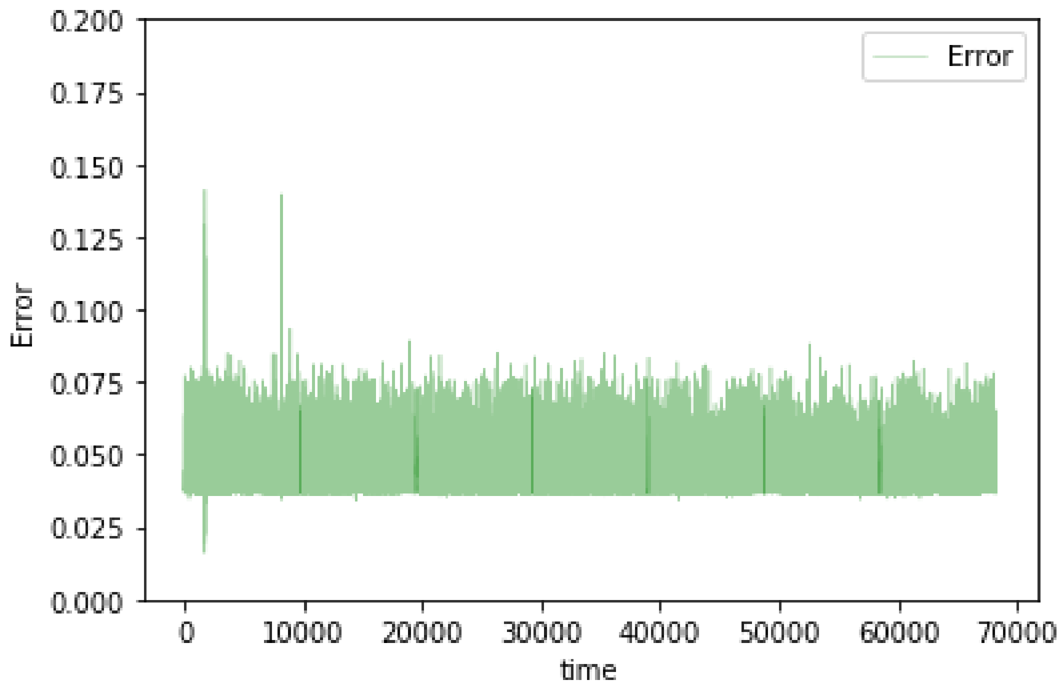
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

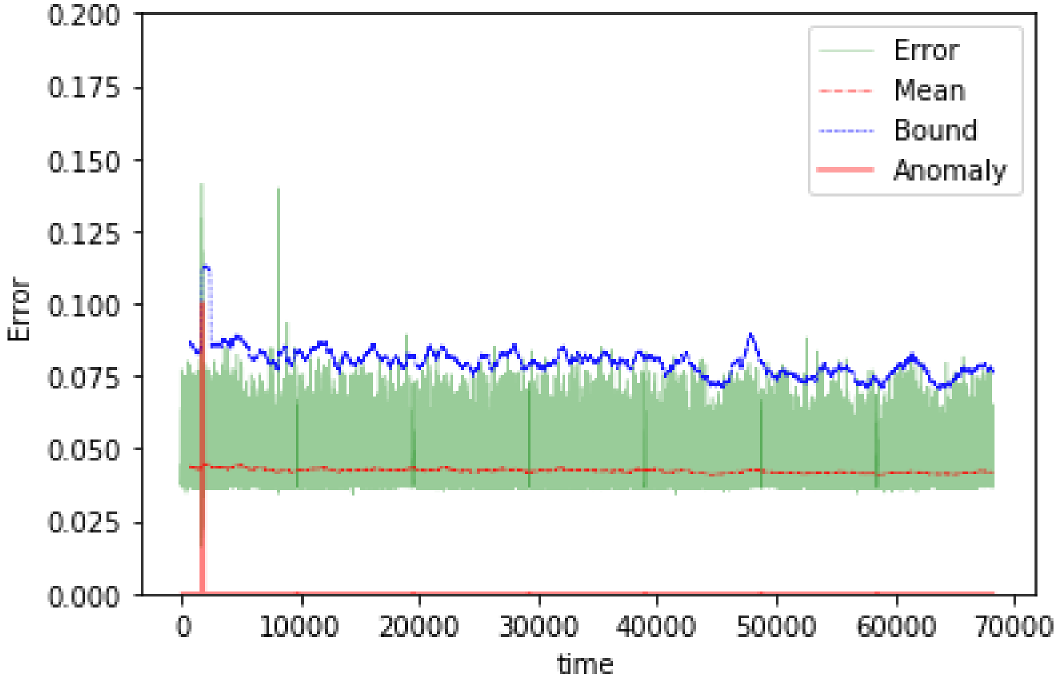


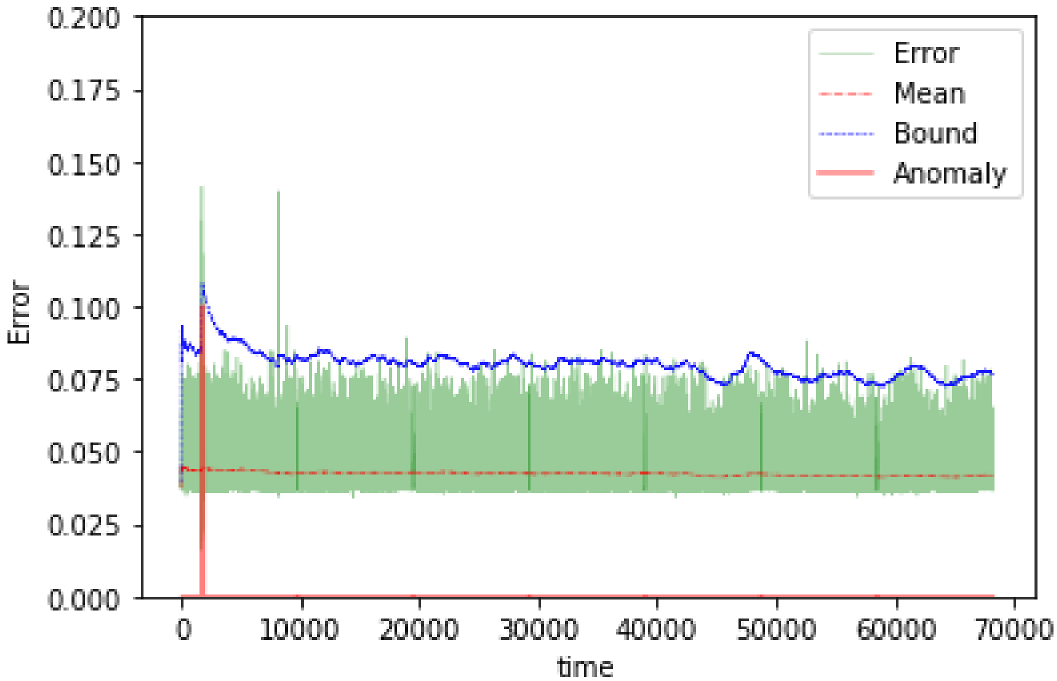
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



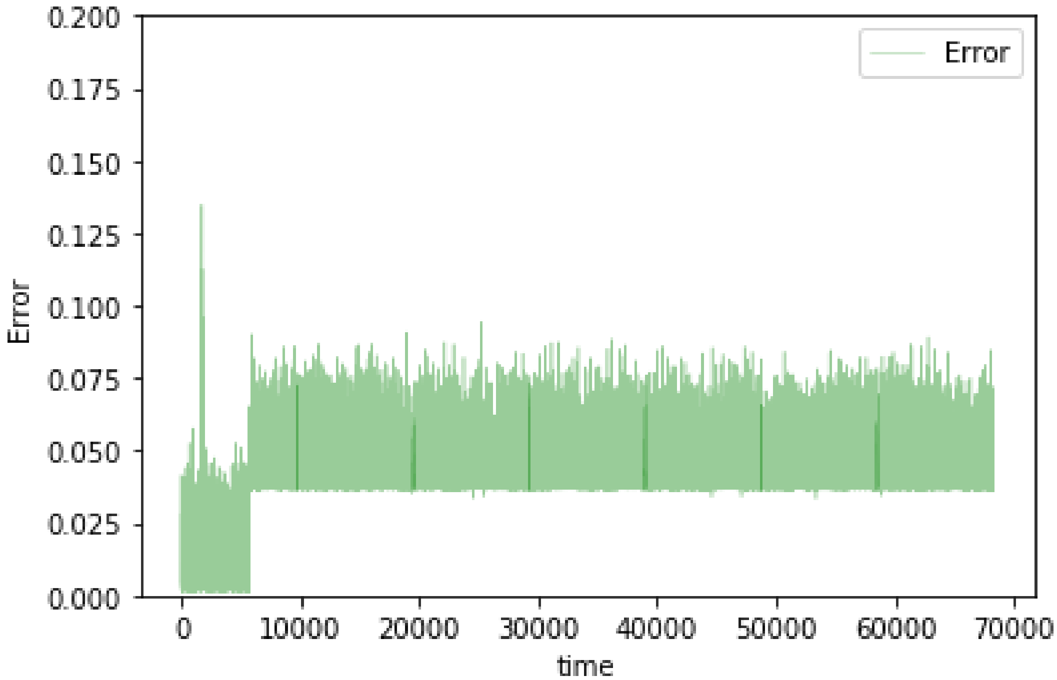

```
The mean error for nn3_20_disk_IO_start_ is 0.018778954764104207 for length 68279
Testing on Avg. load data.
```

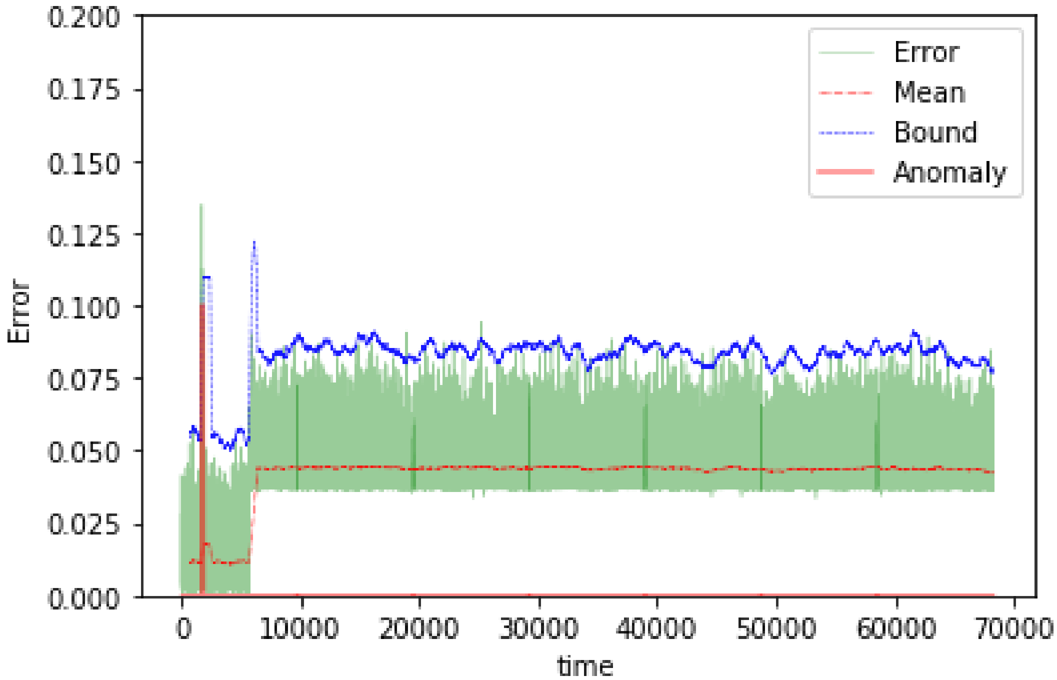


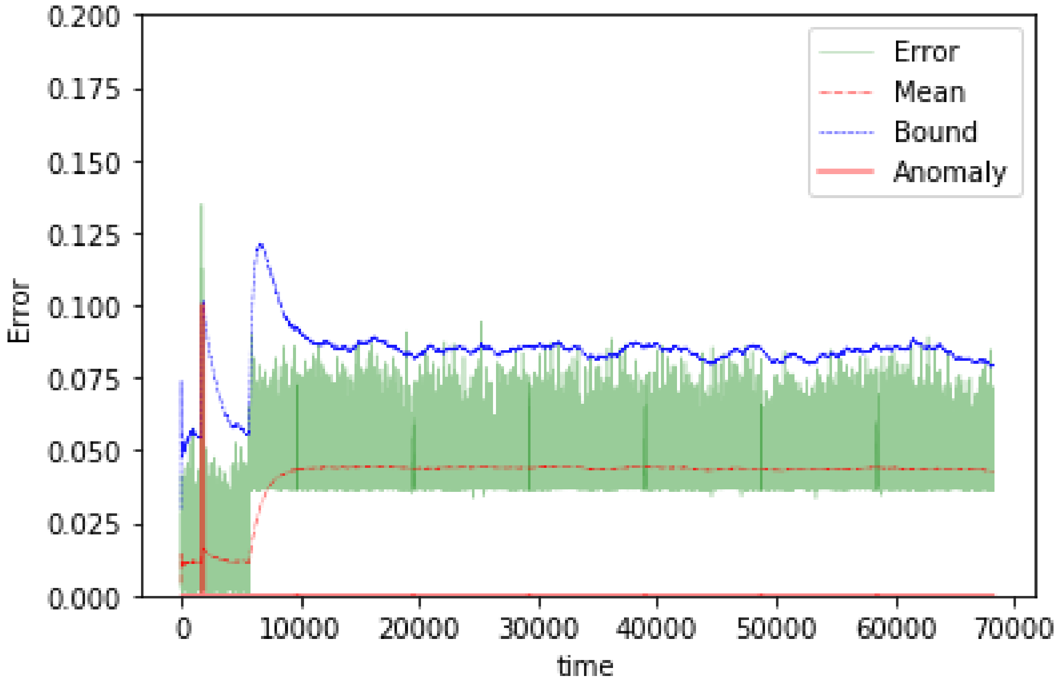

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

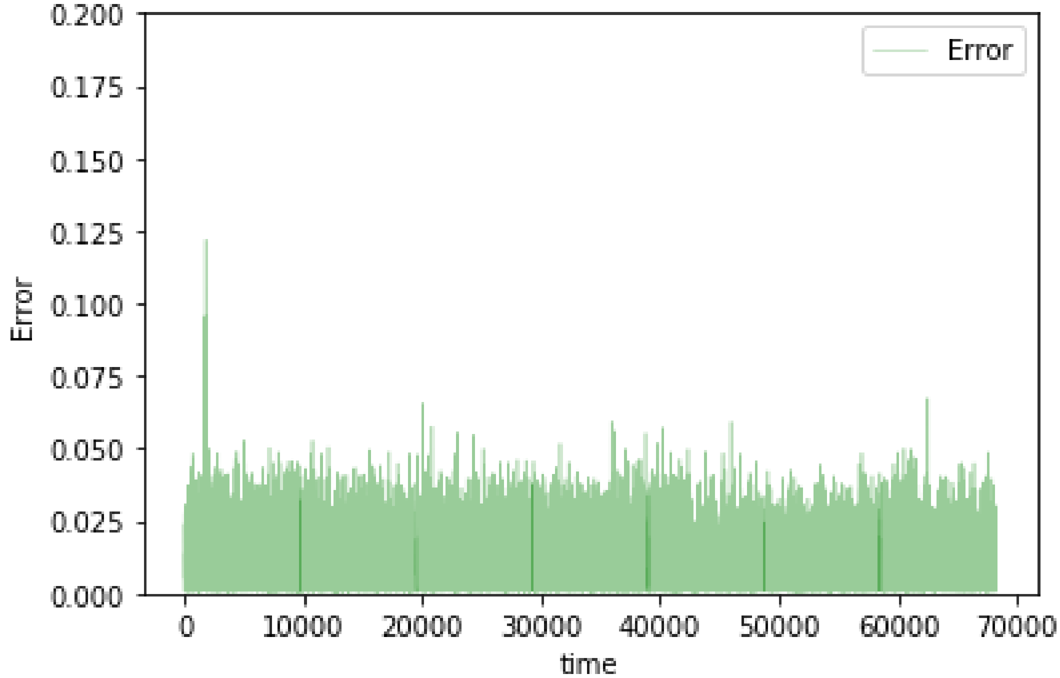
```
The mean error for nn3_20_avg_load_ is 0.04230316349922768 for length 68279
Testing on app change early data.
```

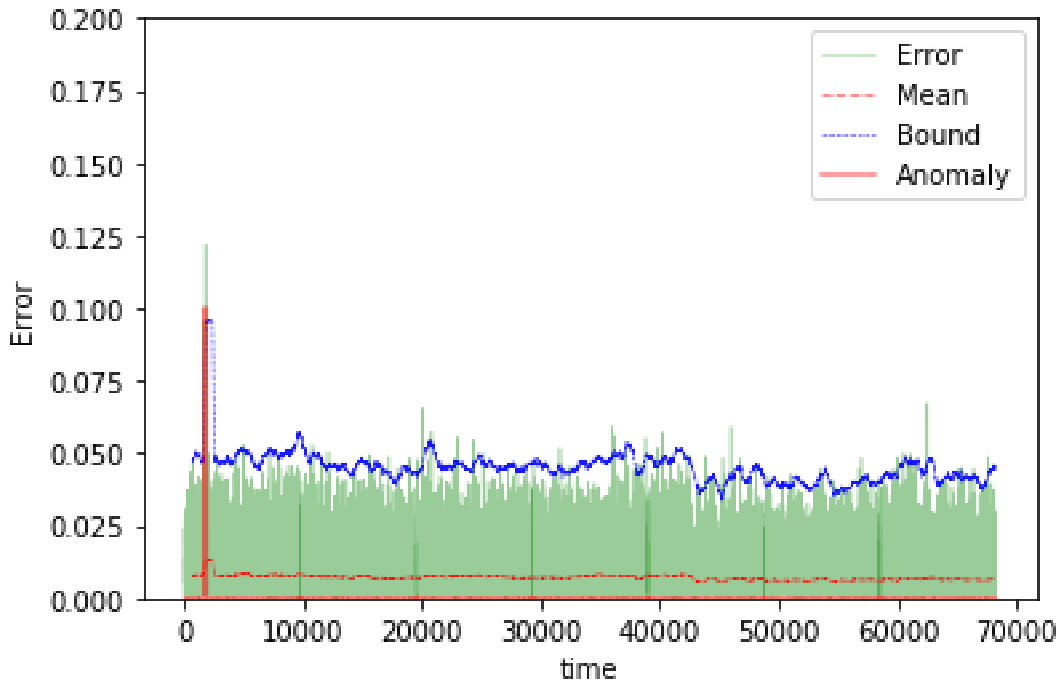


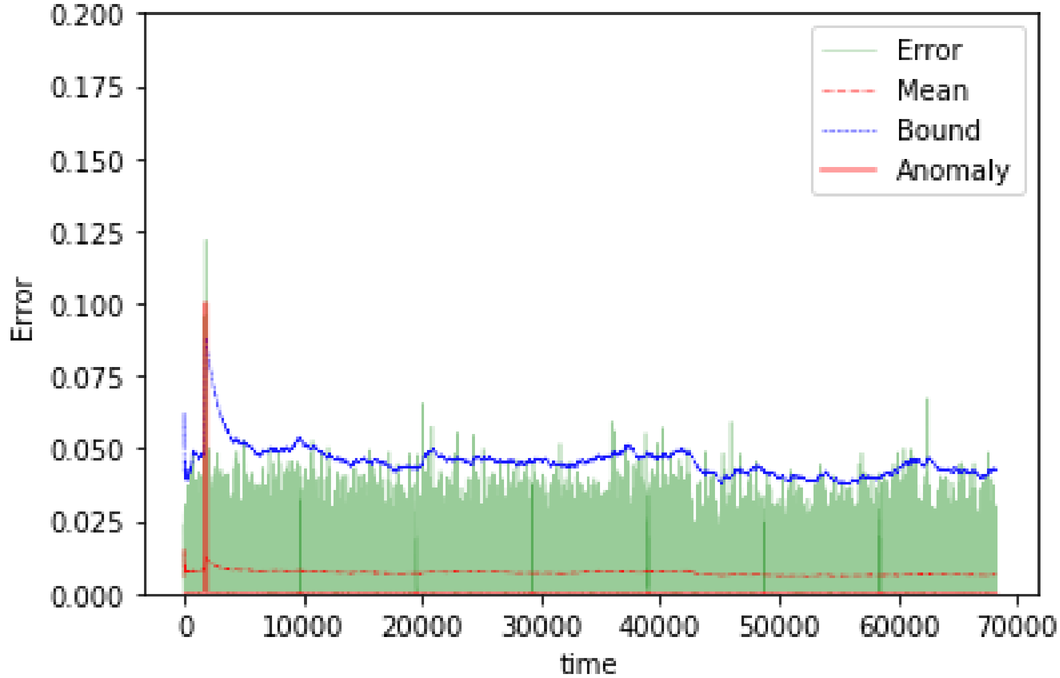





```
The mean error for nn3_20_app_change_early_ is 0.04117207333161687 for length 68279
Testing on Normal data.
```


Error
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

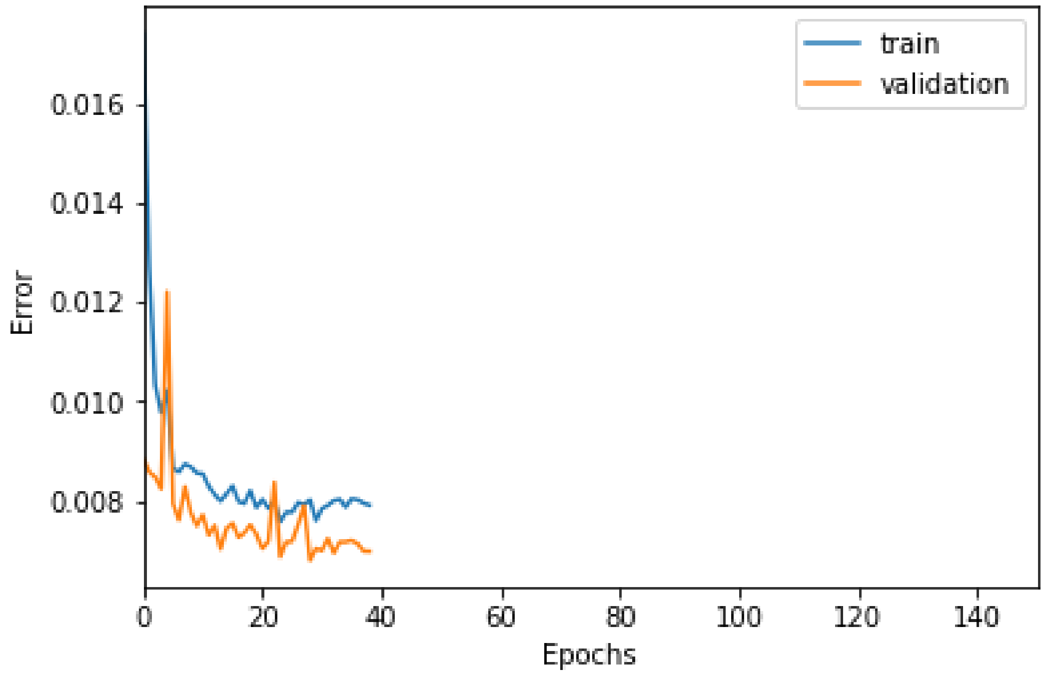

Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000



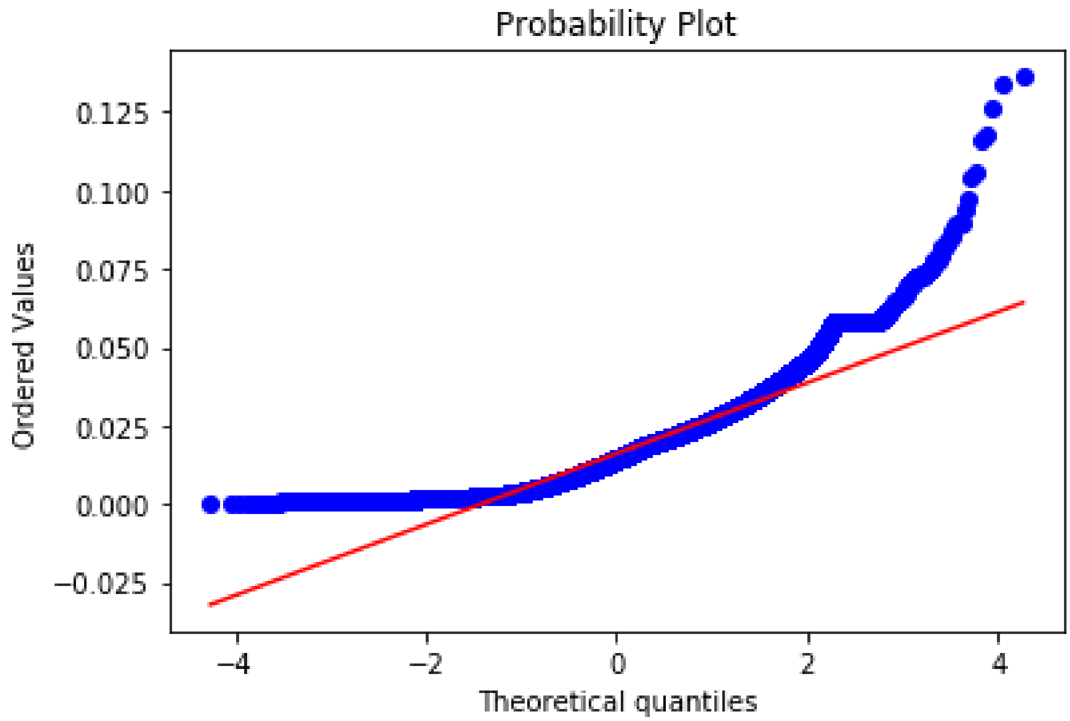

```
The mean error for nn3_20_normal_ is 0.007128392967144289 for length 68279
====================
```

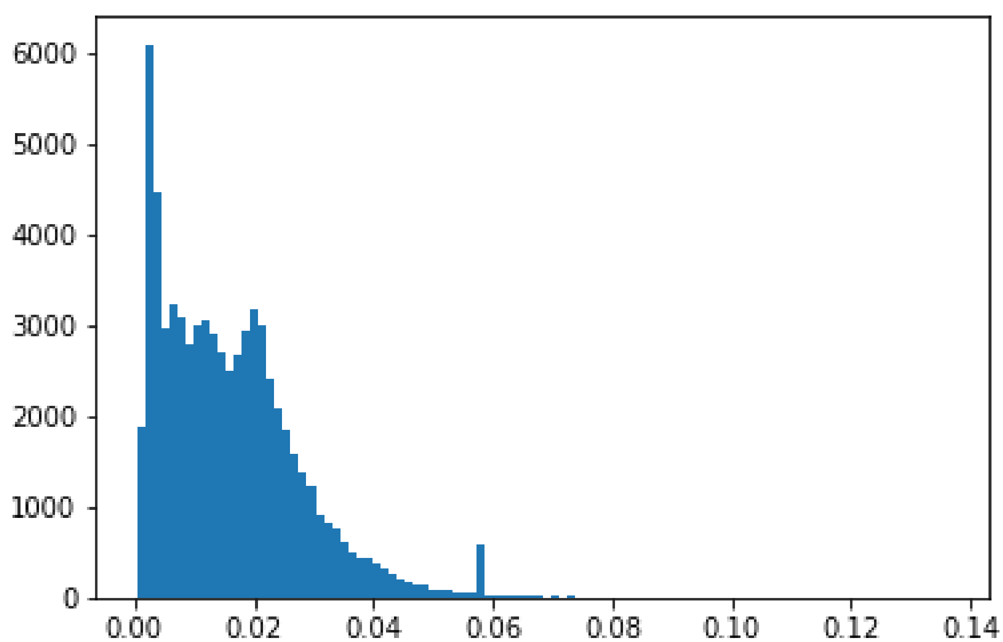
50 steps

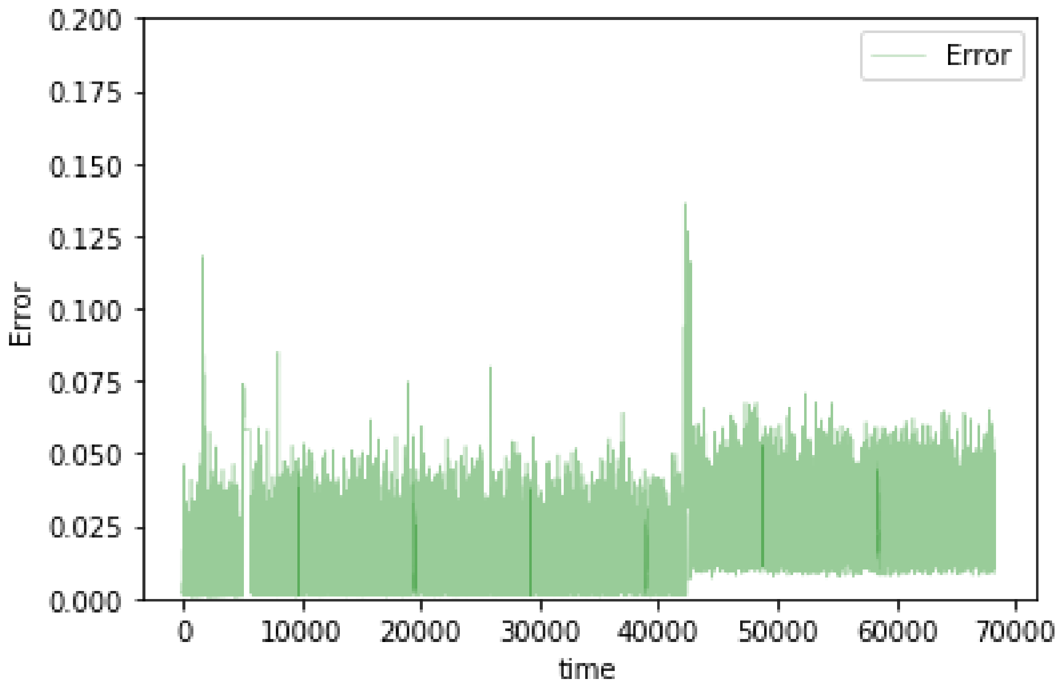
```
In [139]: TIMESTEPS = 50
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS)
          vgen = flat_generator(val_X, TIMESTEPS)
          name = "nn3_50"

In [140]: input_layer = Input(shape=(TIMESTEPS*DIM,))
          hidden = Dense(1000, activation='relu')(input_layer)
          hidden = Dense(500, activation='relu')(hidden)
          hidden = Dense(100, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [141]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [142]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```



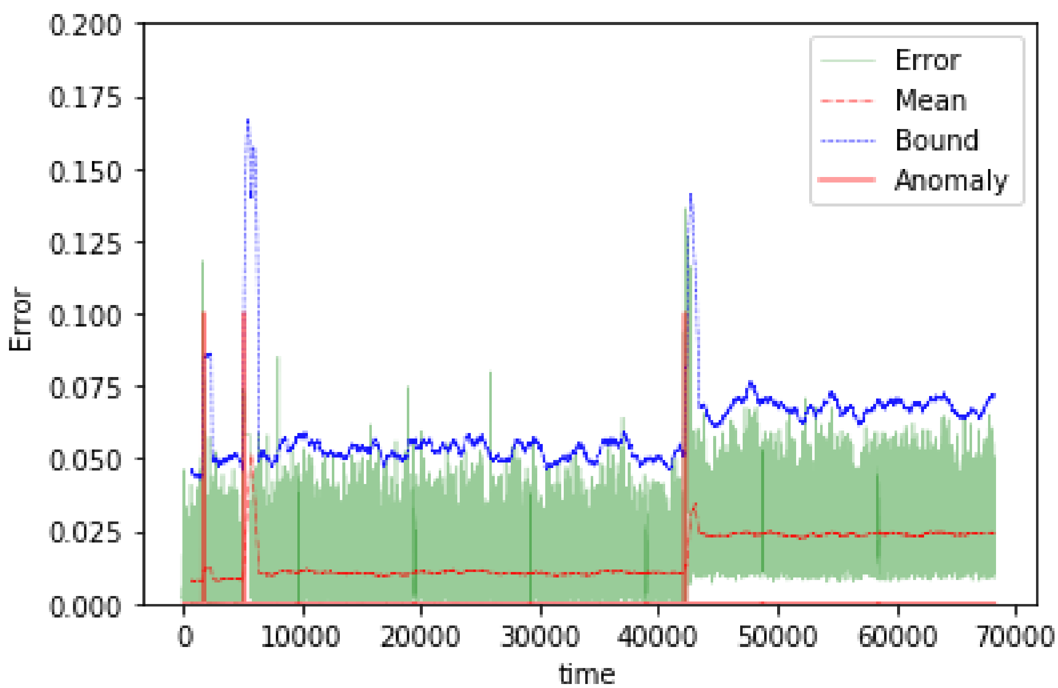

```
Training loss for final epoch is 0.007911833339370787
Validation loss for final epoch is 0.0069839807493845005
---------- Beginning tests for nn3_50 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

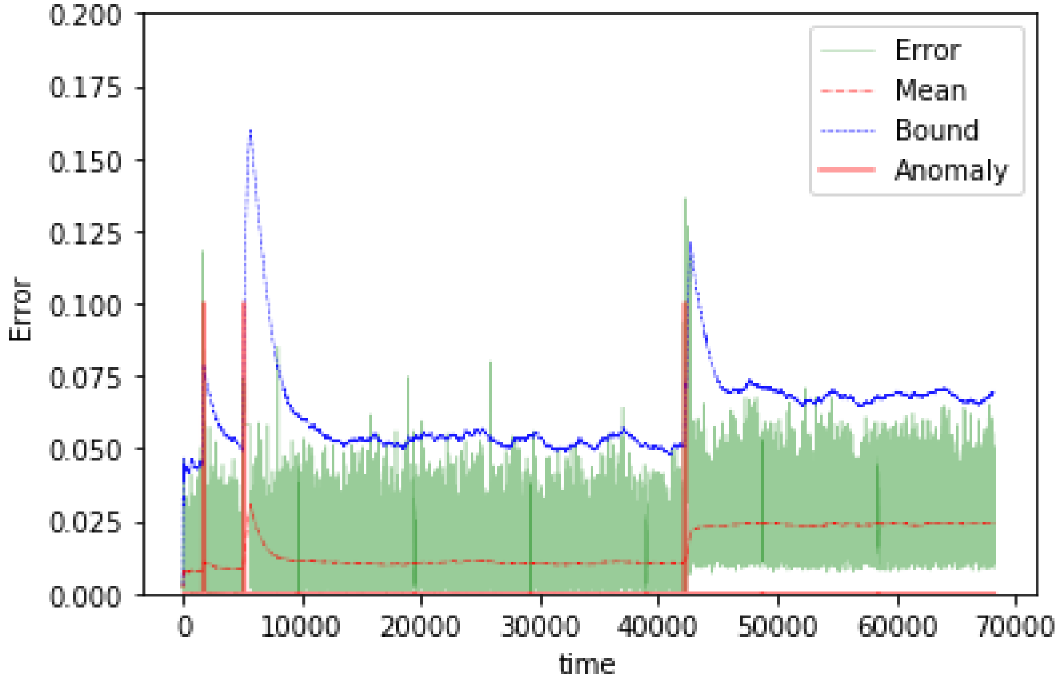

6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14



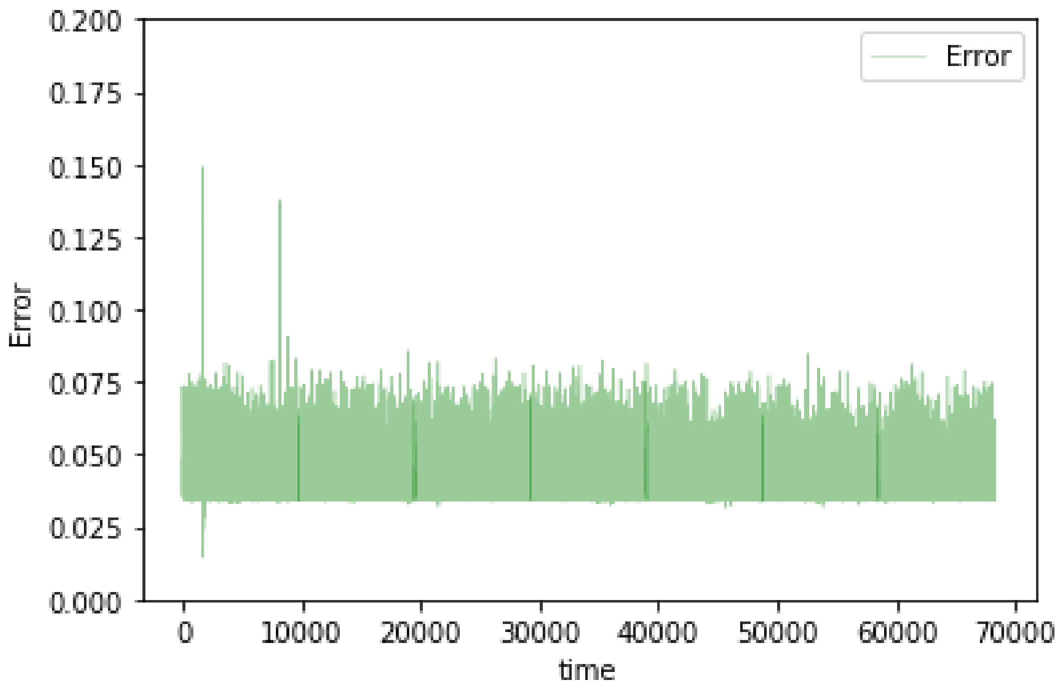
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

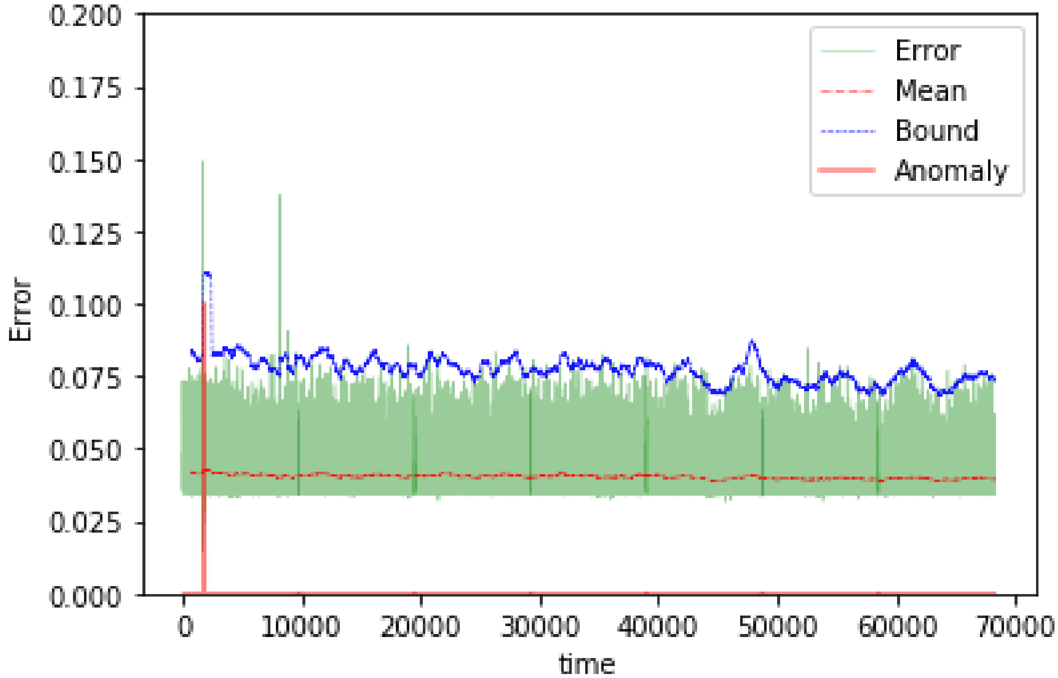


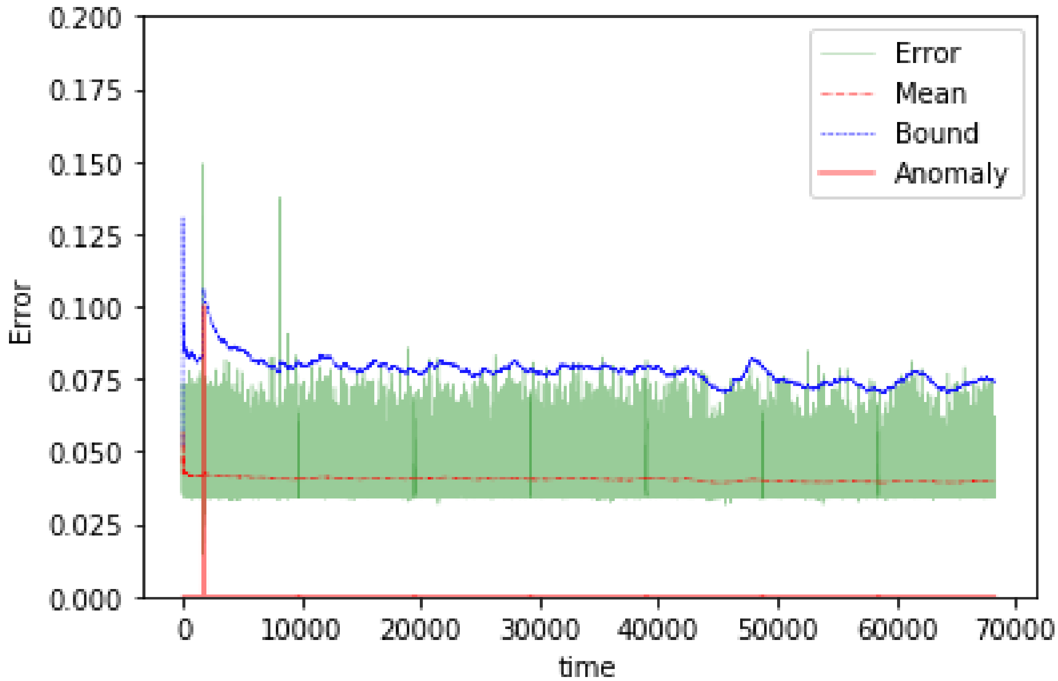
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



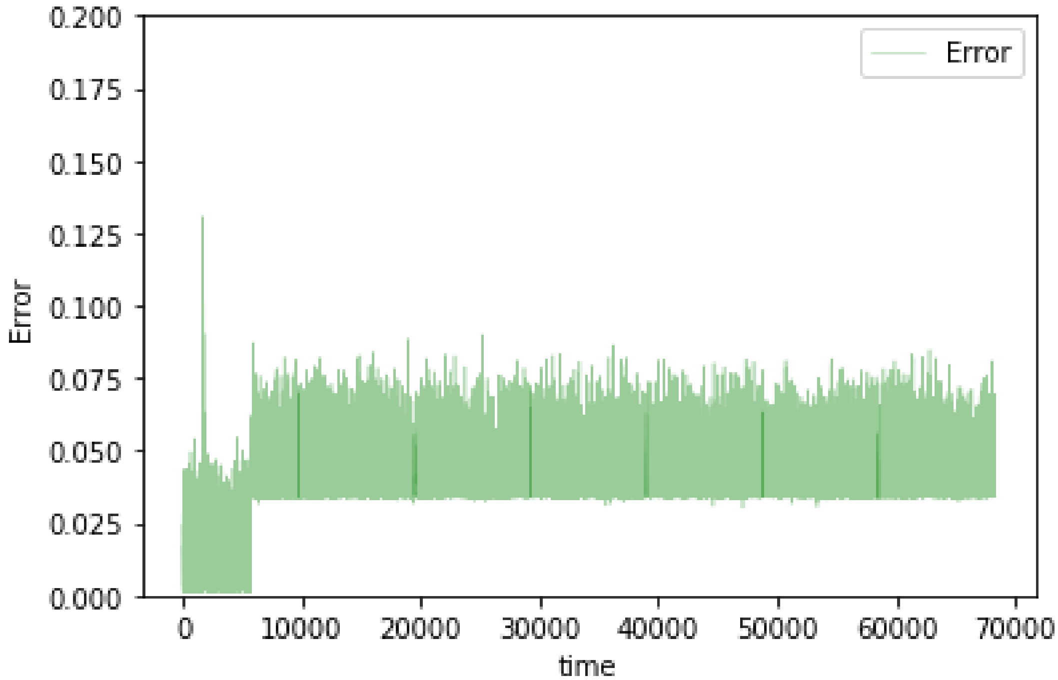

```
The mean error for nn3_50_disk_IO_start_ is 0.016049520924250646 for length 68249
Testing on Avg. load data.
```

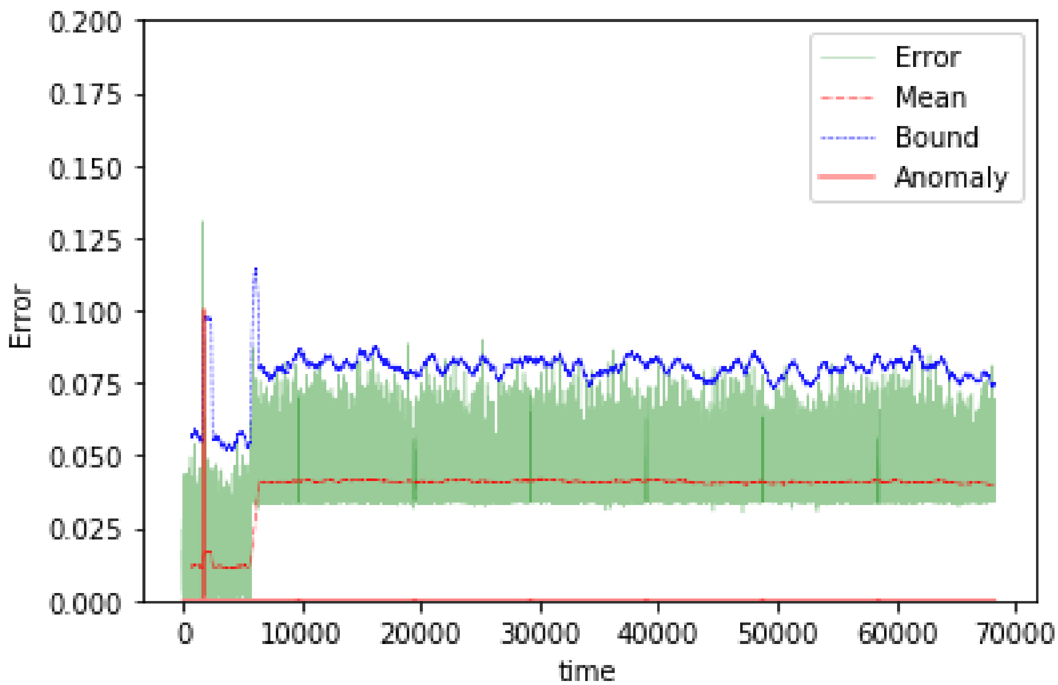


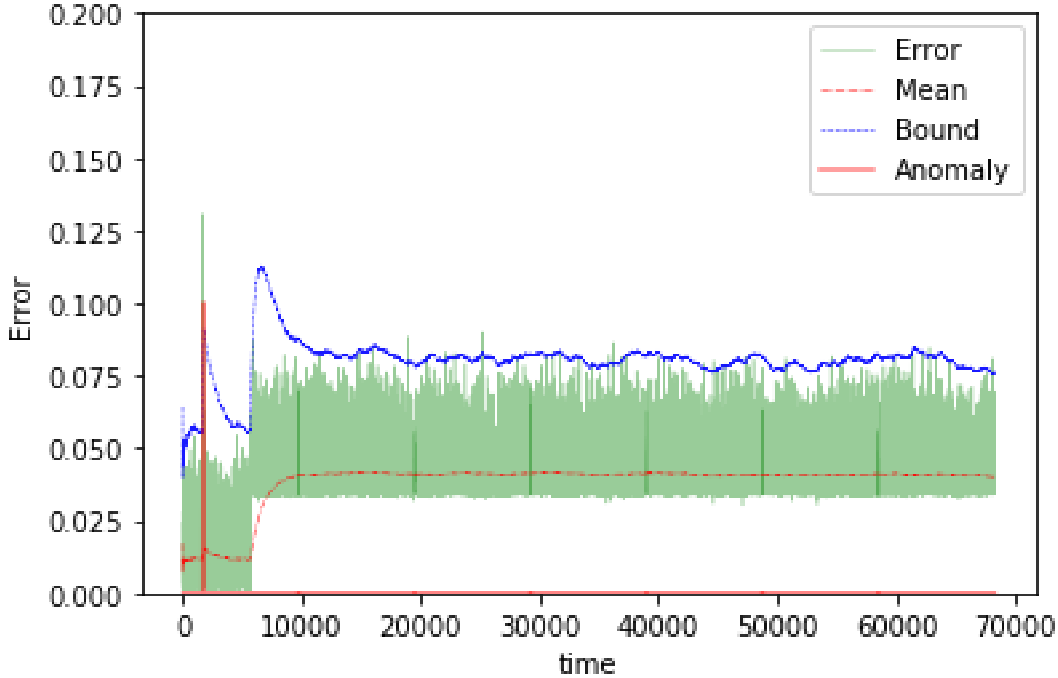

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

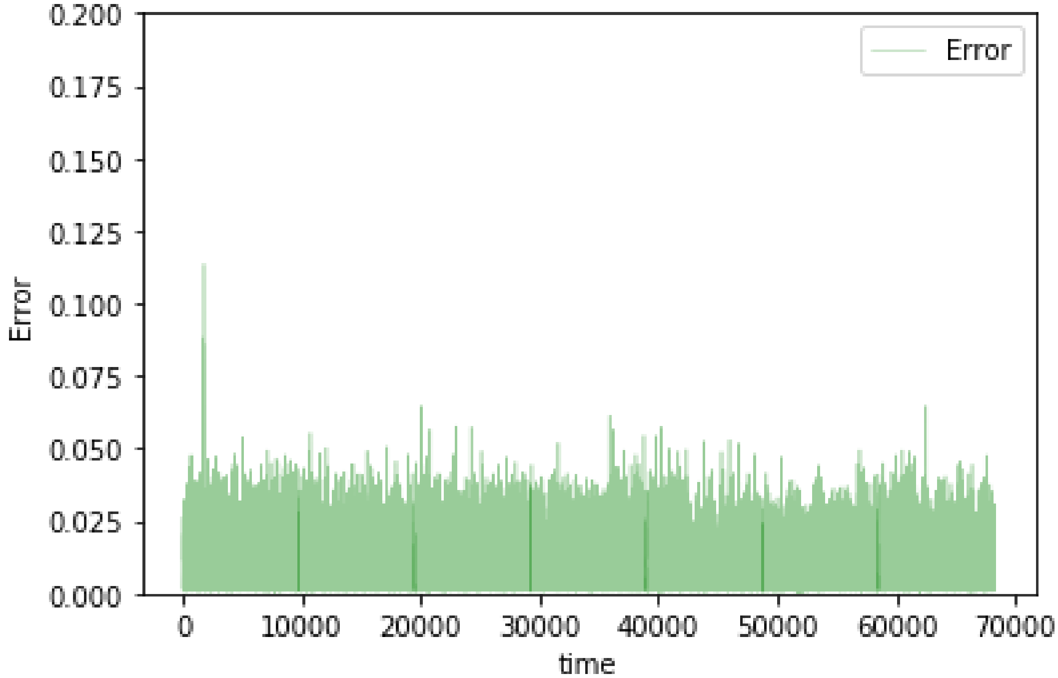
```
The mean error for nn3_50_avg_load_ is 0.040348633778803183 for length 68249
Testing on app change early data.
```



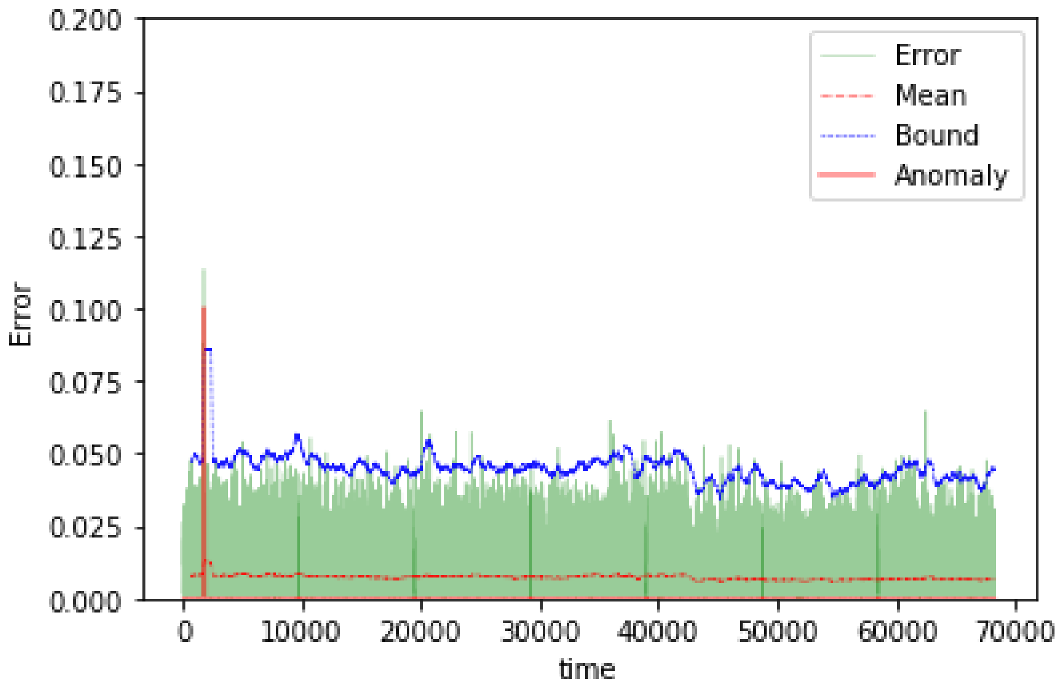




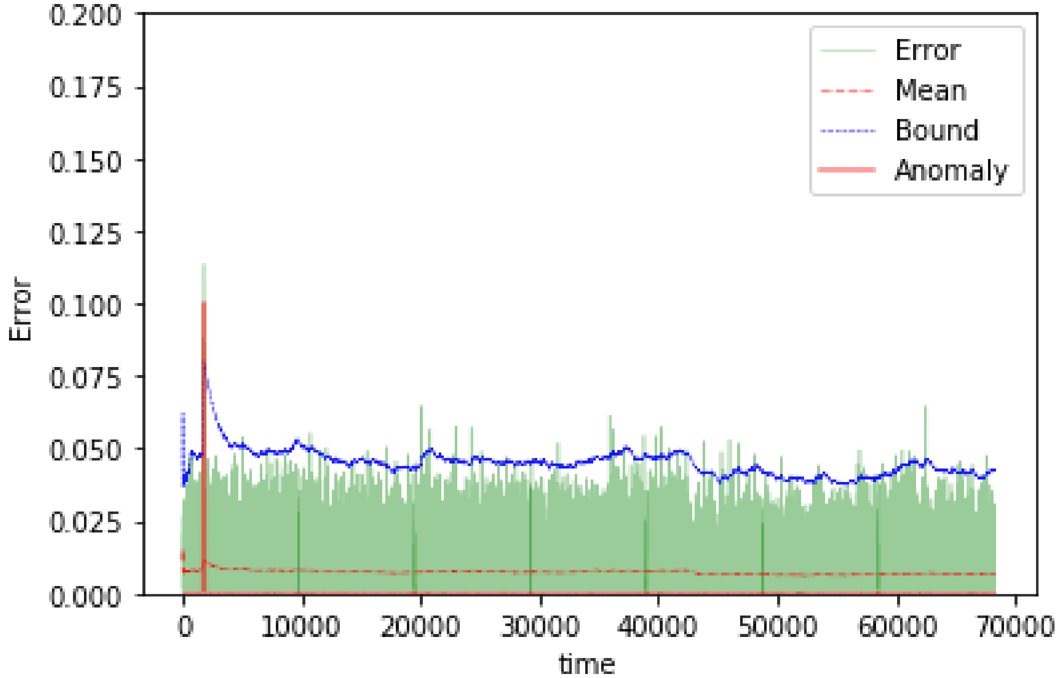

```
The mean error for nn3_50_app_change_early_ is 0.03857272347797367 for length 68249
Testing on Normal data.
```



Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



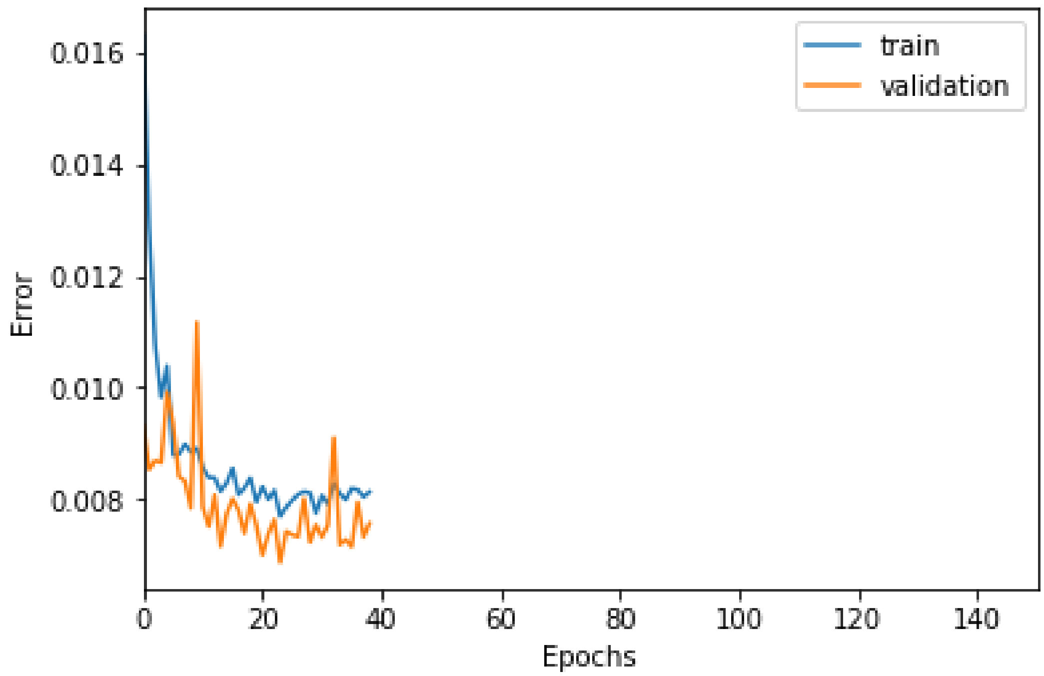
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



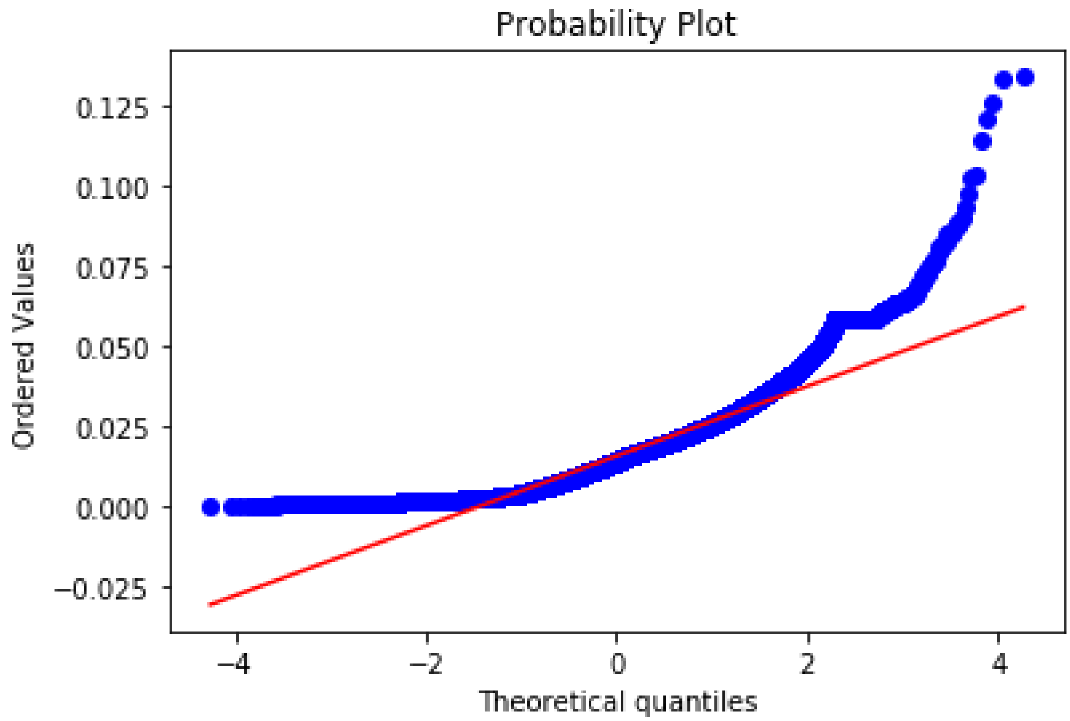

```
The mean error for nn3_50_normal_ is 0.007307085613756603 for length 68249
====================
```

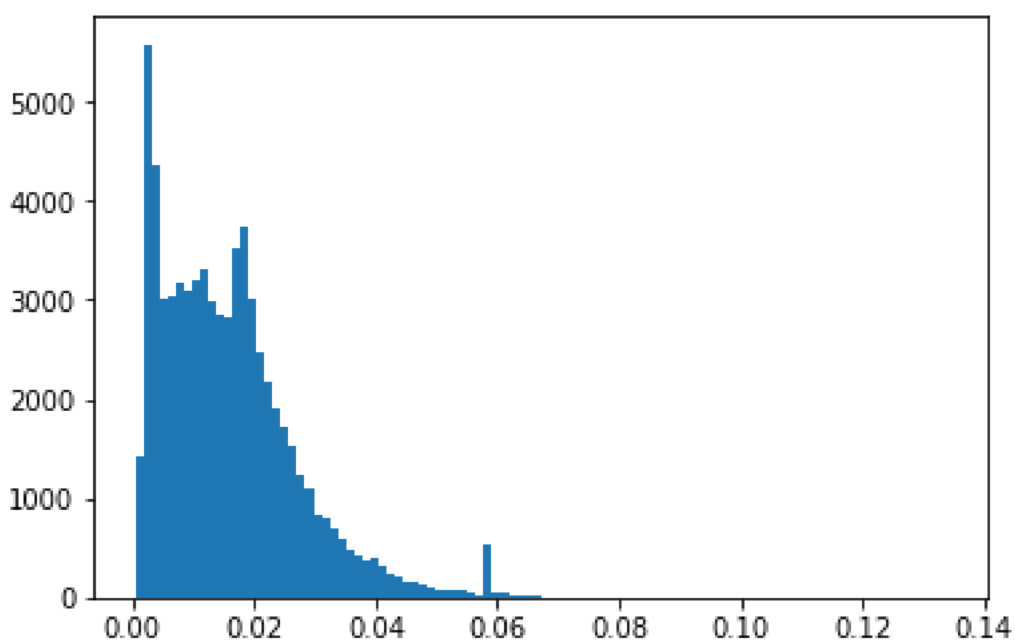
100 steps

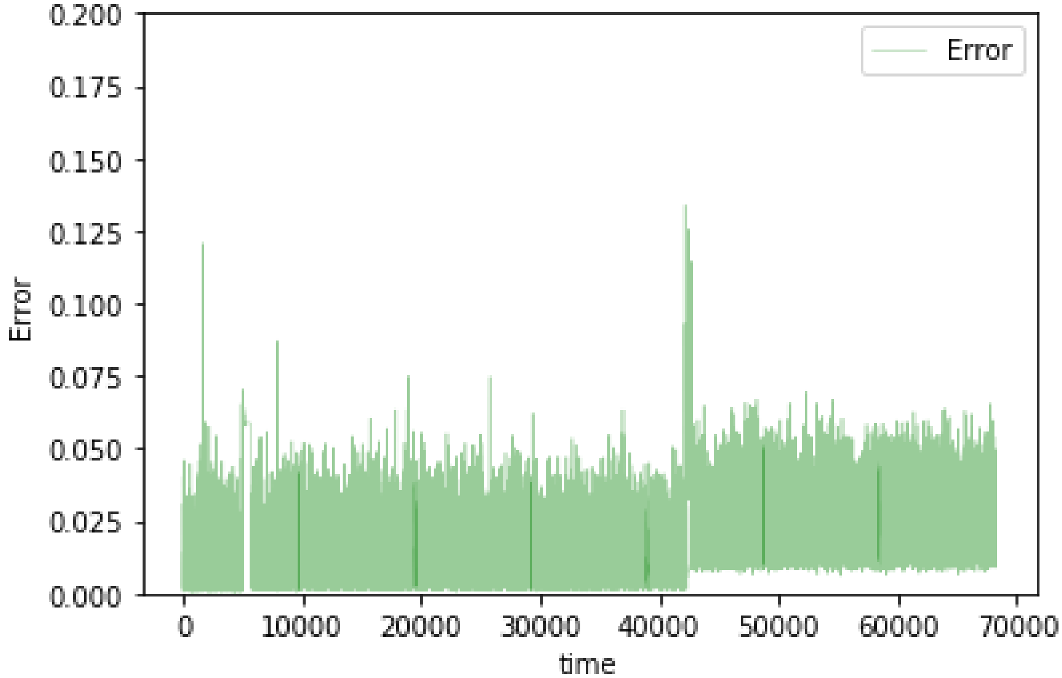
```
In [143]: TIMESTEPS = 100
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS)
          vgen = flat_generator(val_X, TIMESTEPS)
          name = "nn3_100"

In [144]: input_layer = Input(shape=(TIMESTEPS*DIM,))
          hidden = Dense(1000, activation='relu')(input_layer)
          hidden = Dense(500, activation='relu')(hidden)
          hidden = Dense(100, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [145]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [146]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```

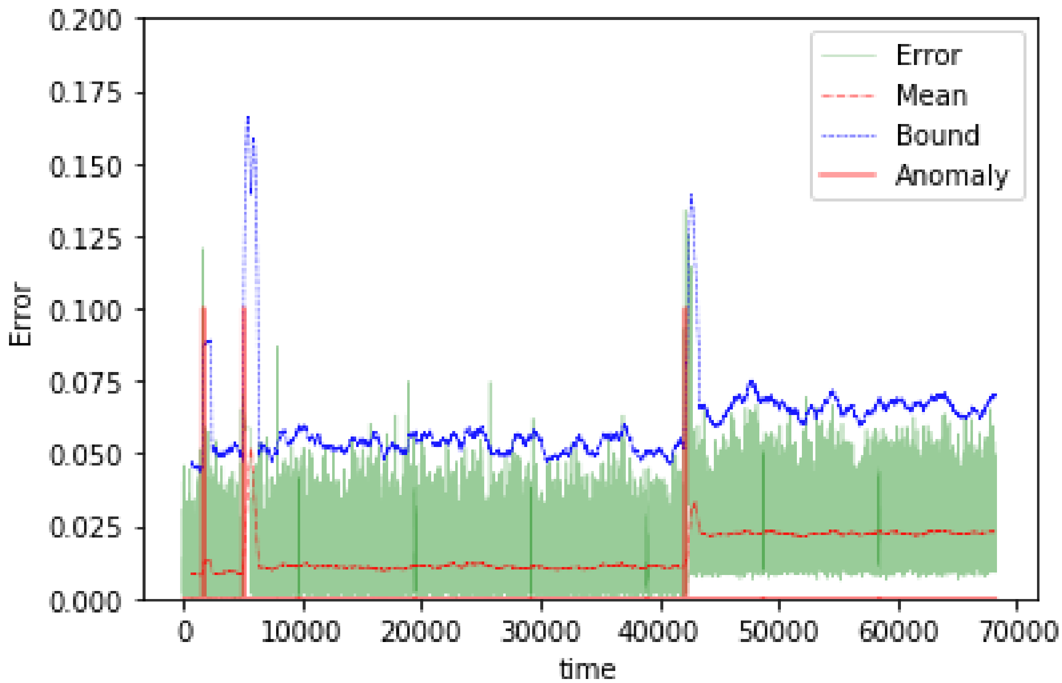


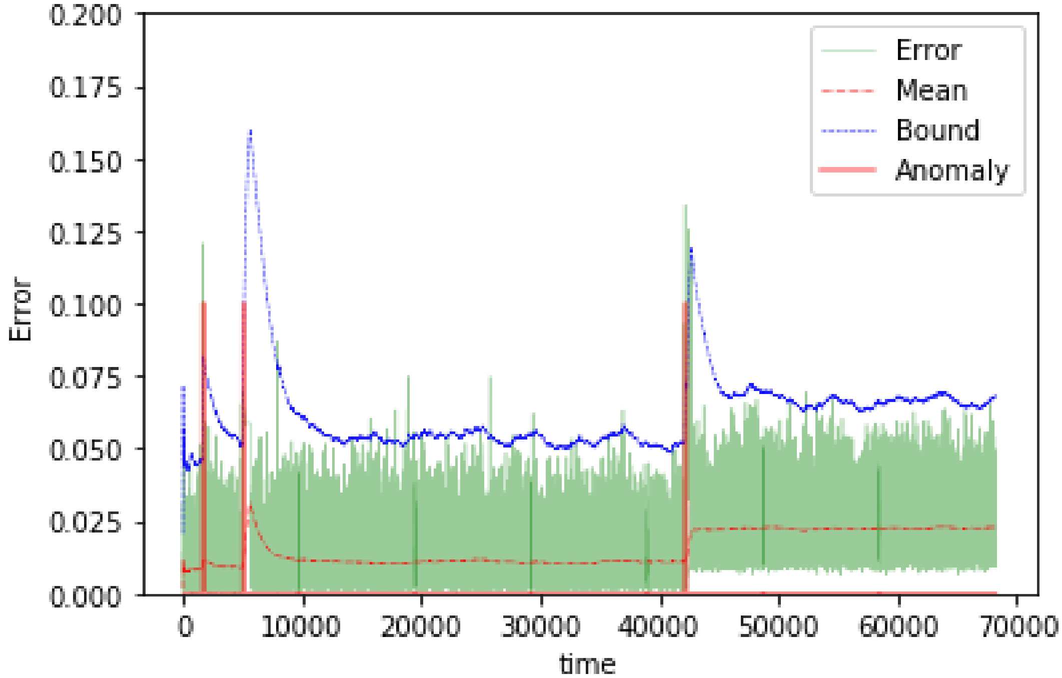

```
Training loss for final epoch is 0.008136635301692877
Validation loss for final epoch is 0.0075755349539685995
---------- Beginning tests for nn3_100 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

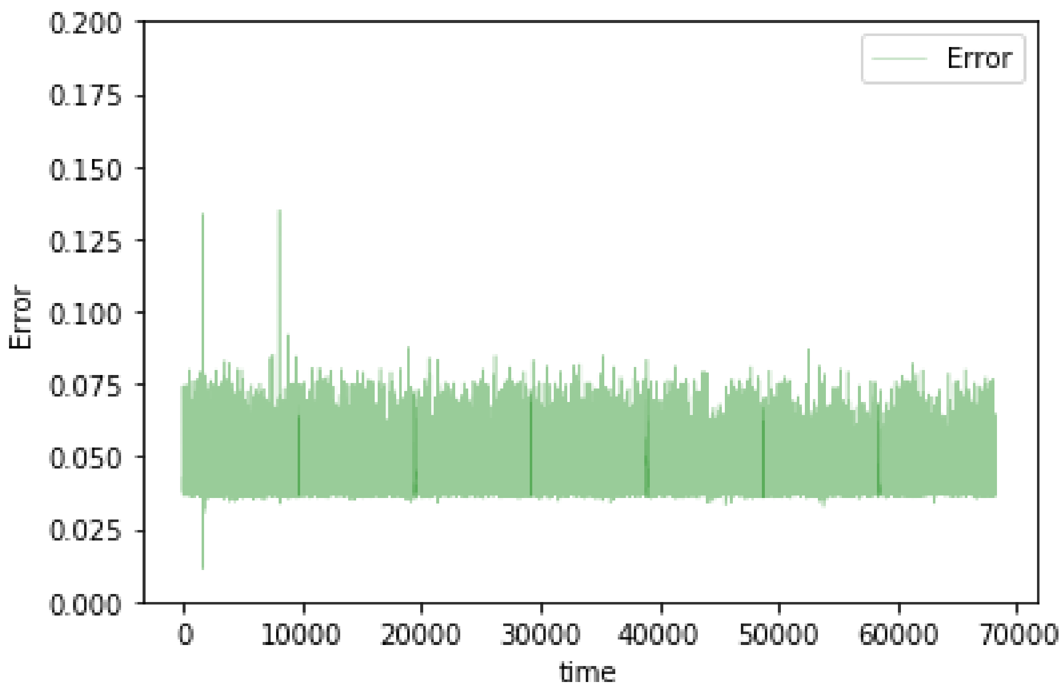

5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

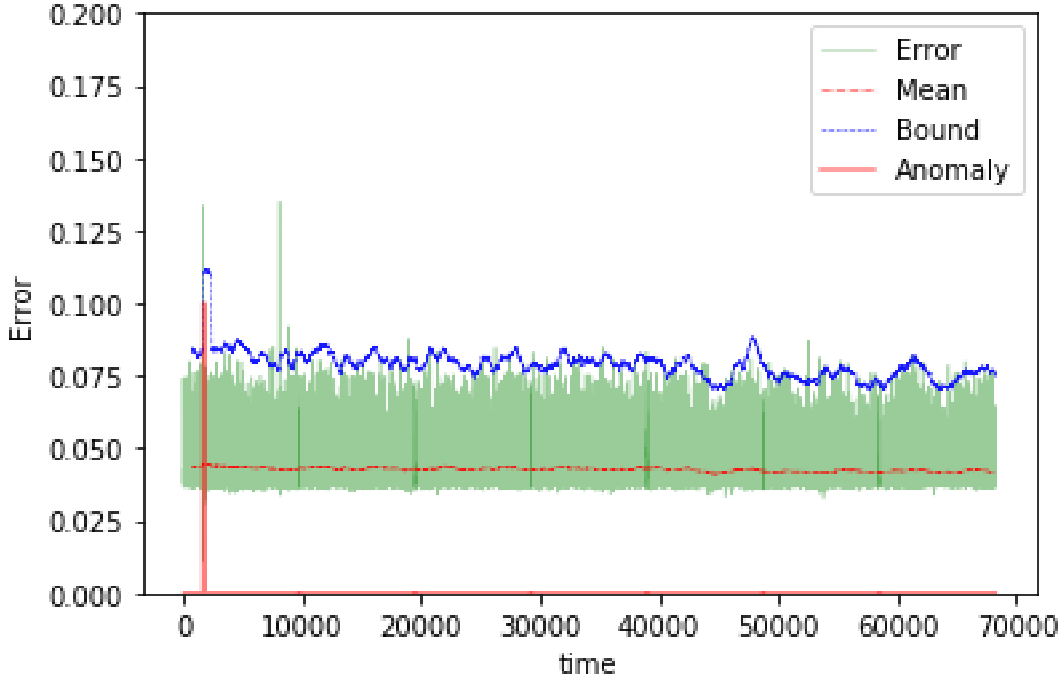

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

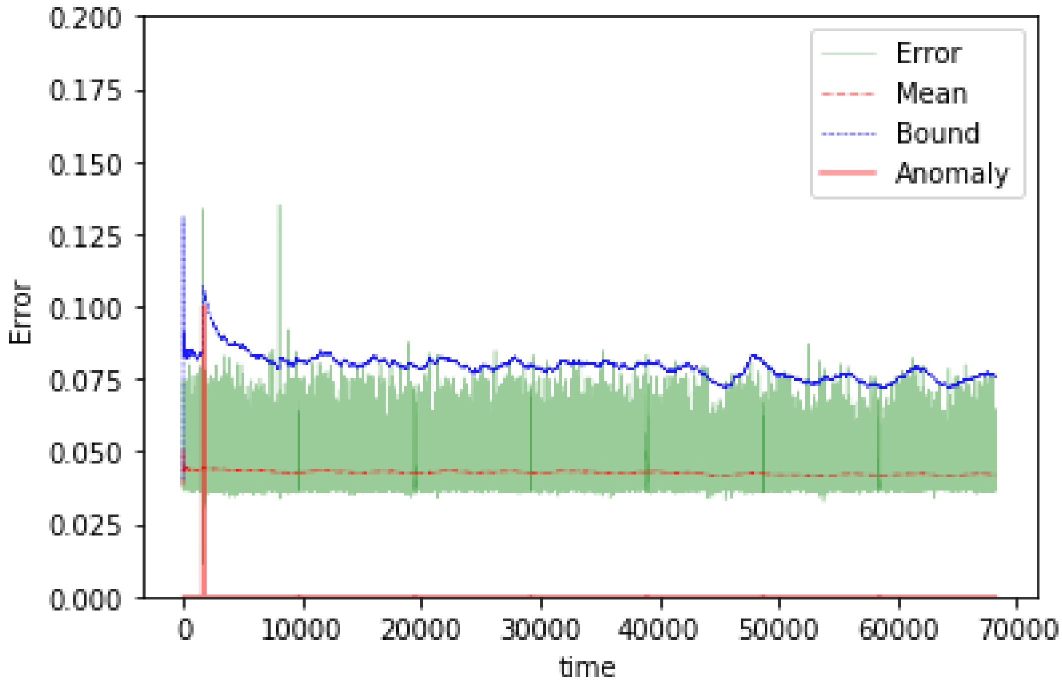

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



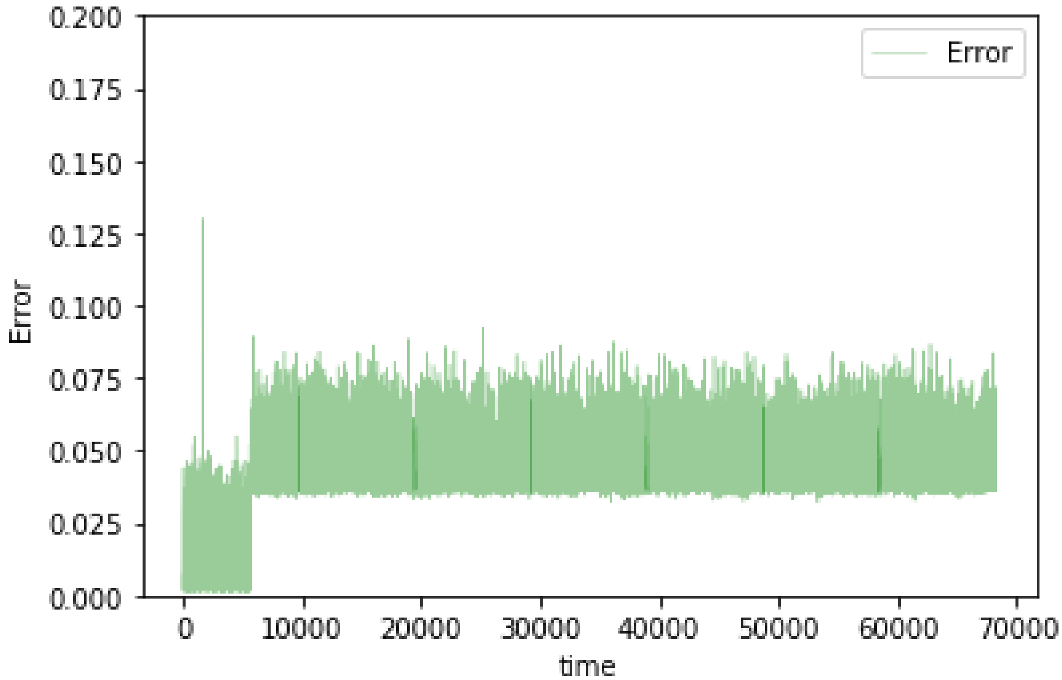

```
The mean error for nn3_100_disk_IO_start_ is 0.01575876378725262 for length 68199
Testing on Avg. load data.
```

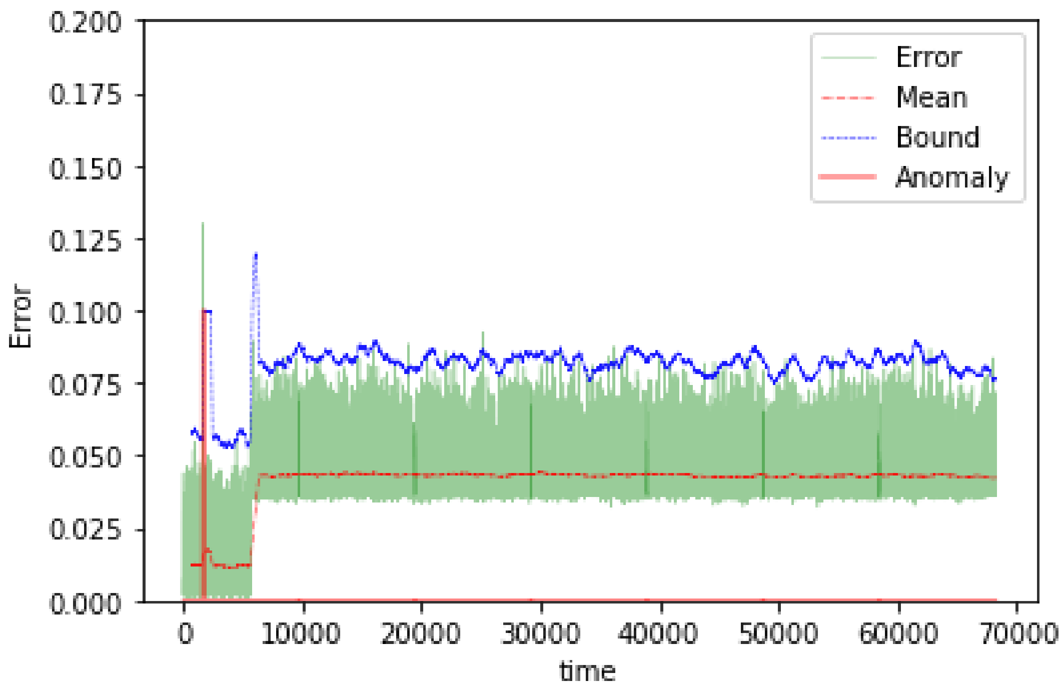


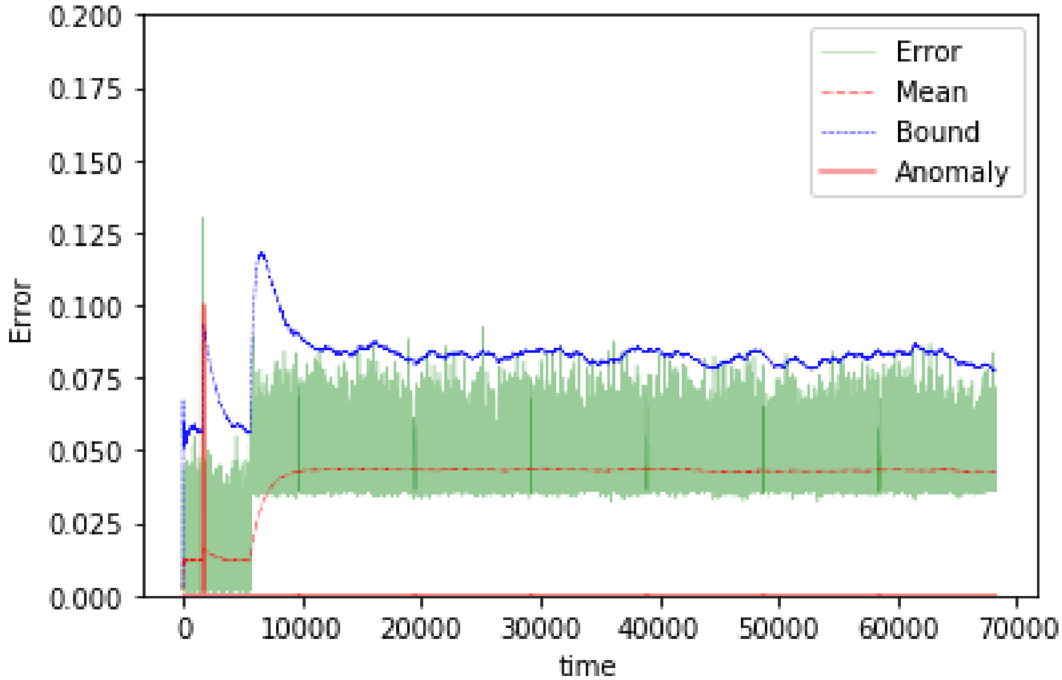

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

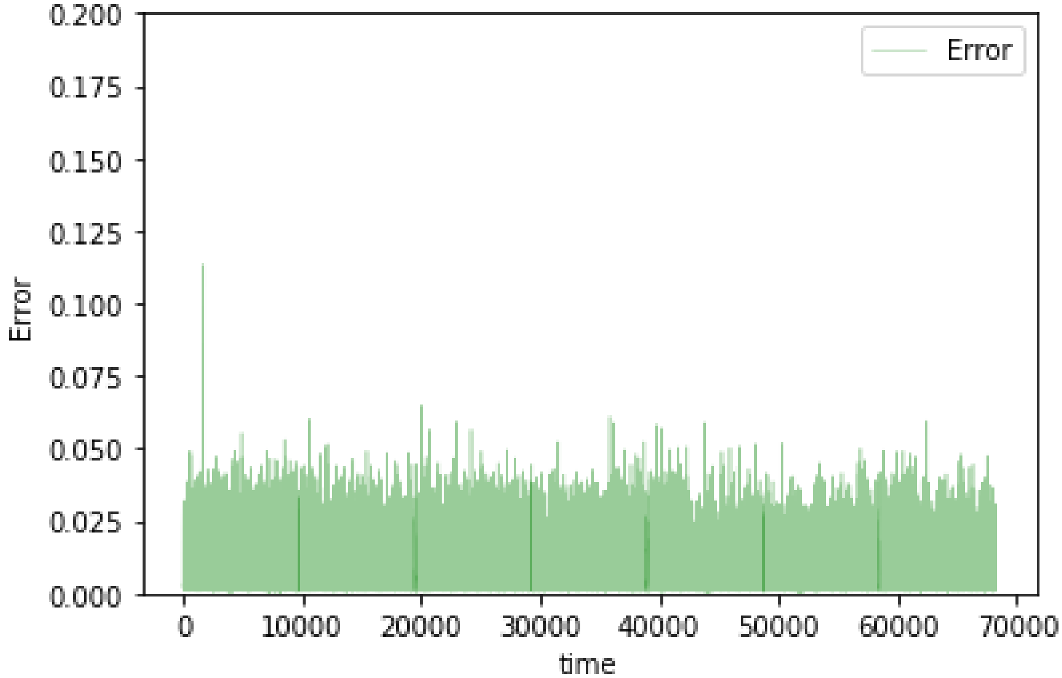
```
The mean error for nn3_100_avg_load_ is 0.04266485634979519 for length 68199
Testing on app change early data.
```

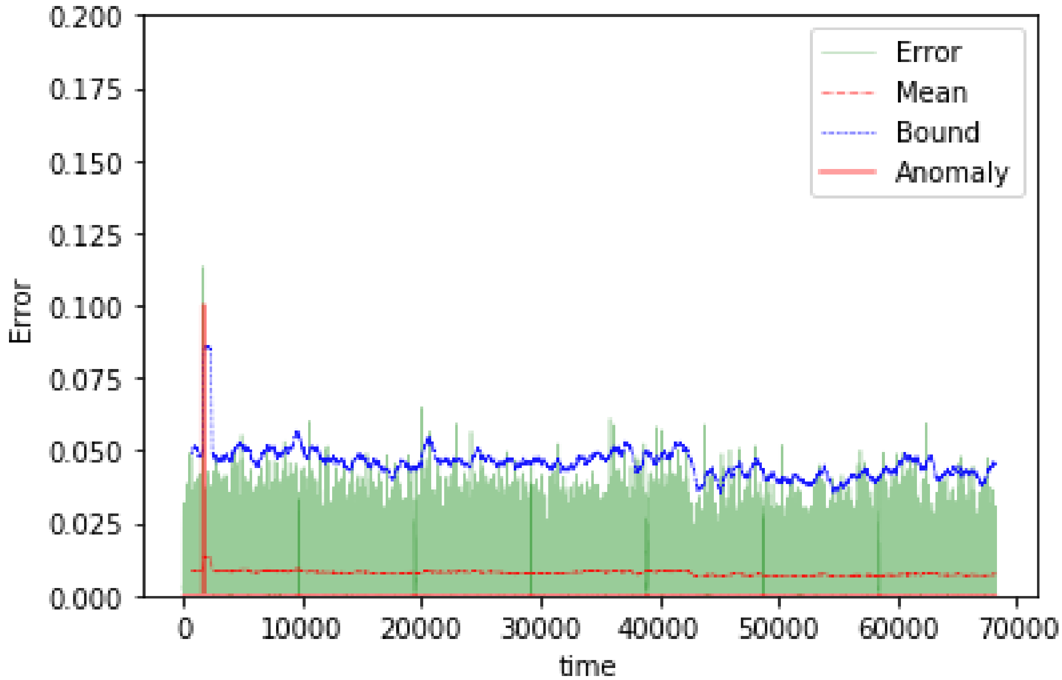


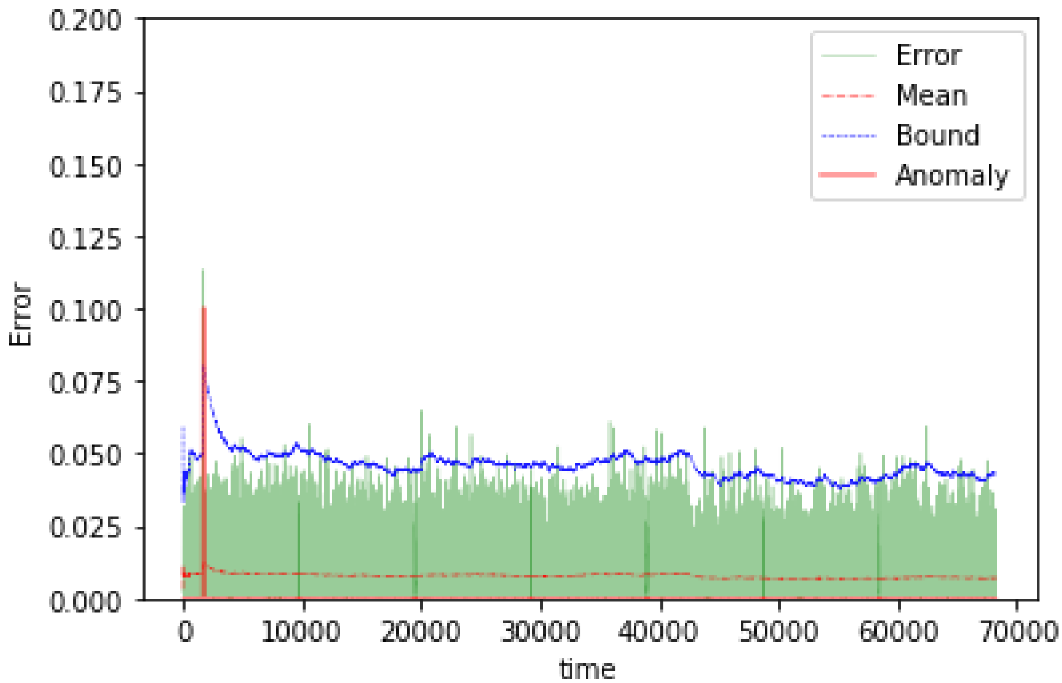





```
The mean error for nn3_100_app_change_early_ is 0.0406628716476895 for length 68199
Testing on Normal data.
```


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
Error
0
10000
20000
30000
40000
50000
60000
70000
time

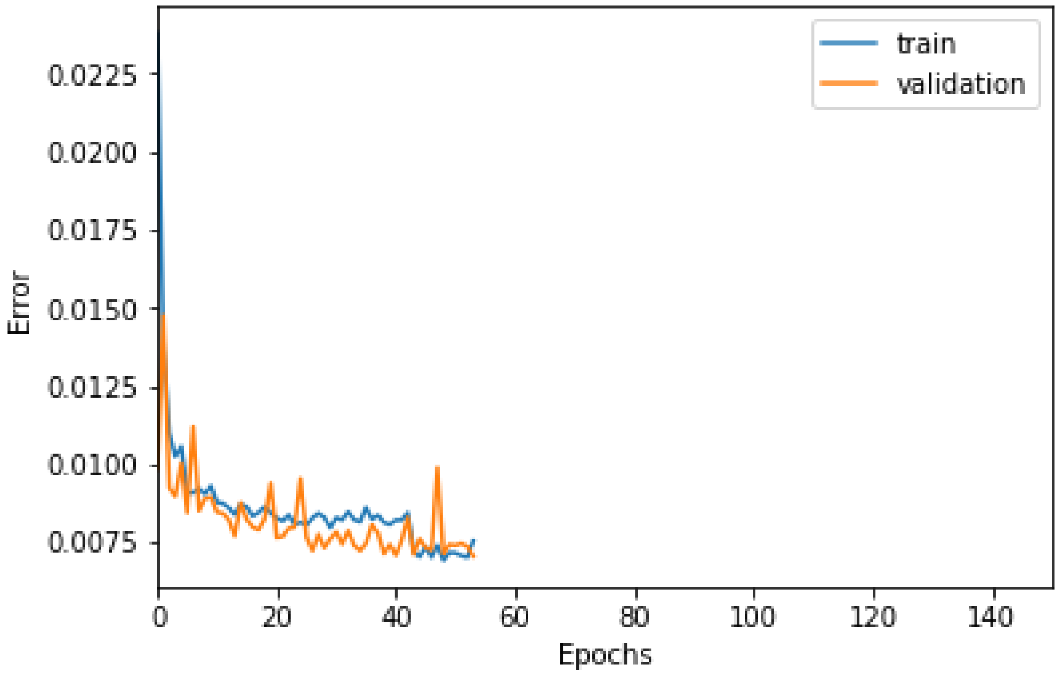

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
Error
Mean
Bound
Anomaly
0
10000
20000
30000
40000
50000
60000
70000
time



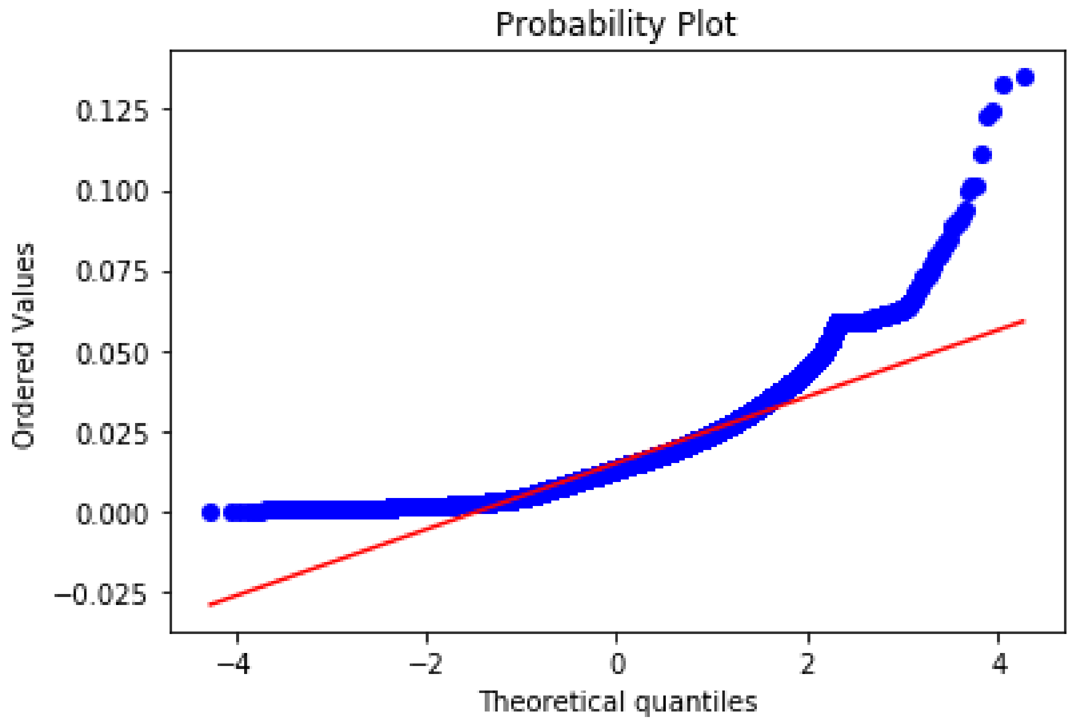

```
The mean error for nn3_100_normal_ is 0.007776899876897526 for length 68199
====================
```

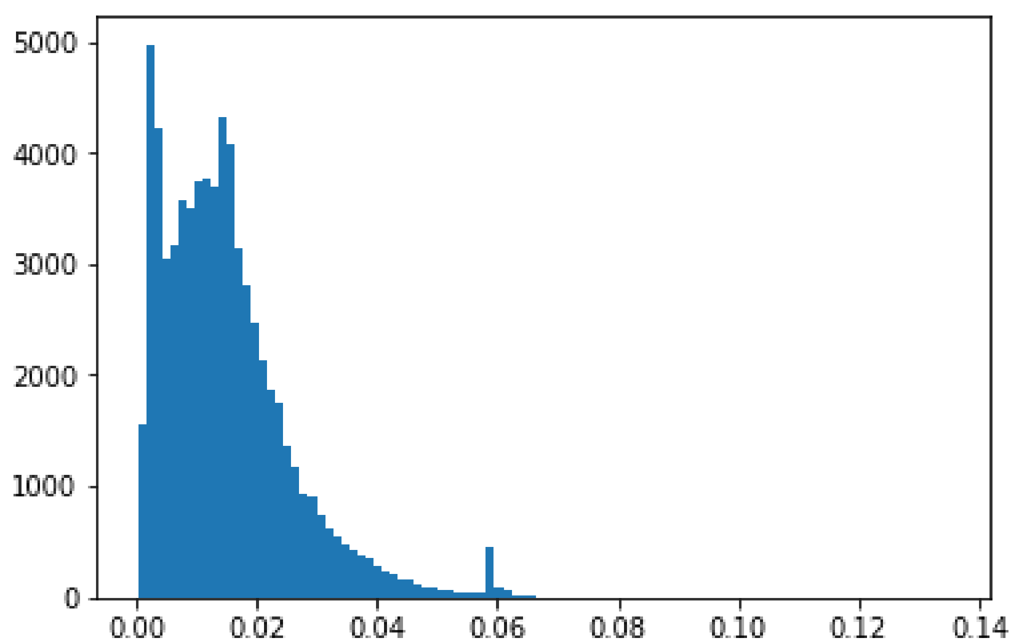
200 steps

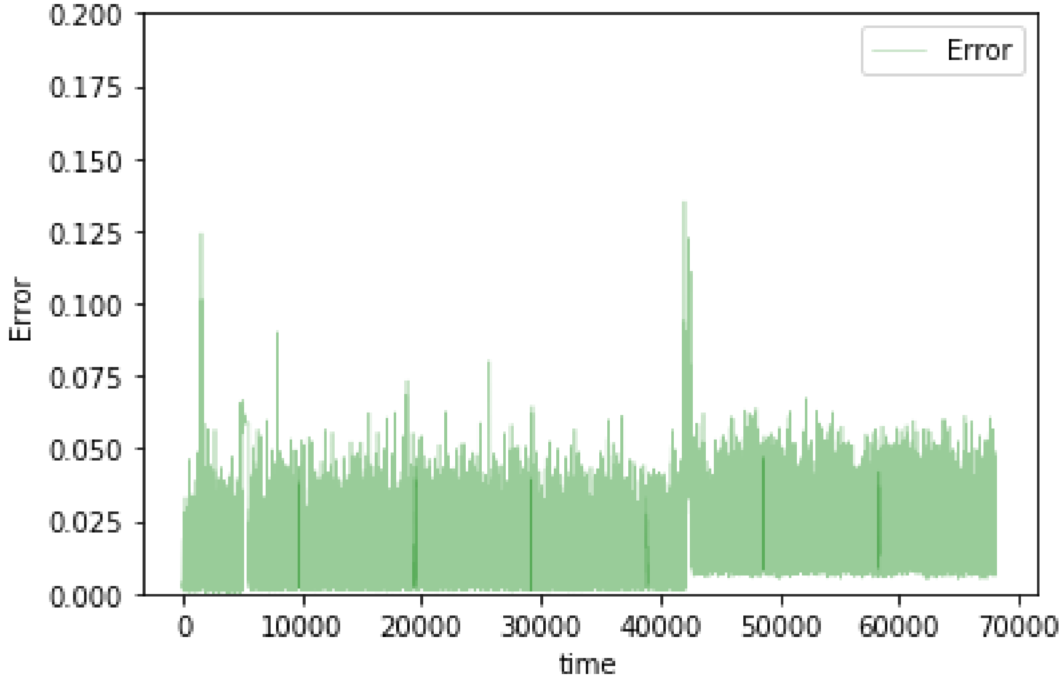
```
In [147]: TIMESTEPS = 200
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS)
          vgen = flat_generator(val_X, TIMESTEPS)
          name = "nn3_200"

In [148]: input_layer = Input(shape=(TIMESTEPS*DIM,))
          hidden = Dense(1000, activation='relu')(input_layer)
          hidden = Dense(500, activation='relu')(hidden)
          hidden = Dense(100, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [149]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [150]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```

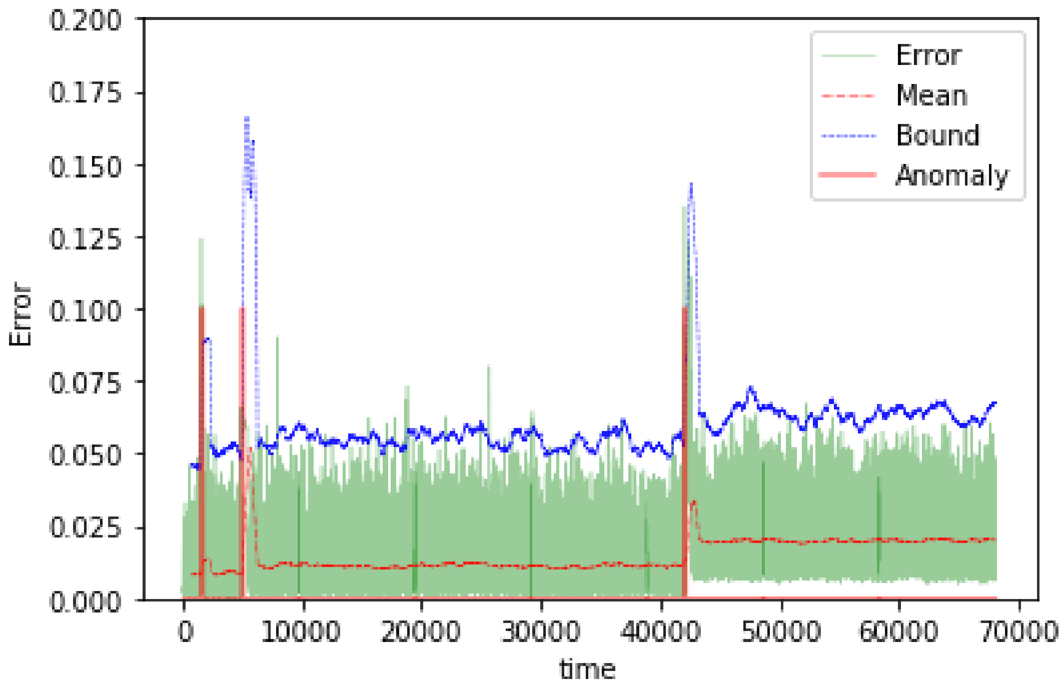


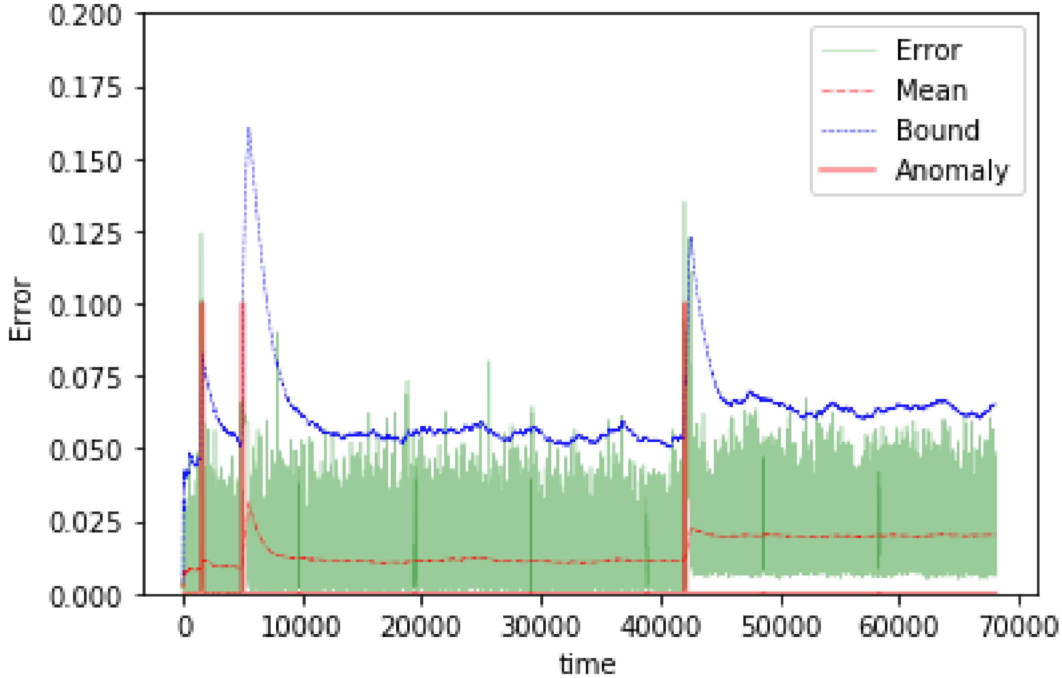

```
Training loss for final epoch is 0.007537590371677652
Validation loss for final epoch is 0.007050825907150284
---------- Beginning tests for nn3_200 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−4
−2
0
2
4

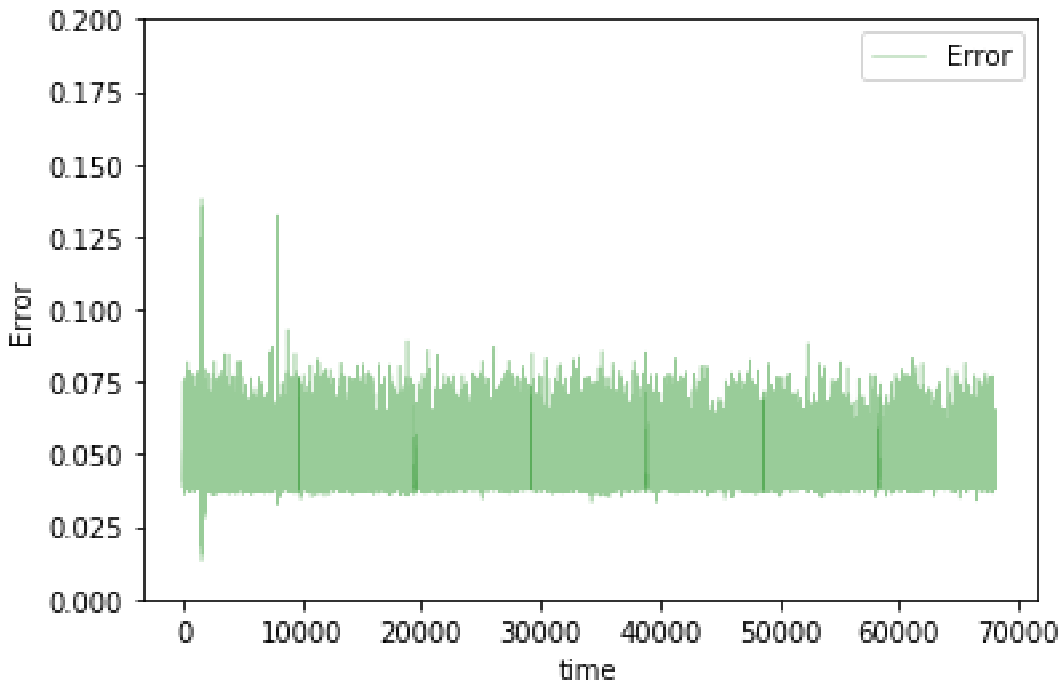

5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

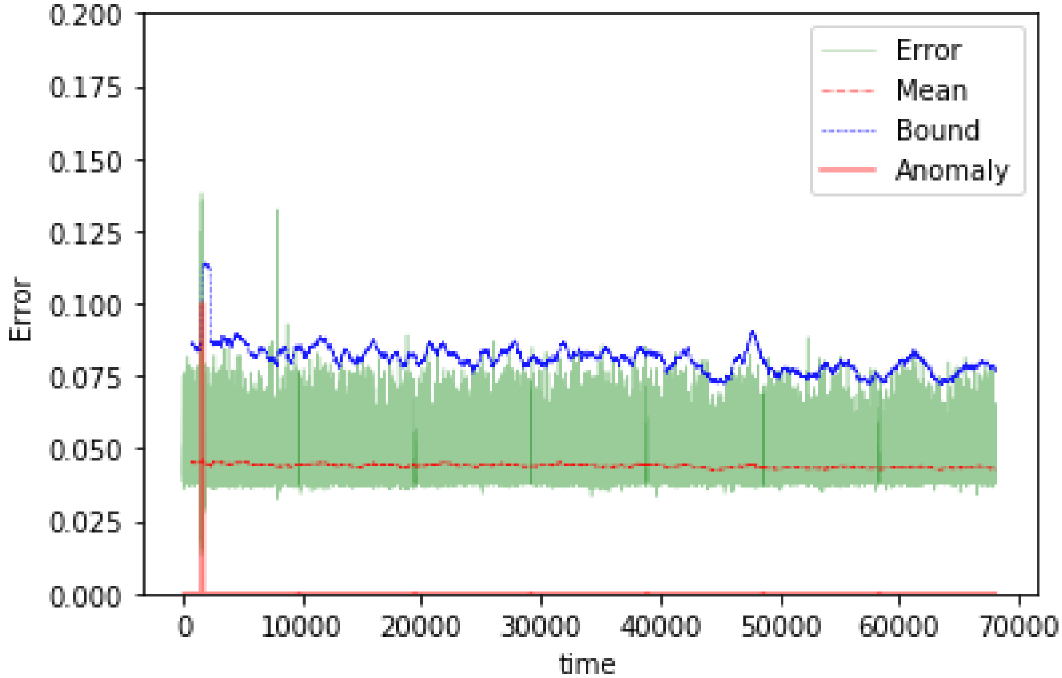

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error

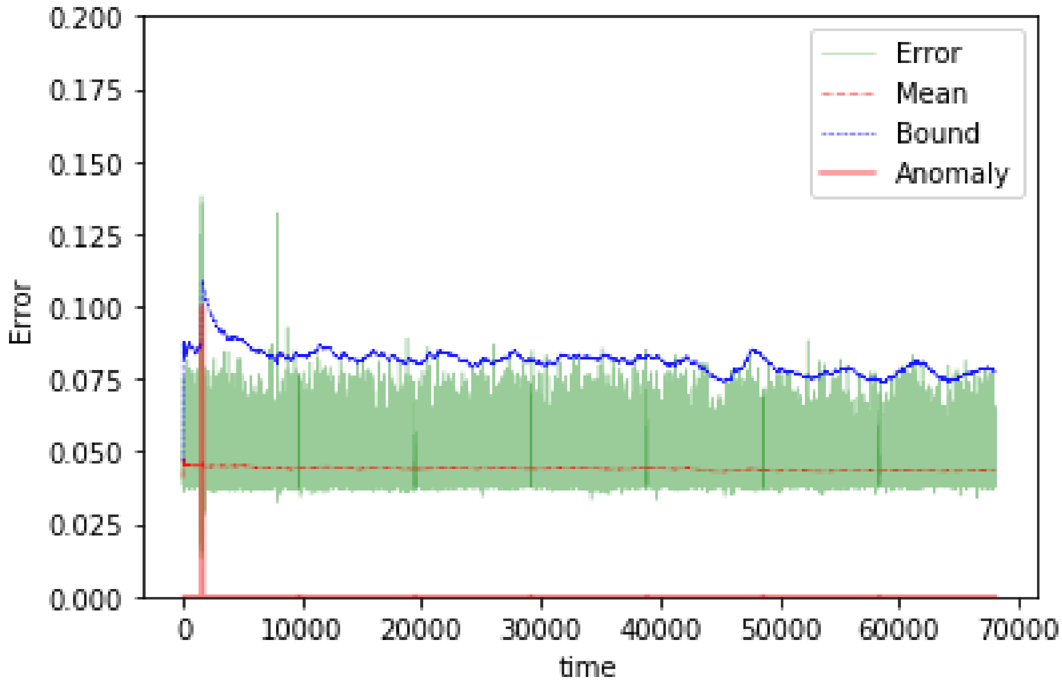

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly



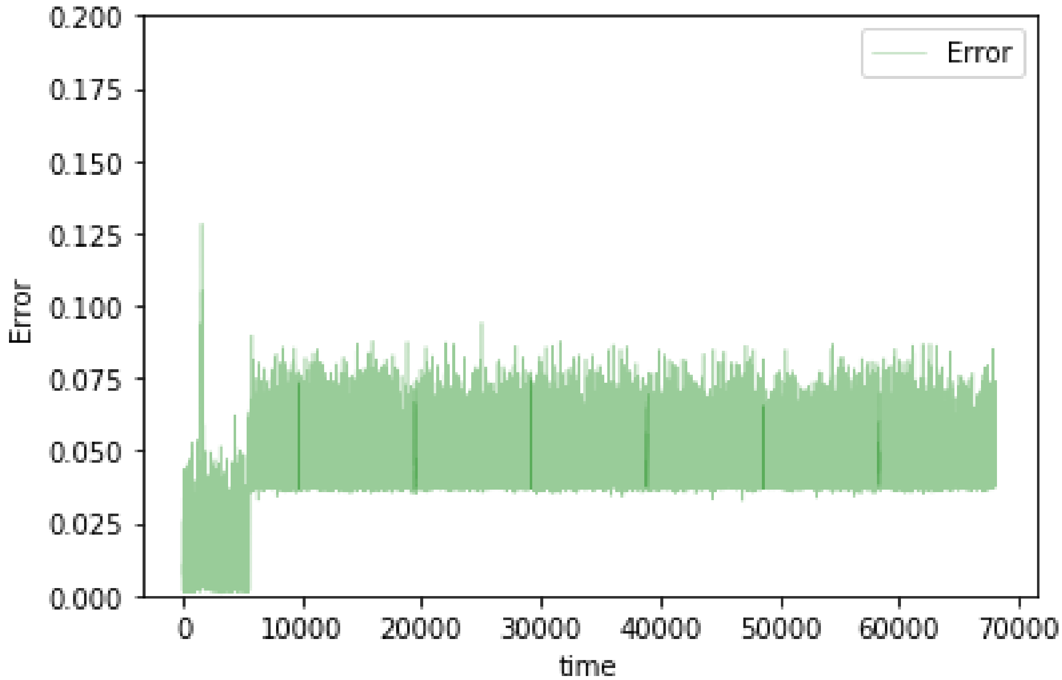

```
The mean error for nn3_200_disk_IO_start_ is 0.015023076592114598 for length 68099
Testing on Avg. load data.
```

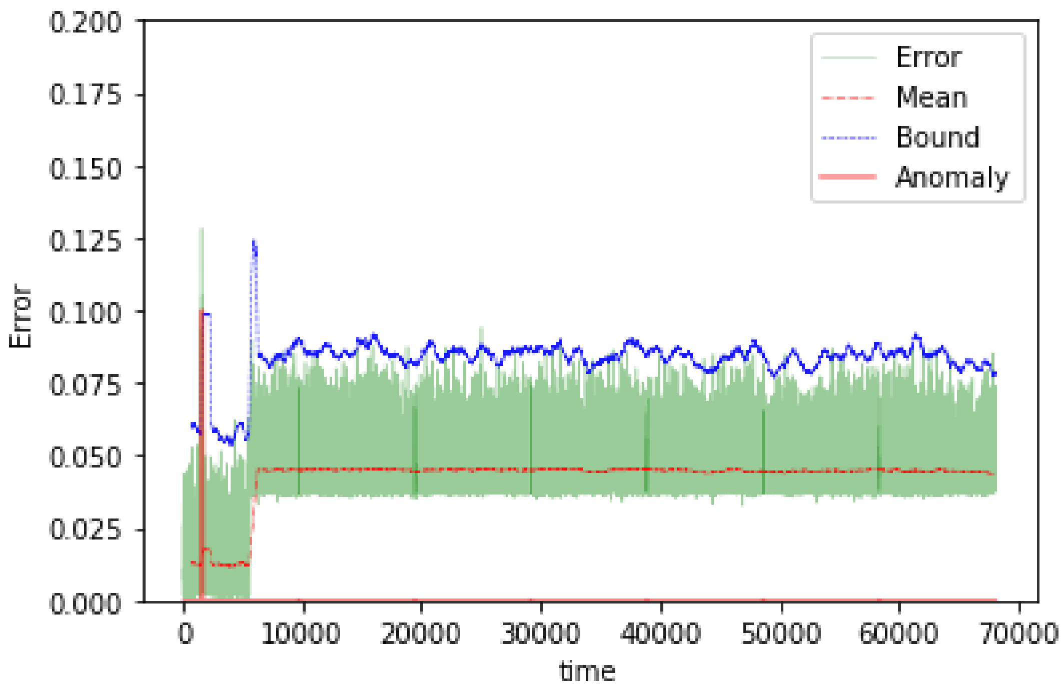


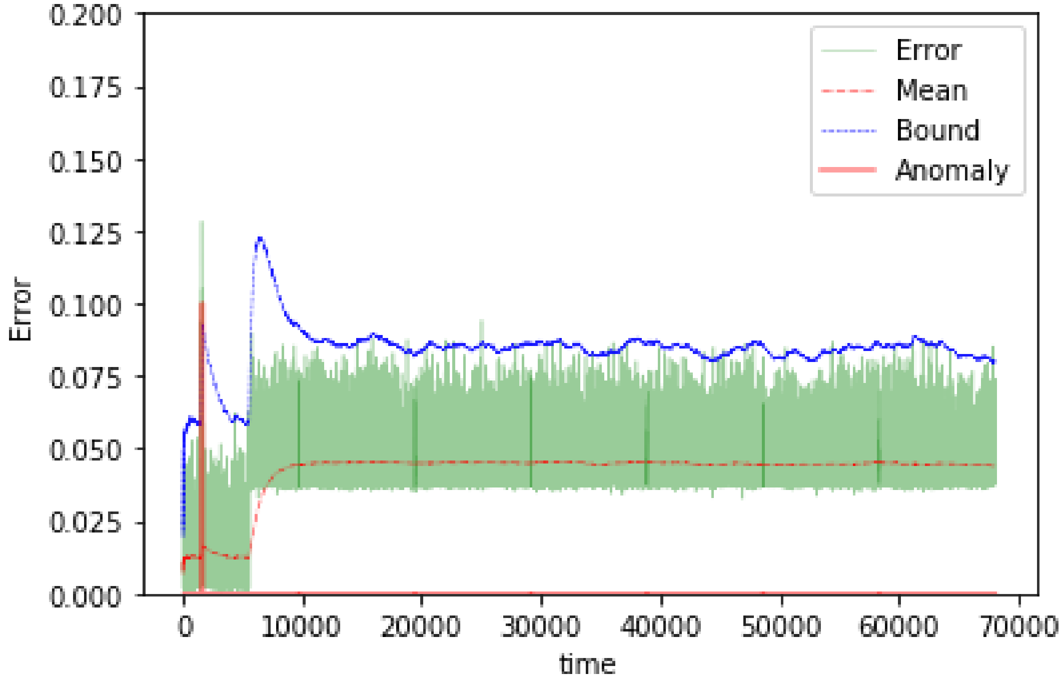

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

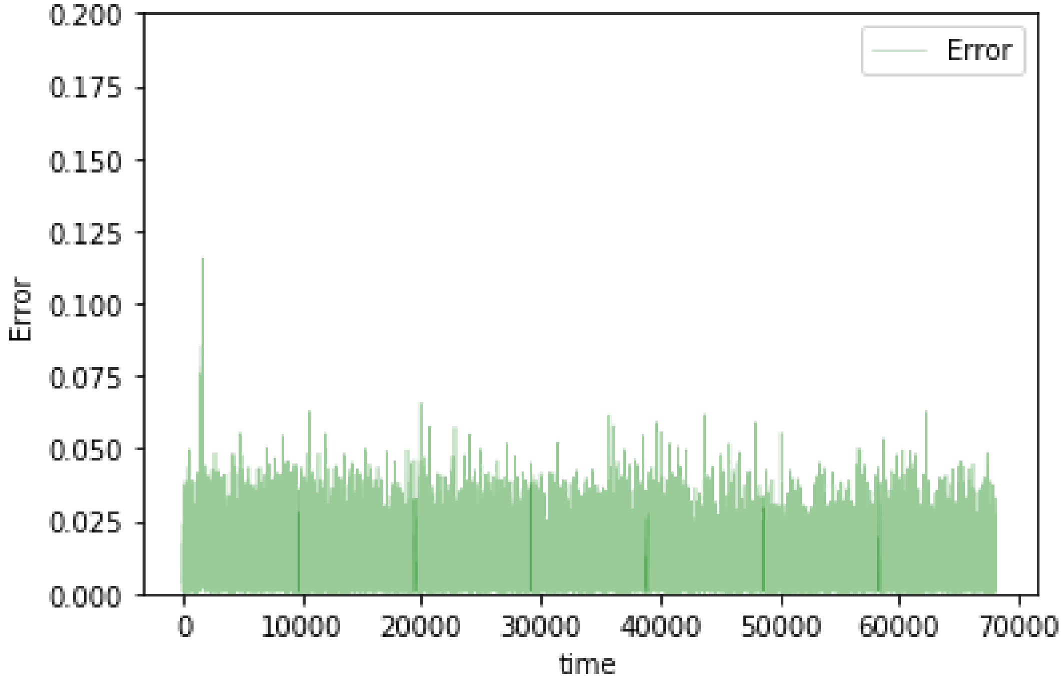
```
The mean error for nn3_200_avg_load_ is 0.044001592992706 for length 68099
Testing on app change early data.
```

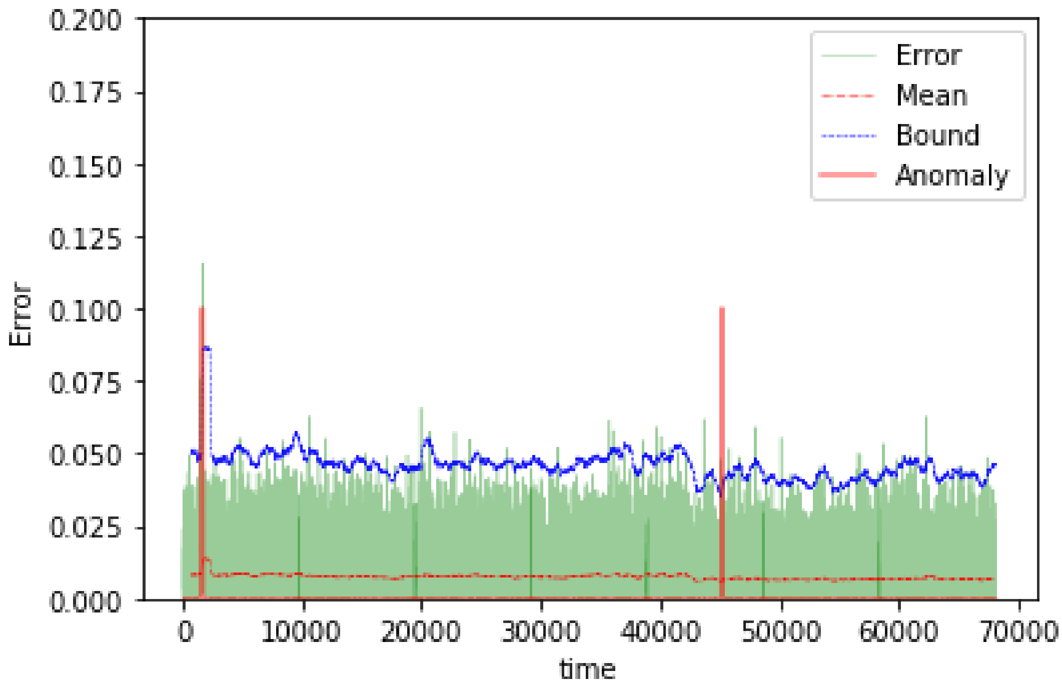


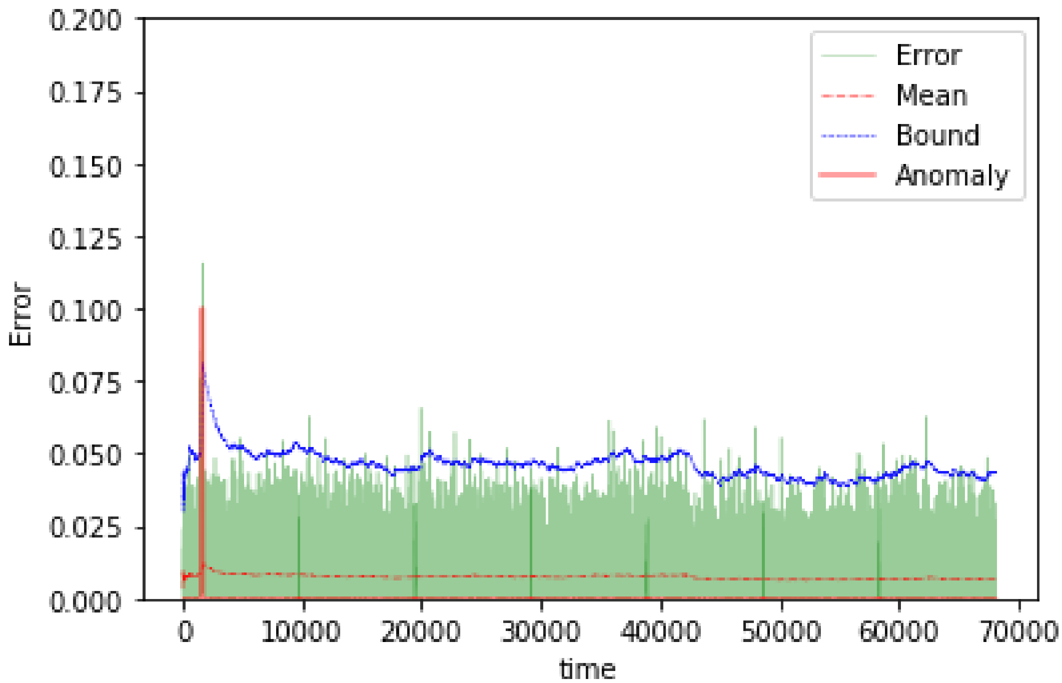





```
The mean error for nn3_200_app_change_early_ is 0.04228741419757072 for length 68099
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error




```
The mean error for nn3_200_normal_ is 0.007413899040705531 for length 68099
====================
```

1.11.5 RNN with 1 GRU layers

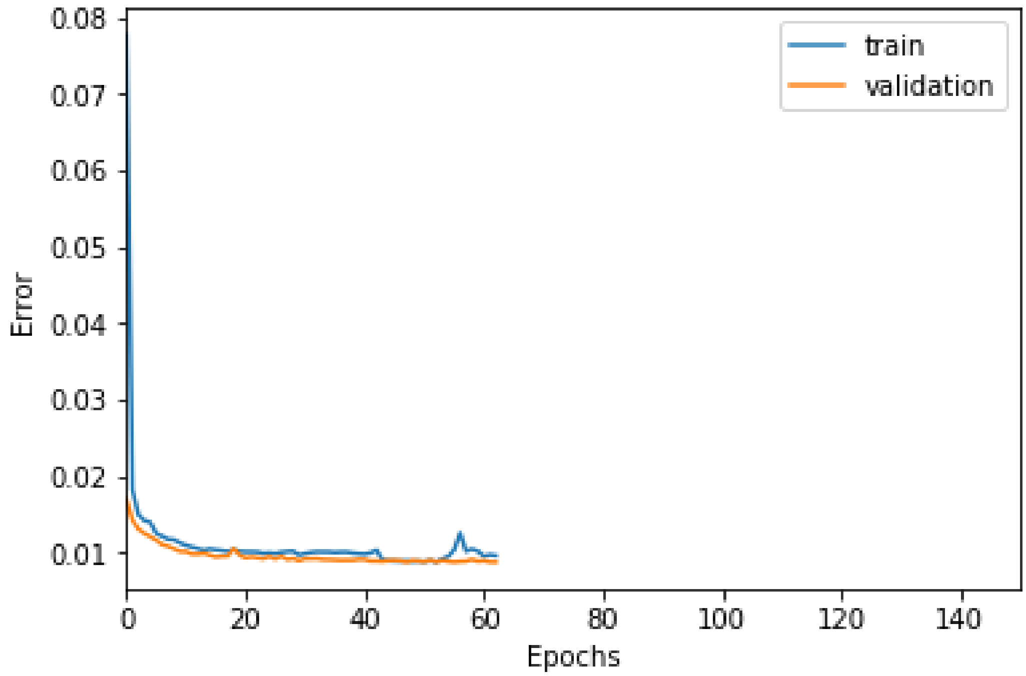
2 steps

```
In [151]: TIMESTEPS = 2
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru1_2"
```

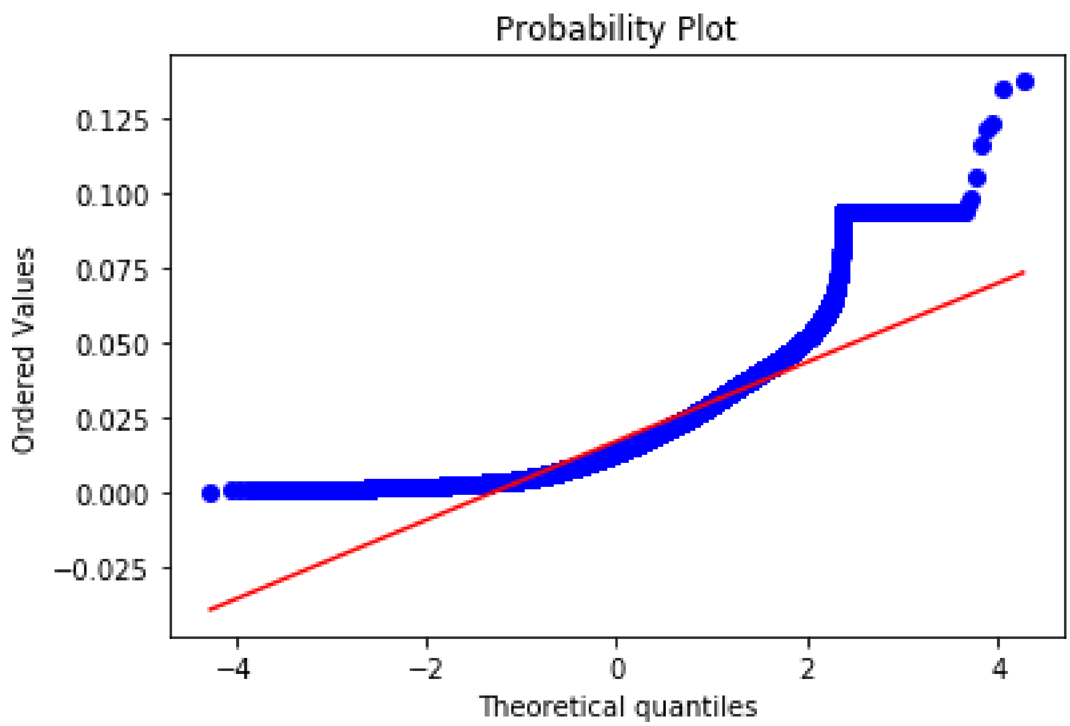
```
In [152]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu')(input_layer)
          output = Dense(DIM, activation='sigmoid')(hidden)
```

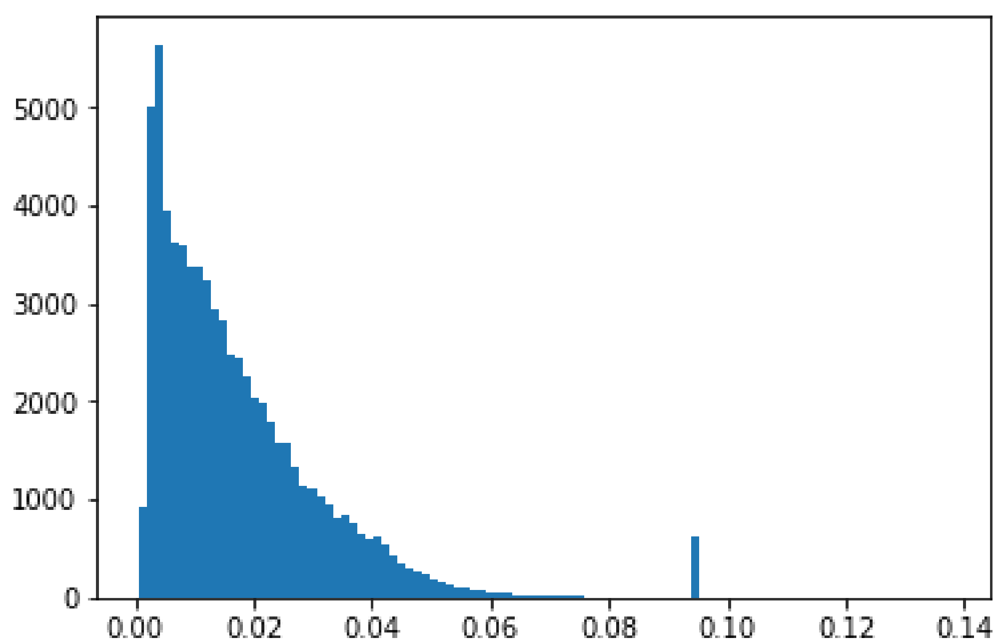
```
In [153]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])
```

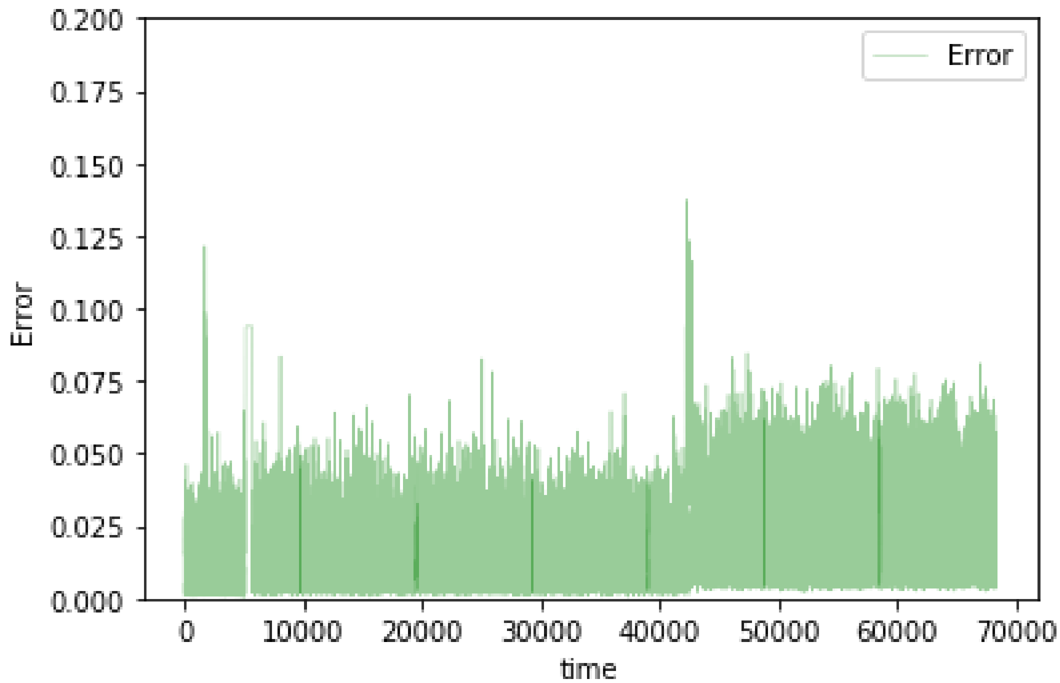
```
In [154]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

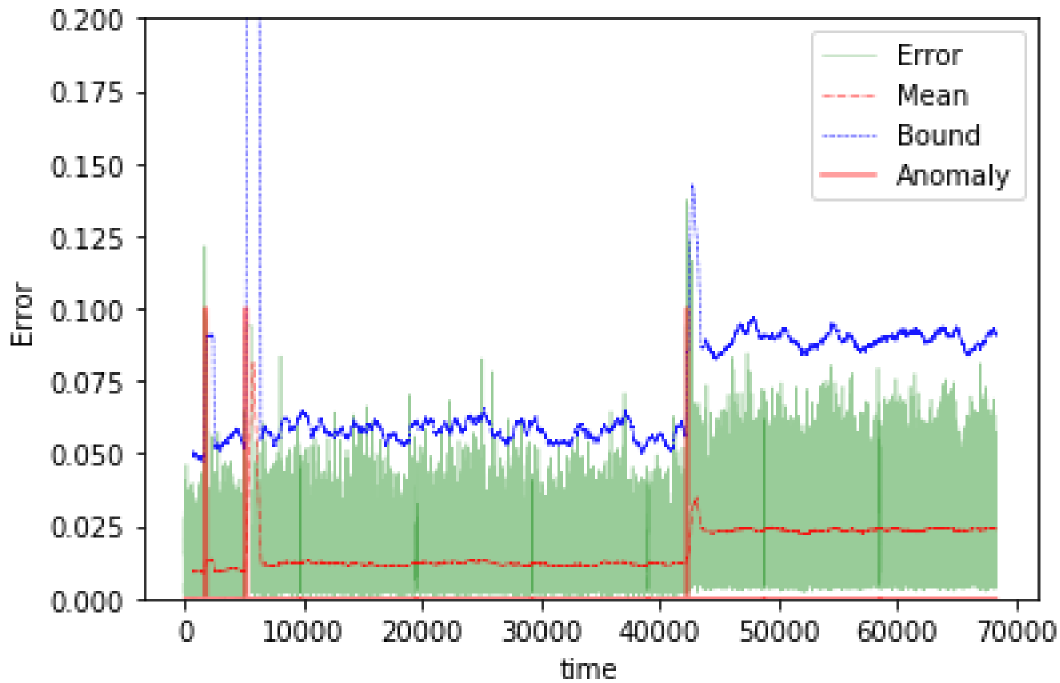


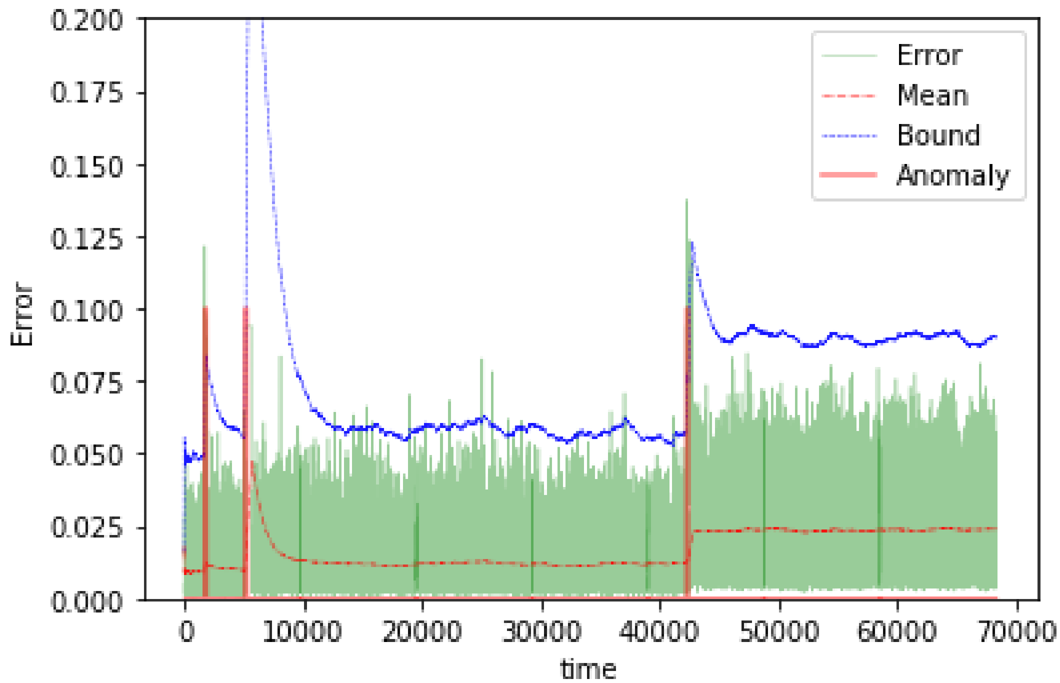

```
Training loss for final epoch is 0.009565628632553853
Validation loss for final epoch is 0.008720844502095133
---------- Beginning tests for gru1_2 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

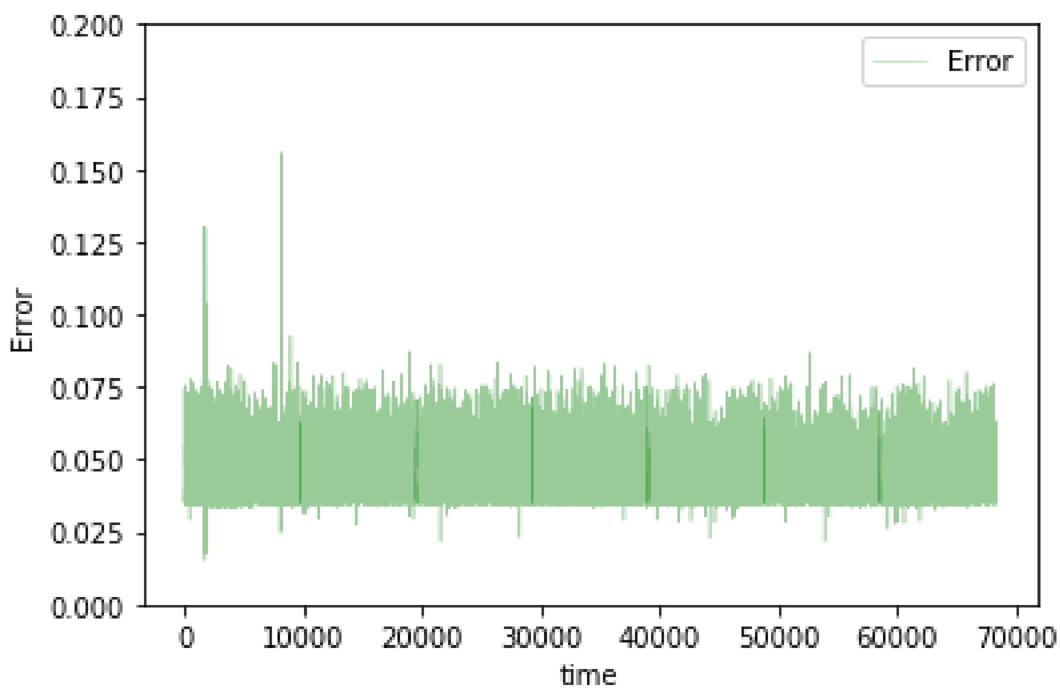

5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

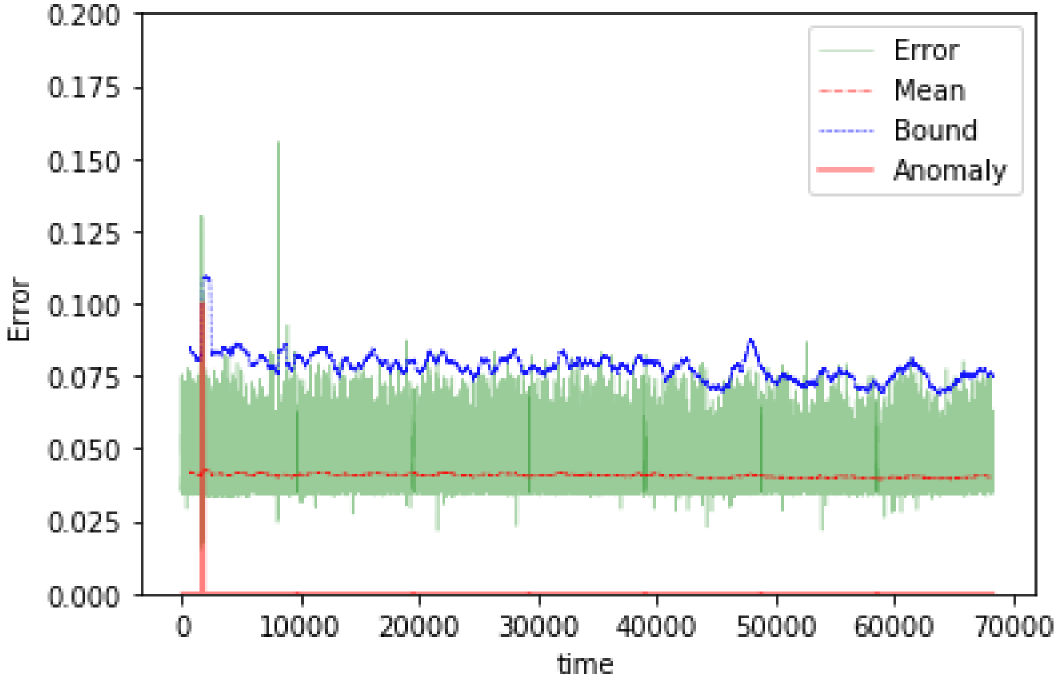

Error
Error
time

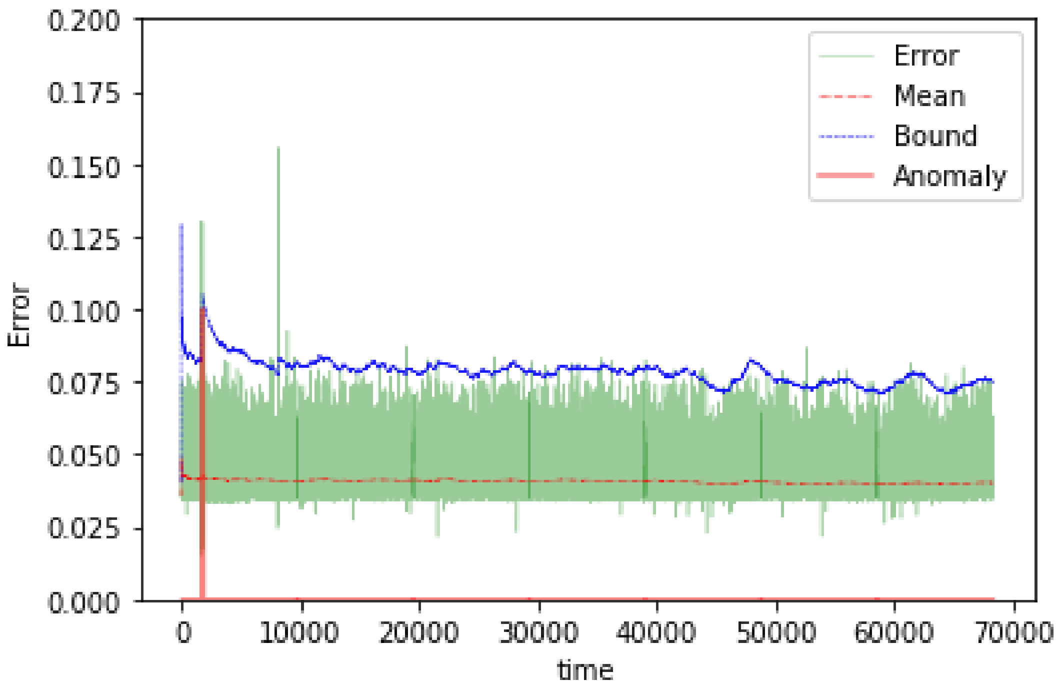

Error
Mean
Bound
Anomaly
Error
time



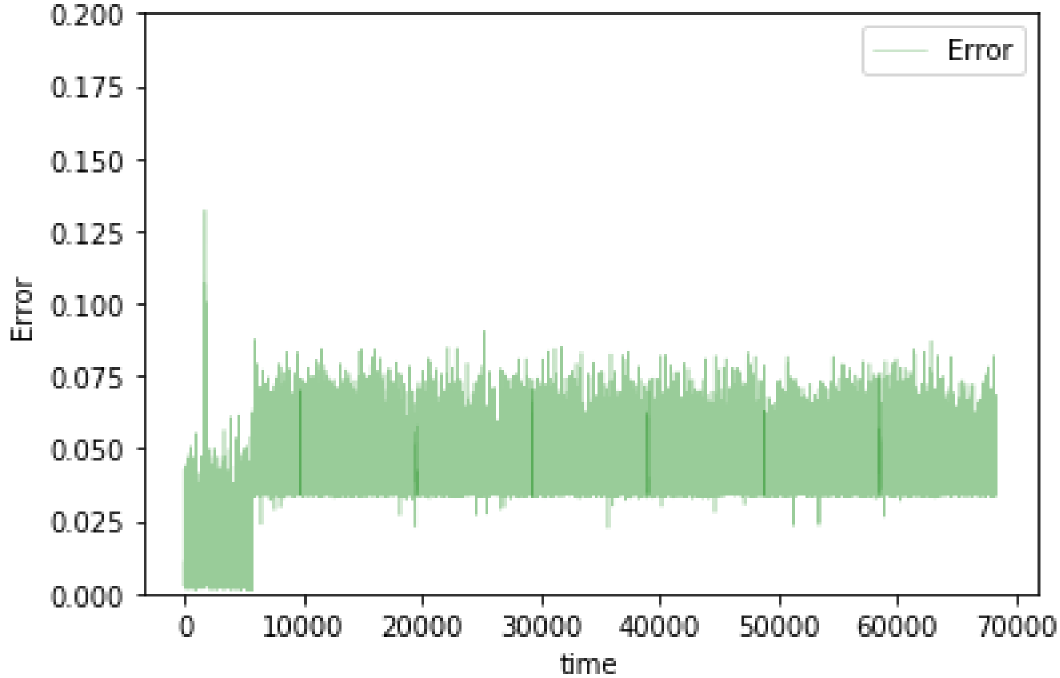

```
The mean error for gru1_2_disk_IO_start_ is 0.017109448006642844 for length 68297
Testing on Avg. load data.
```

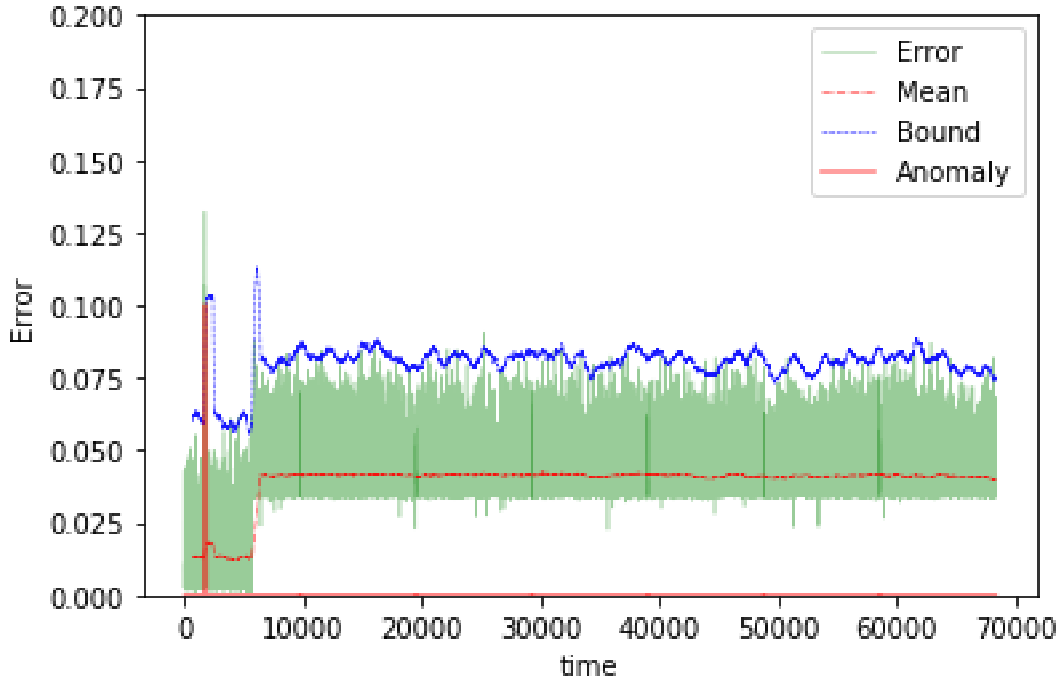


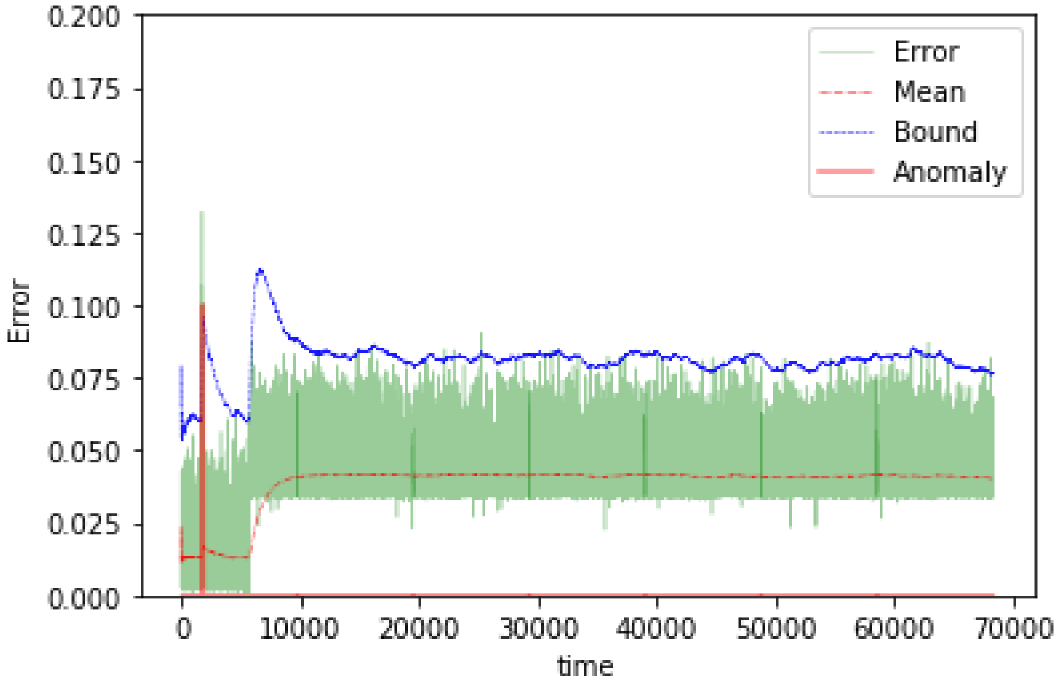

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

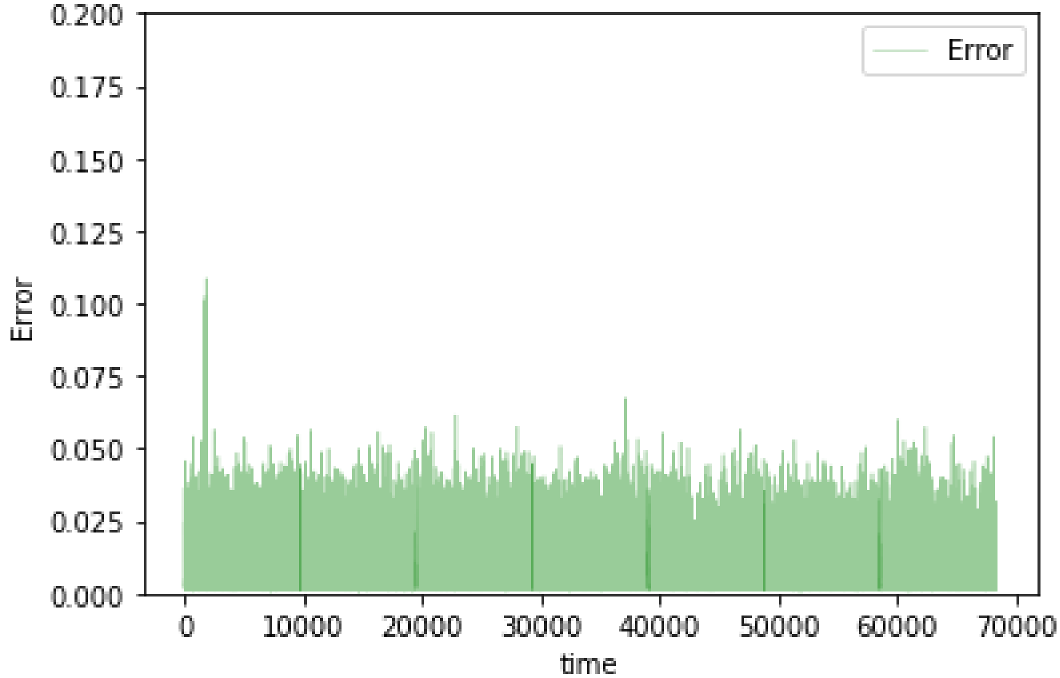
```
The mean error for gru1_2_avg_load_ is 0.040670835961593294 for length 68297
Testing on app change early data.
```

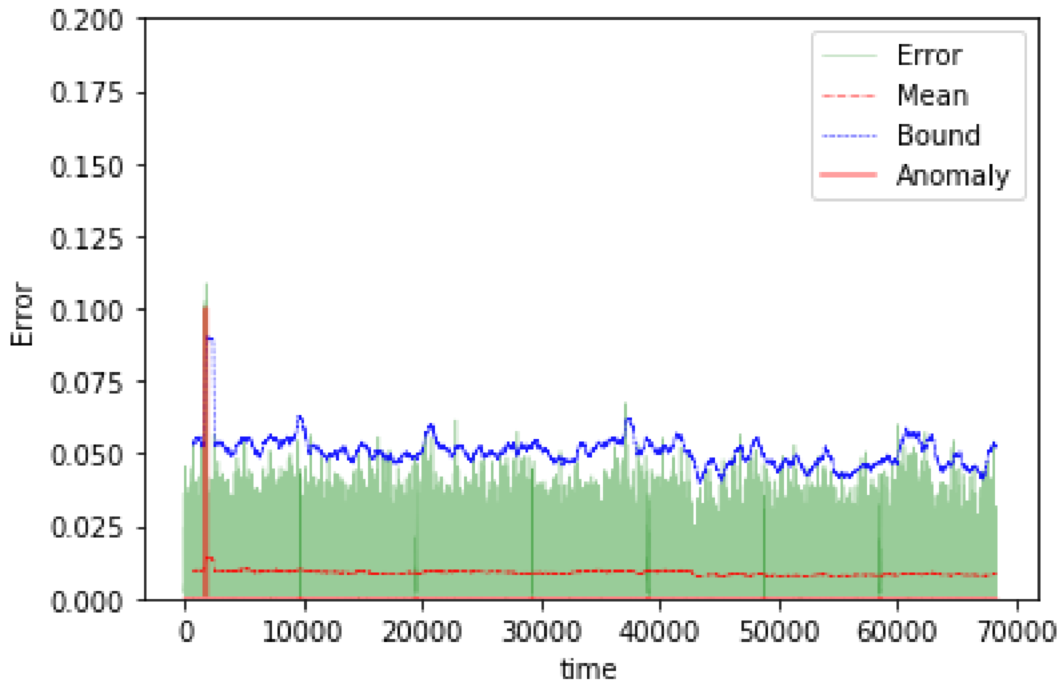


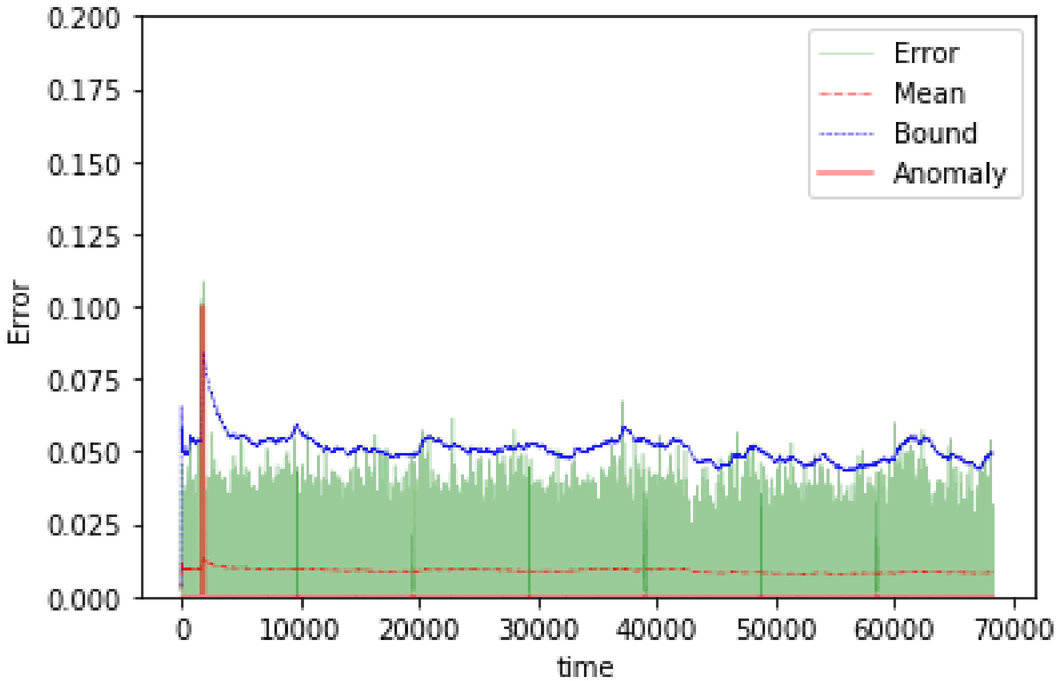





```
The mean error for gru1_2_app_change_early_ is 0.038922377772049405 for length 68297
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

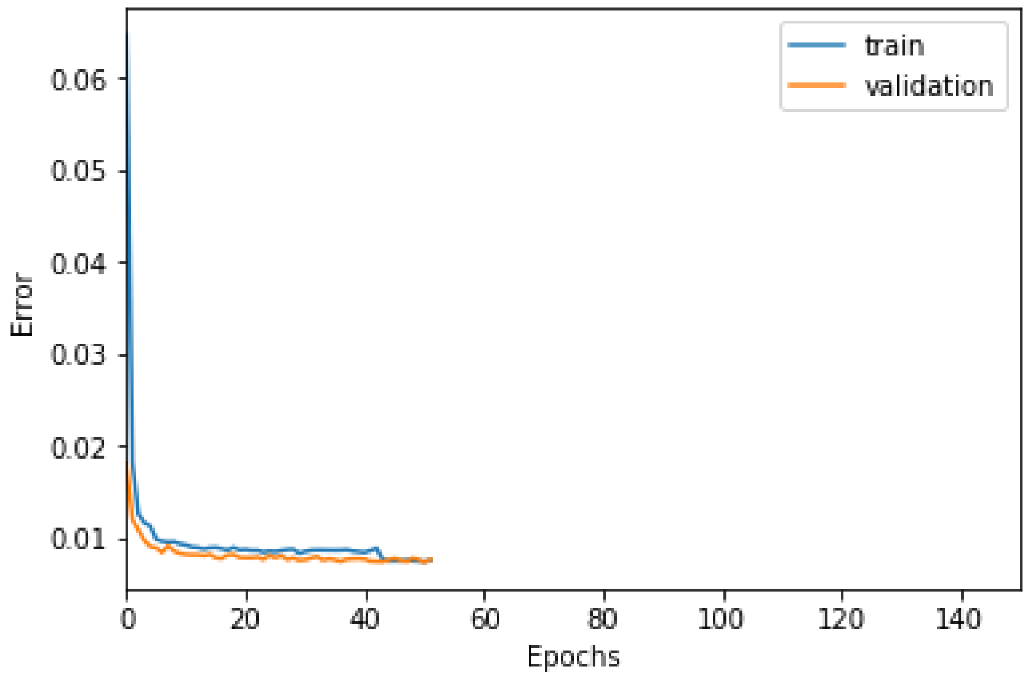

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



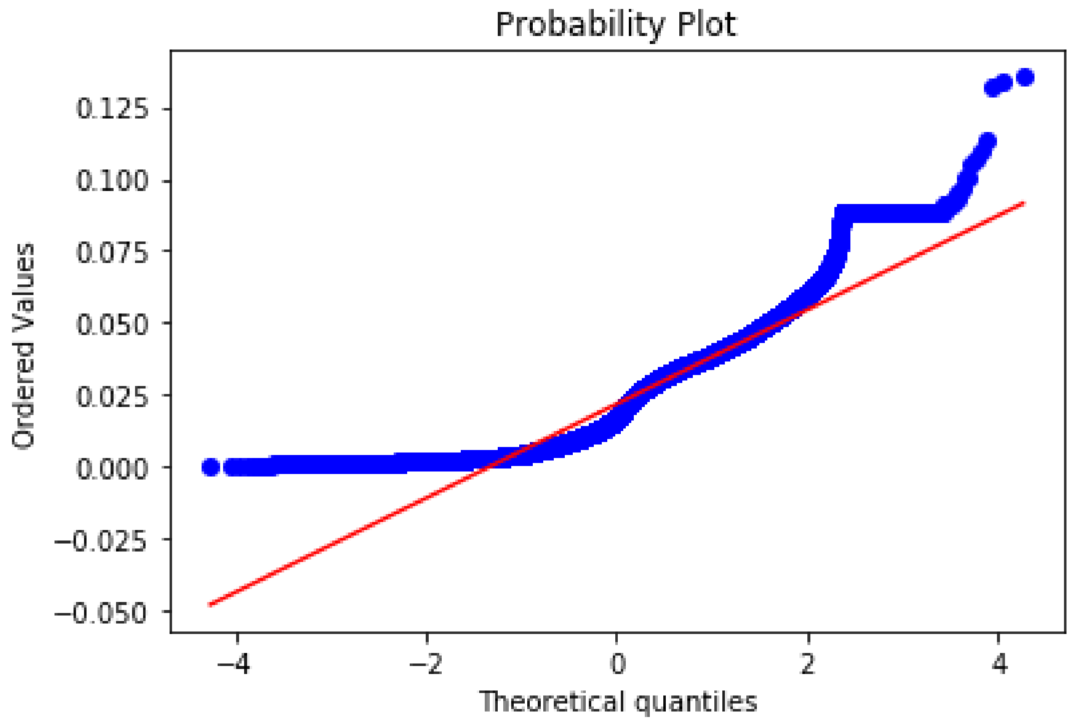

```
The mean error for gru1_2_normal_ is 0.008856913935017825 for length 68297
====================
```

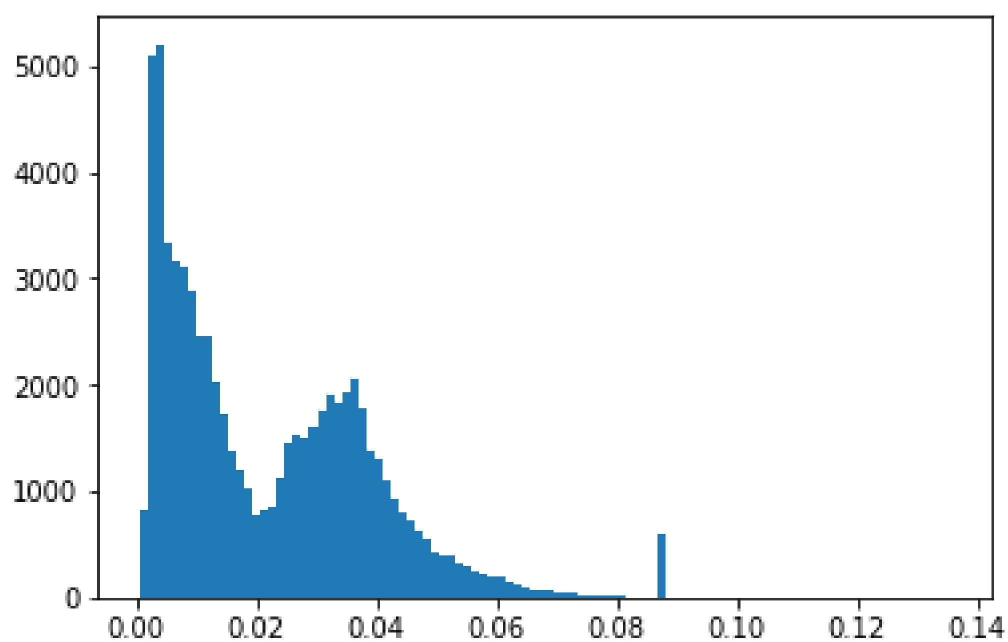
5 steps

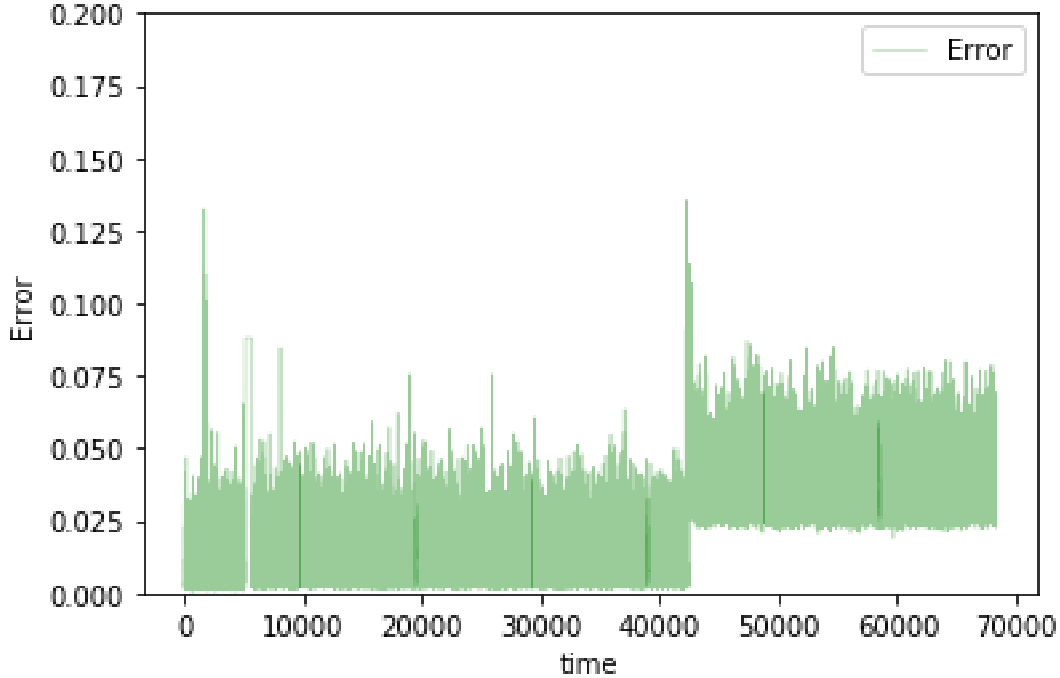
```
In [155]: TIMESTEPS = 5
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS, 0)
          name = "gru1_5"

In [156]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu')(input_layer)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [157]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [158]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

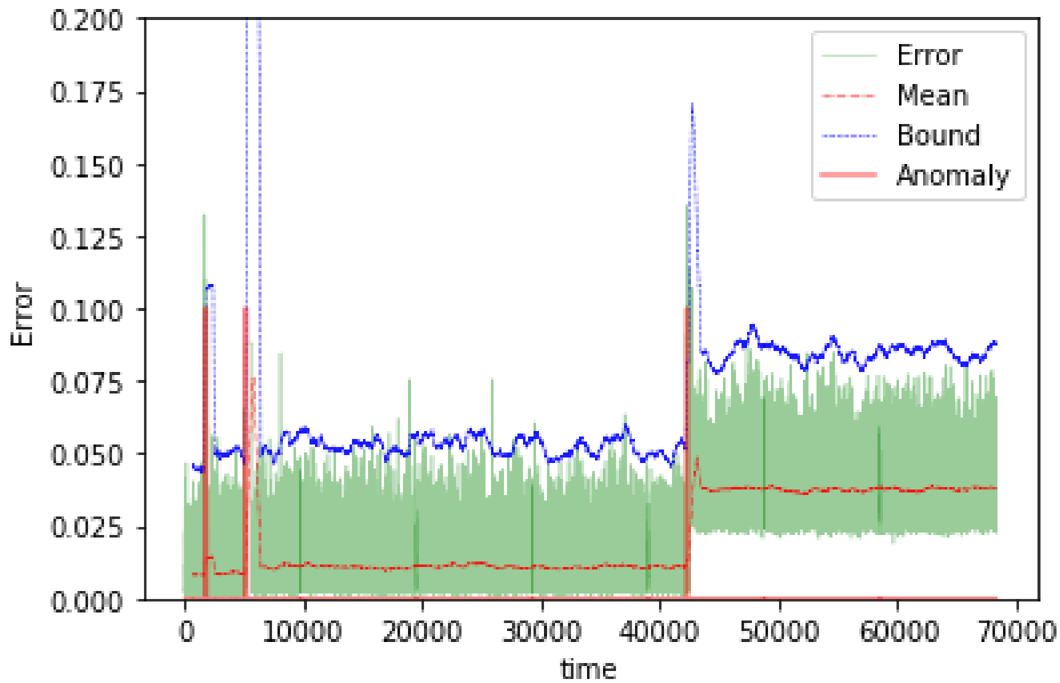


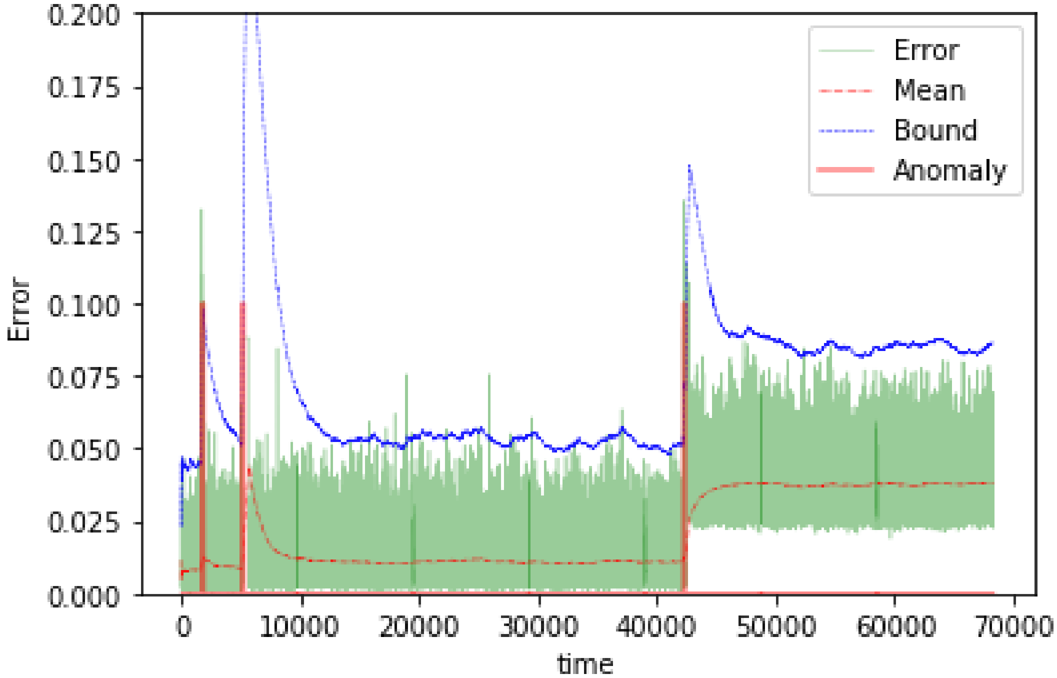

```
Training loss for final epoch is 0.007556810664447869
Validation loss for final epoch is 0.007623770359437913
---------- Beginning tests for gru1_5 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

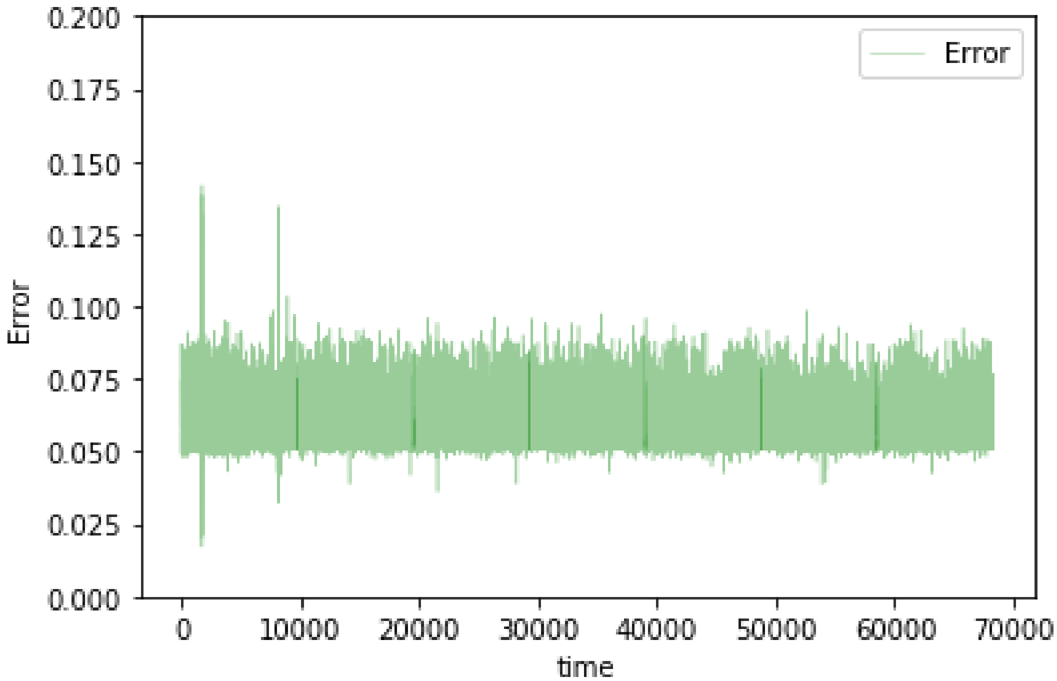

5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

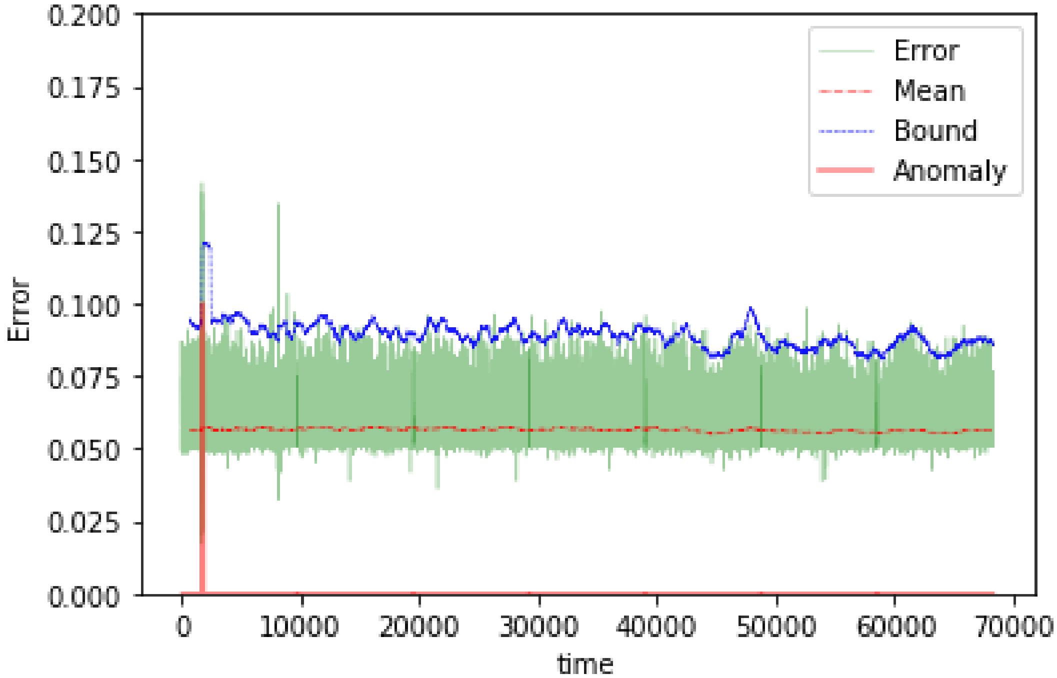

Error
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

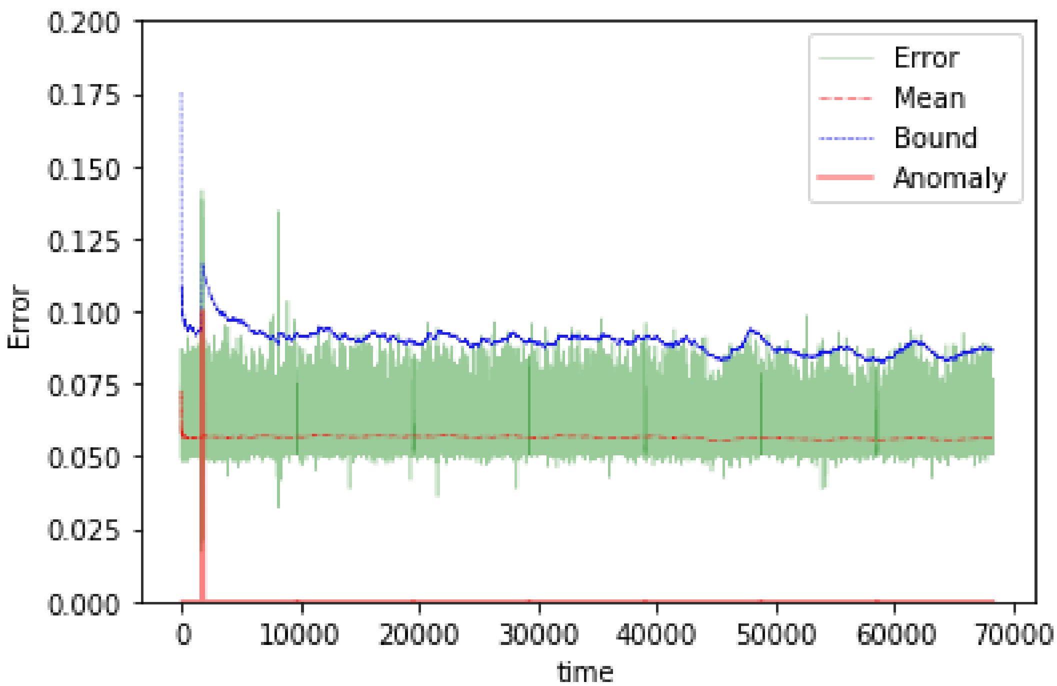

Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000



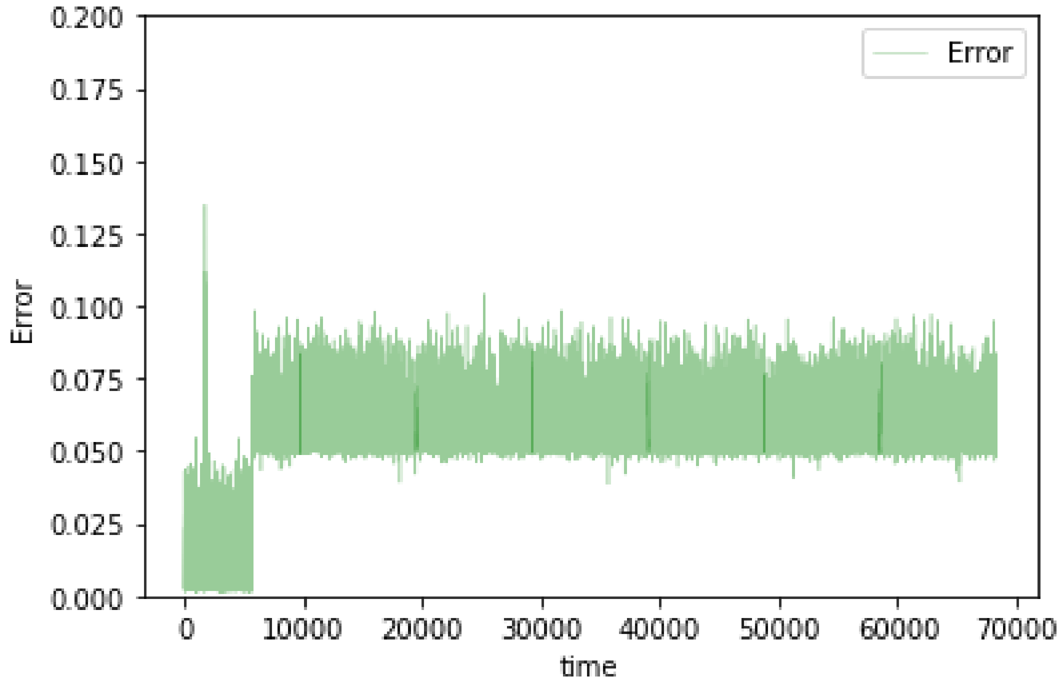

```
The mean error for gru1_5_disk_IO_start_ is 0.0217754101191509 for length 68294
Testing on Avg. load data.
```

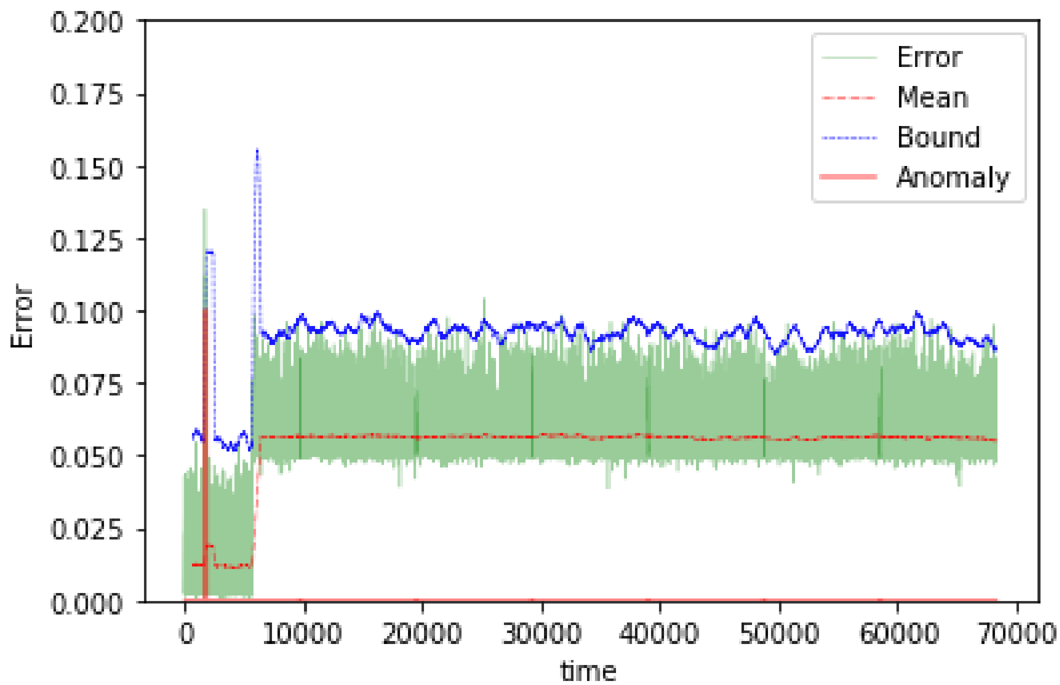


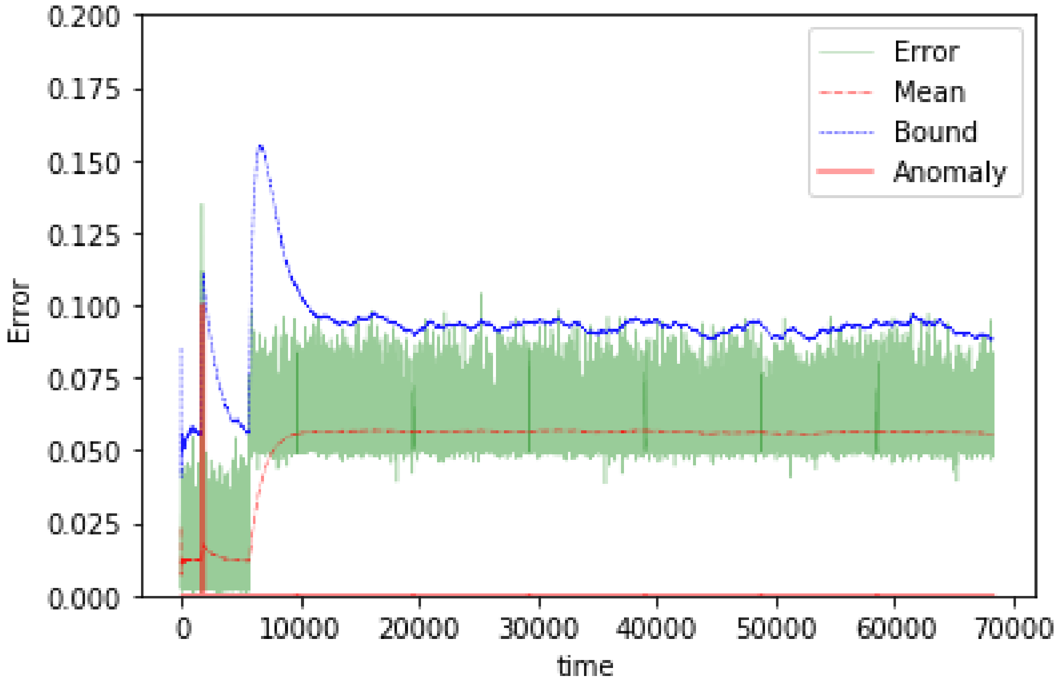

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

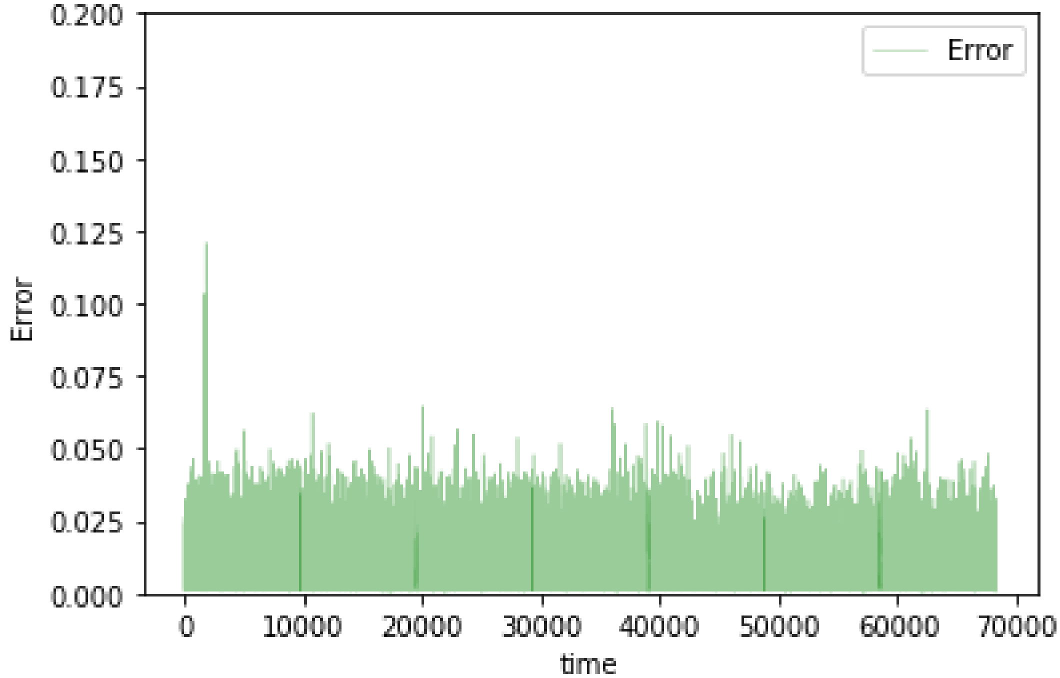
```
The mean error for gru1_5_avg_load_ is 0.05649569085109779 for length 68294
Testing on app change early data.
```

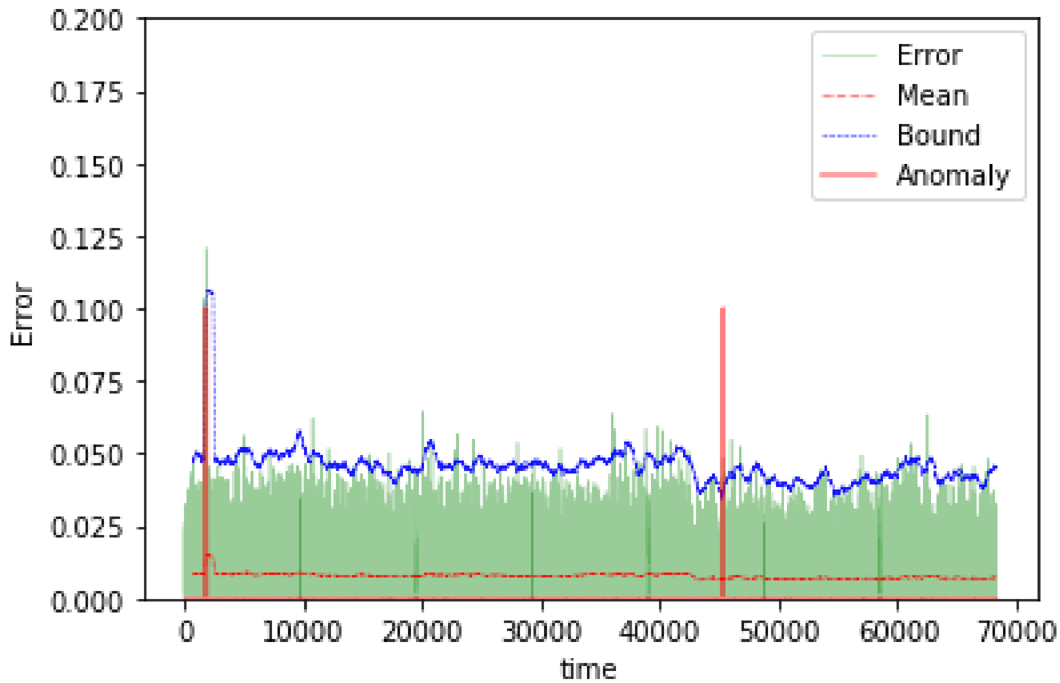


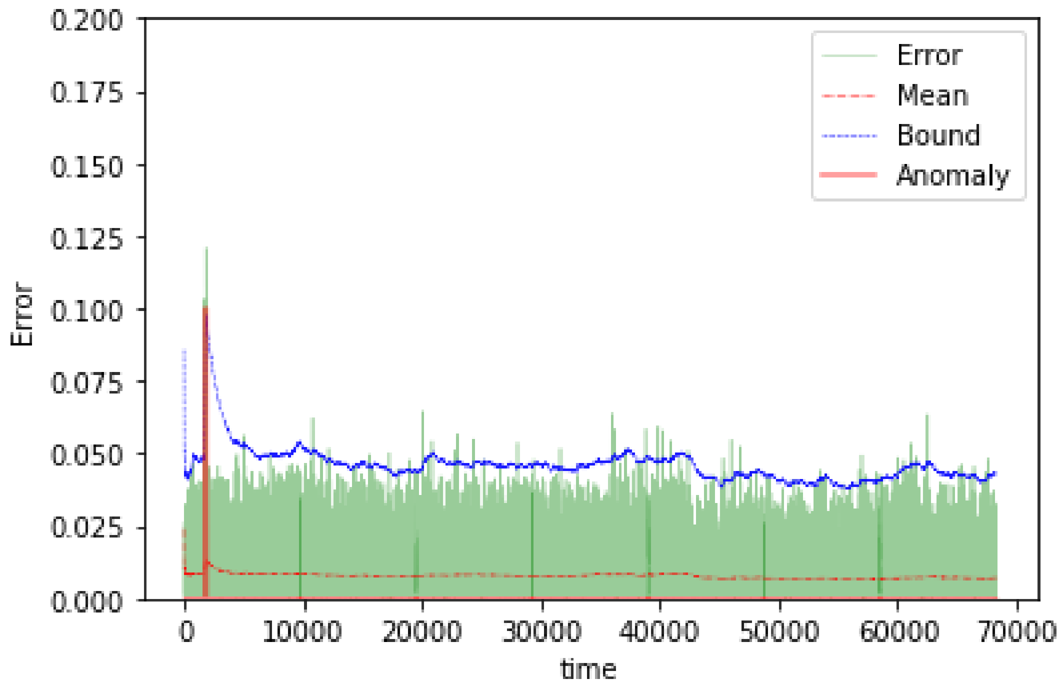





```
The mean error for gru1_5_app_change_early_ is 0.052691220497685515 for length 68294
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

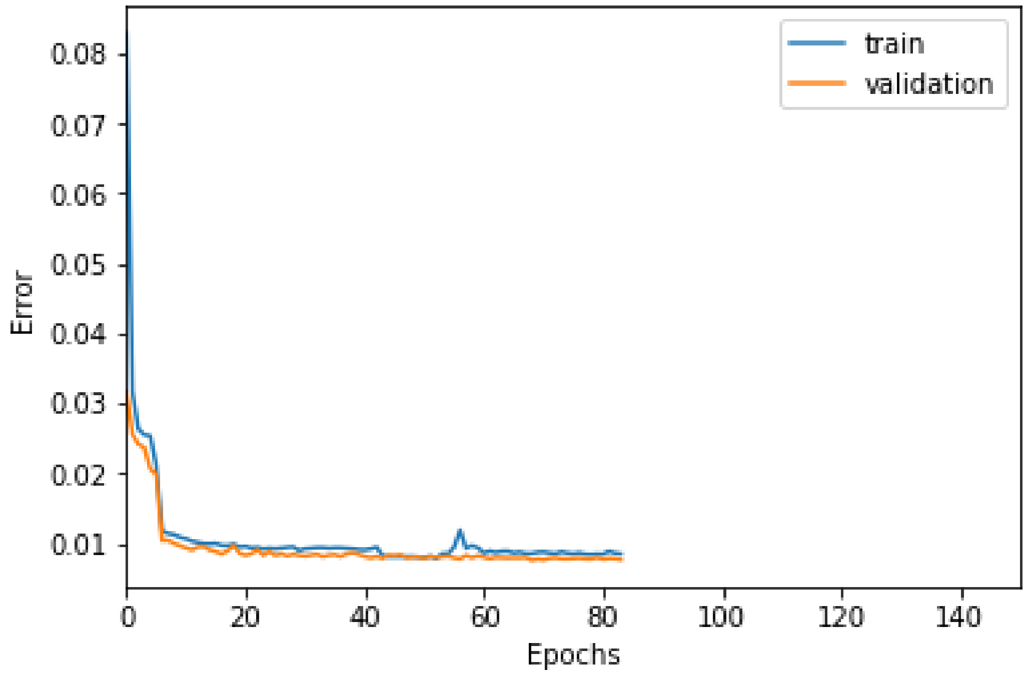

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



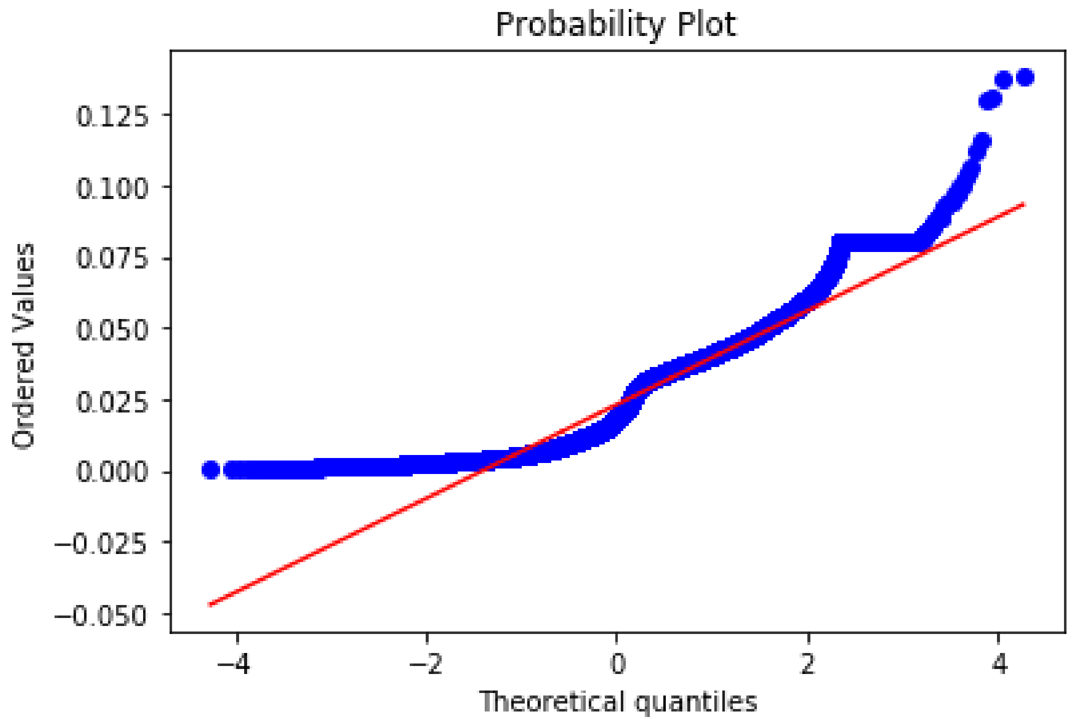

```
The mean error for gru1_5_normal_ is 0.007753765841026808 for length 68294
====================
```

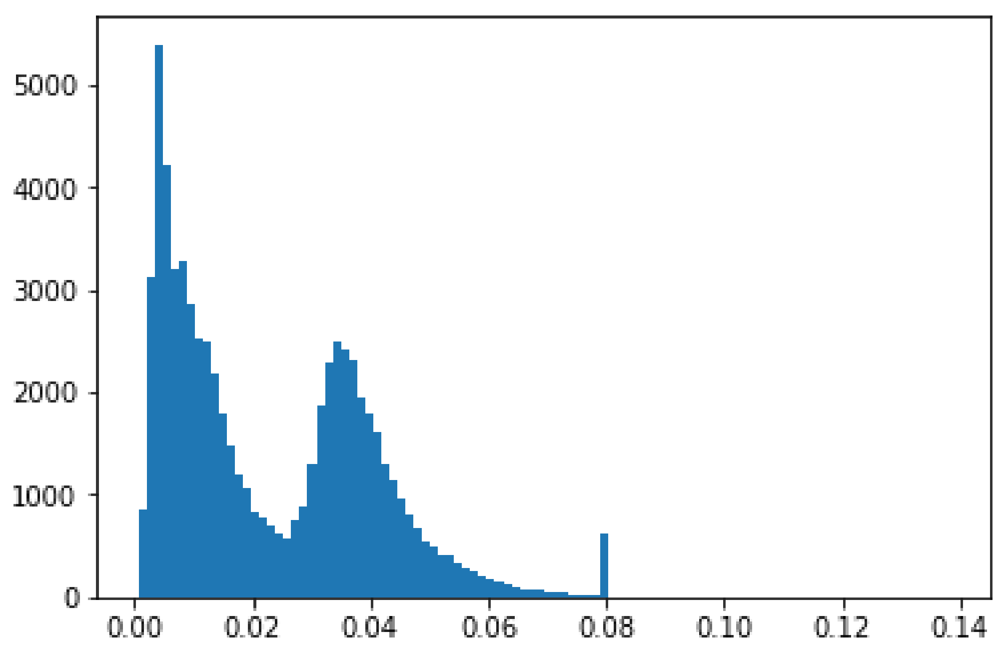
10 steps

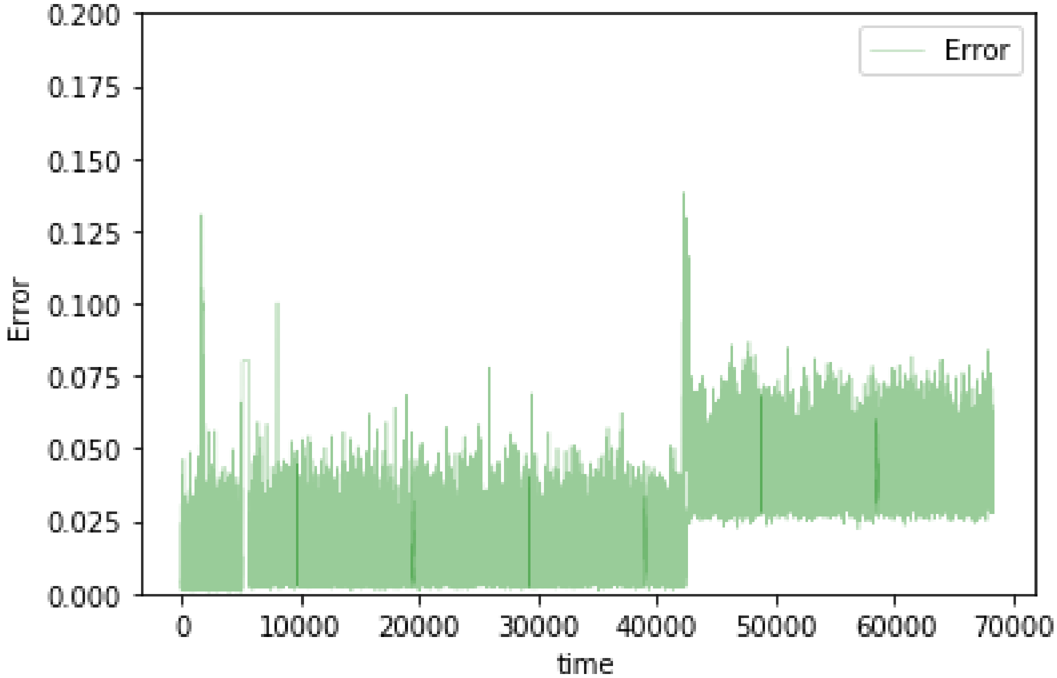
```
In [159]: TIMESTEPS = 10
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS, 0)
          vgen = flat_generator(val_X, TIMESTEPS, 0)
          name = "gru1_10"

In [160]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu')(input_layer)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [161]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [162]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```



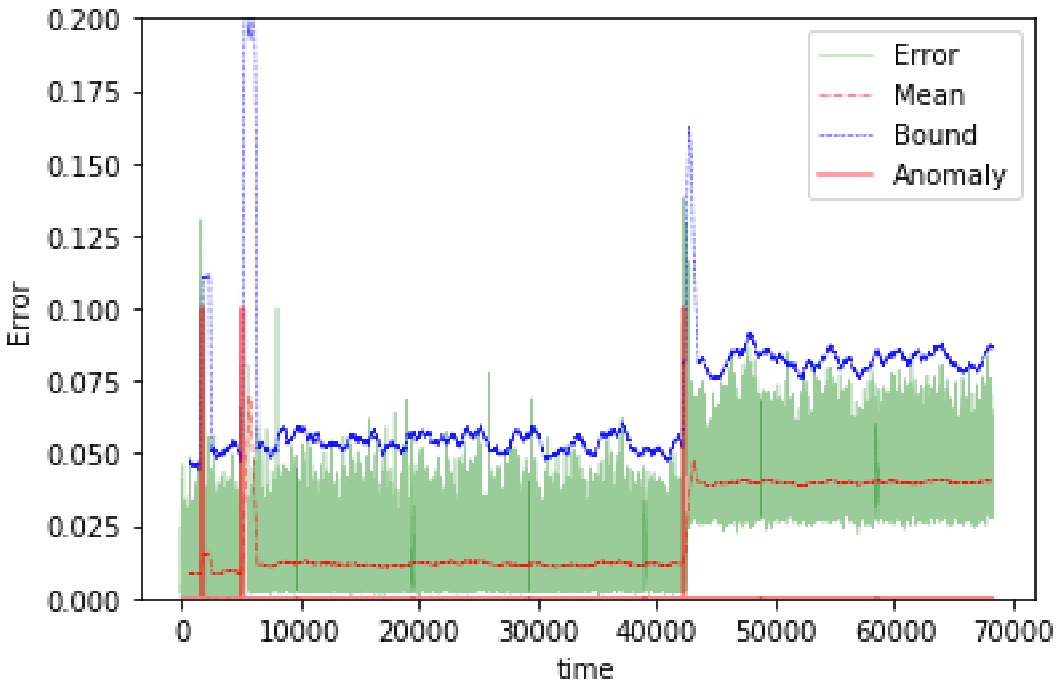

```
Training loss for final epoch is 0.008462999998708256
Validation loss for final epoch is 0.007635489291744306
---------- Beginning tests for gru1_10 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−4
−2
0
2
4
Theoretical quantiles

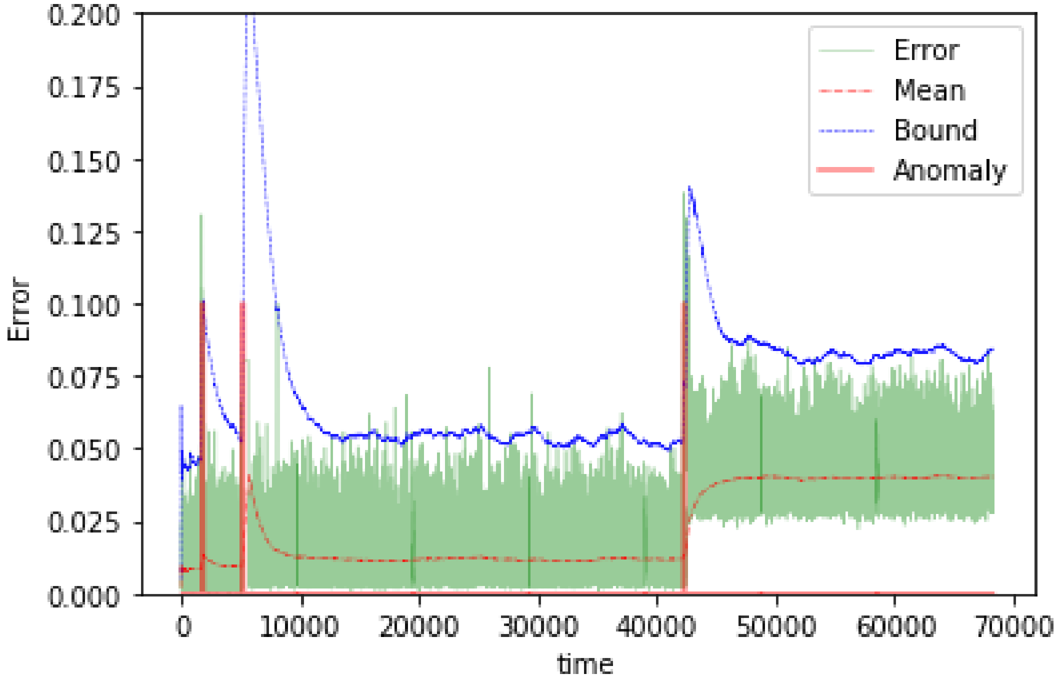

5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14



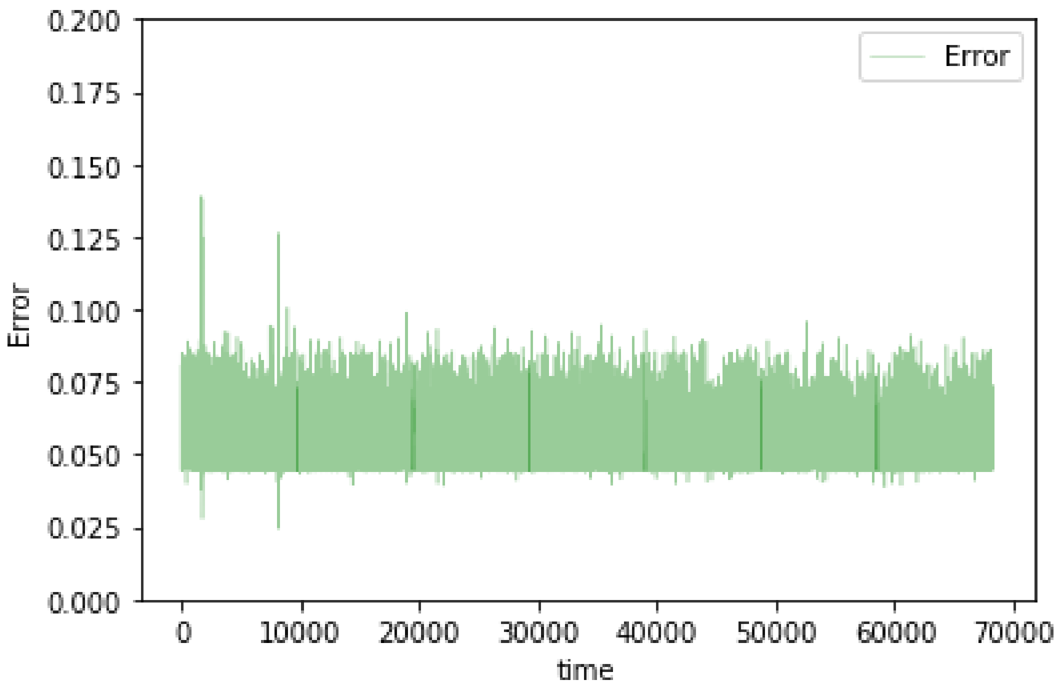
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

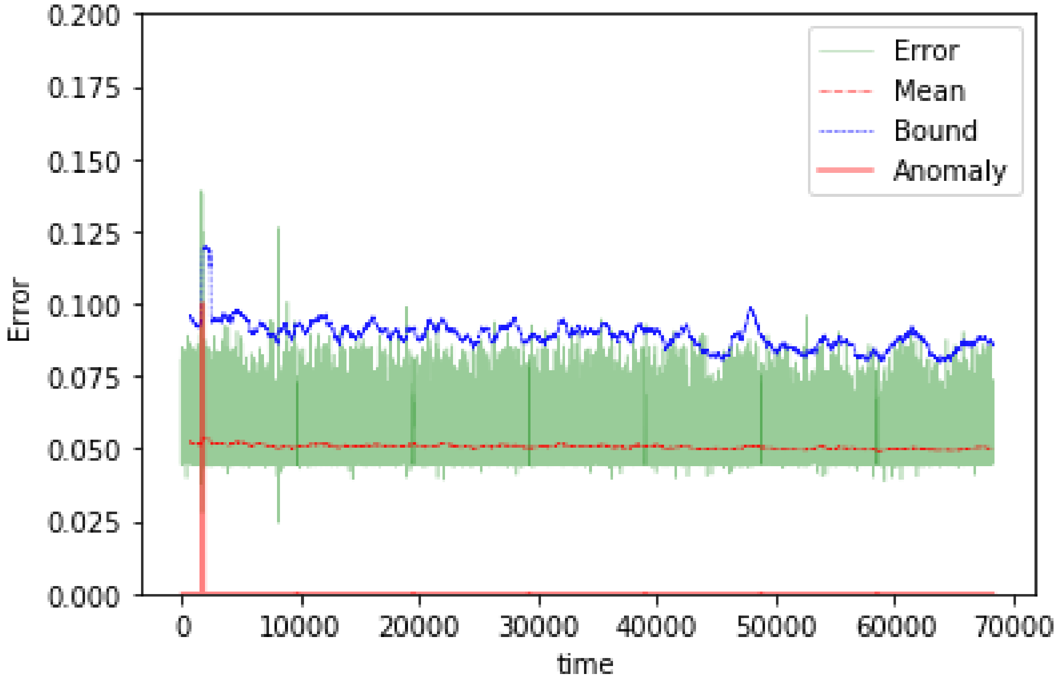


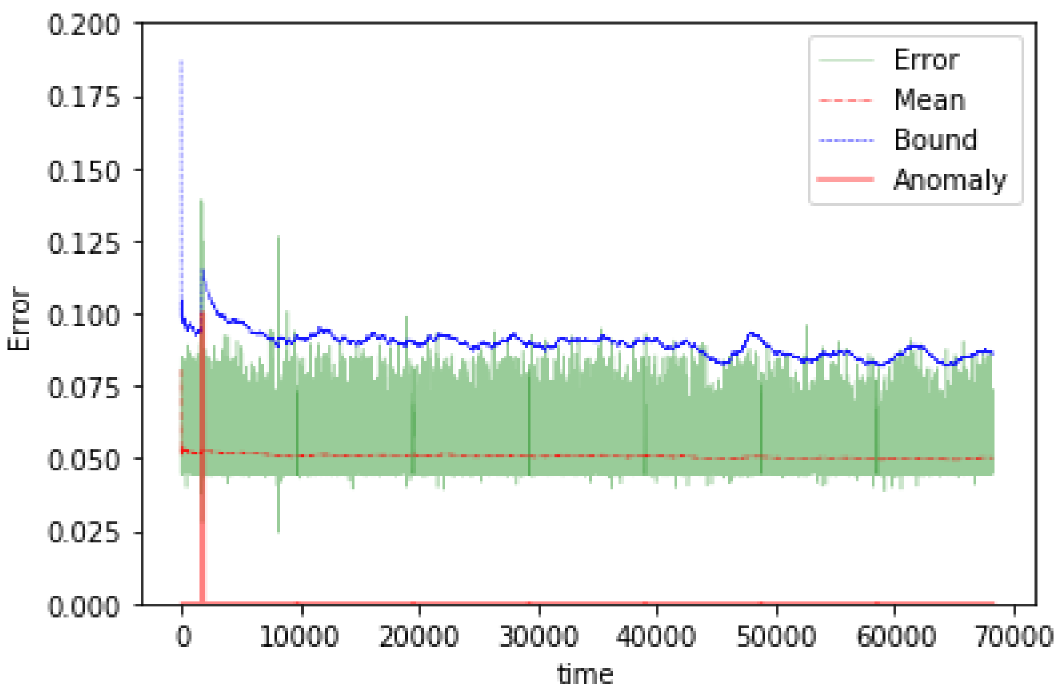
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



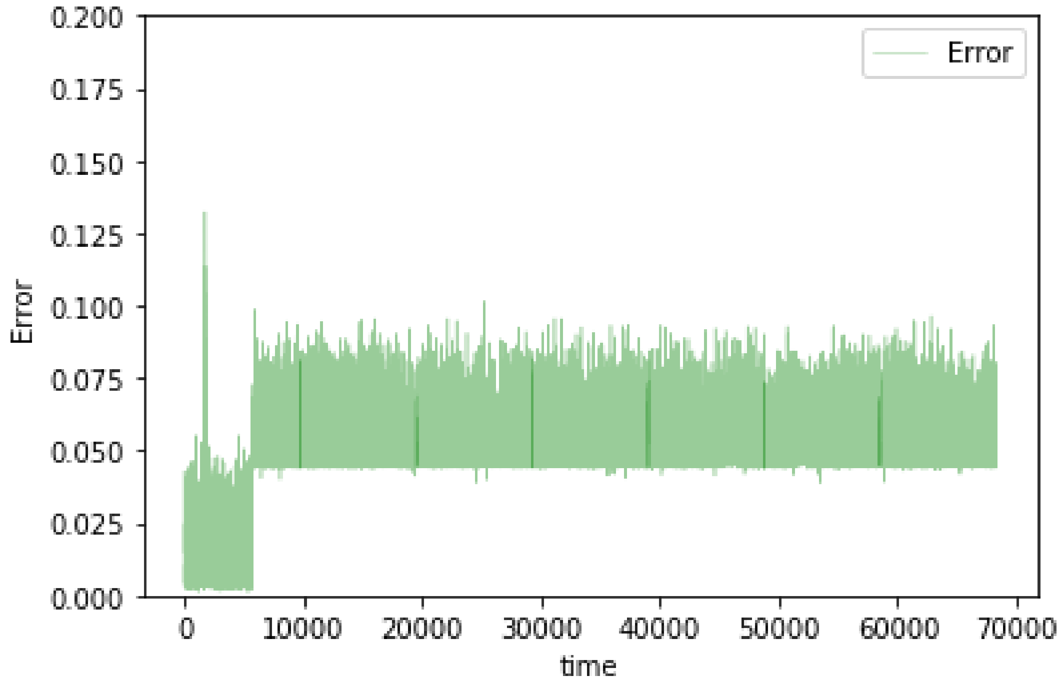

```
The mean error for gru1_10_disk_IO_start_ is 0.023003706276915312 for length 68289
Testing on Avg. load data.
```

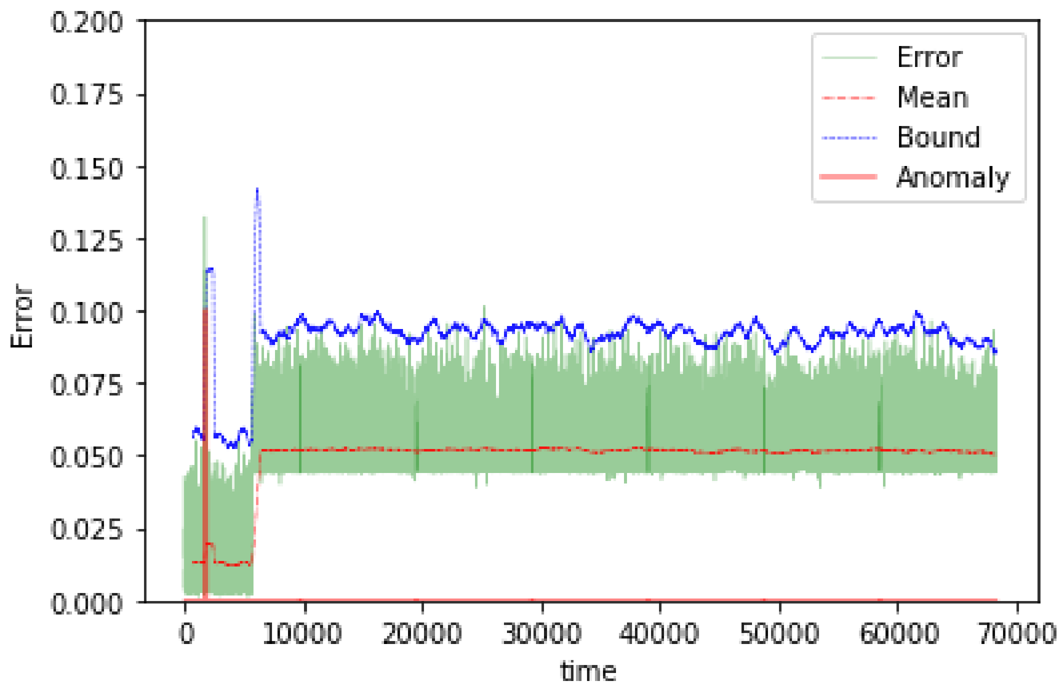


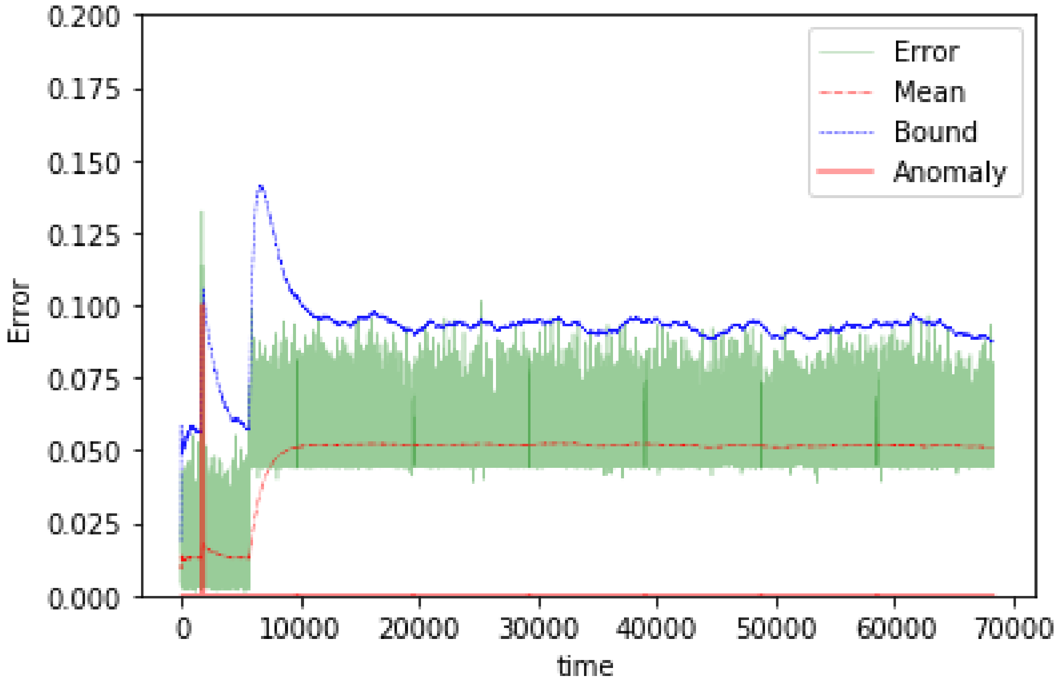

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

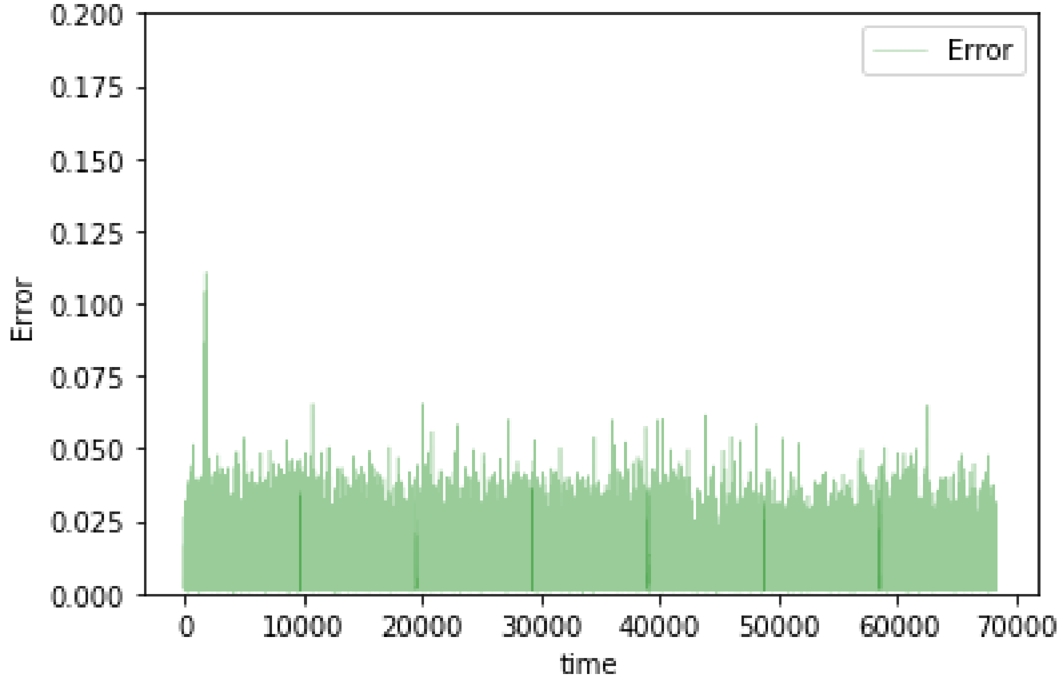
```
The mean error for gru1_10_avg_load_ is 0.05067138774458834 for length 68289
Testing on app change early data.
```



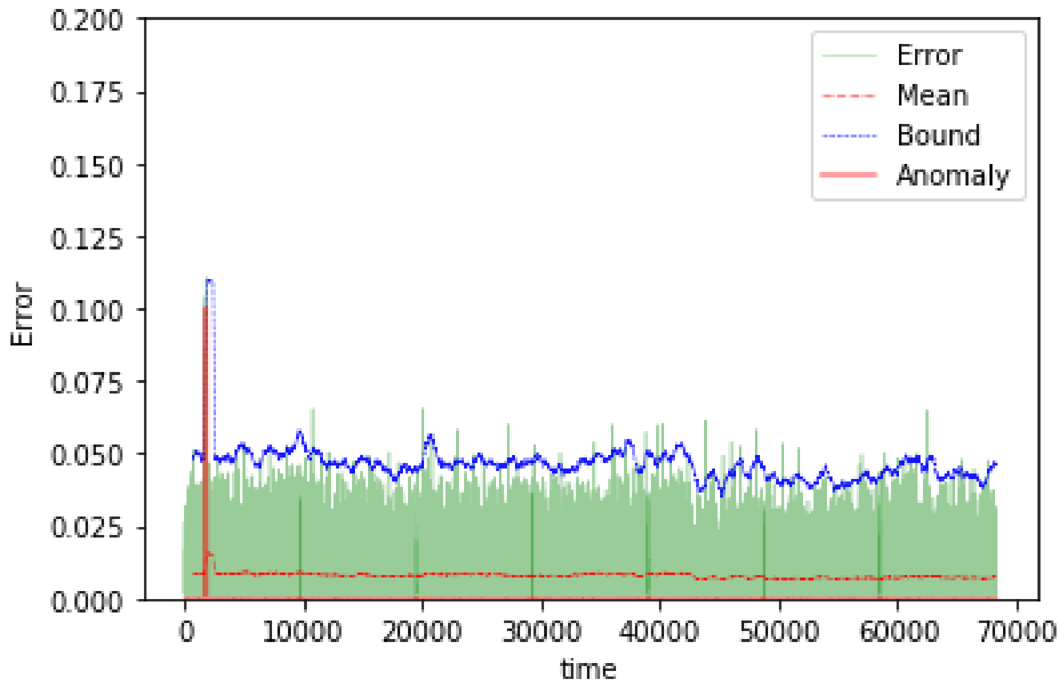




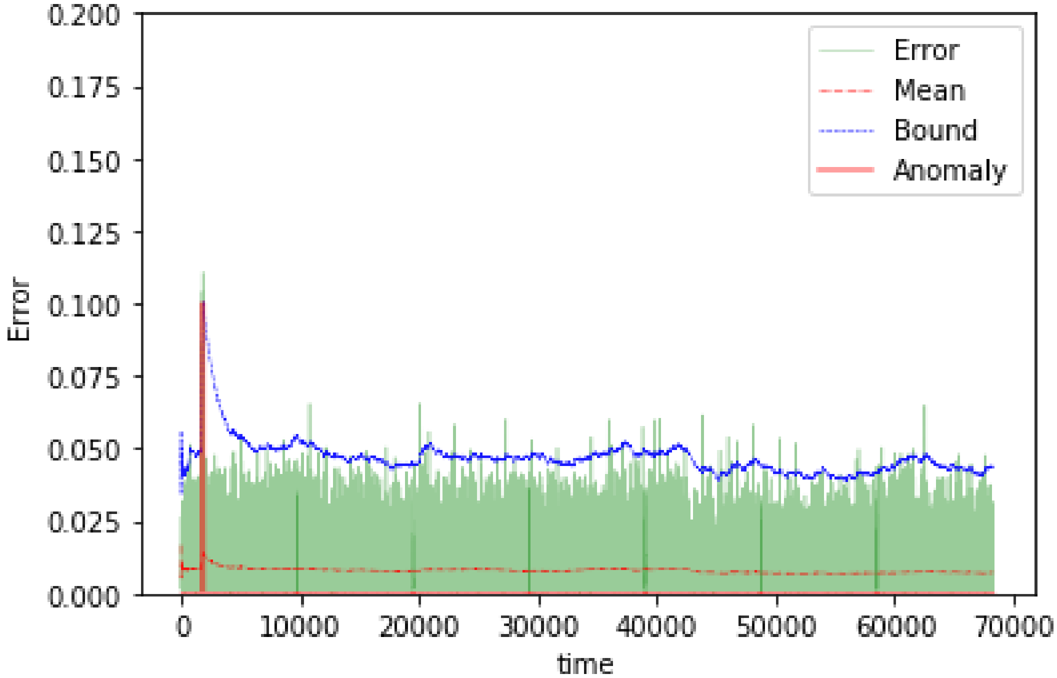

```
The mean error for gru1_10_app_change_early_ is 0.048560904075556445 for length 68289
Testing on Normal data.
```



0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error



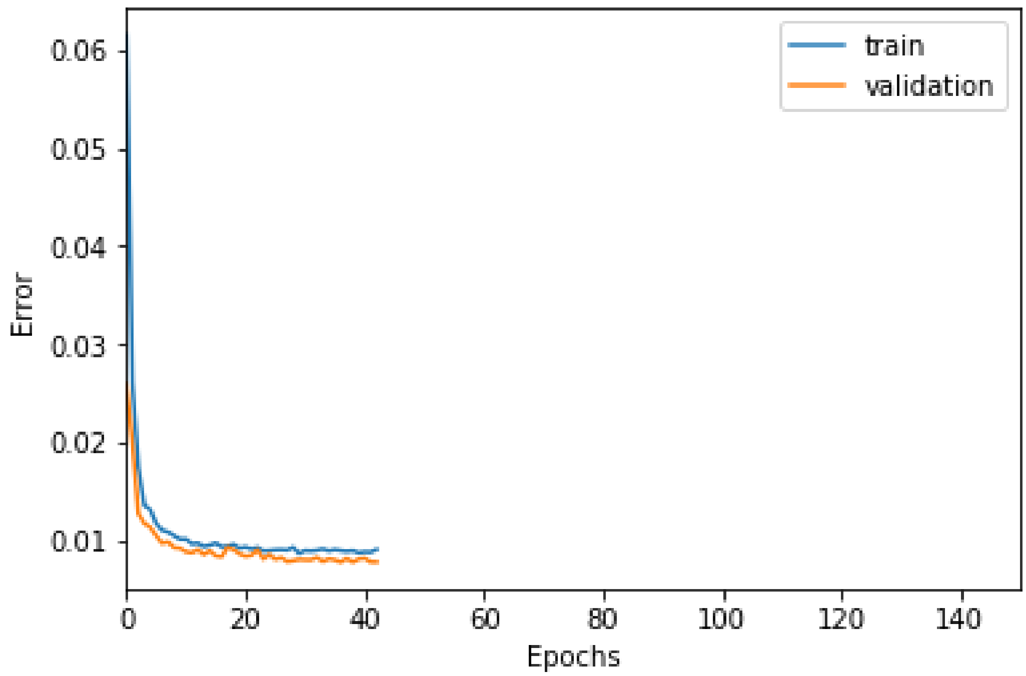
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly



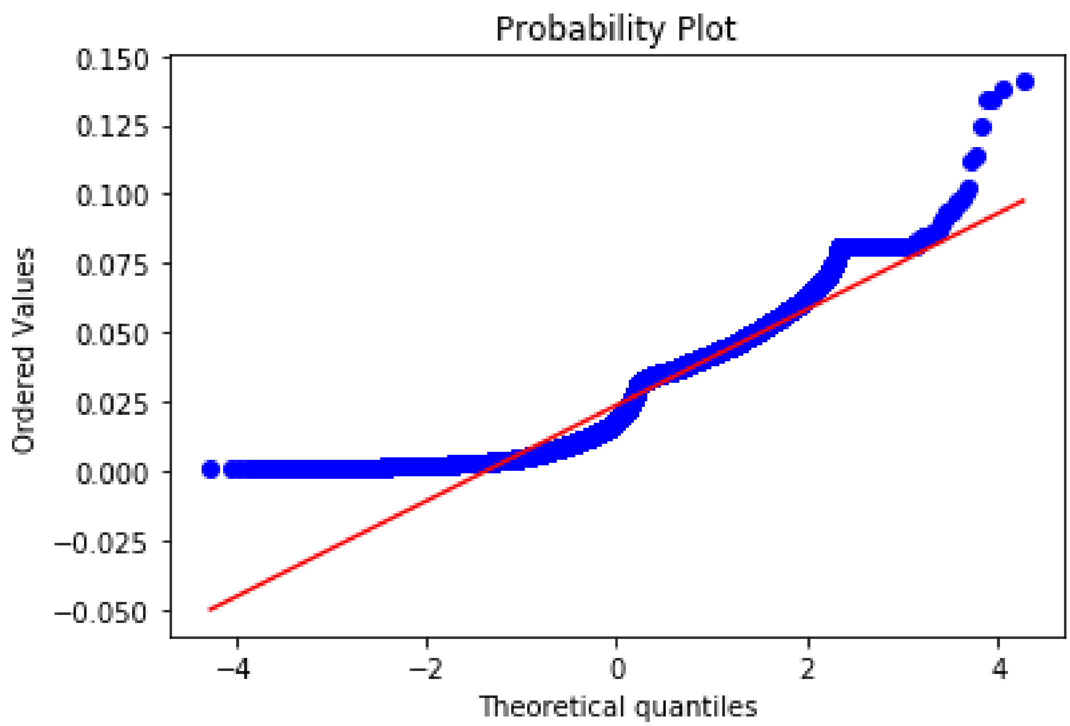

```
The mean error for gru1_10_normal_ is 0.00790596551006325 for length 68289
====================
```

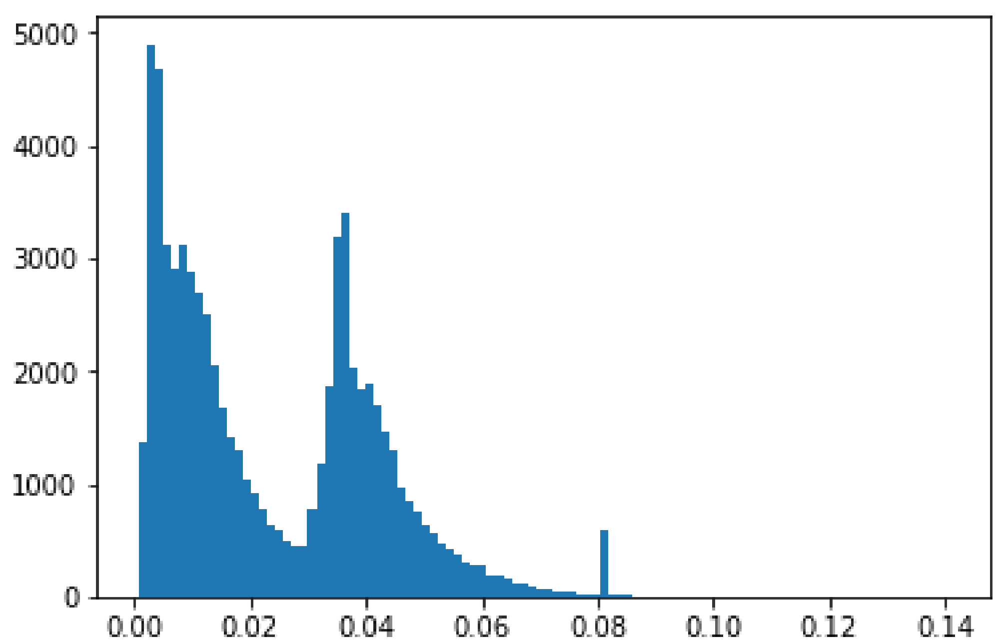
20 steps

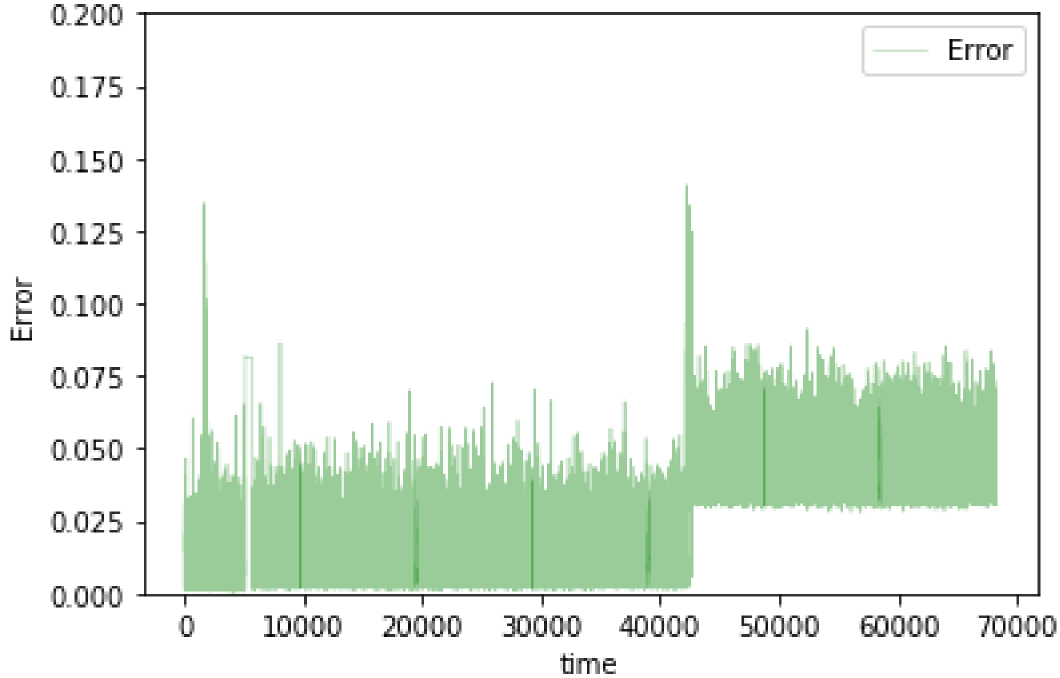
```
In [163]: TIMESTEPS = 20
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru1_20"

In [164]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu')(input_layer)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [165]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [166]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```



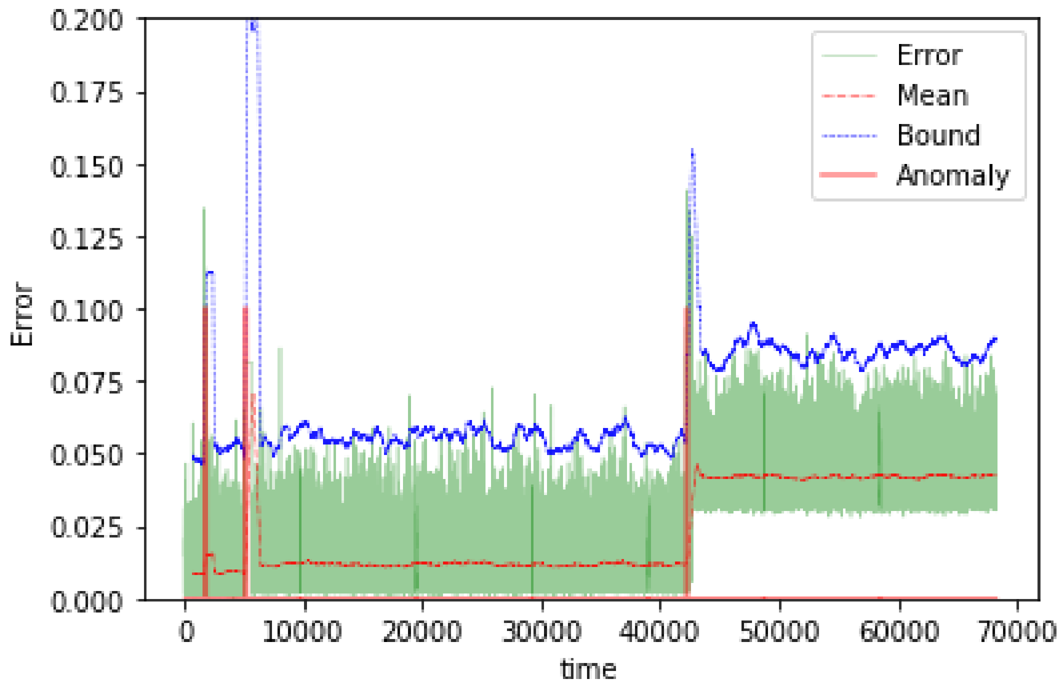

```
Training loss for final epoch is 0.00910140510473866
Validation loss for final epoch is 0.007817412640433759
---------- Beginning tests for gru1_20 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−4
−2
0
2
4

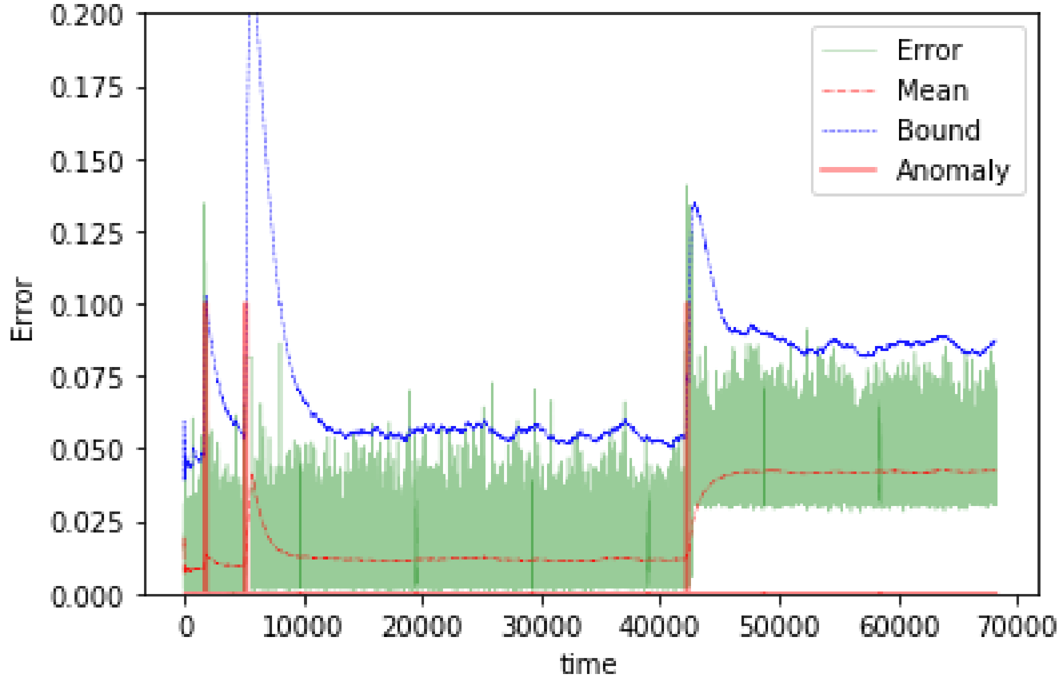

5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14



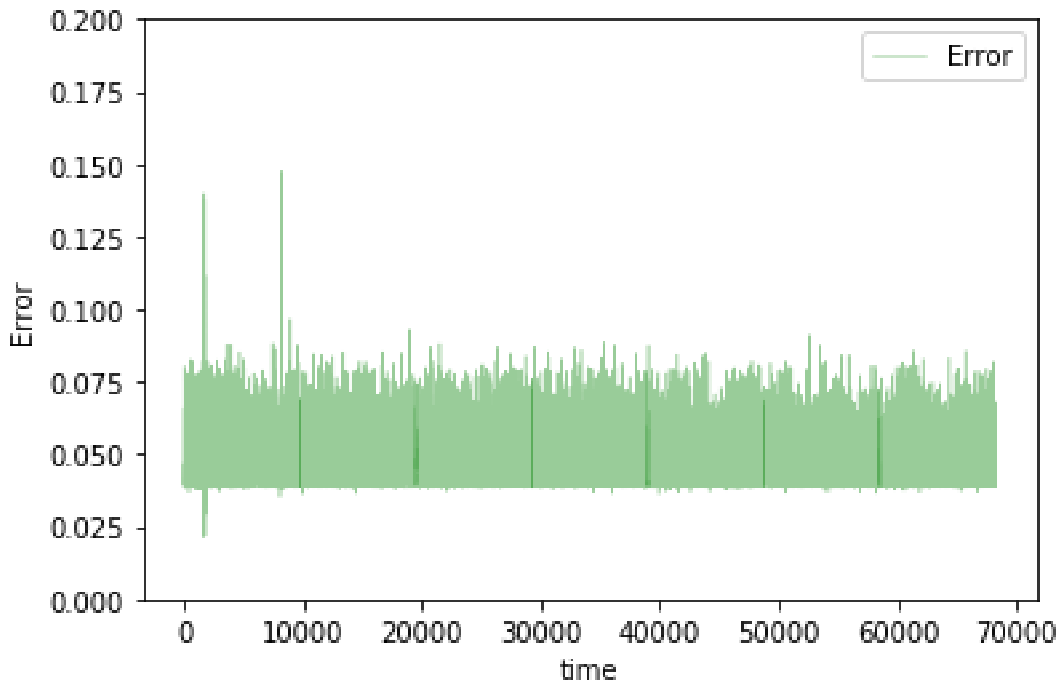
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

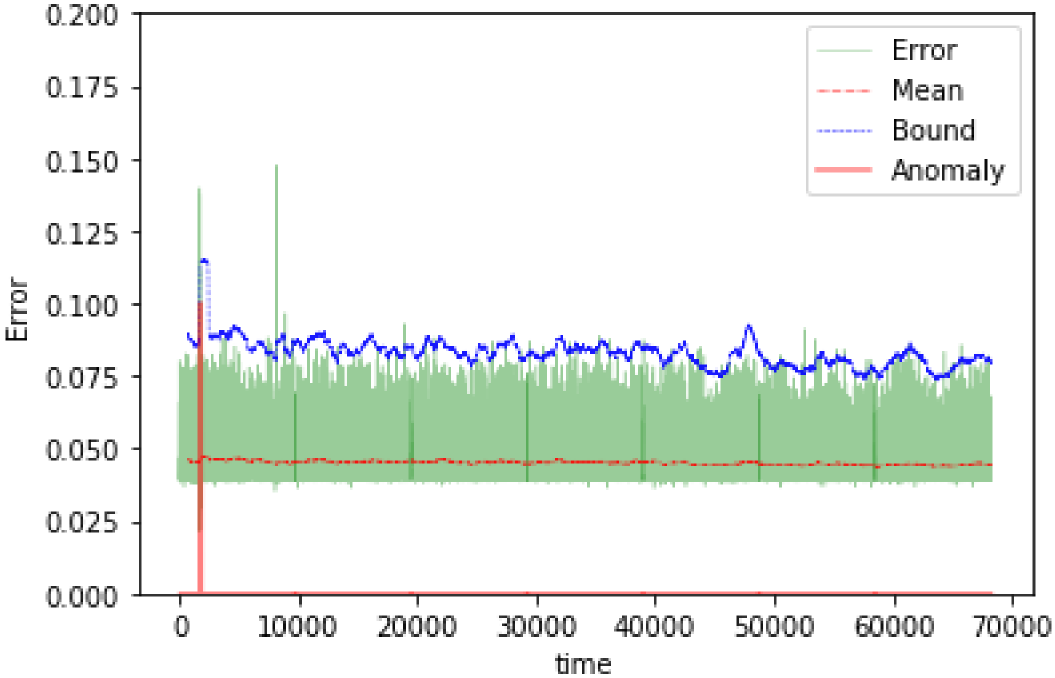


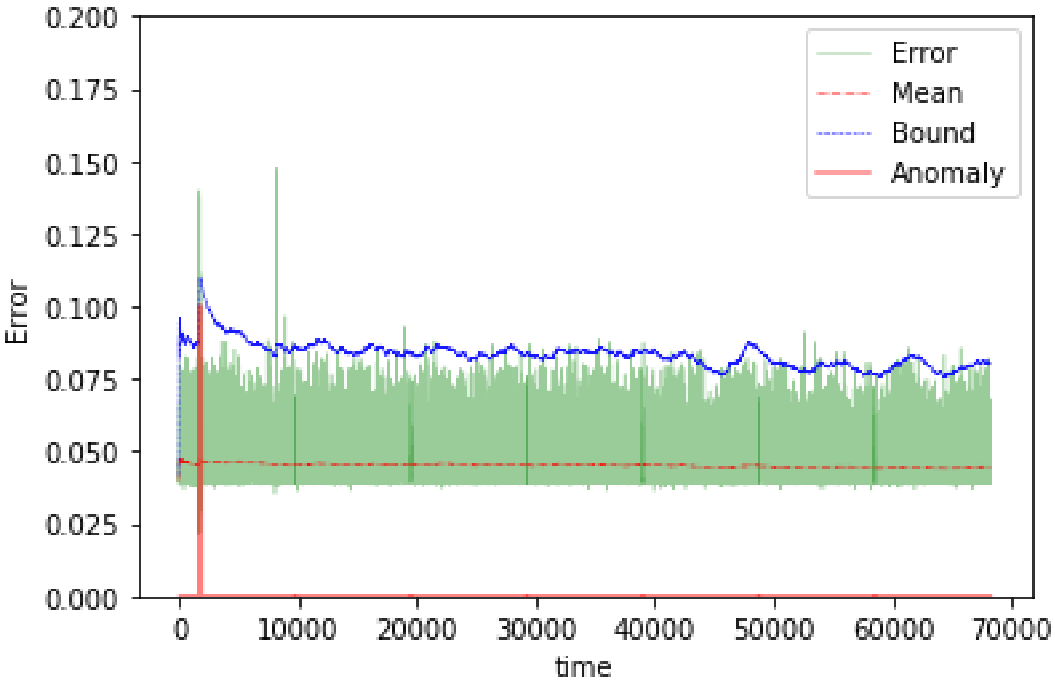
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



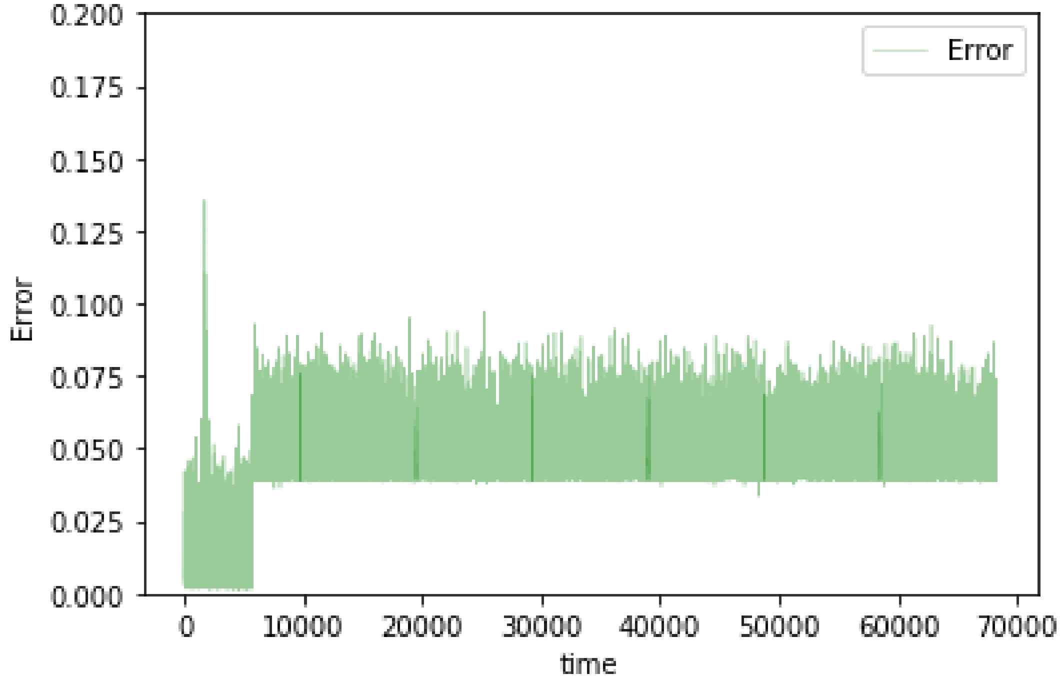

```
The mean error for gru1_20_disk_IO_start_ is 0.023751951840785414 for length 68279
Testing on Avg. load data.
```

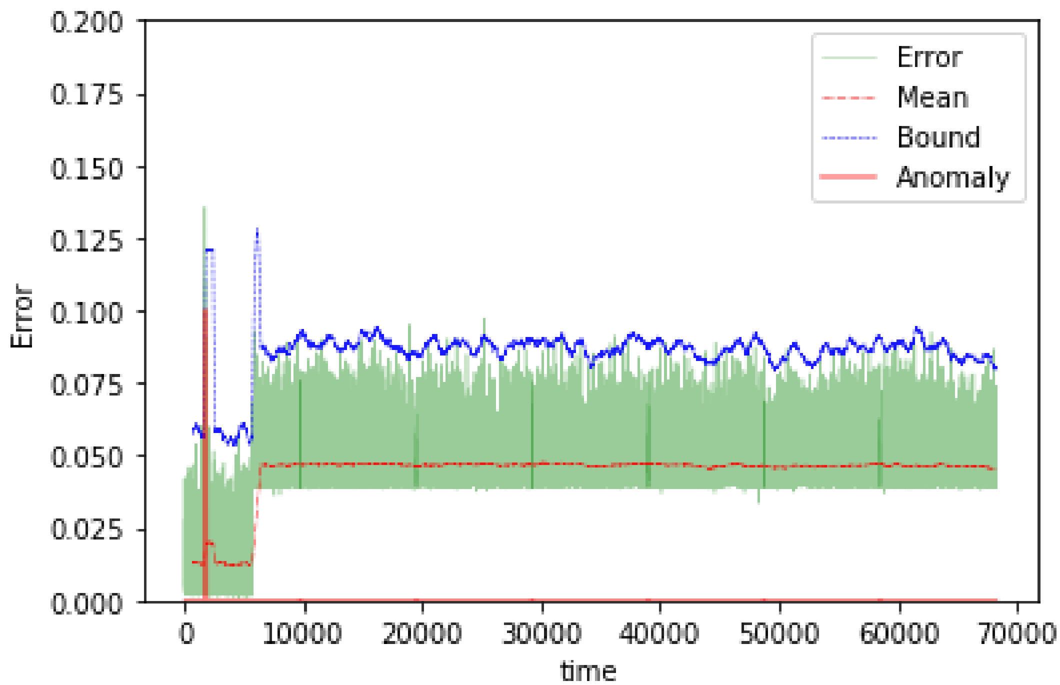


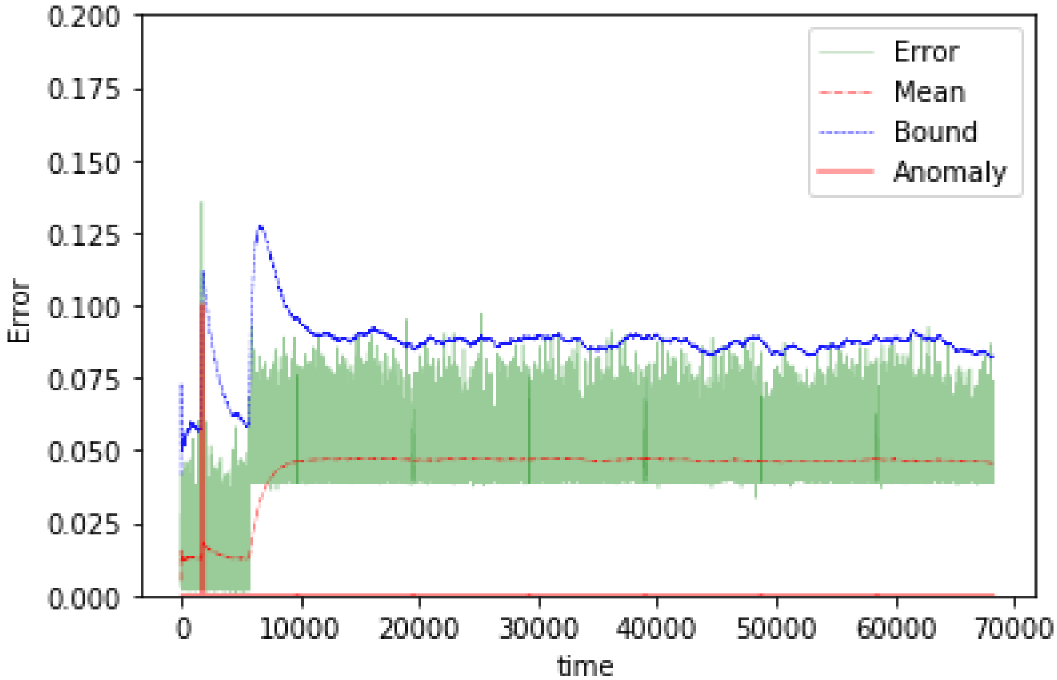

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

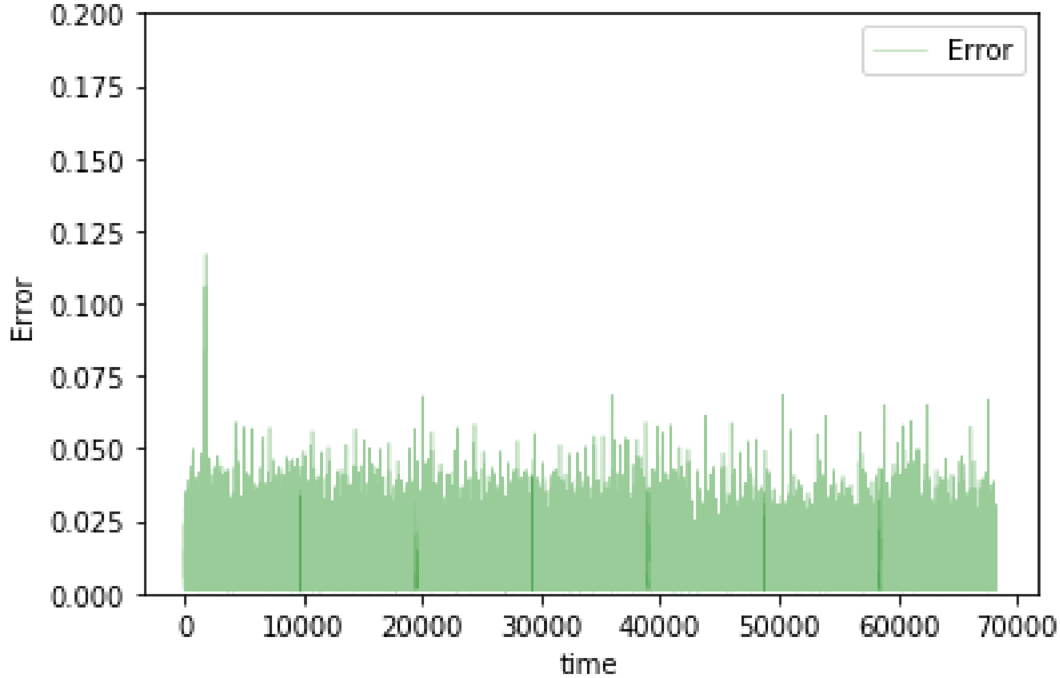
```
The mean error for gru1_20_avg_load_ is 0.04510235397174501 for length 68279
Testing on app change early data.
```



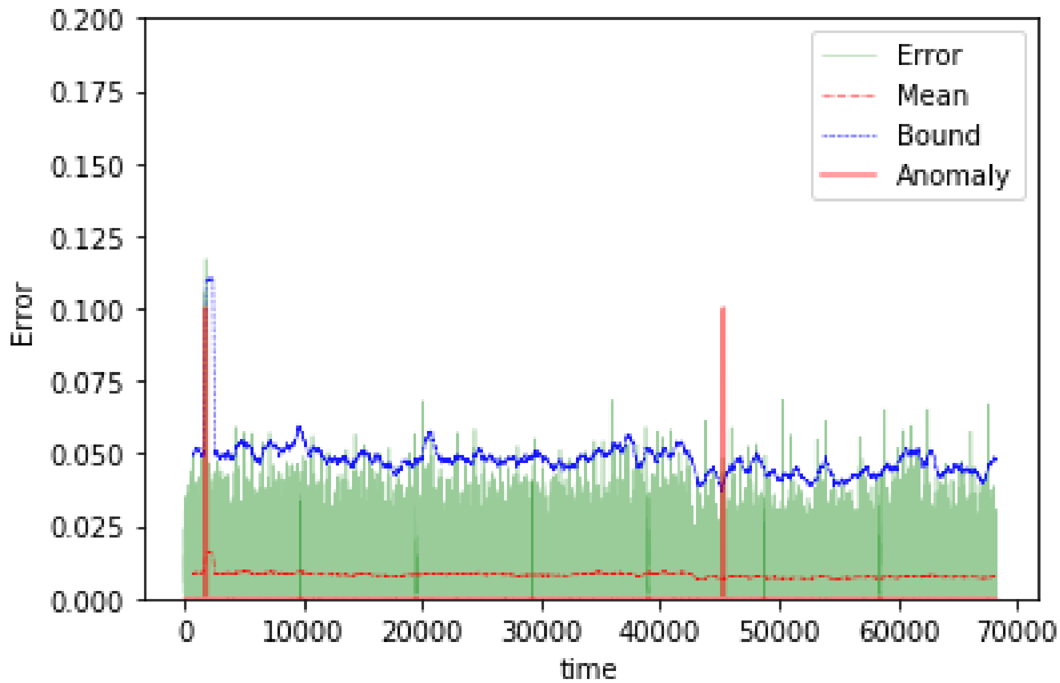




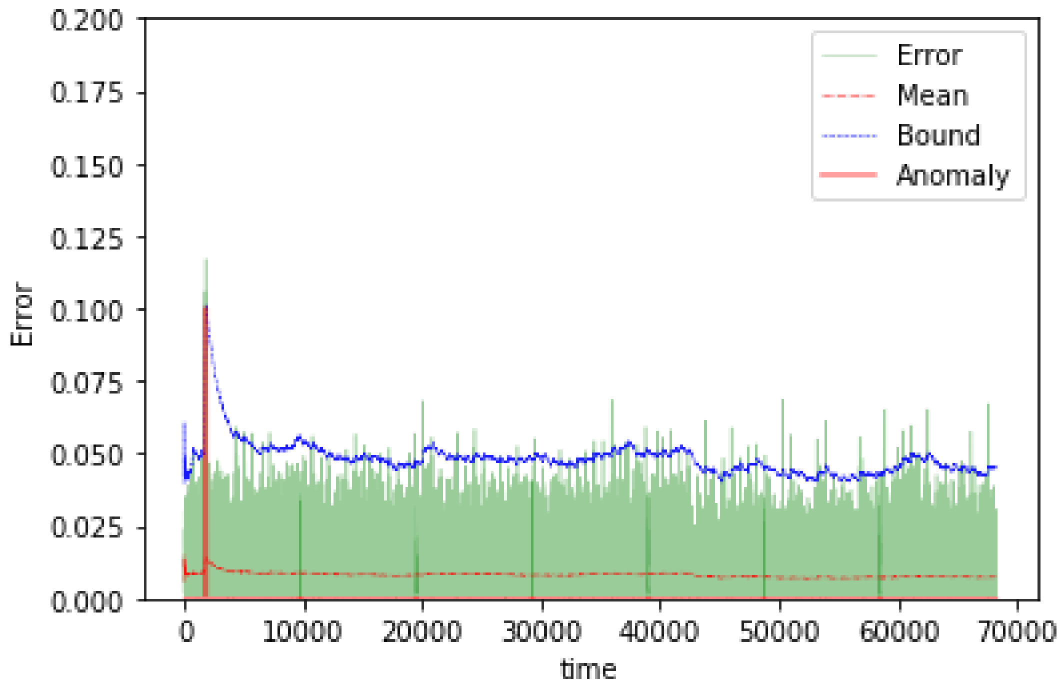

```
The mean error for gru1_20_app_change_early_ is 0.04388599918228171 for length 68279
Testing on Normal data.
```



Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



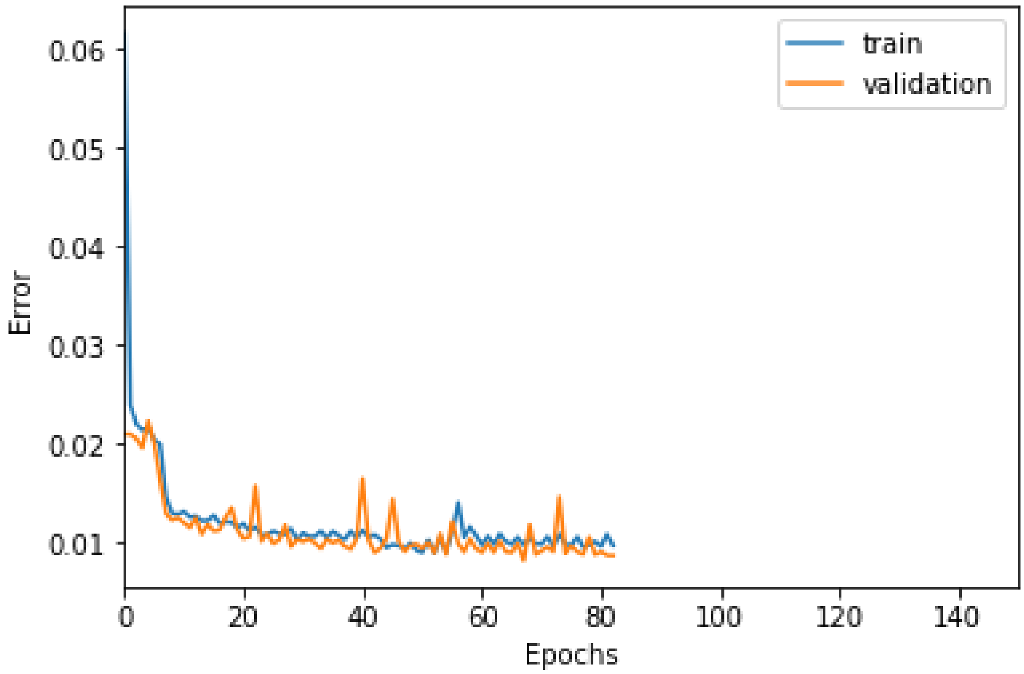
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



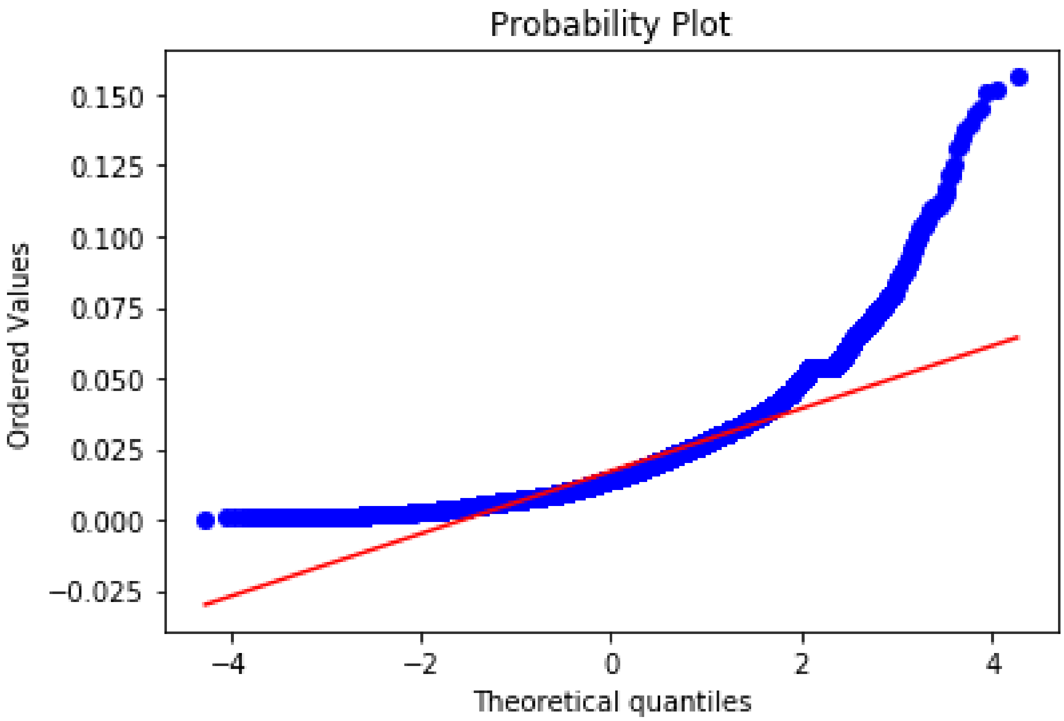

```
The mean error for gru1_20_normal_ is 0.008096794426425791 for length 68279
====================
```

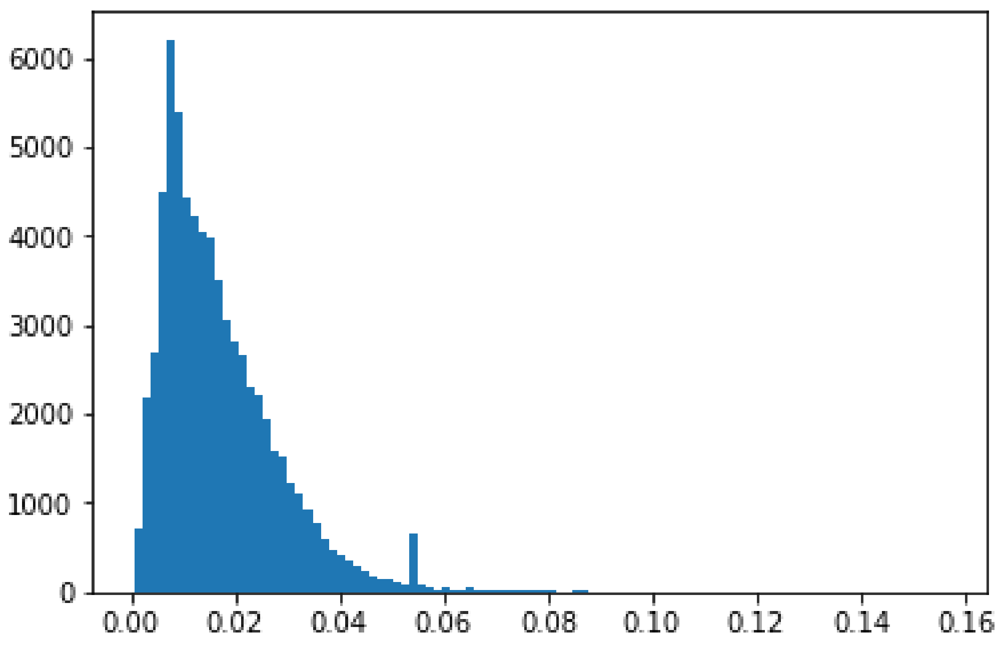
50 steps

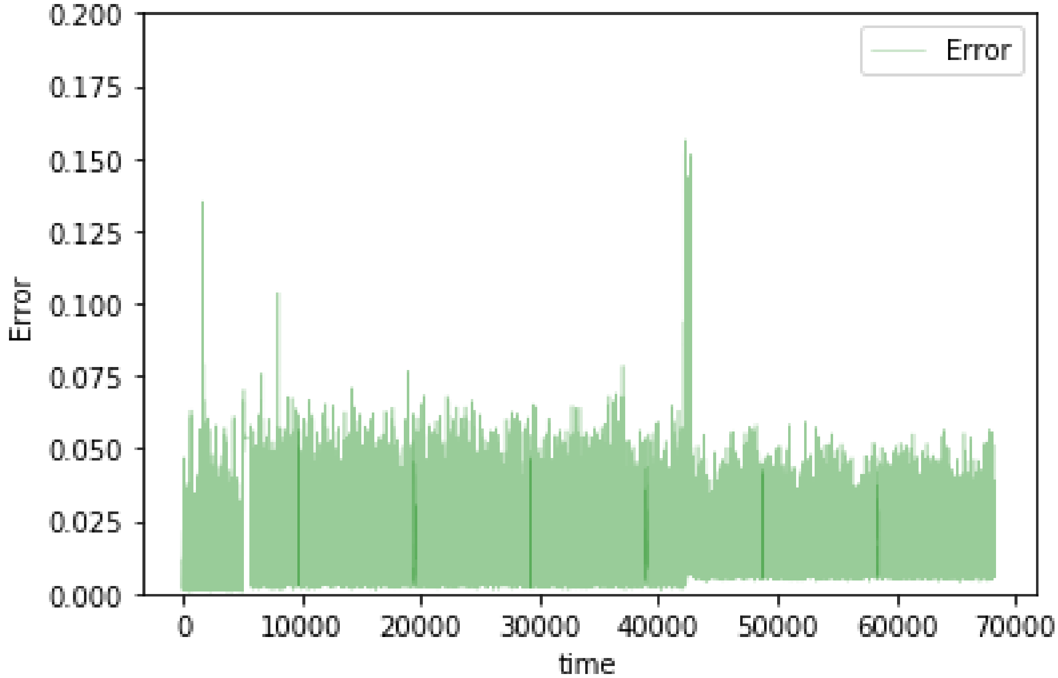
```
In [167]: TIMESTEPS = 50
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru1_50"

In [168]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu')(input_layer)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [169]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [170]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

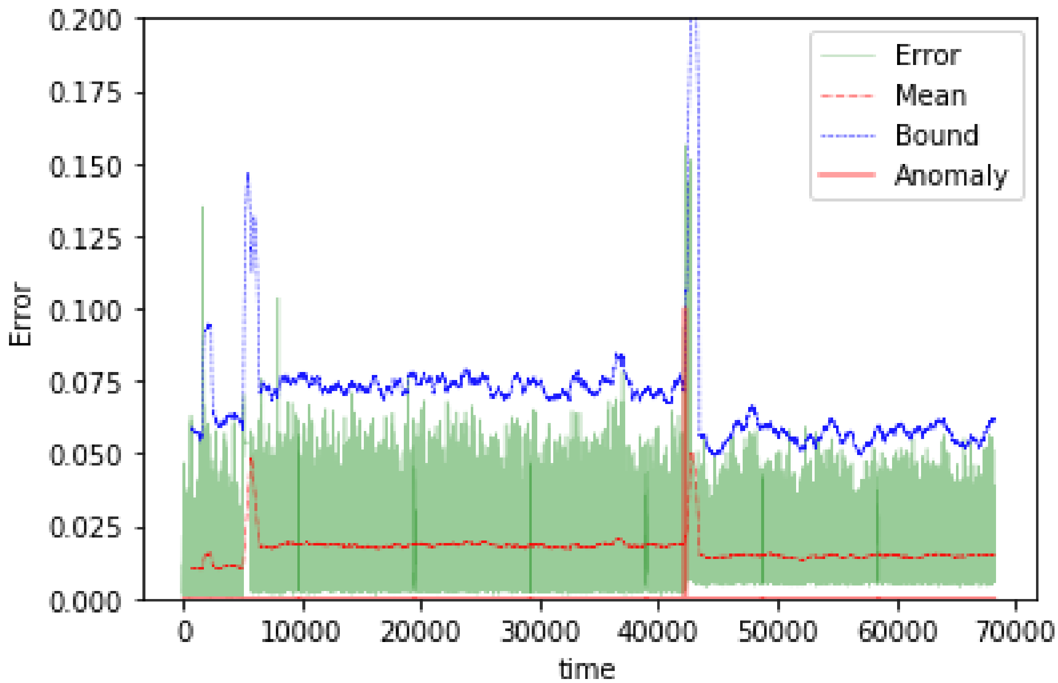


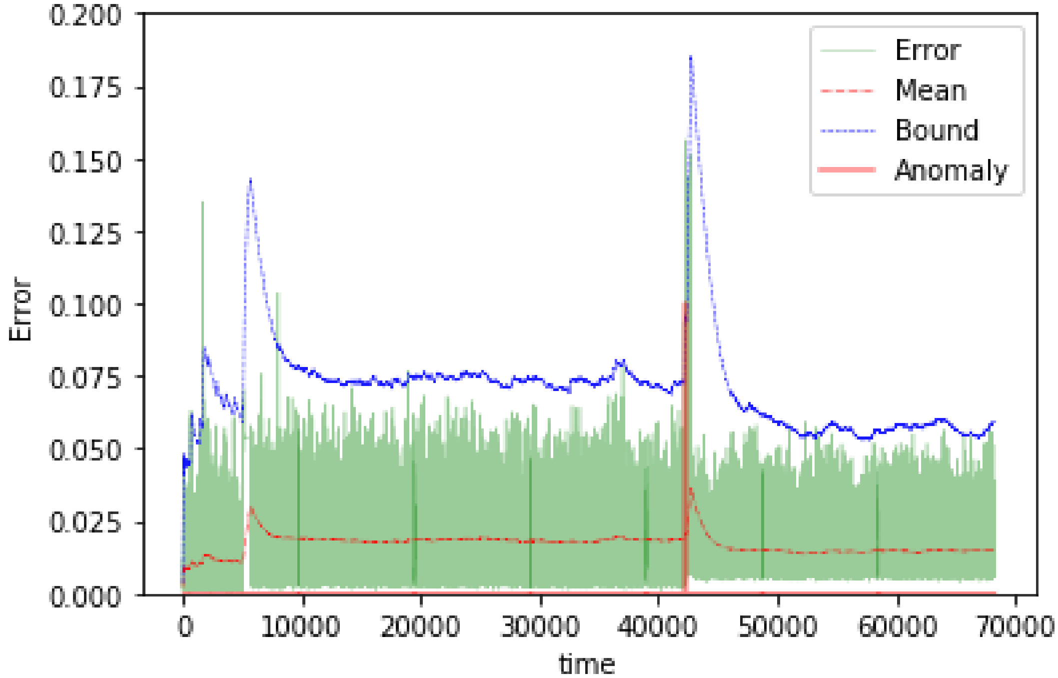

```
Training loss for final epoch is 0.009711743056424893
Validation loss for final epoch is 0.008679575216956437
---------- Beginning tests for gru1_50 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

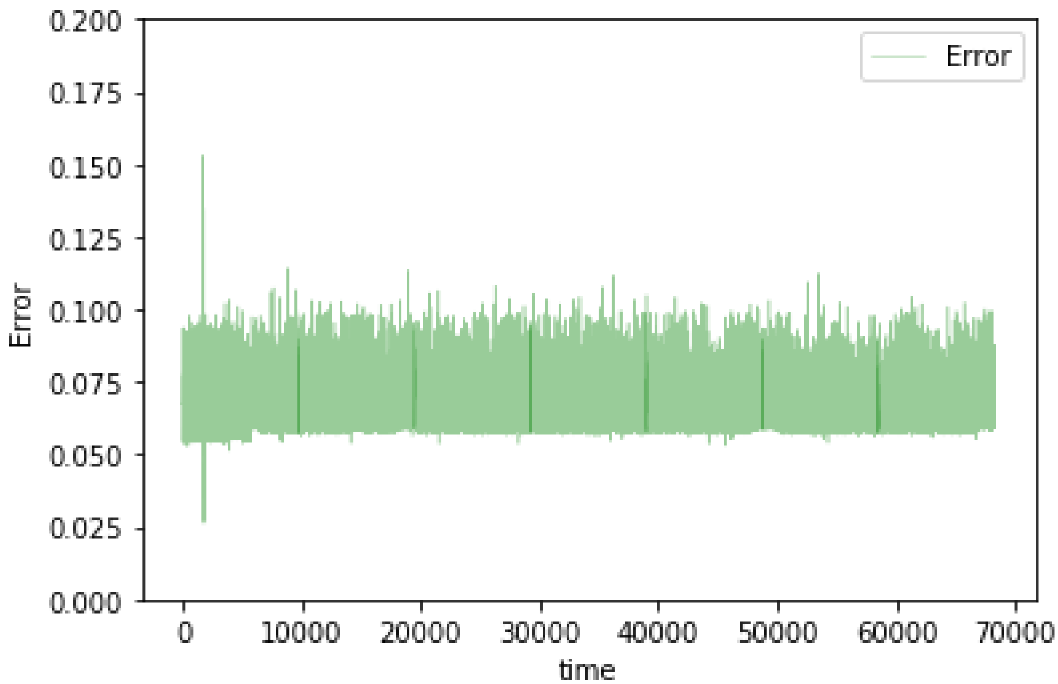

6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16

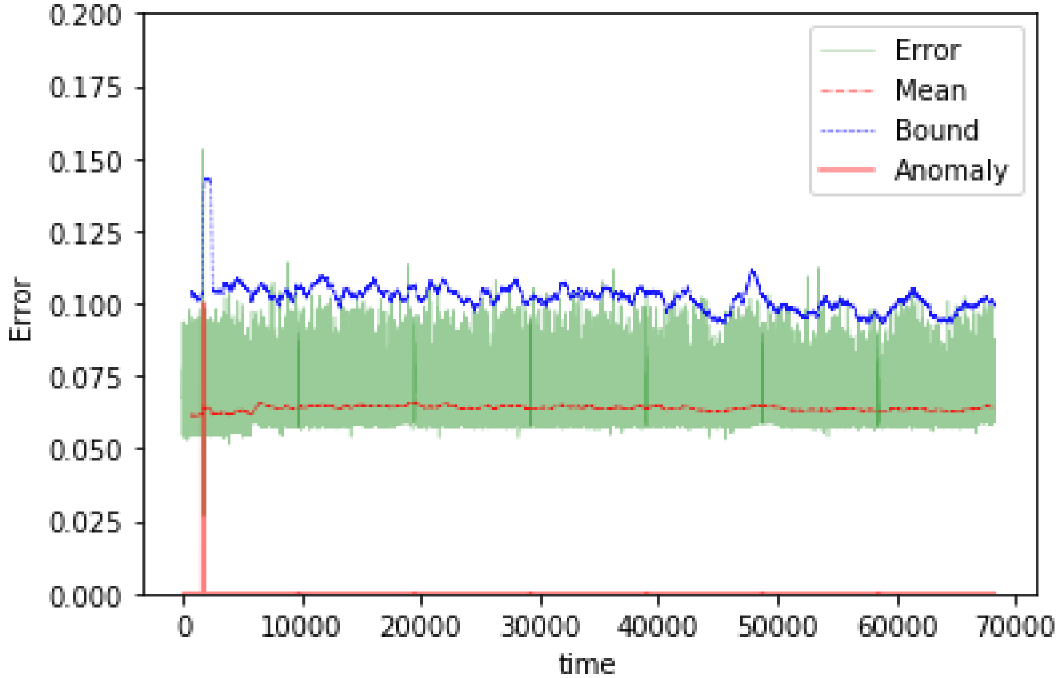

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

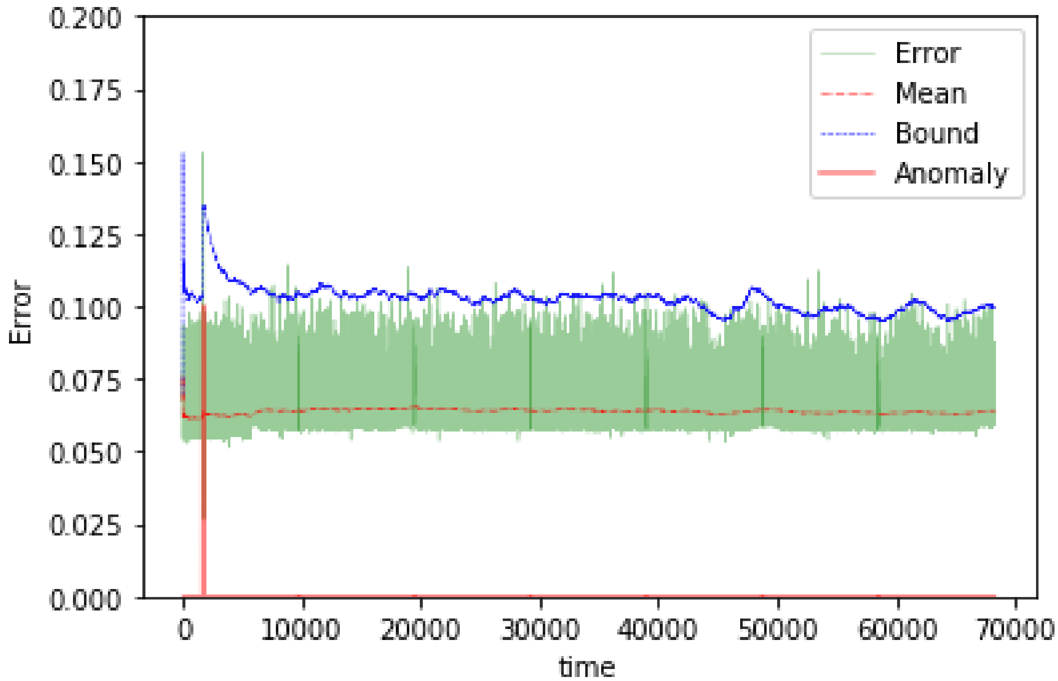

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



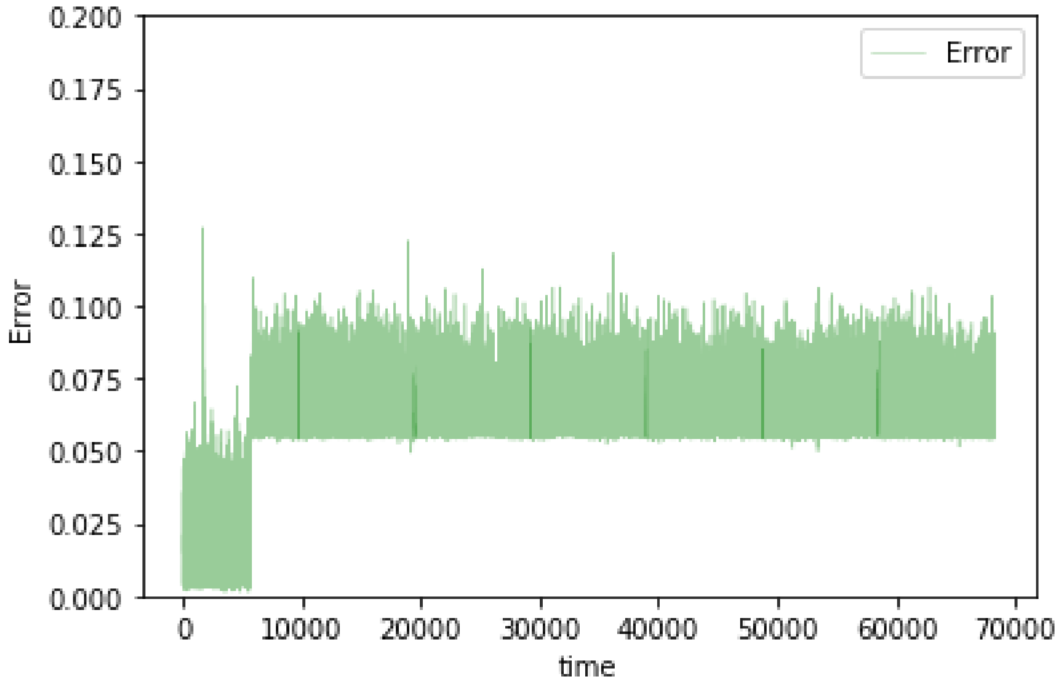

```
The mean error for gru1_50_disk_IO_start_ is 0.017100427512291436 for length 68249
Testing on Avg. load data.
```

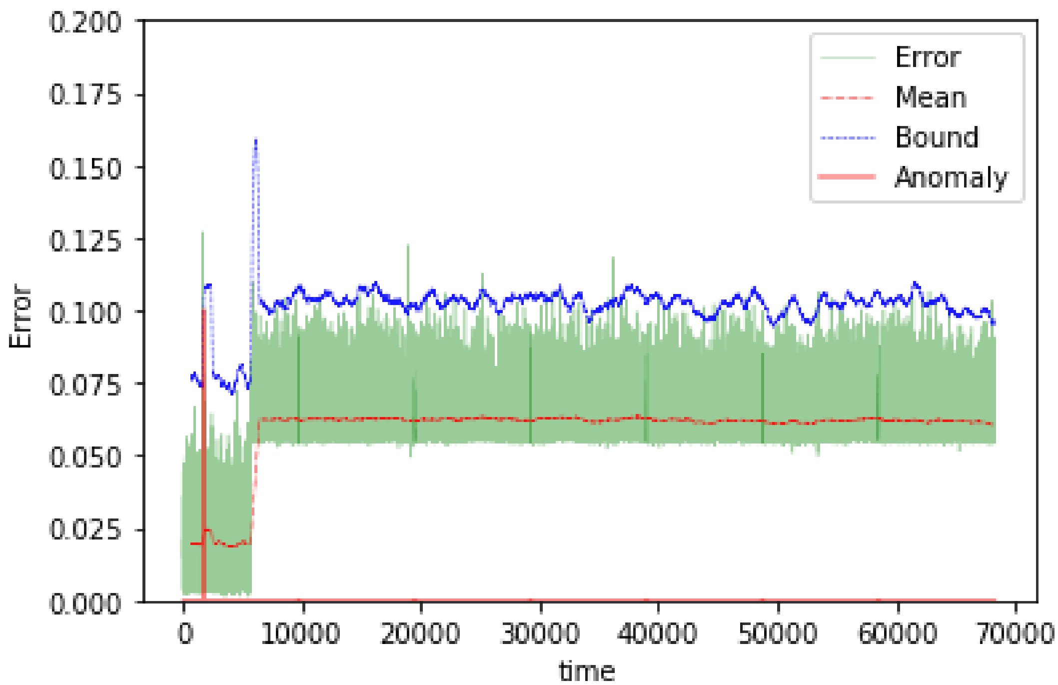


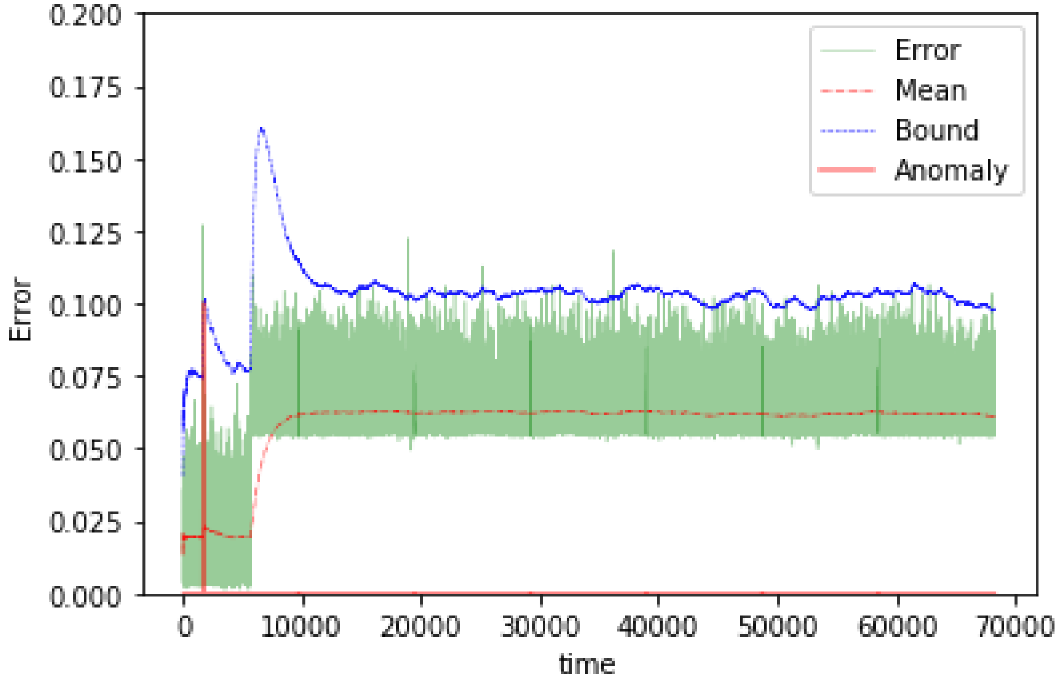

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

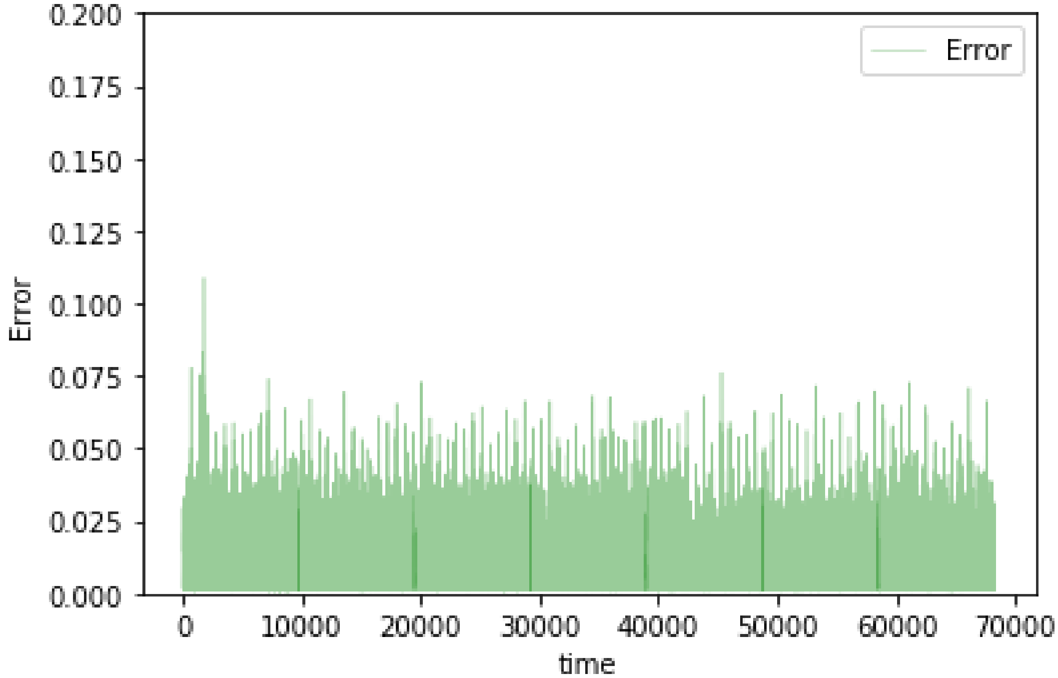
```
The mean error for gru1_50_avg_load_ is 0.06384542983166507 for length 68249
Testing on app change early data.
```

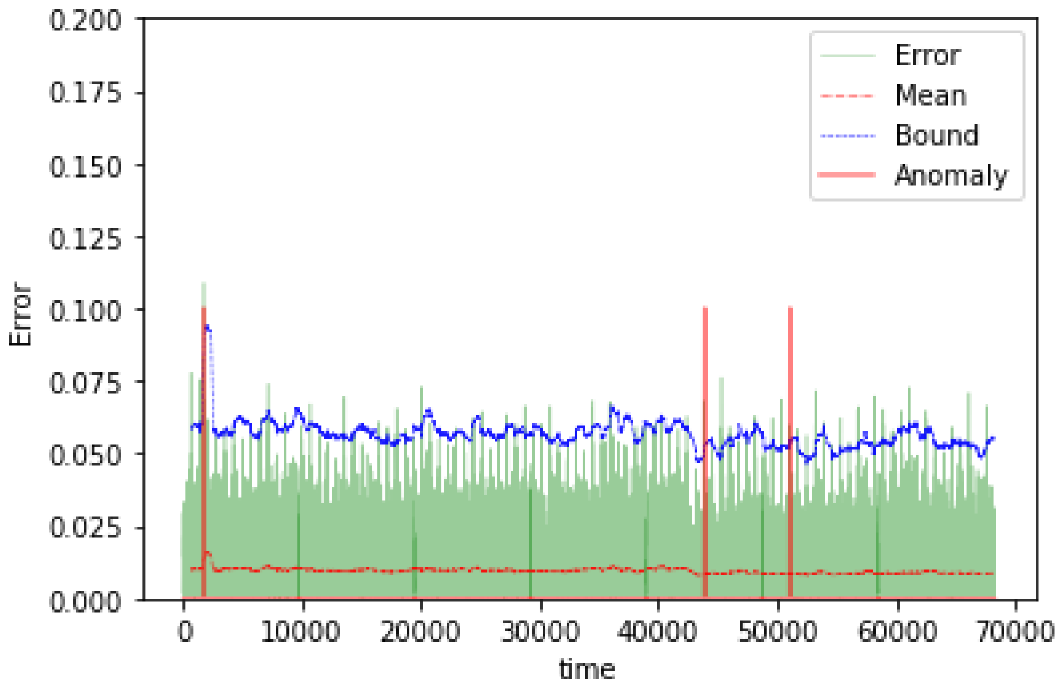


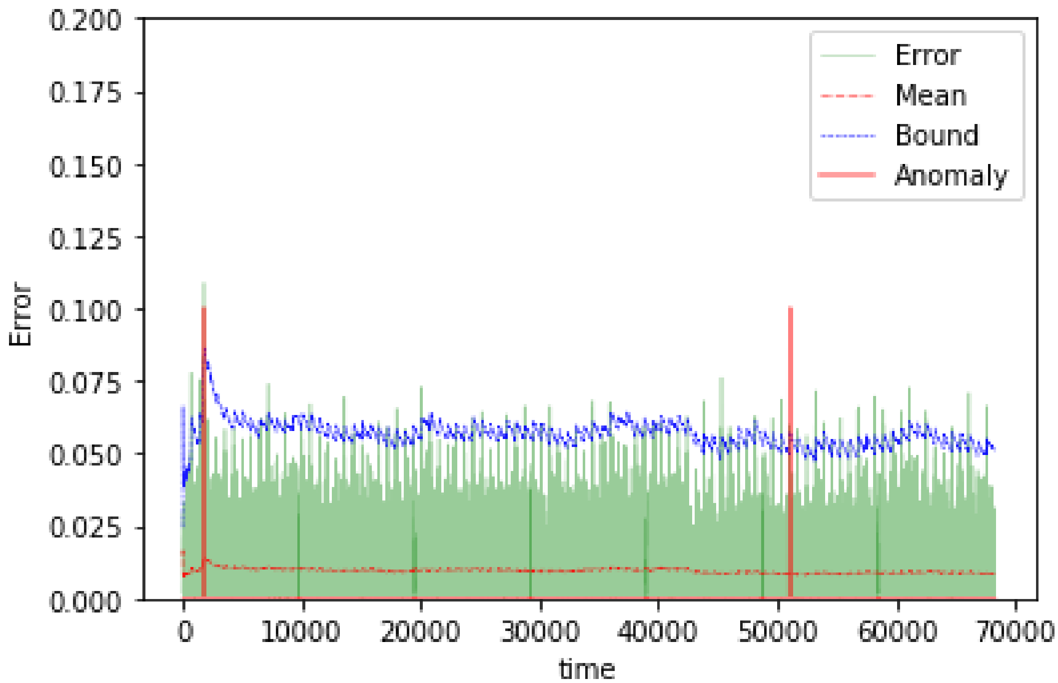





```
The mean error for gru1_50_app_change_early_ is 0.05864576726478017 for length 68249
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

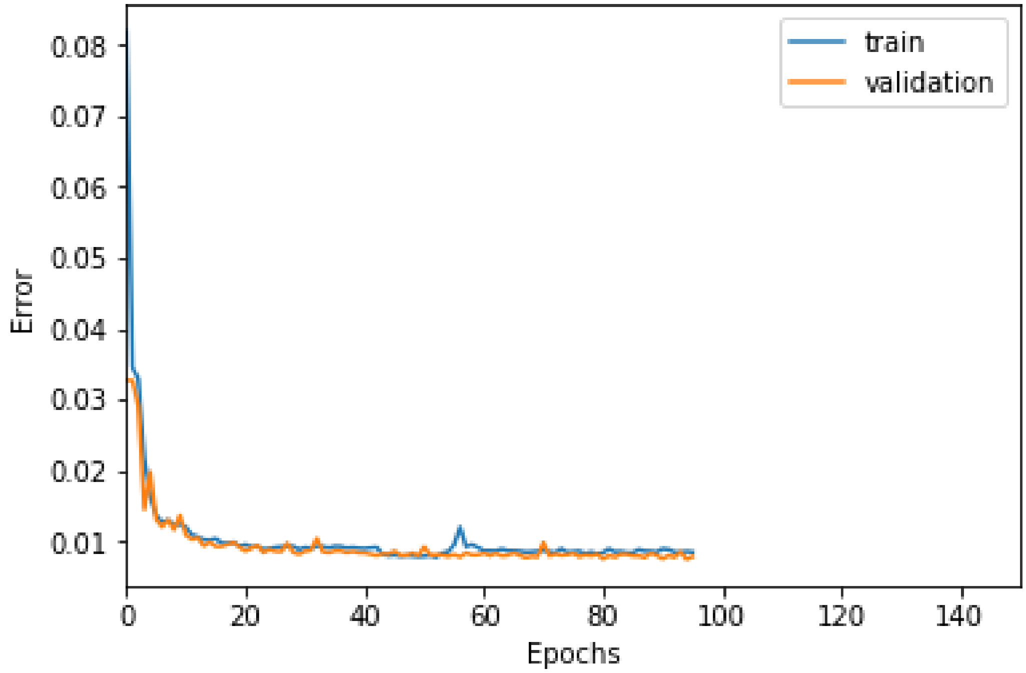

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



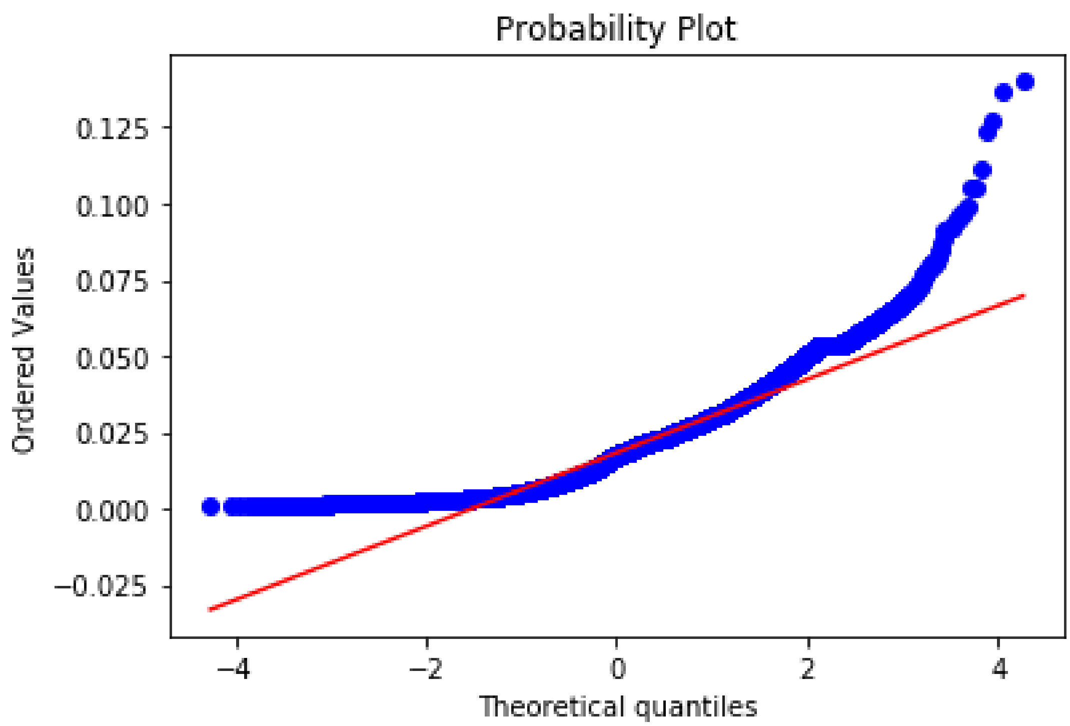

```
The mean error for gru1_50_normal_ is 0.009517379304388564 for length 68249
====================
```

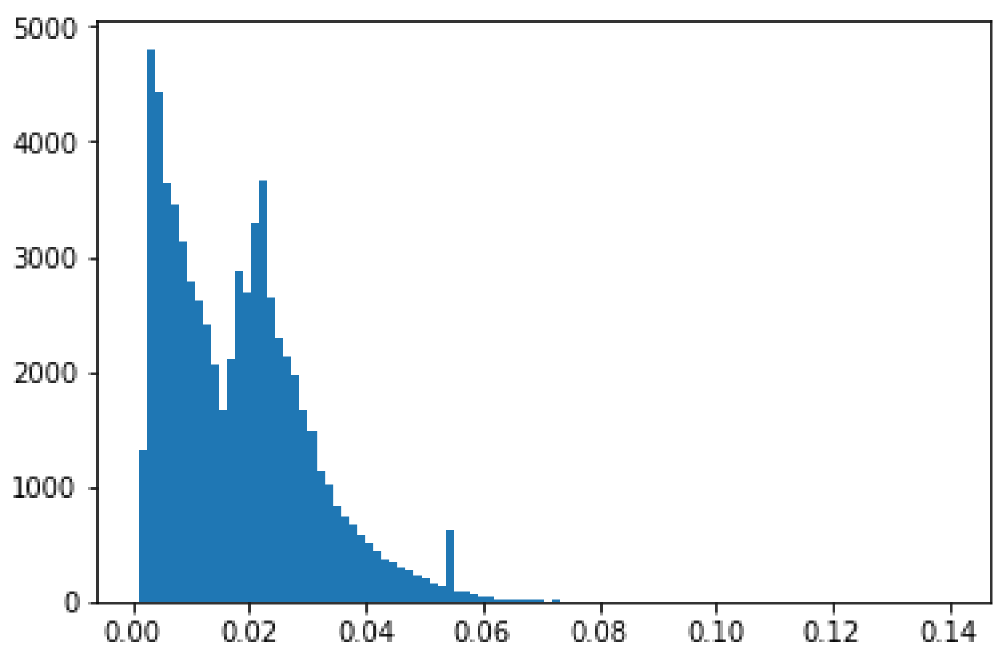
100 steps

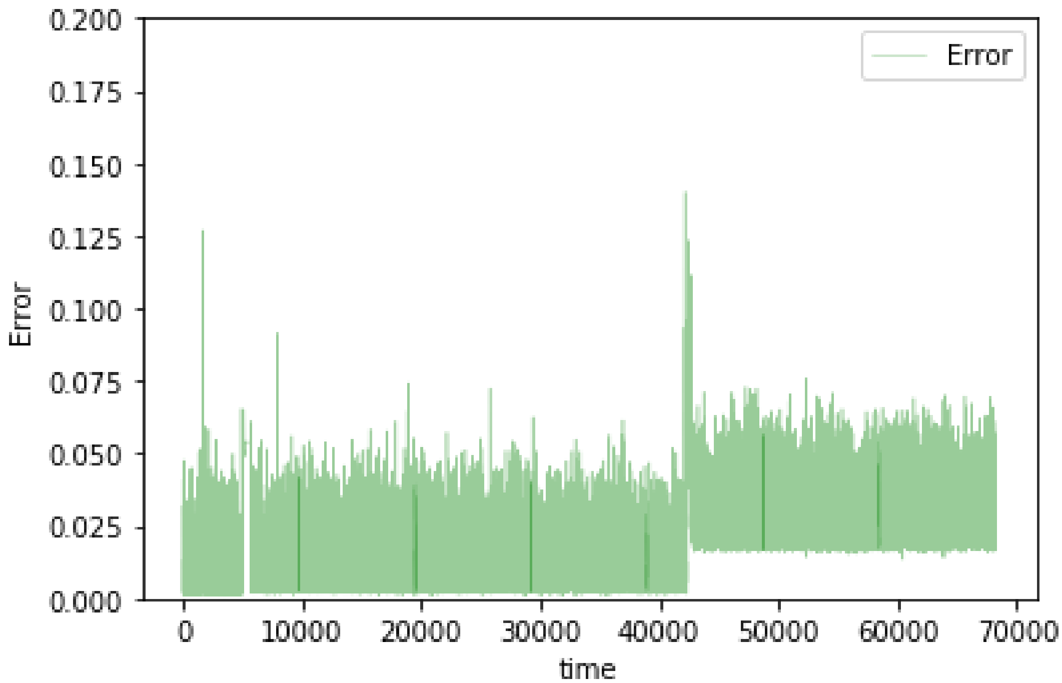
```
In [171]: TIMESTEPS = 100
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru1_100"

In [172]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu')(input_layer)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [173]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [174]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

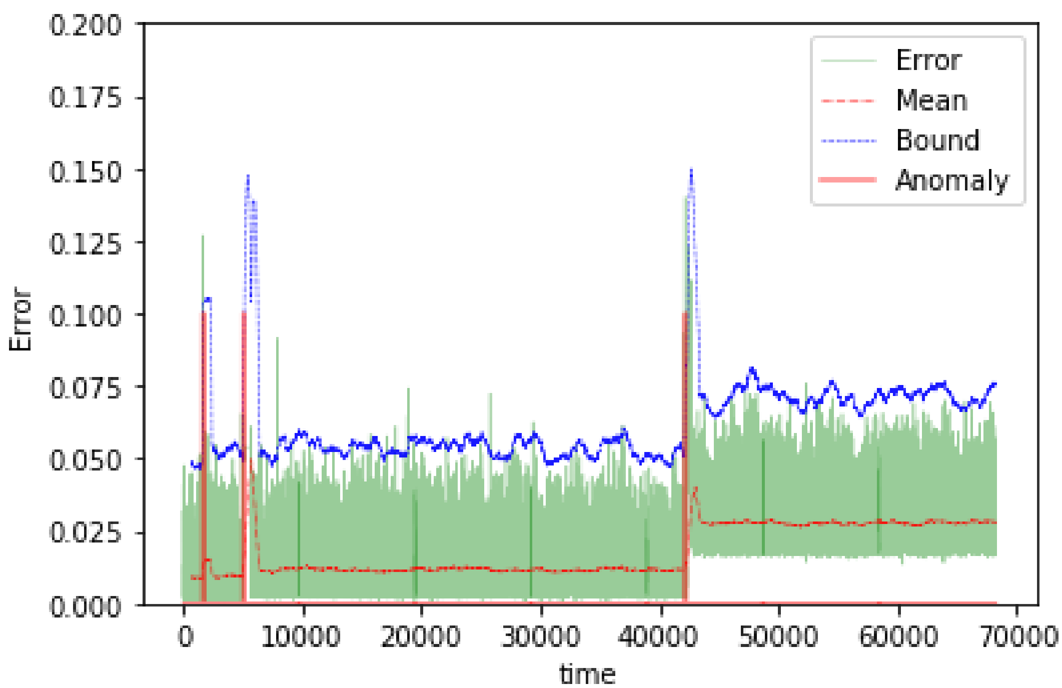


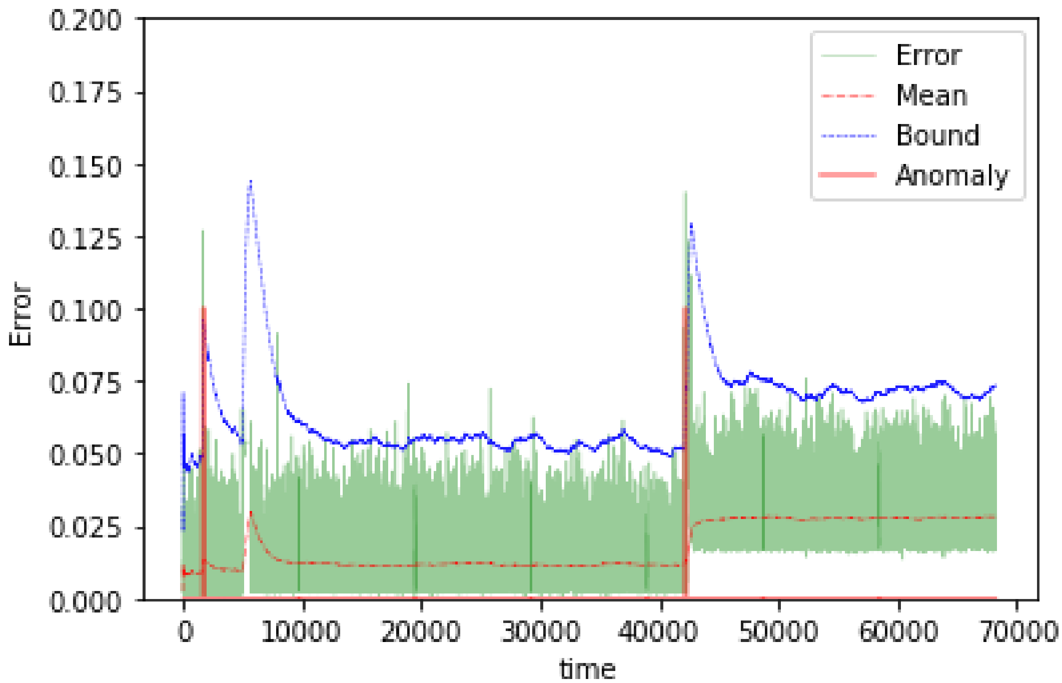

```
Training loss for final epoch is 0.008486375066800974
Validation loss for final epoch is 0.007837170831742696
---------- Beginning tests for gru1_100 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−4
−2
0
2
4
Theoretical quantiles

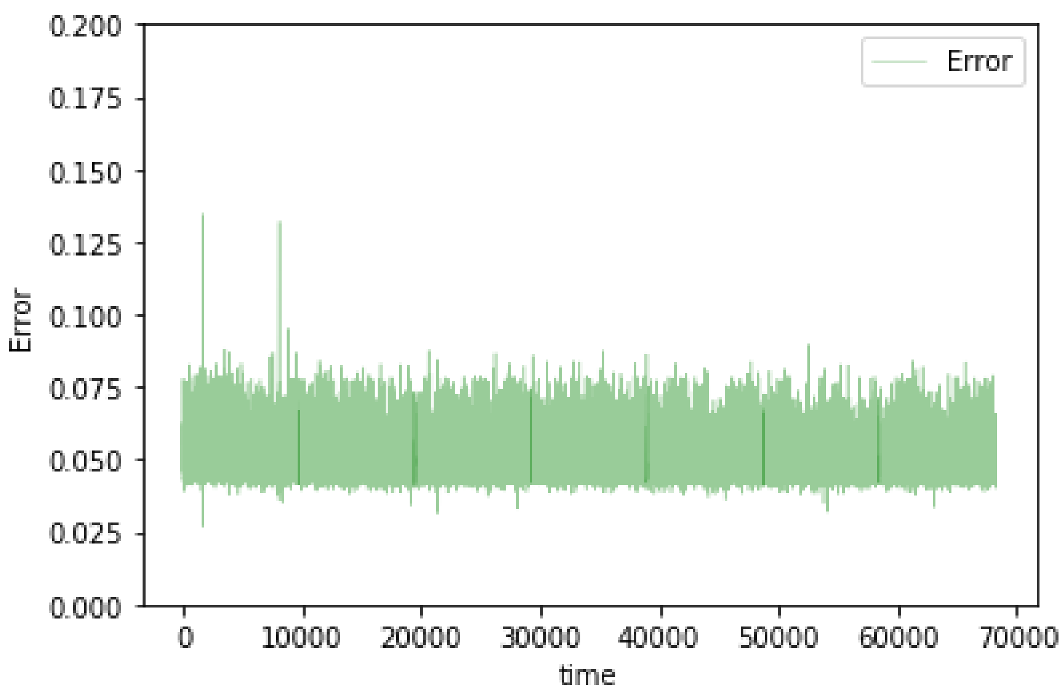

5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

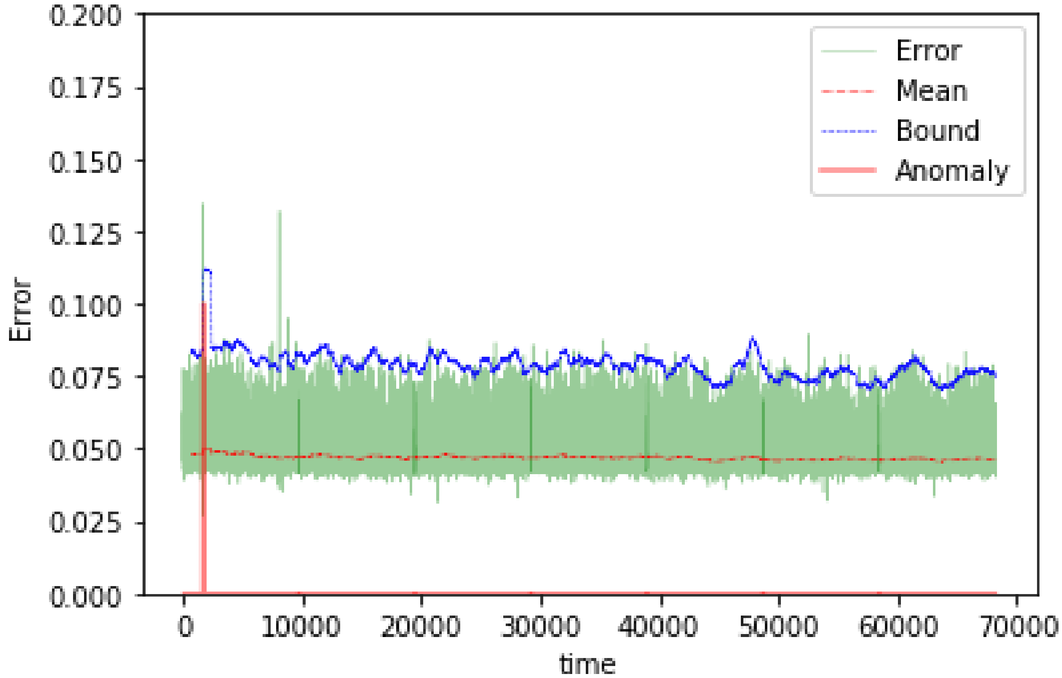

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

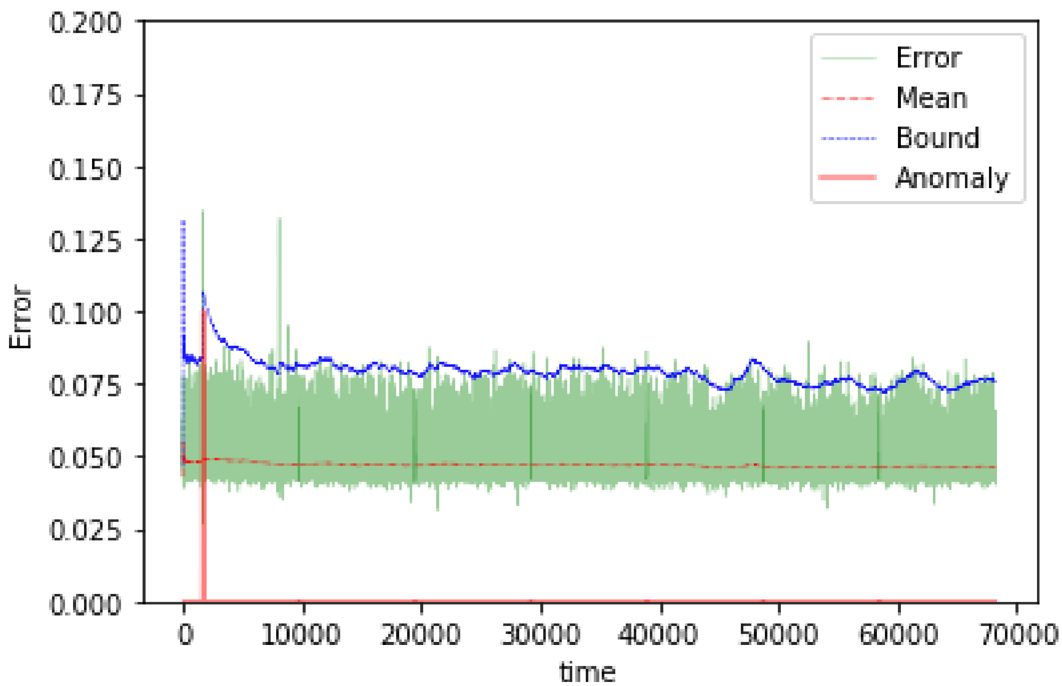

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



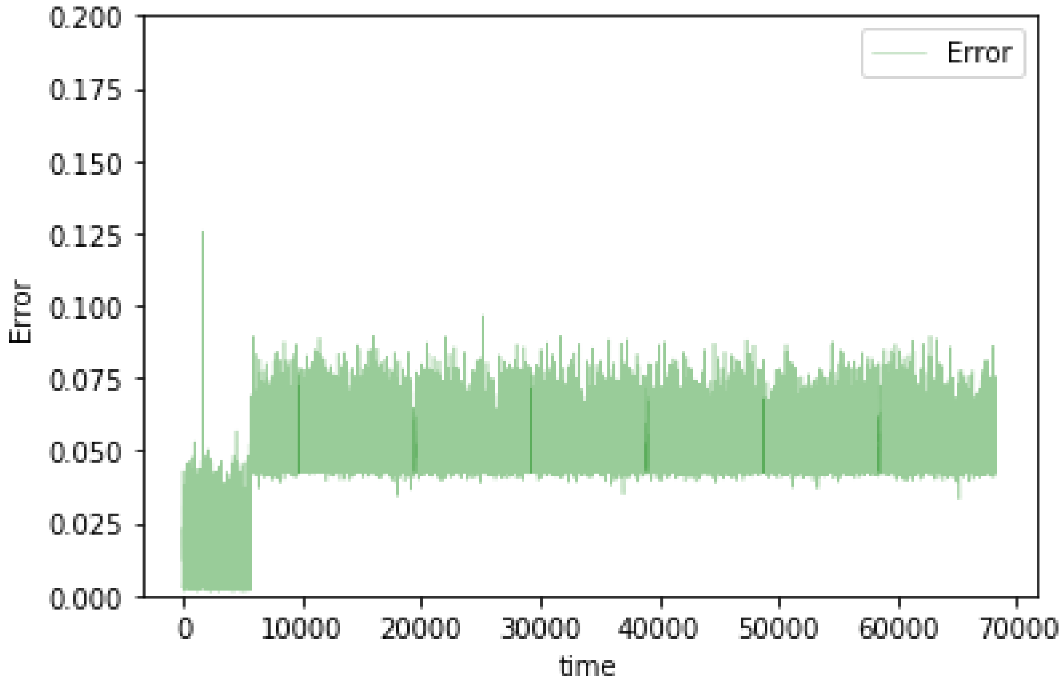

```
The mean error for gru1_100_disk_IO_start_ is 0.01821920647102366 for length 68199
Testing on Avg. load data.
```

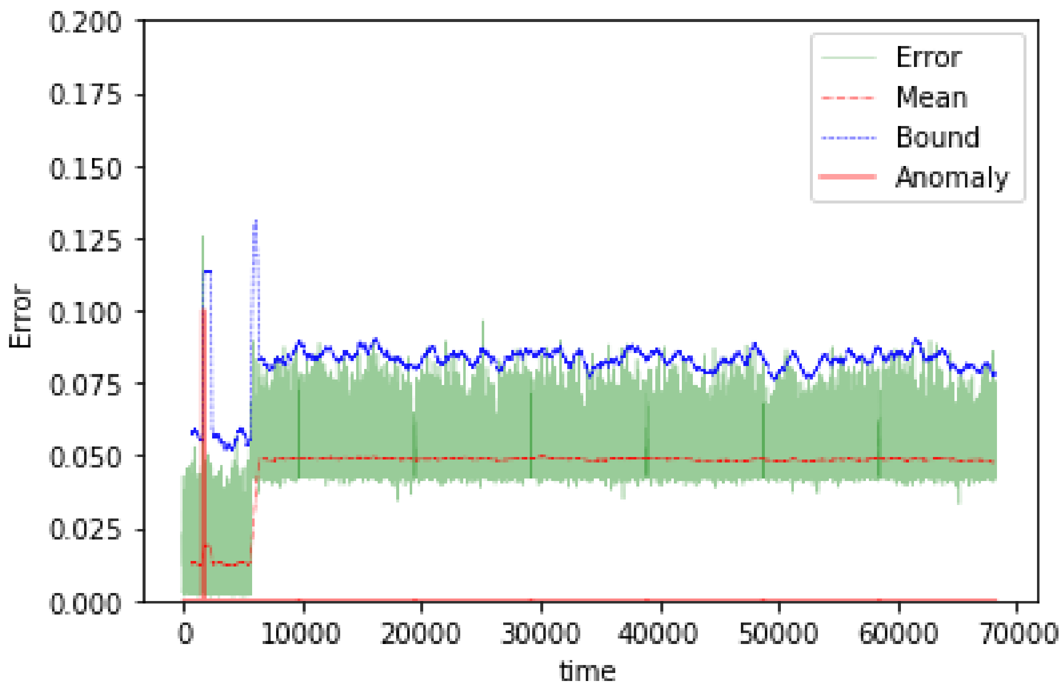


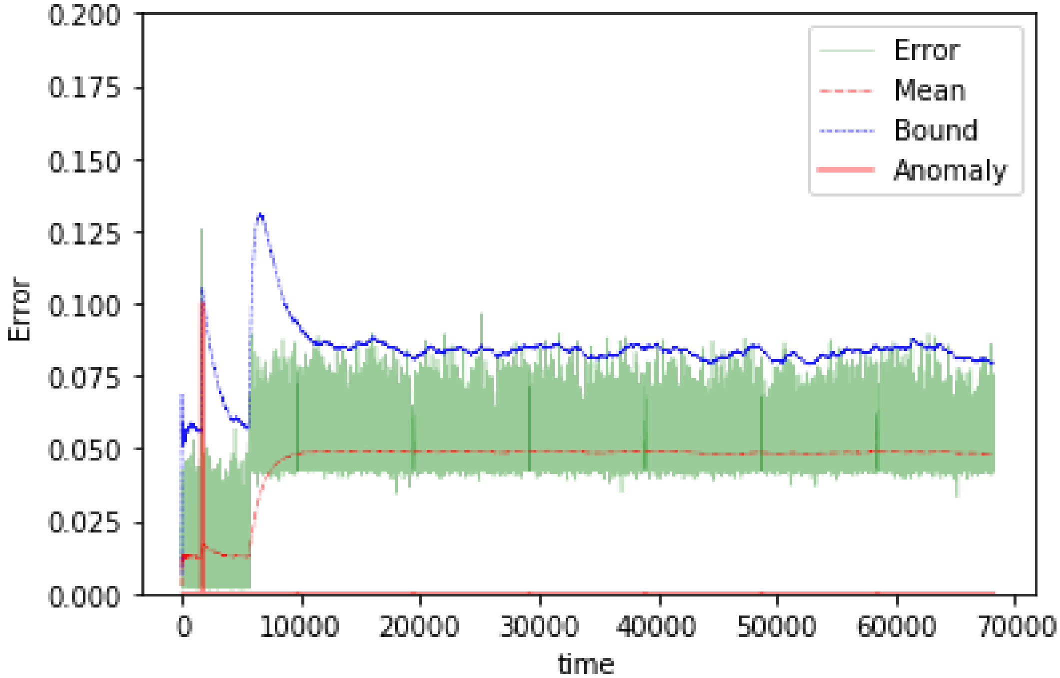

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

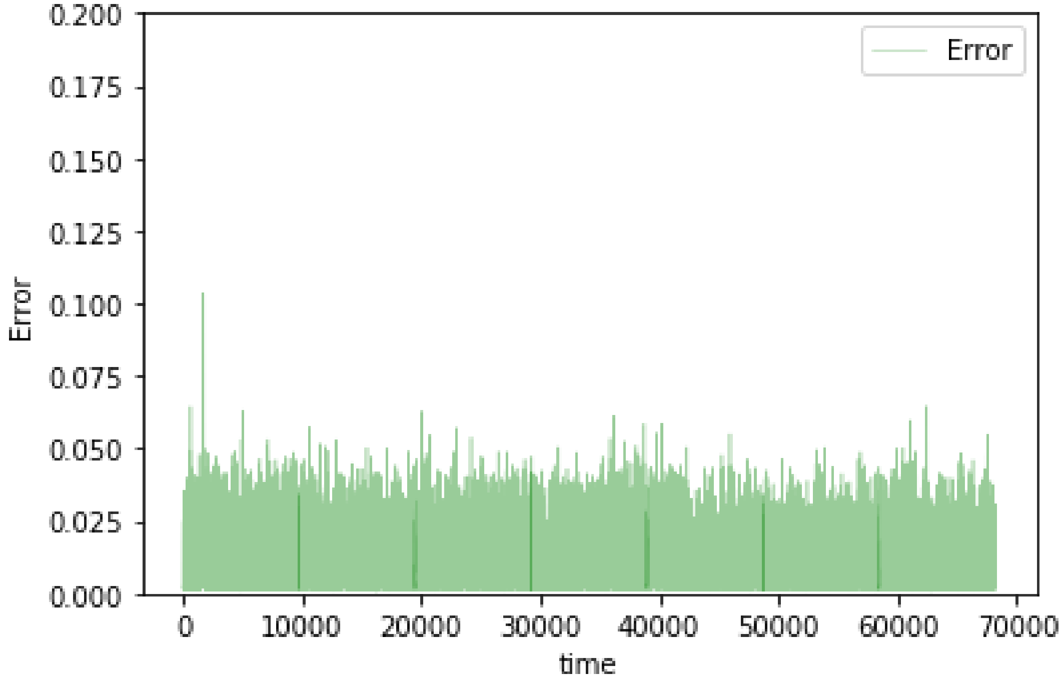
```
The mean error for gru1_100_avg_load_ is 0.04696257223546437 for length 68199
Testing on app change early data.
```

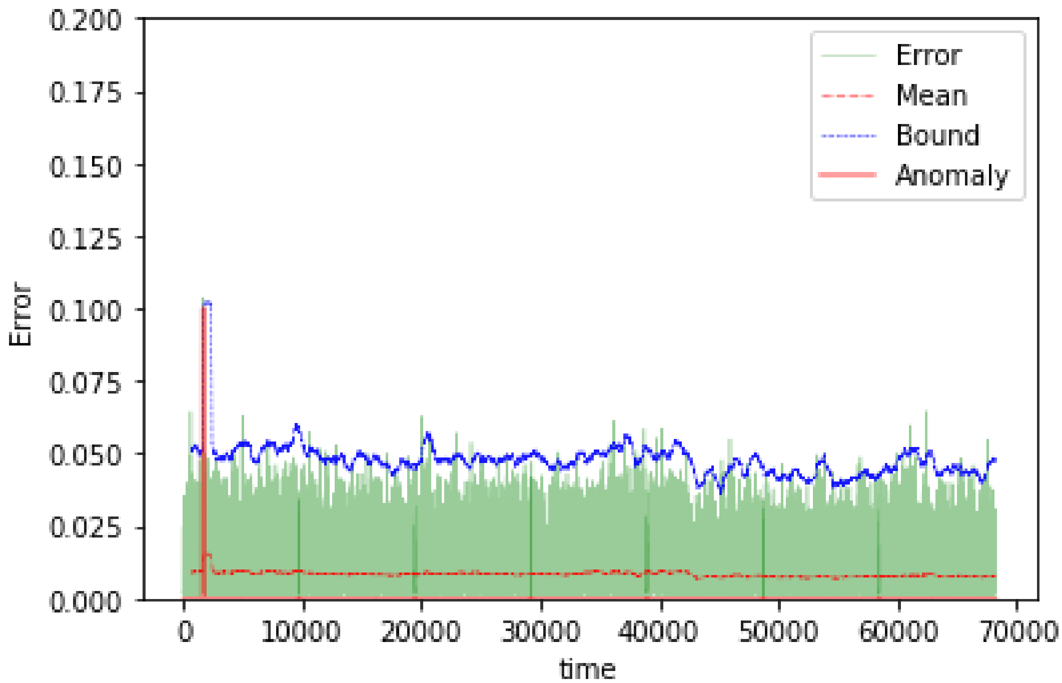


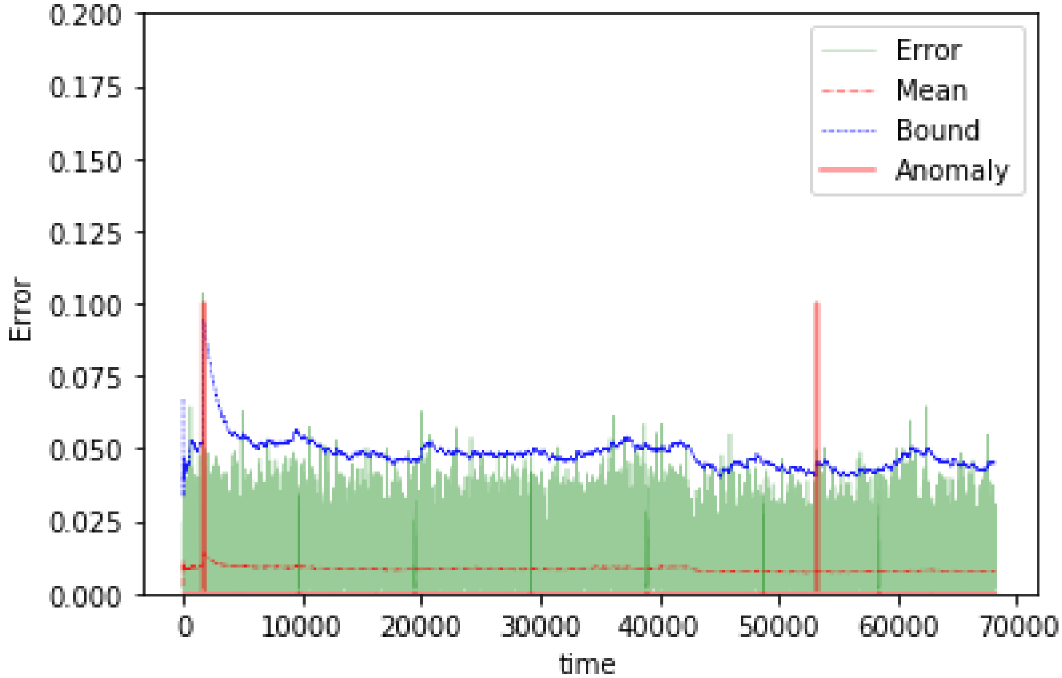





```
The mean error for gru1_100_app_change_early_ is 0.04581654476393923 for length 68199
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

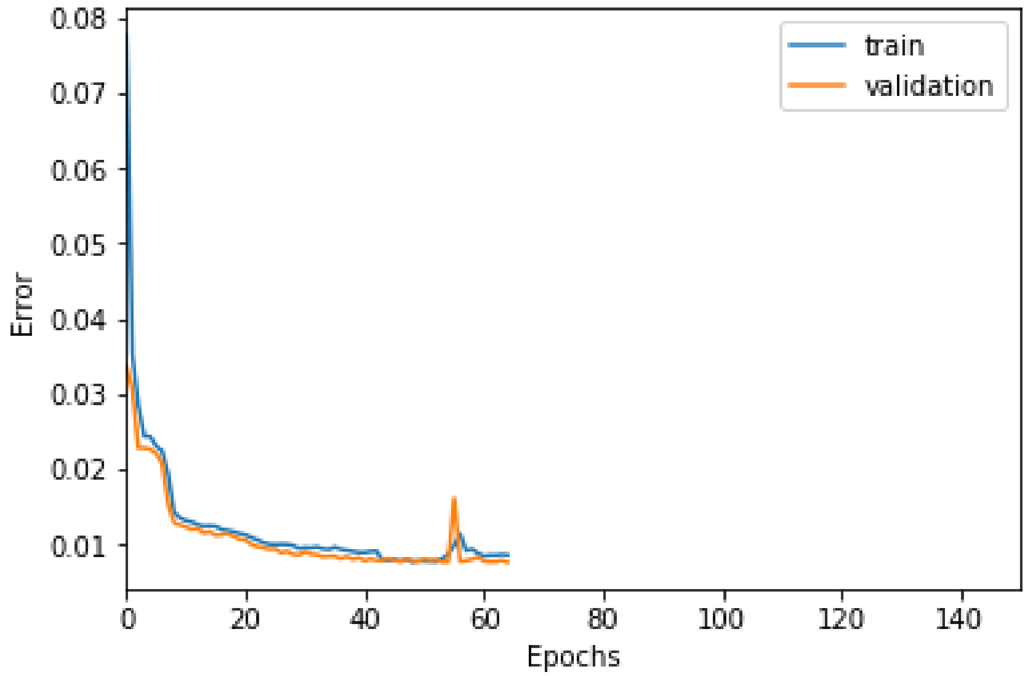

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



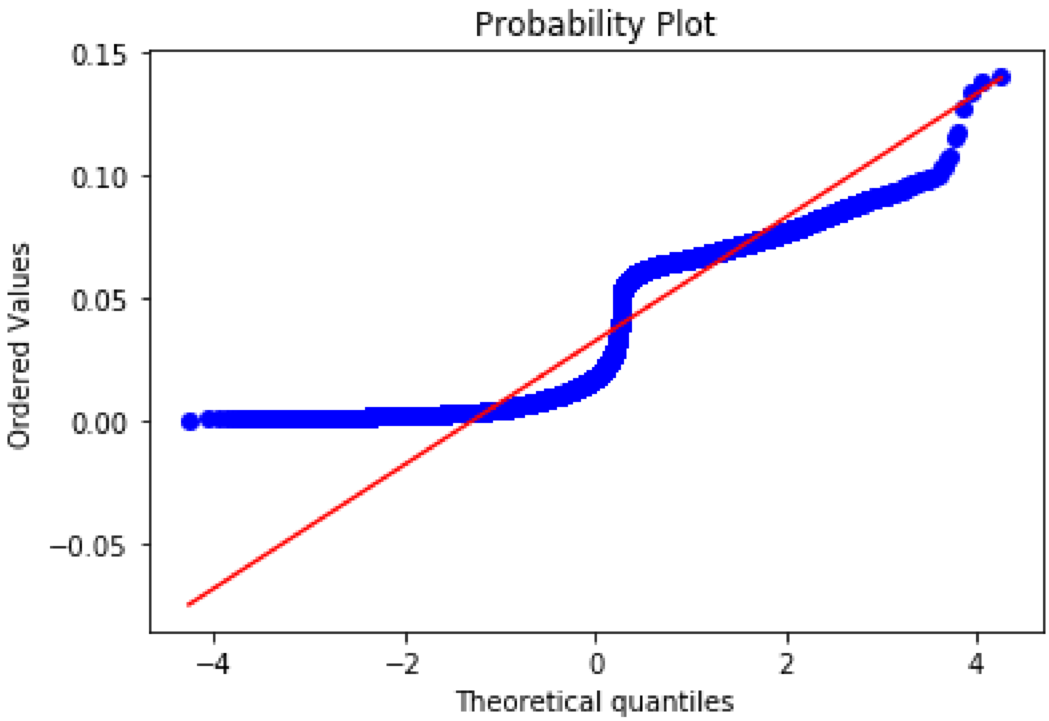

```
The mean error for gru1_100_normal_ is 0.008366218991236529 for length 68199
====================
```

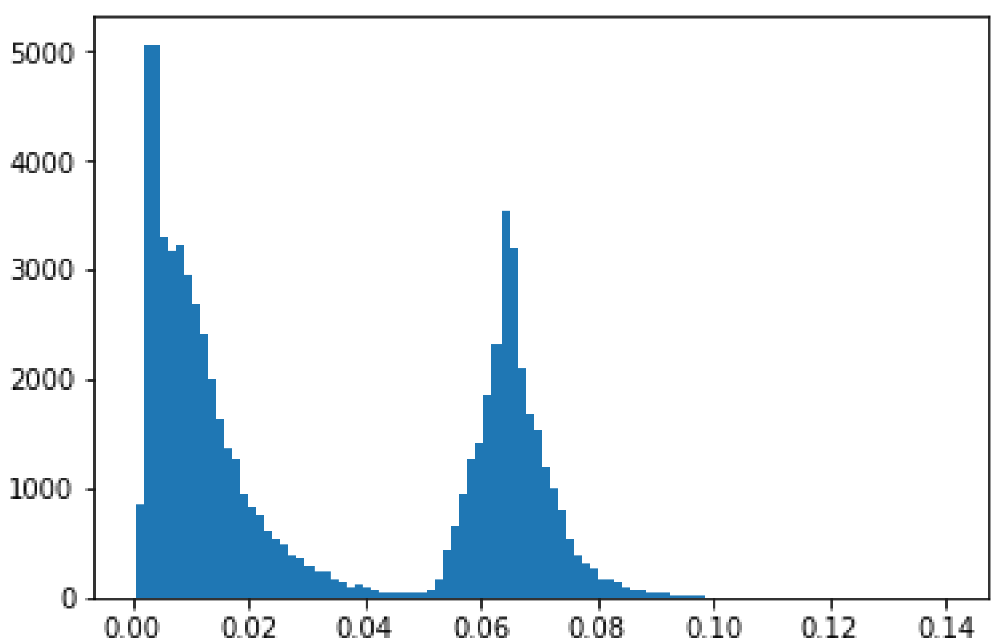
200 steps

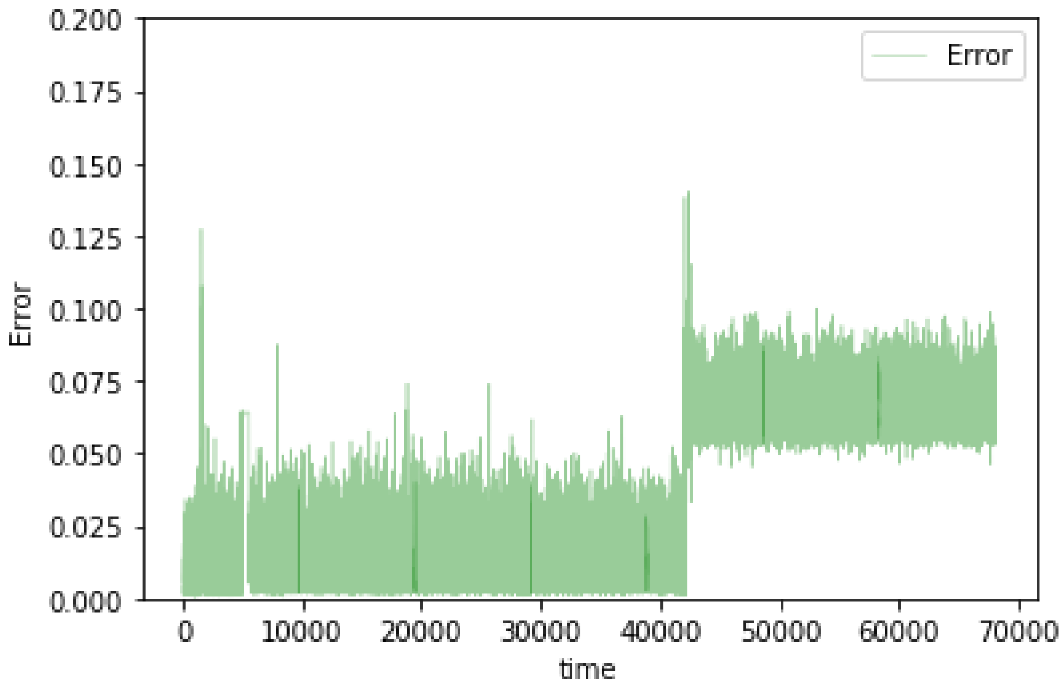
```
In [175]: TIMESTEPS = 200
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru1_200"

In [176]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu')(input_layer)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [177]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [178]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```



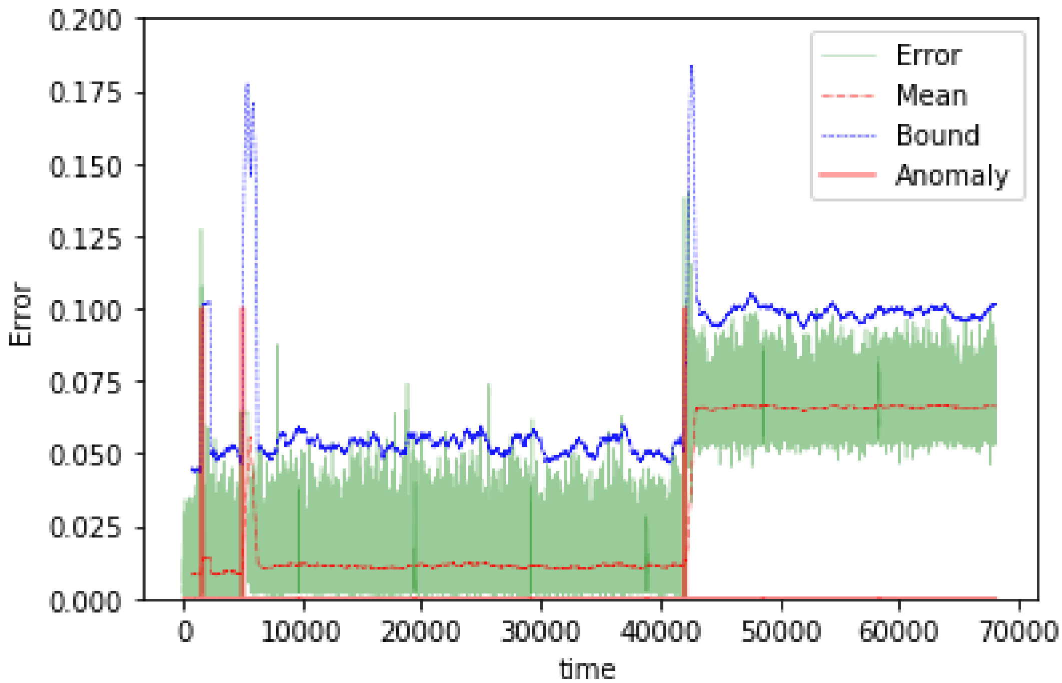

```
Training loss for final epoch is 0.008576546949800104
Validation loss for final epoch is 0.007635805290774443
---------- Beginning tests for gru1_200 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.15
0.10
0.05
0.00
−0.05
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

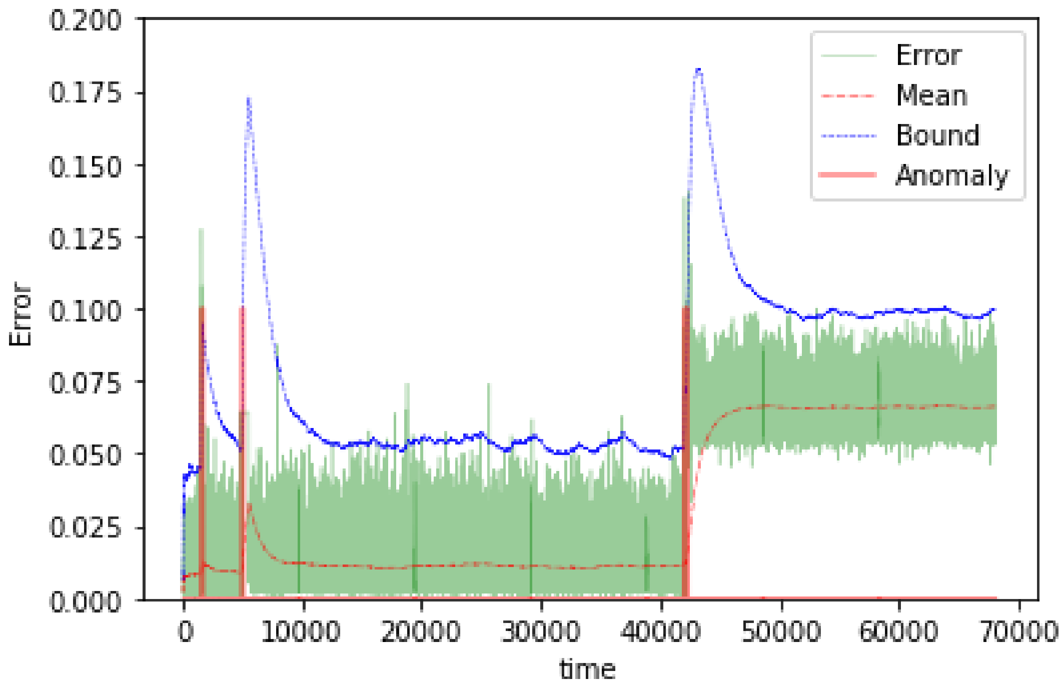

5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14



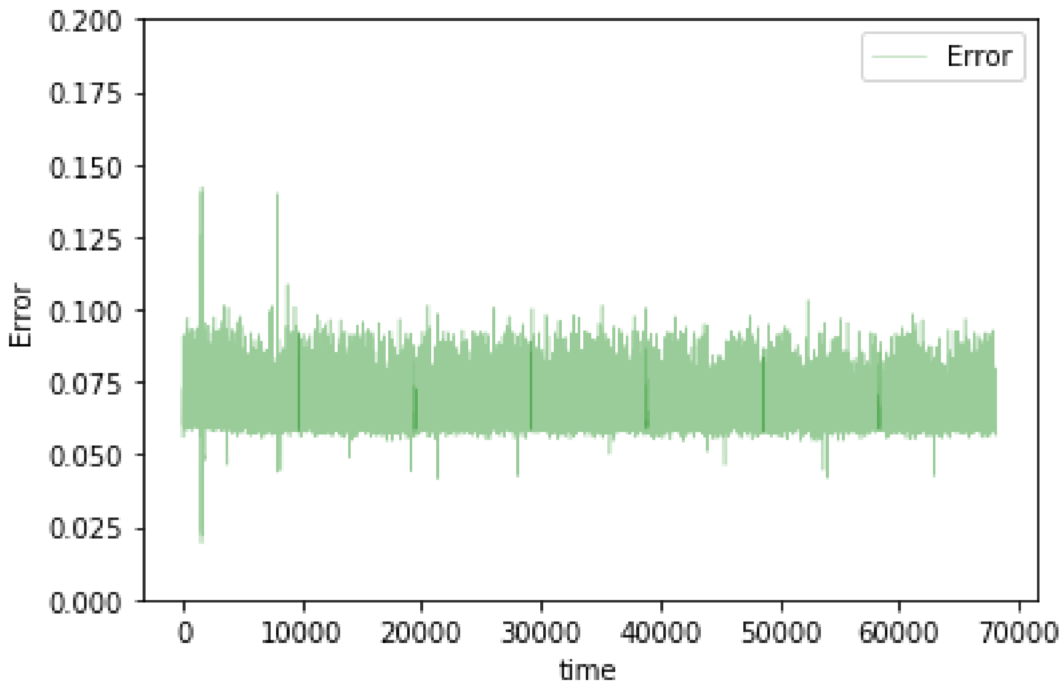
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

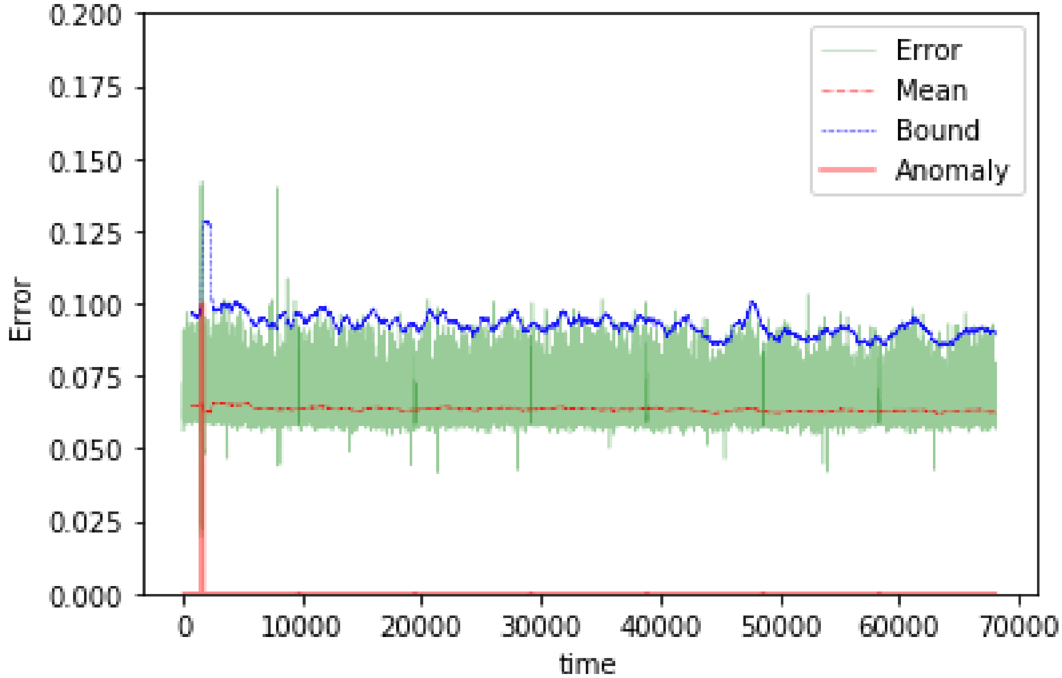


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

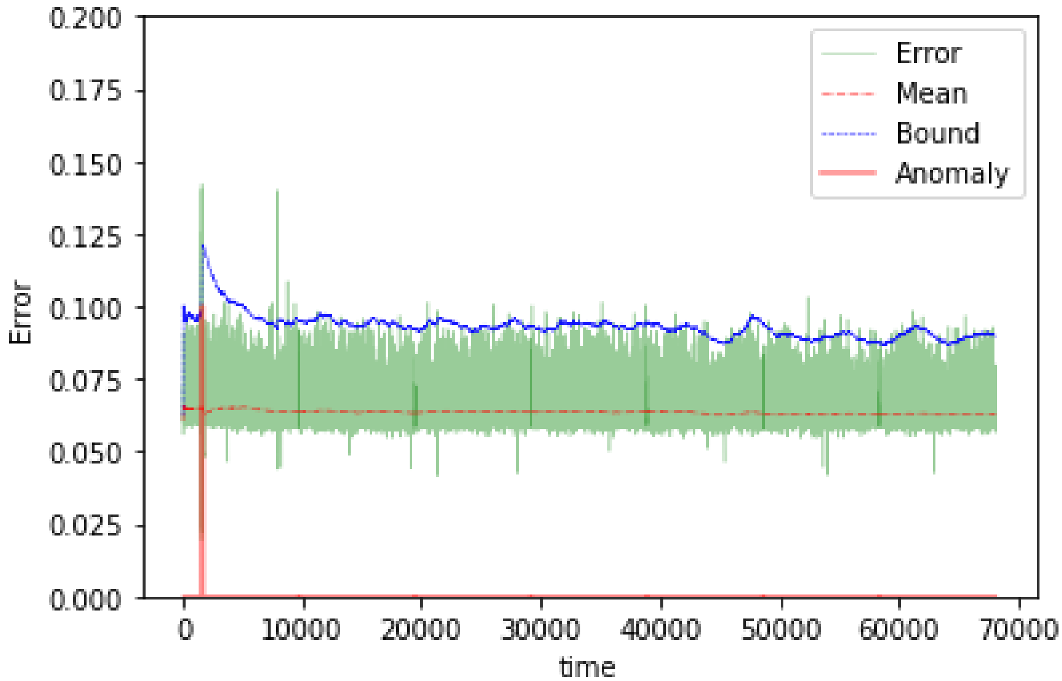



```
The mean error for gru1_200_disk_IO_start_ is 0.03232633895944938 for length 68099
Testing on Avg. load data.
```



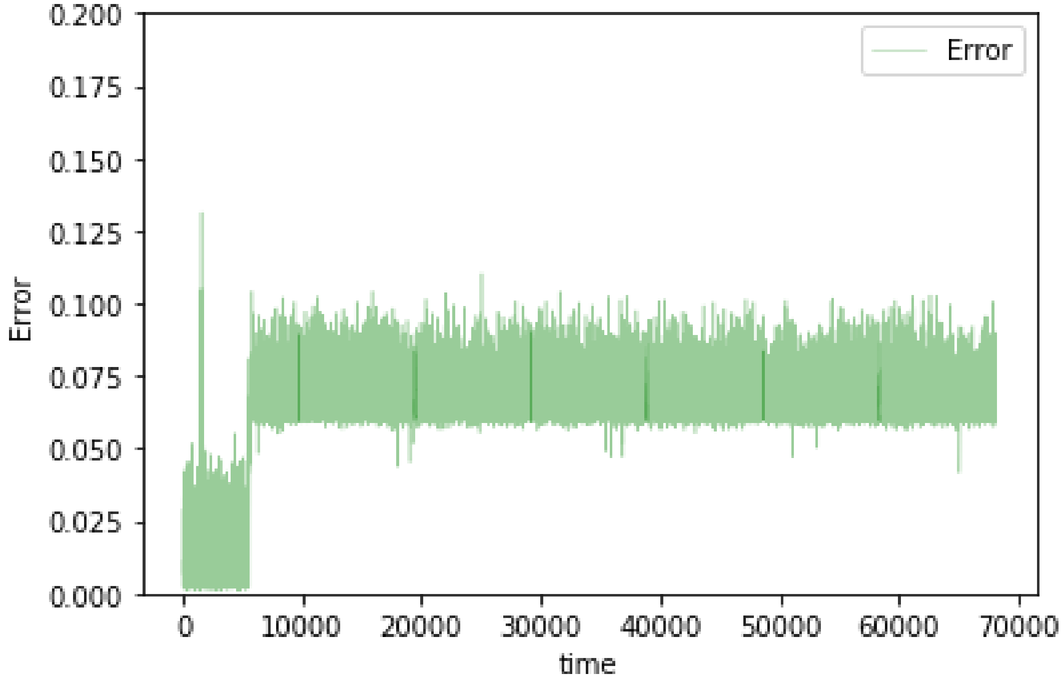


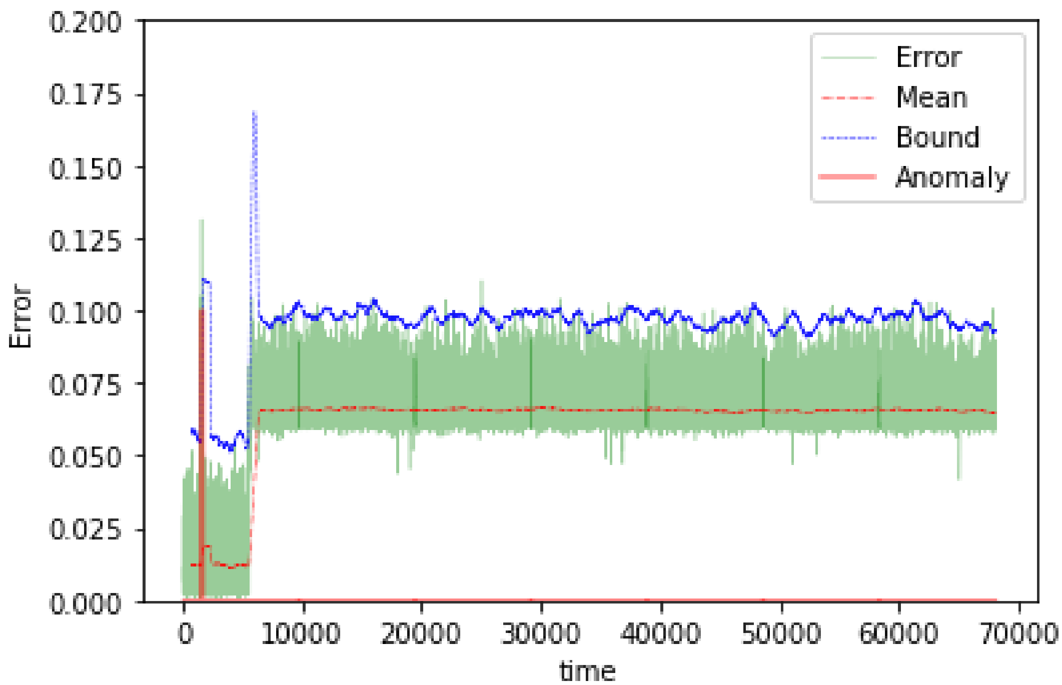
Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

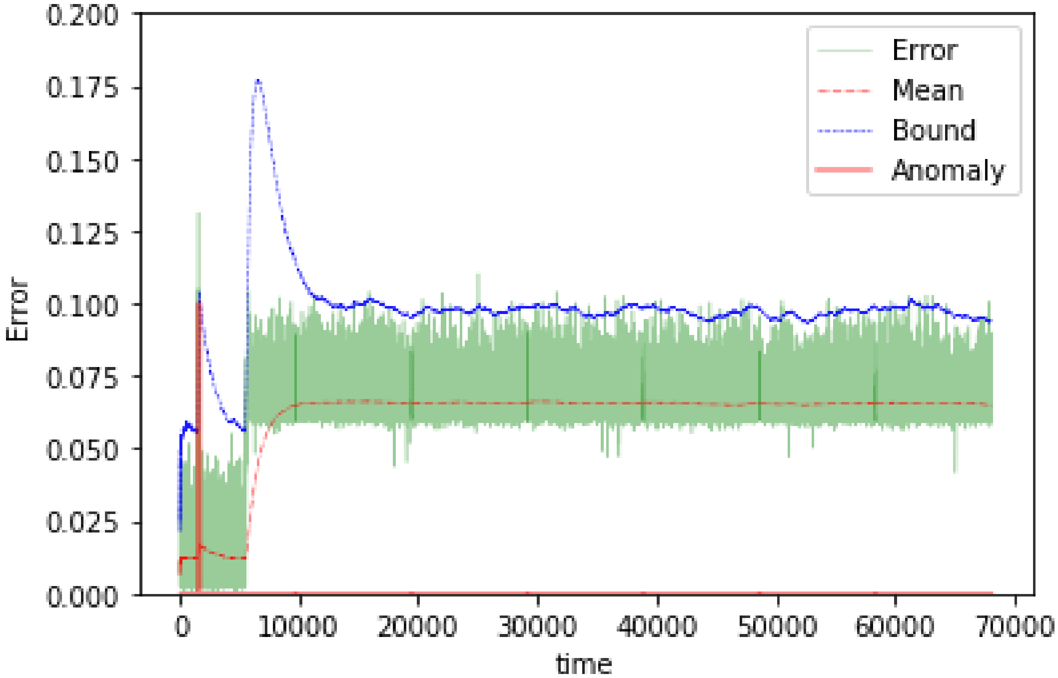


Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

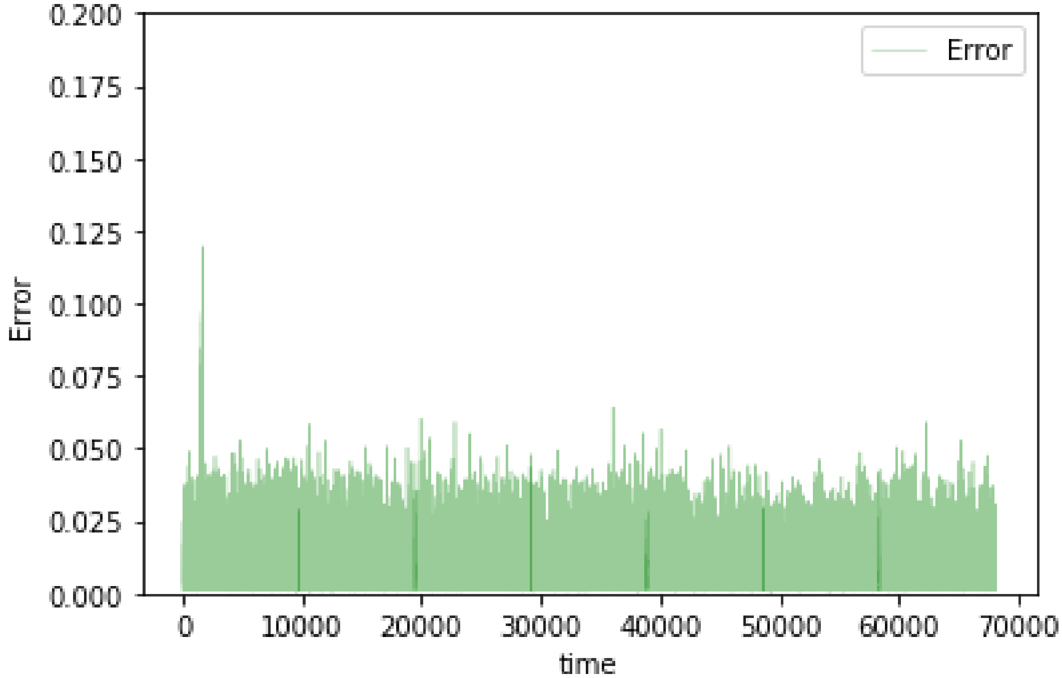
```
The mean error for gru1_200_avg_load_ is 0.06345686335379157 for length 68099
Testing on app change early data.
```

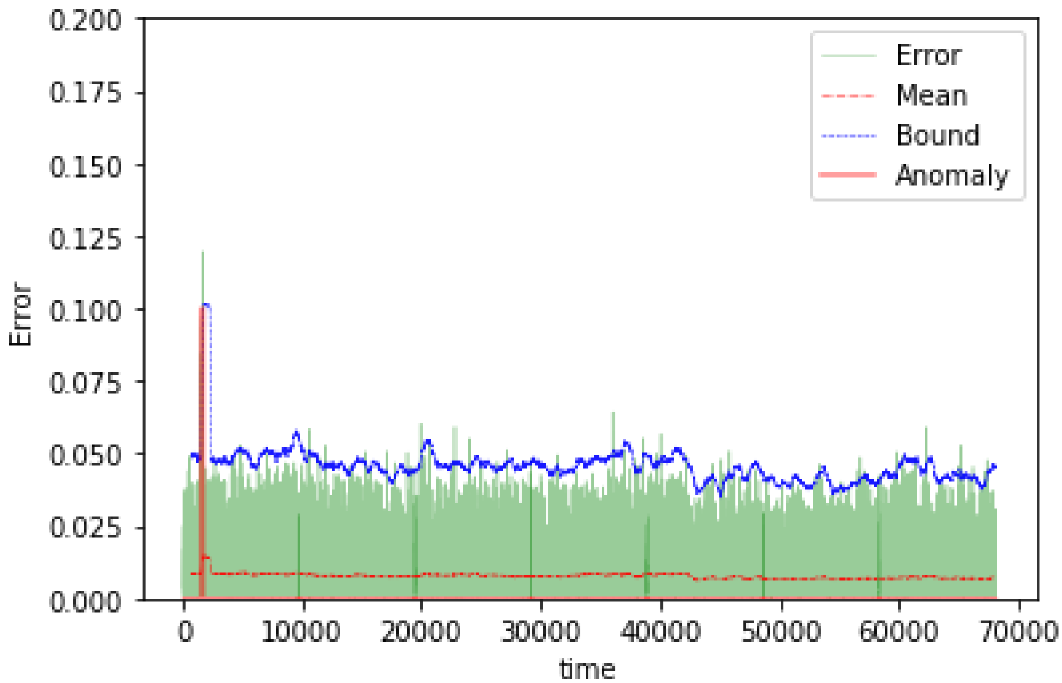


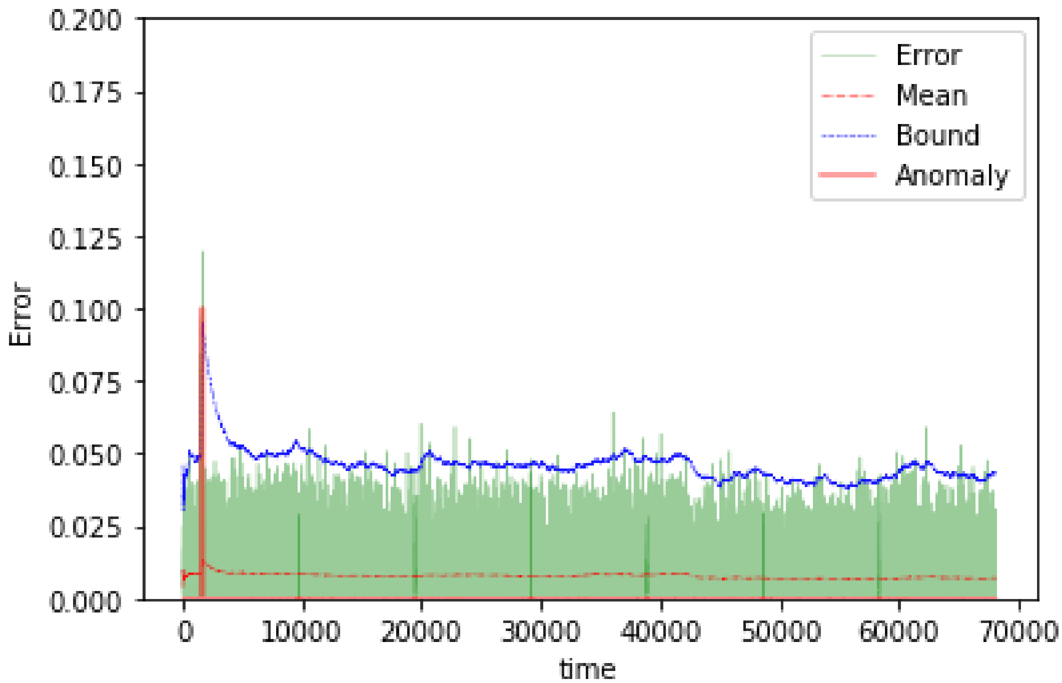





```
The mean error for gru1_200_app_change_early_ is 0.061280428947724175 for length 68099
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



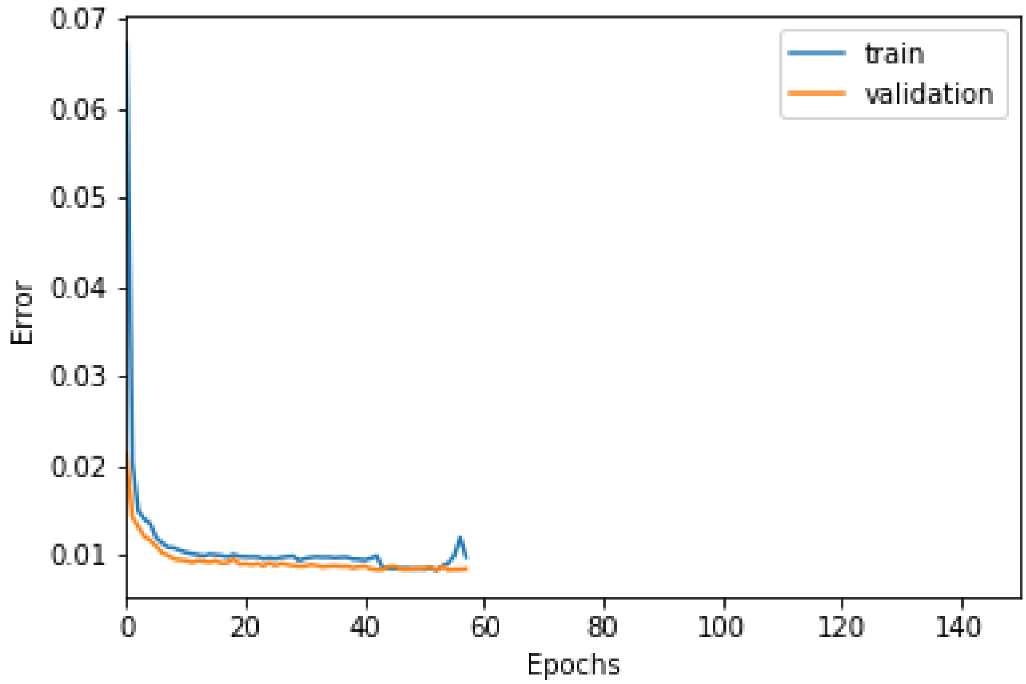

```
The mean error for gru1_200_normal_ is 0.007739639112848501 for length 68099
====================
```

1.11.6 RNN with 2 GRU layers

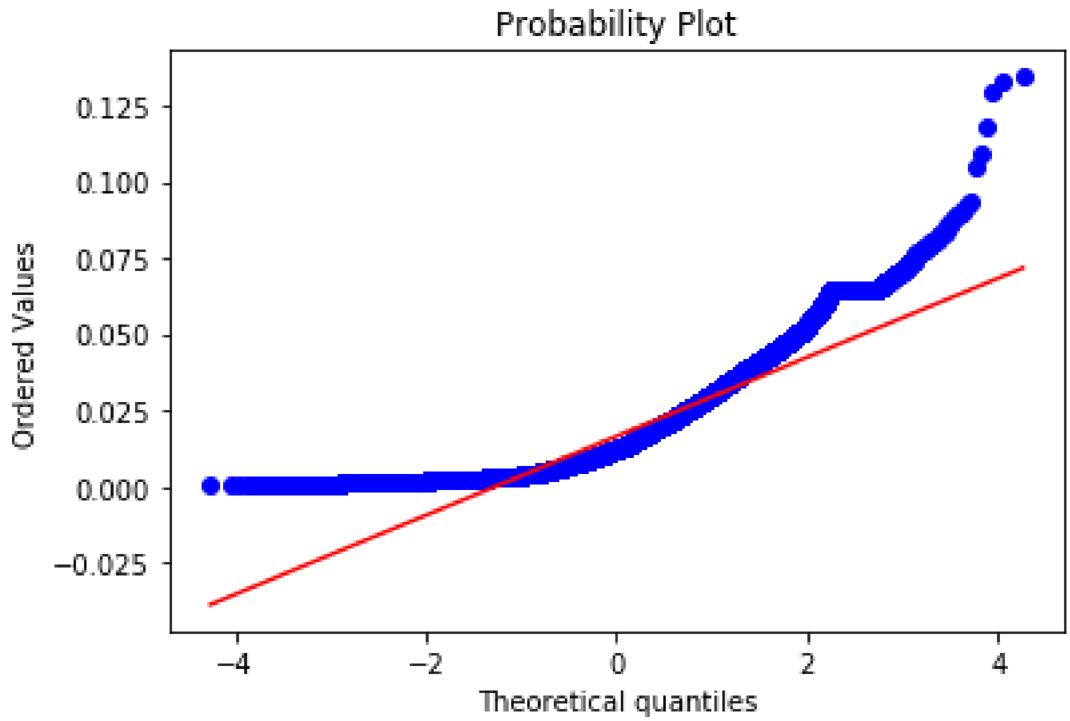
2 steps

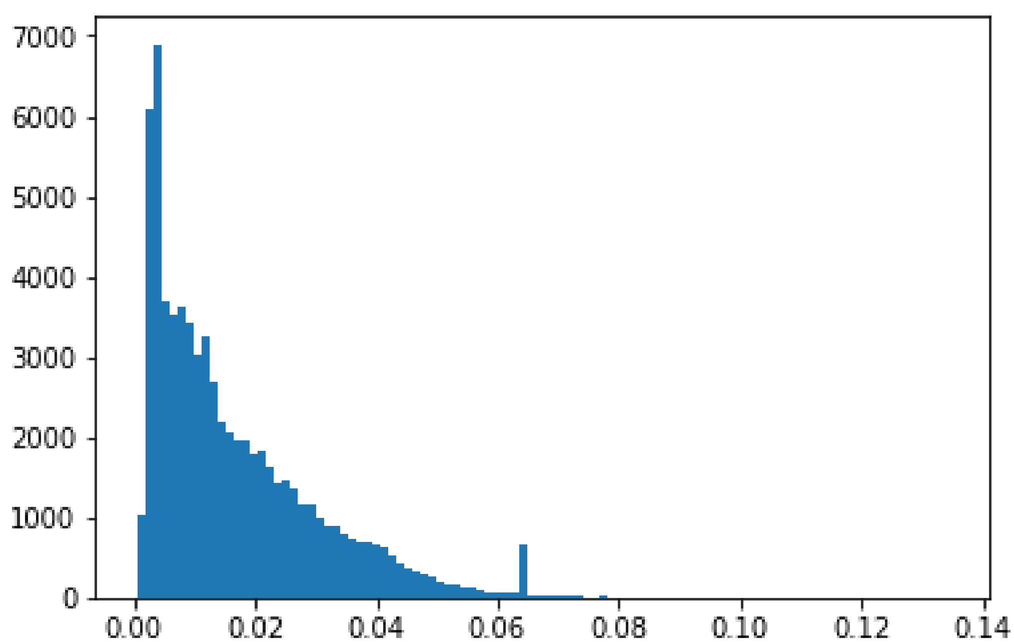
```
In [179]: TIMESTEPS = 2
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru2_2"

In [180]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(10, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [181]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])
```

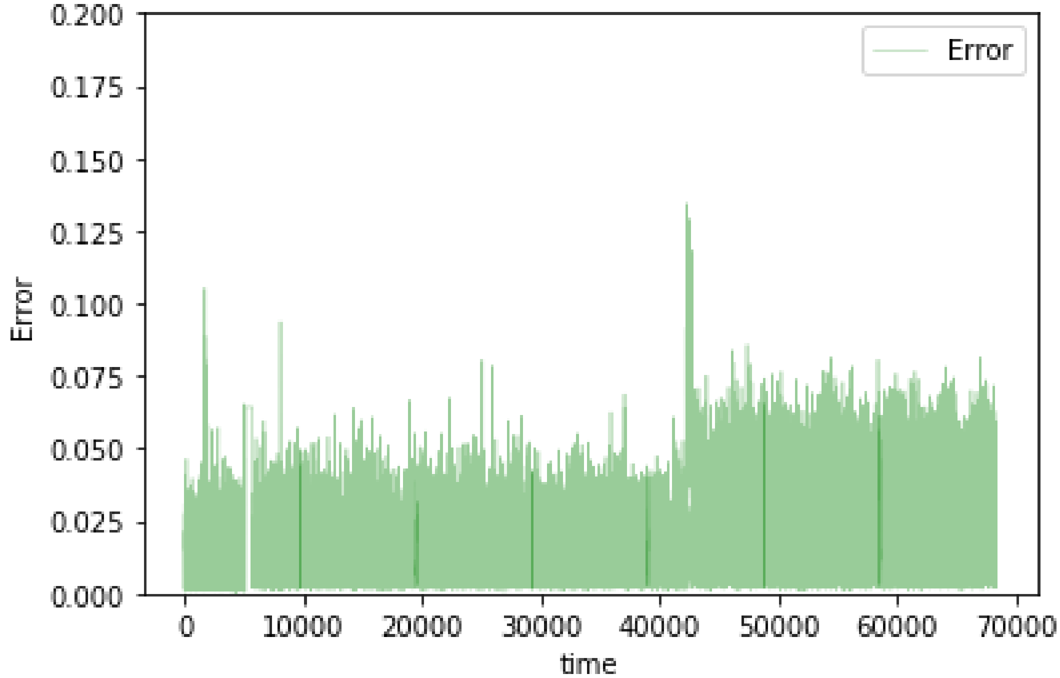
```
In [182]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

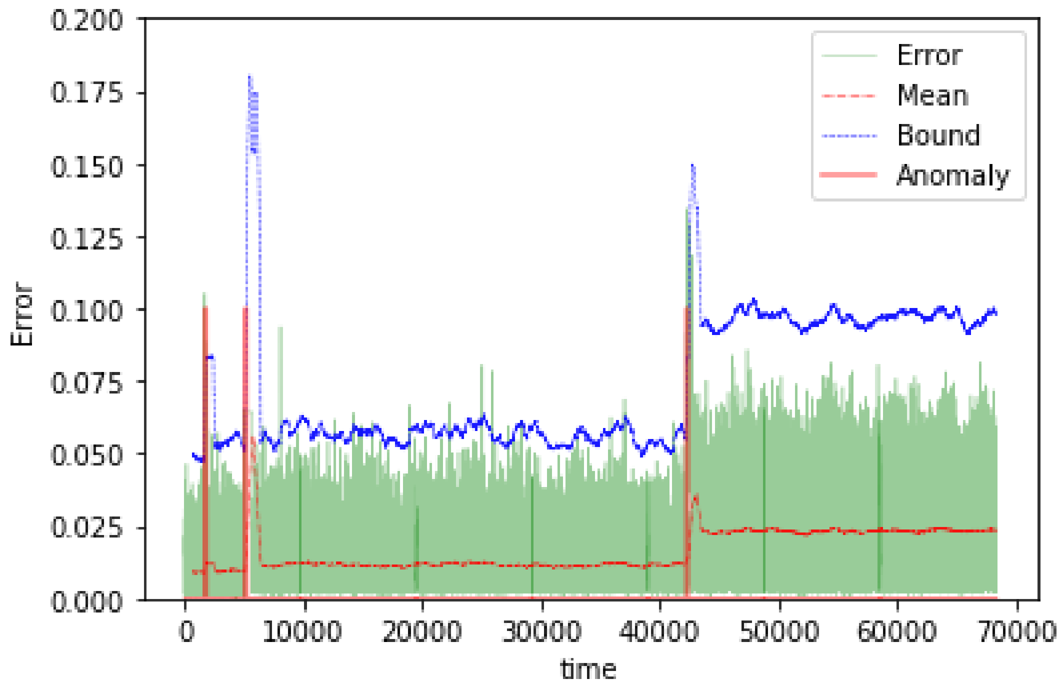


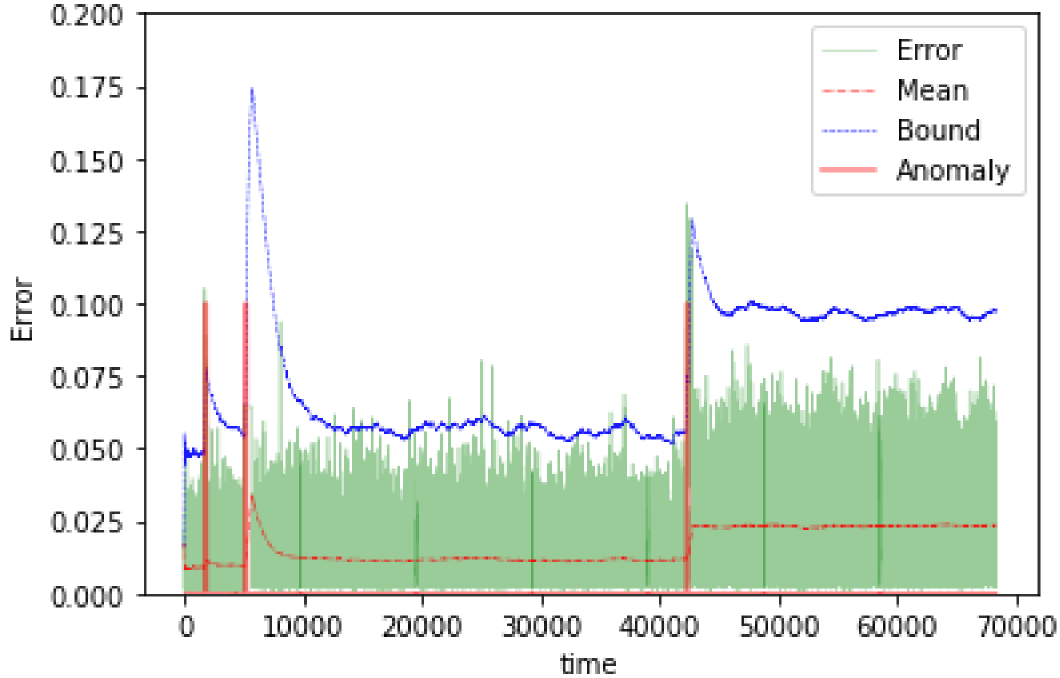

```
Training loss for final epoch is 0.009734550825553015
Validation loss for final epoch is 0.008419790519634261
---------- Beginning tests for gru2_2 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−4
−2
0
2
4

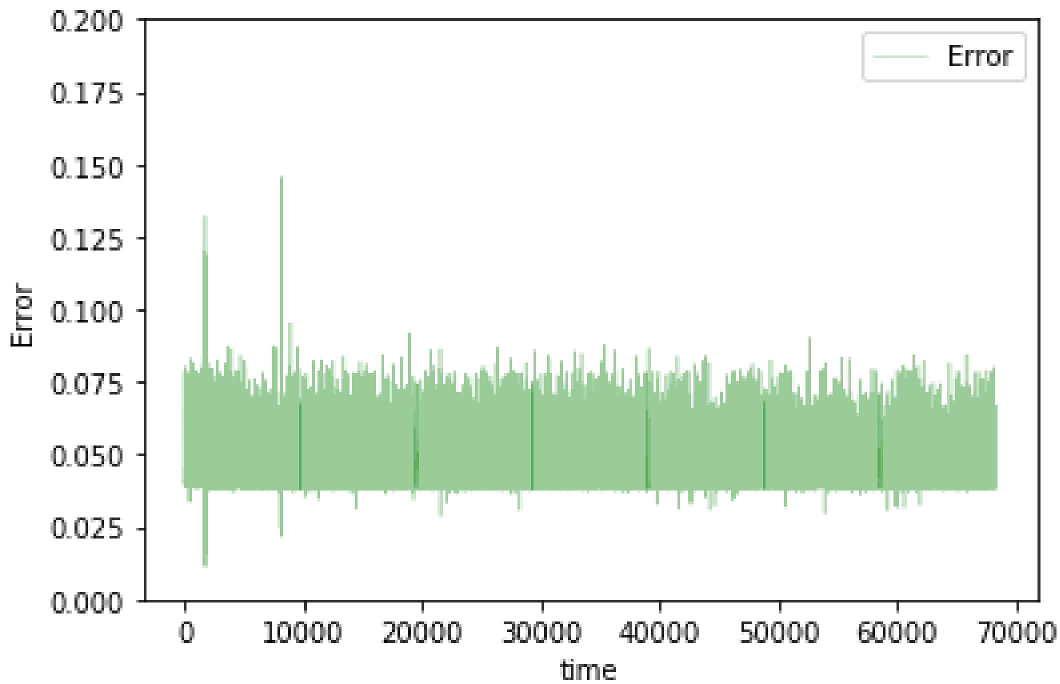

7000
6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

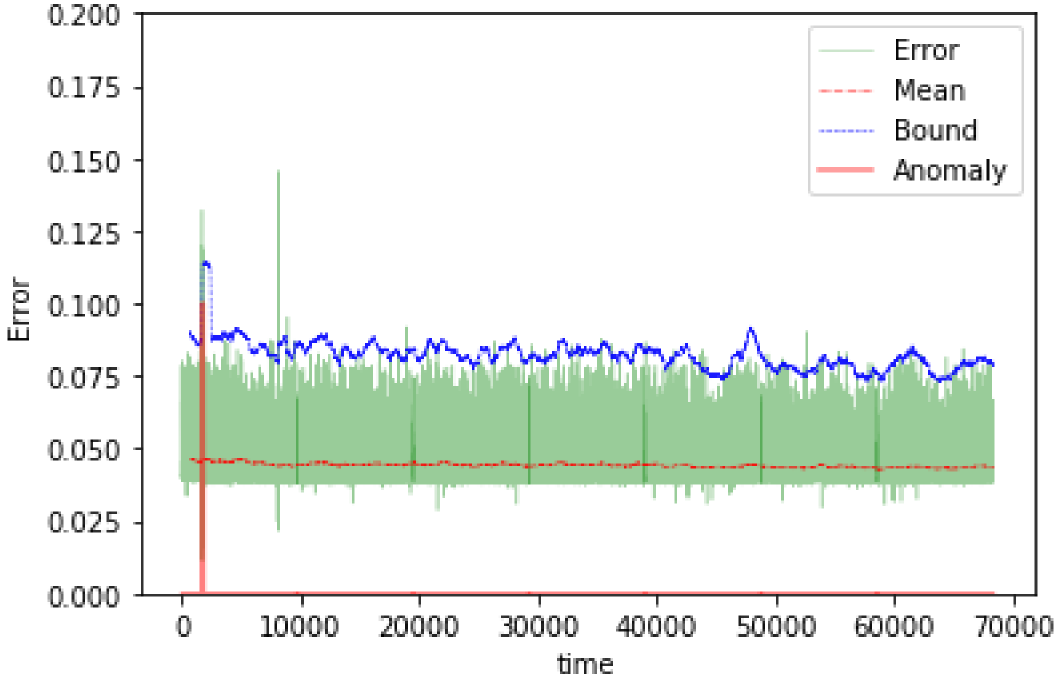

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

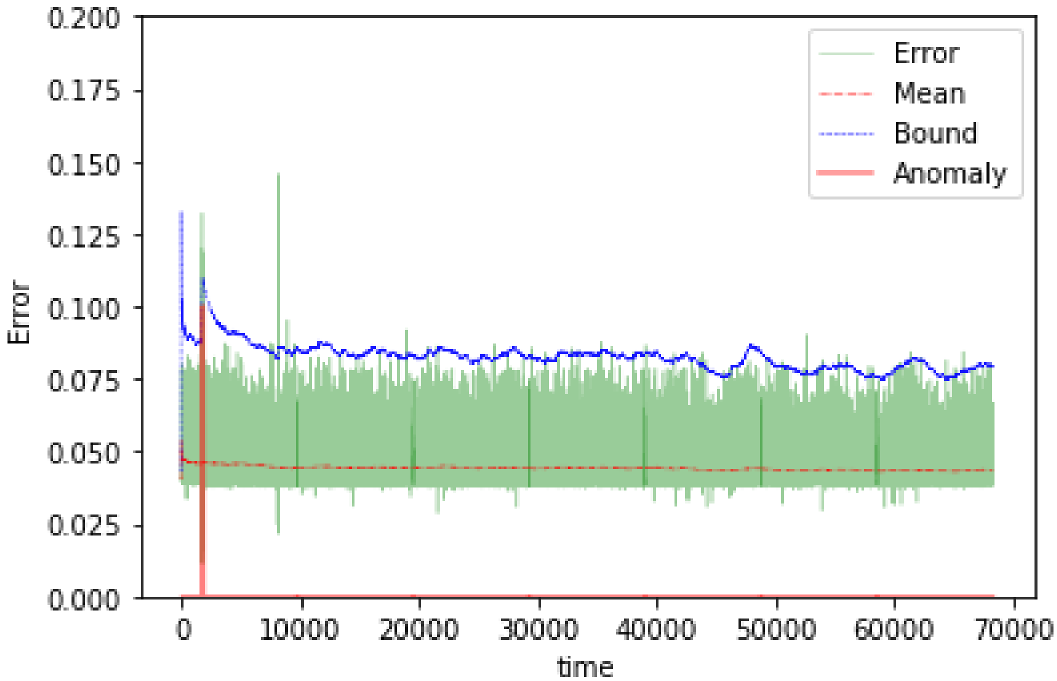

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



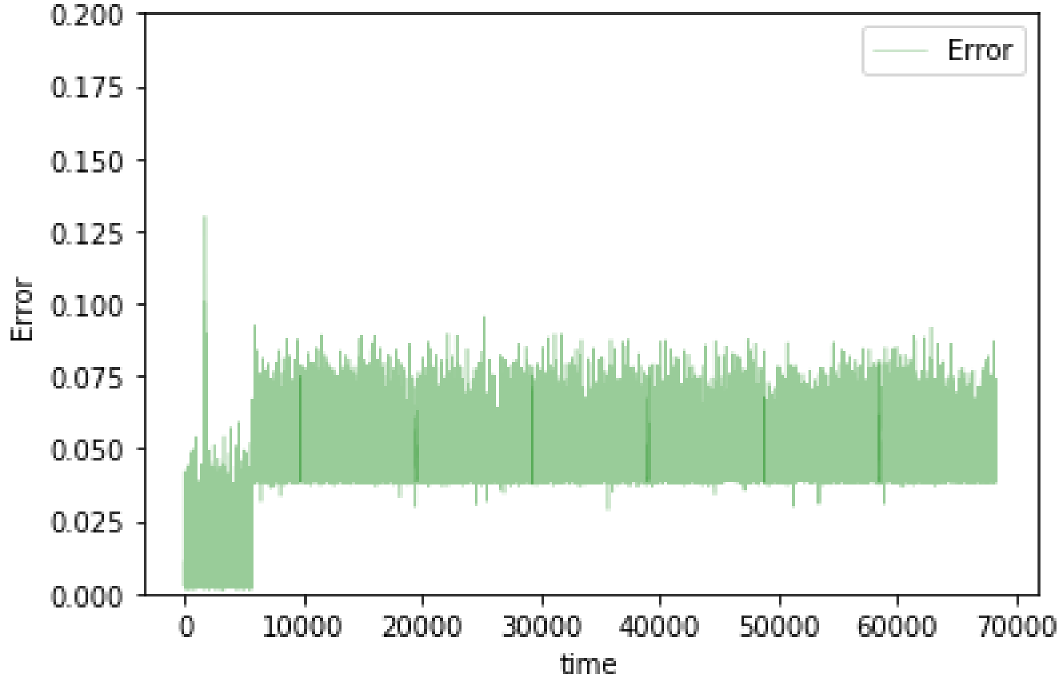

```
The mean error for gru2_2_disk_IO_start_ is 0.016514152125920705 for length 68297
Testing on Avg. load data.
```

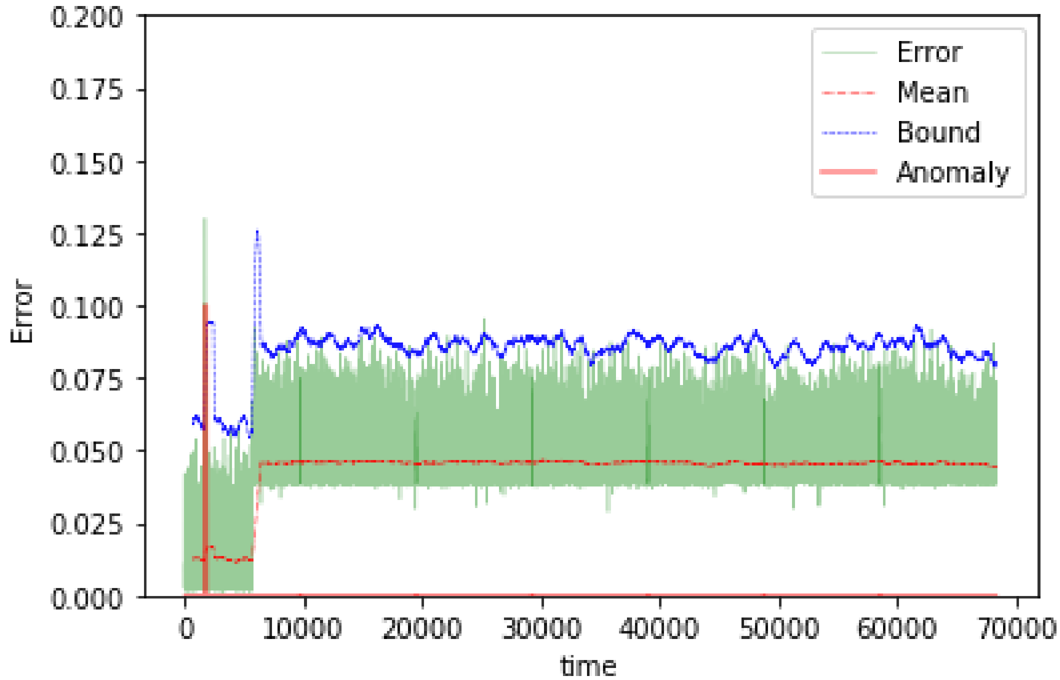


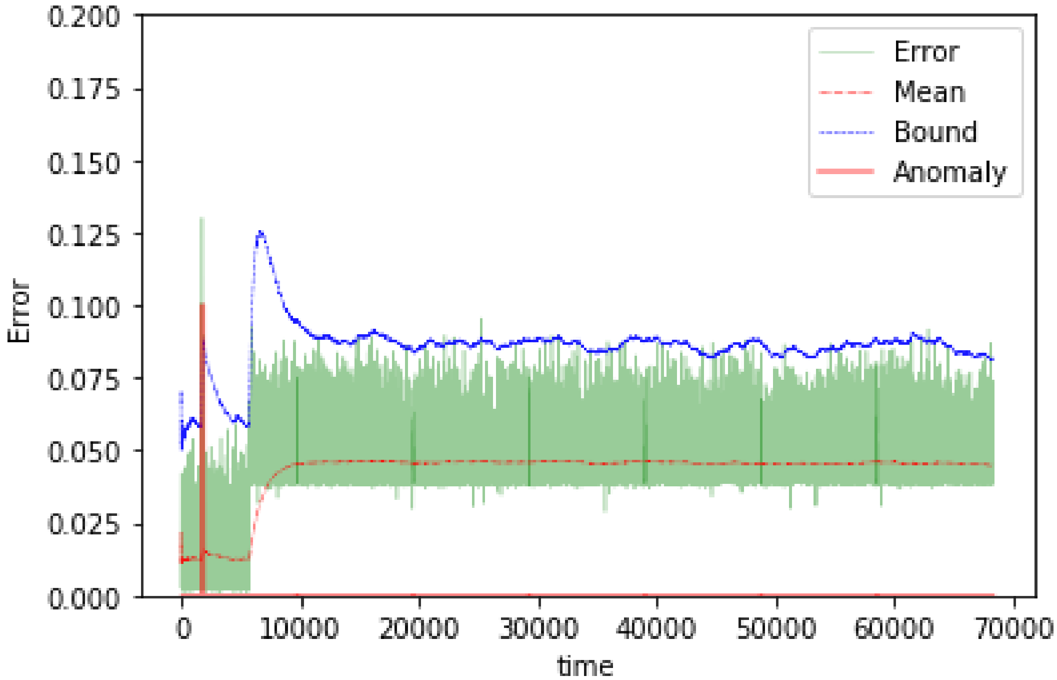

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

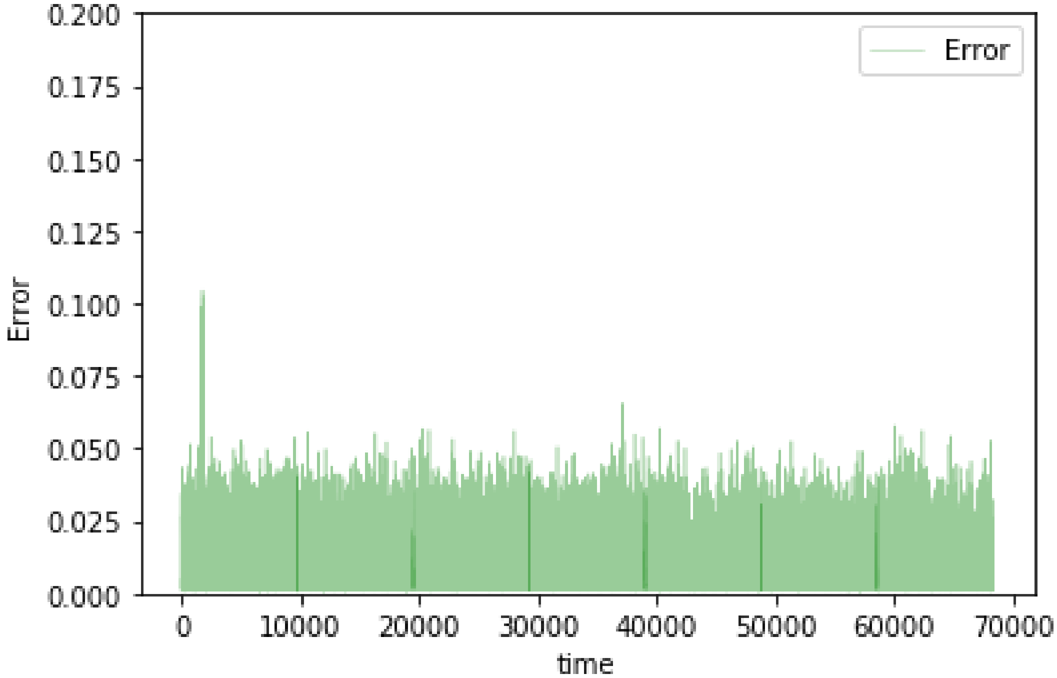
```
The mean error for gru2_2_avg_load_ is 0.04425034266363133 for length 68297
Testing on app change early data.
```

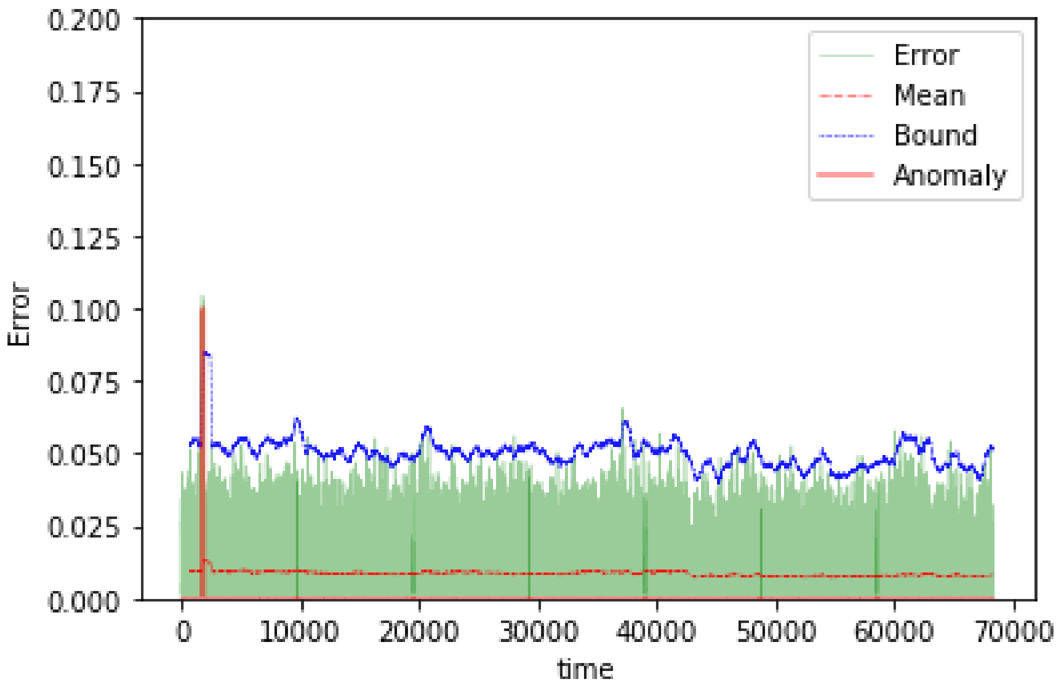


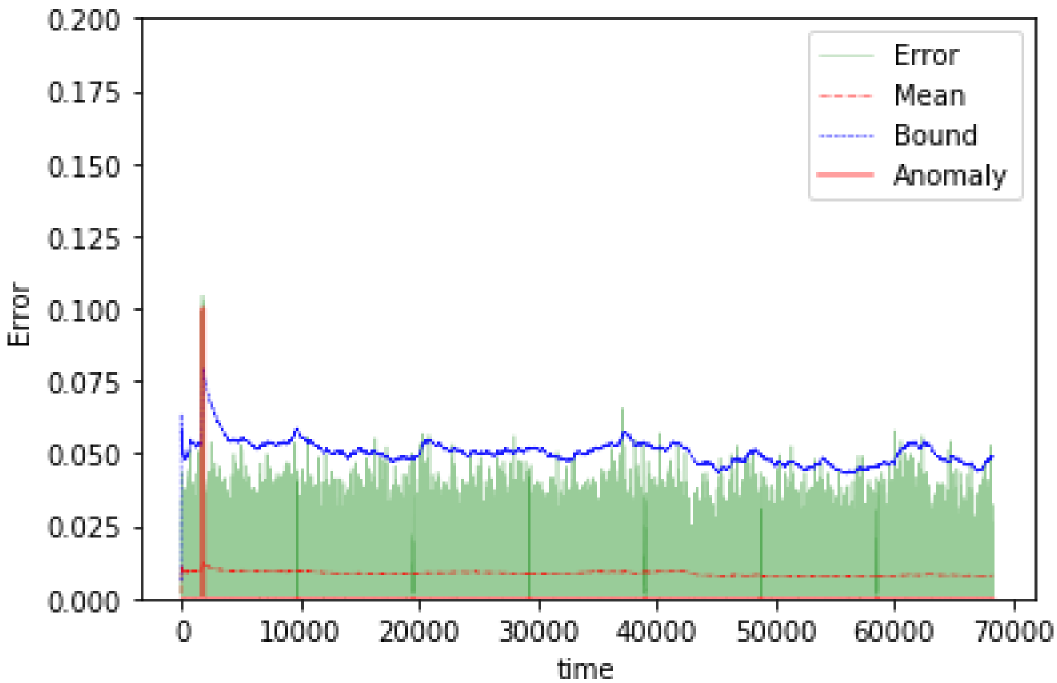





```
The mean error for gru2_2_app_change_early_ is 0.042993442072310165 for length 68297
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

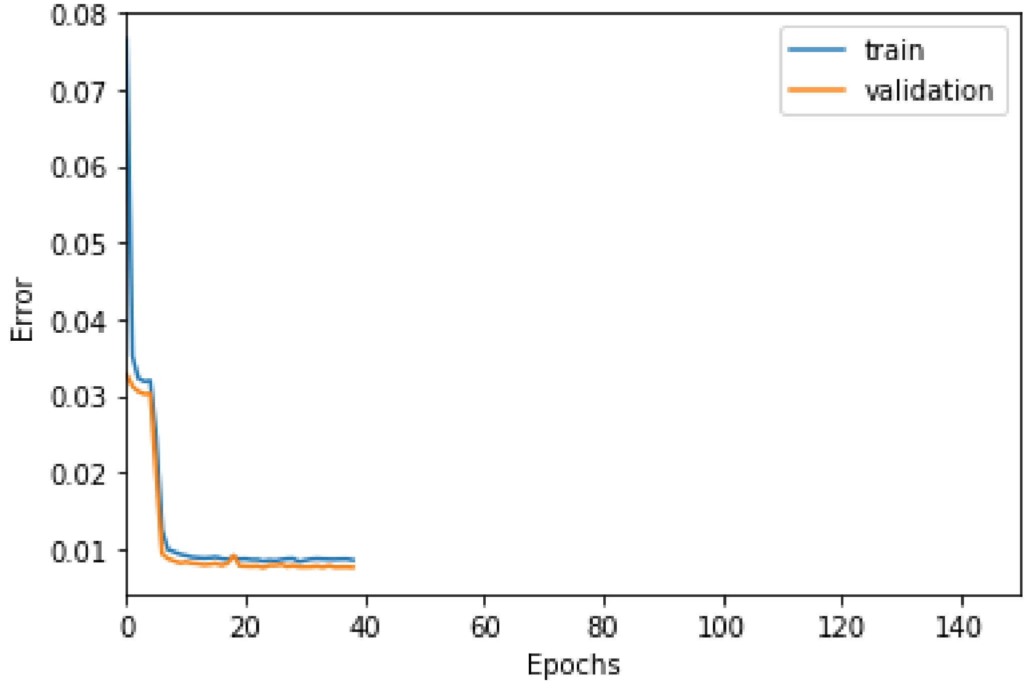

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



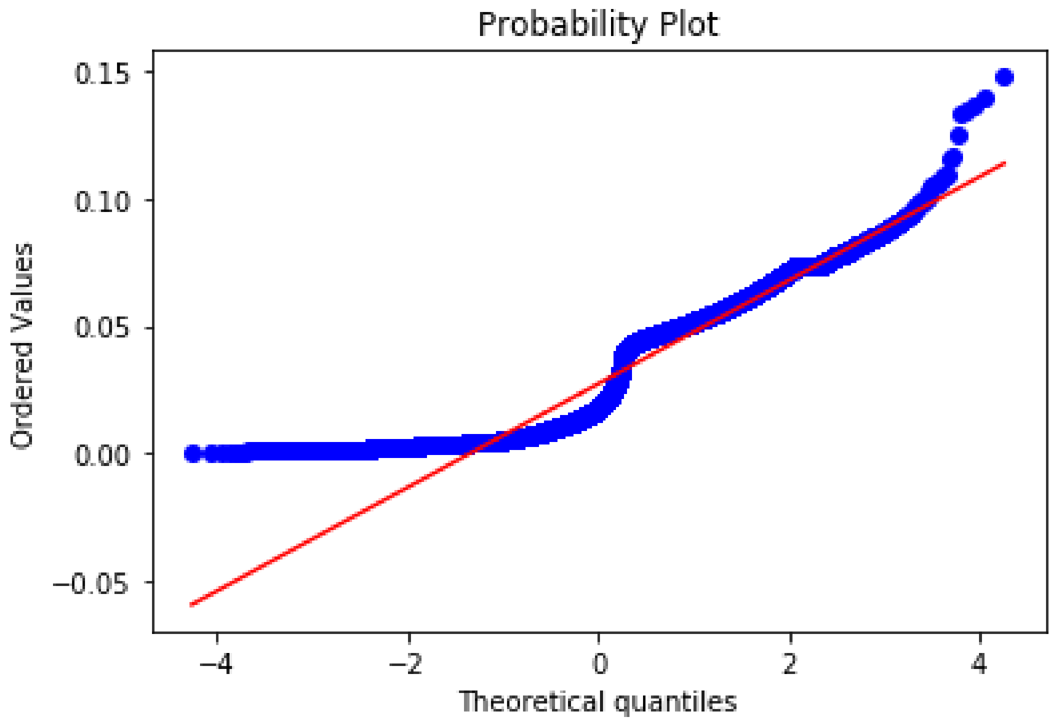

```
The mean error for gru2_2_normal_ is 0.008584455705773803 for length 68297
====================
```

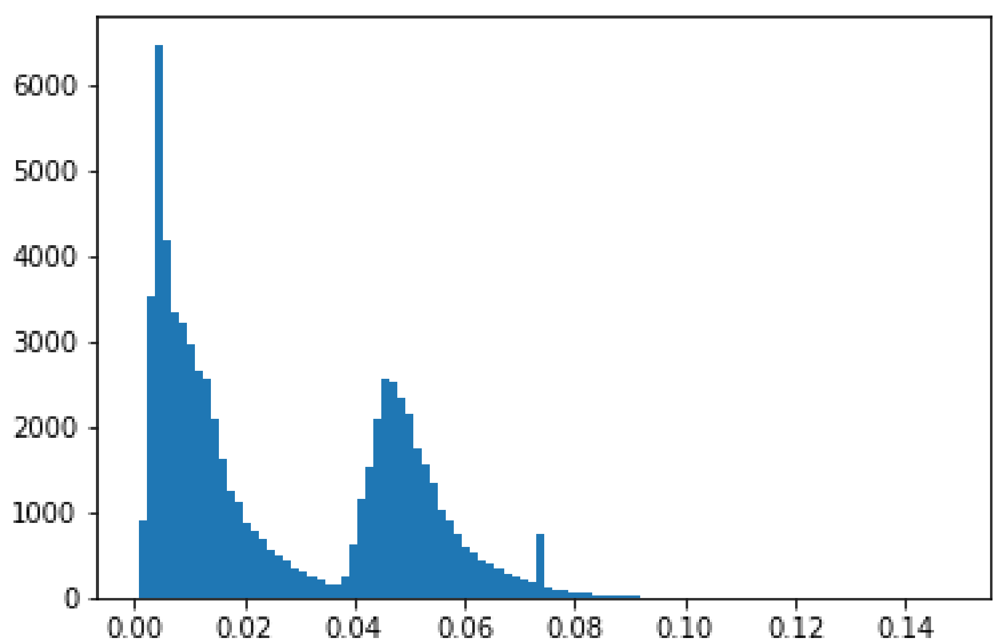
5 steps

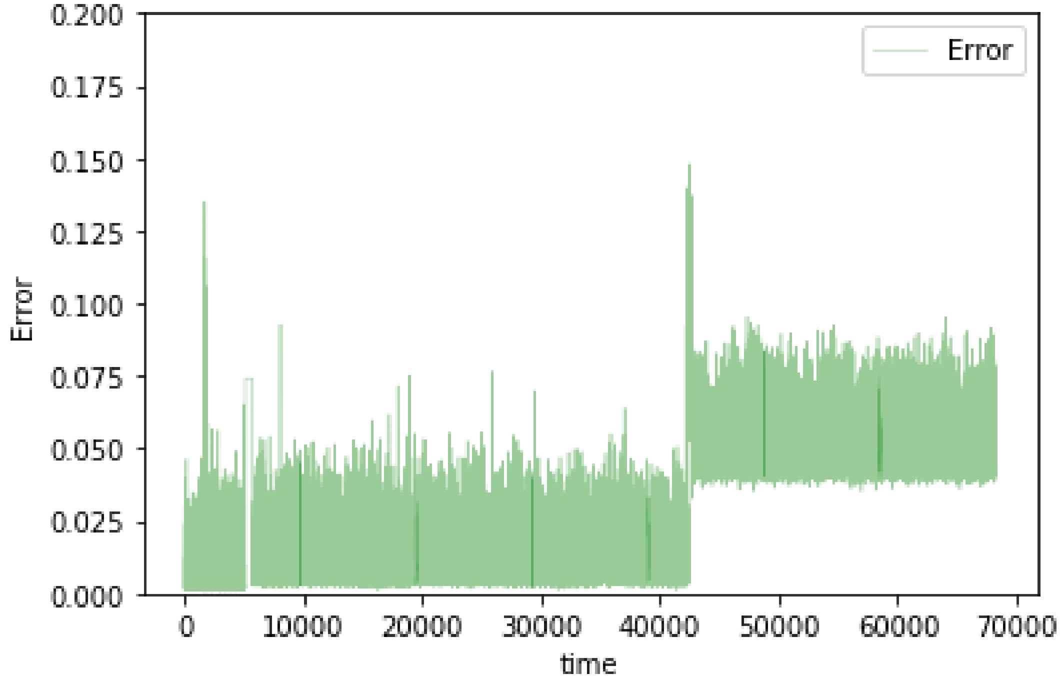
```
In [183]: TIMESTEPS = 5
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS, 0)
          name = "gru2_5"

In [184]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(10, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [185]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [186]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

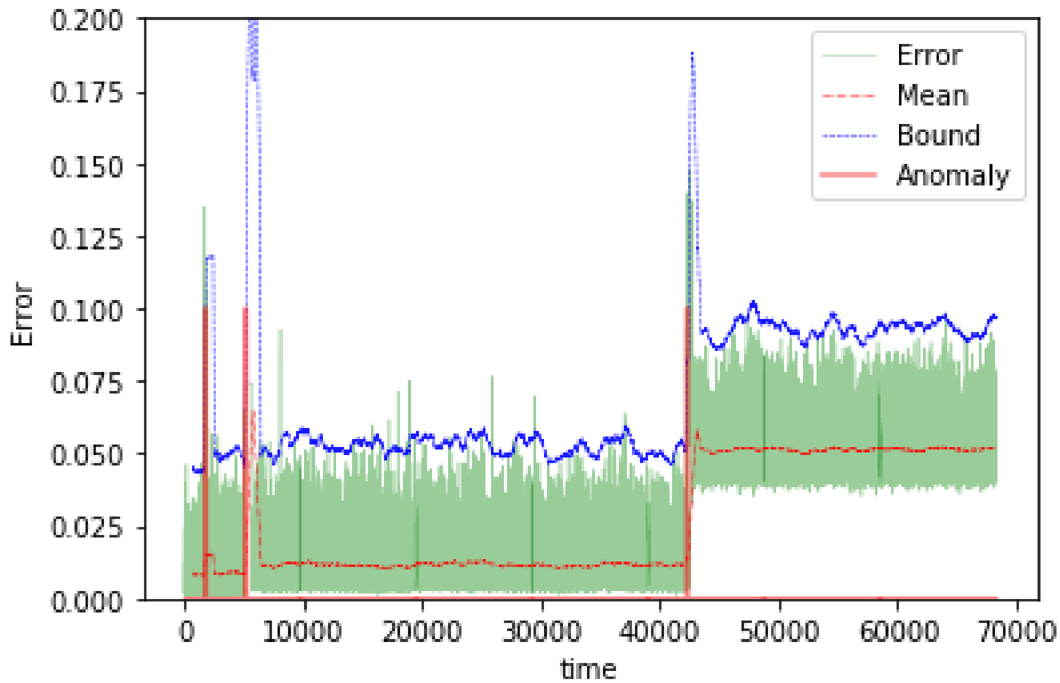


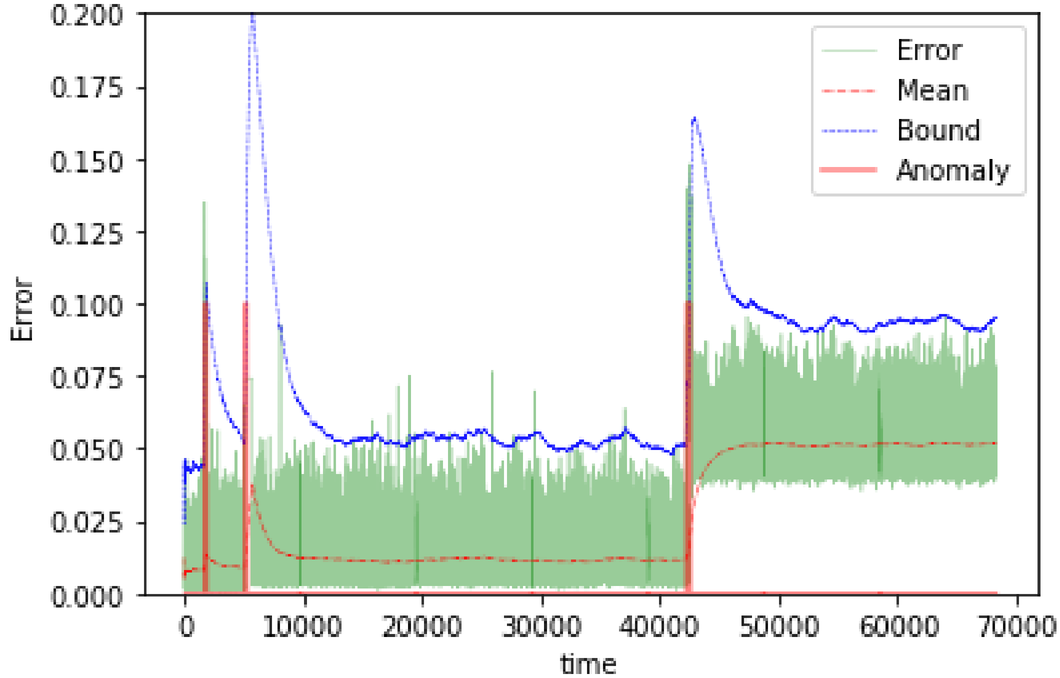

```
Training loss for final epoch is 0.008491841300623491
Validation loss for final epoch is 0.007539087241864763
---------- Beginning tests for gru2_5 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.15
0.10
0.05
0.00
−0.05
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

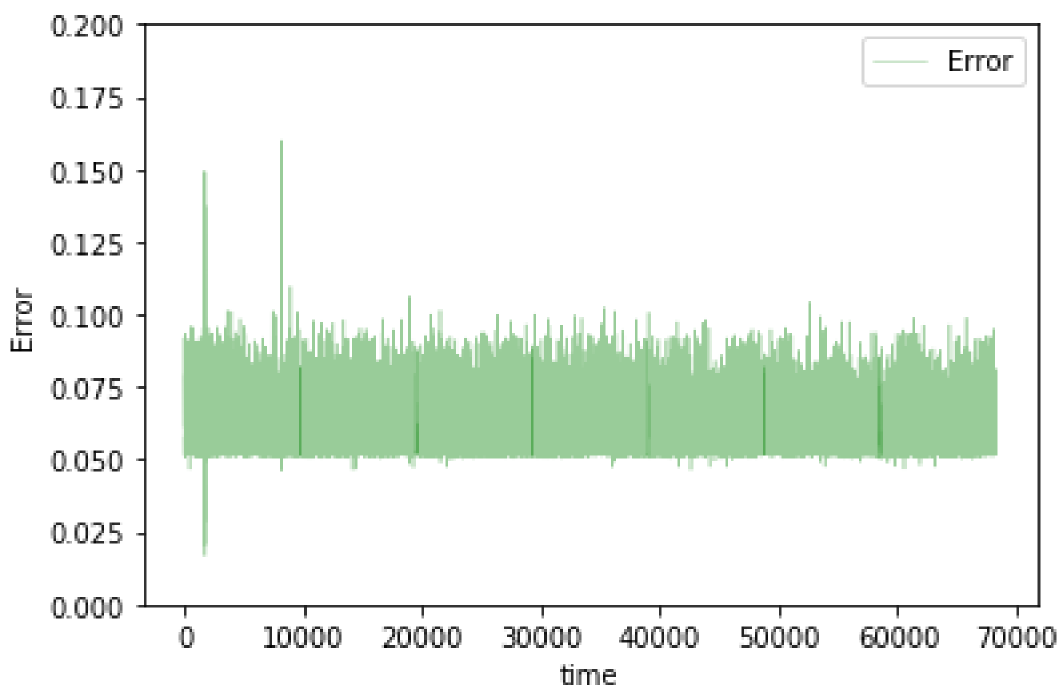

6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

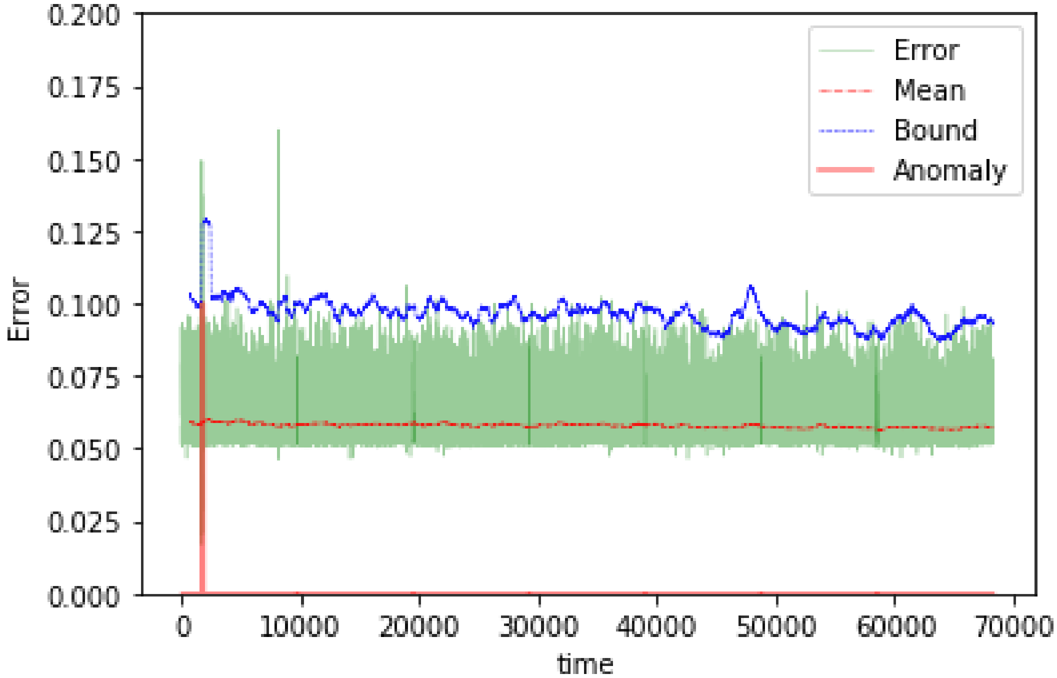

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

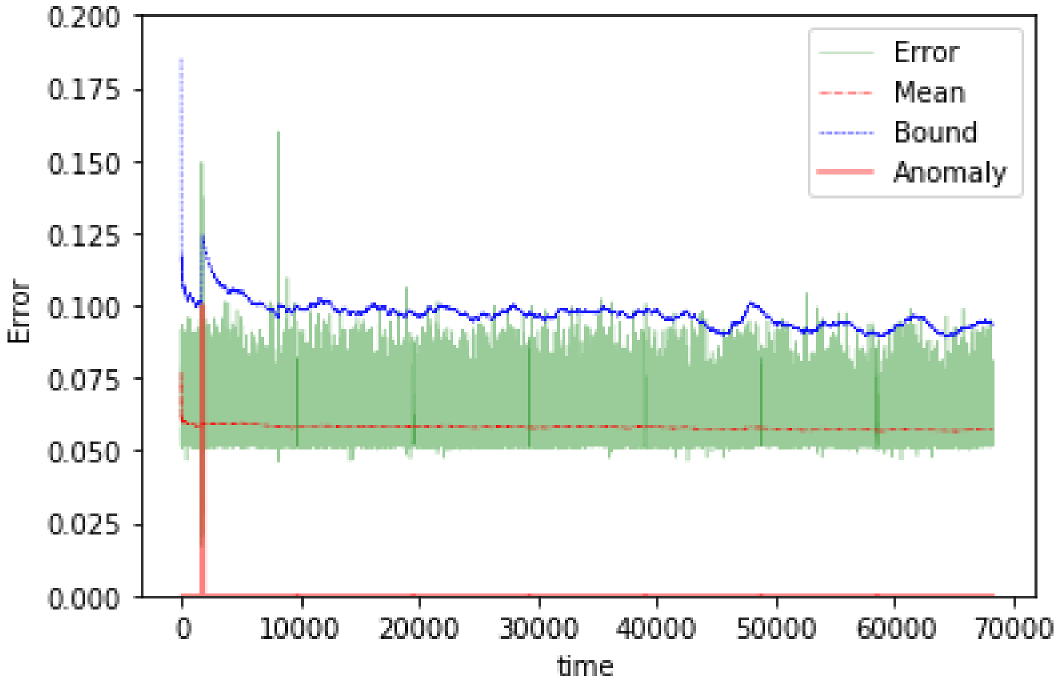

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



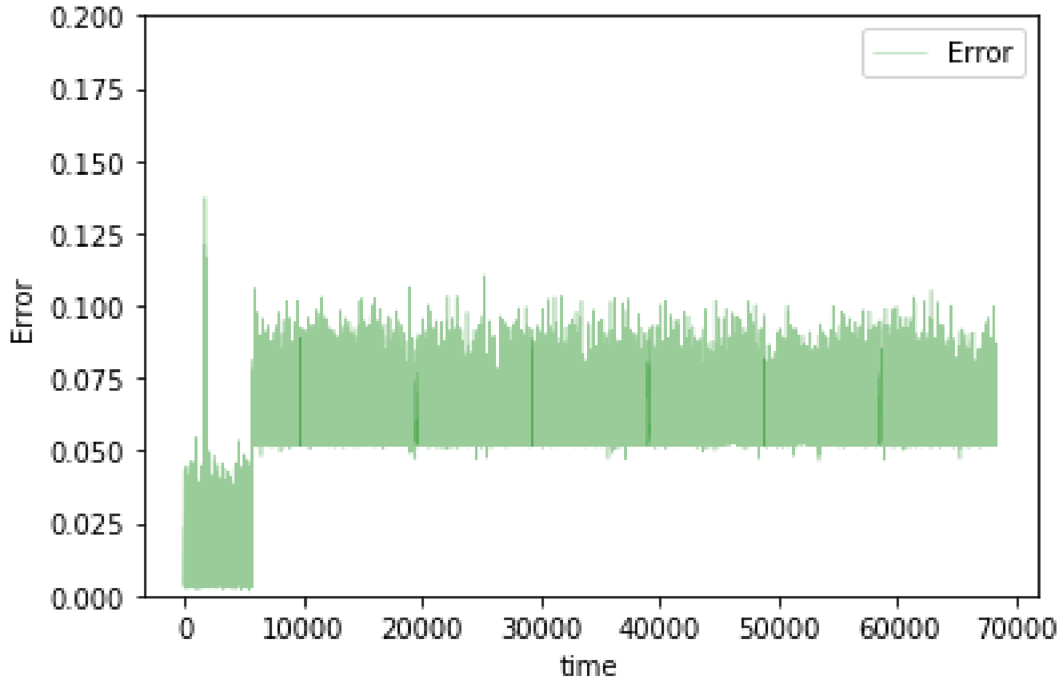

```
The mean error for gru2_5_disk_IO_start_ is 0.027149631138310915 for length 68294
Testing on Avg. load data.
```

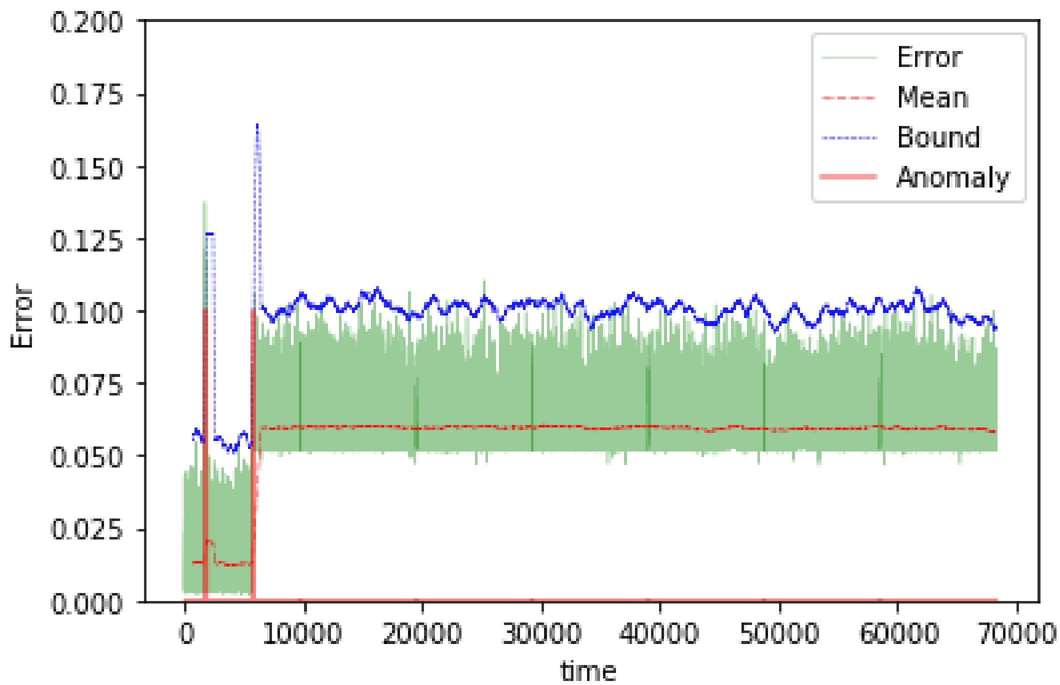


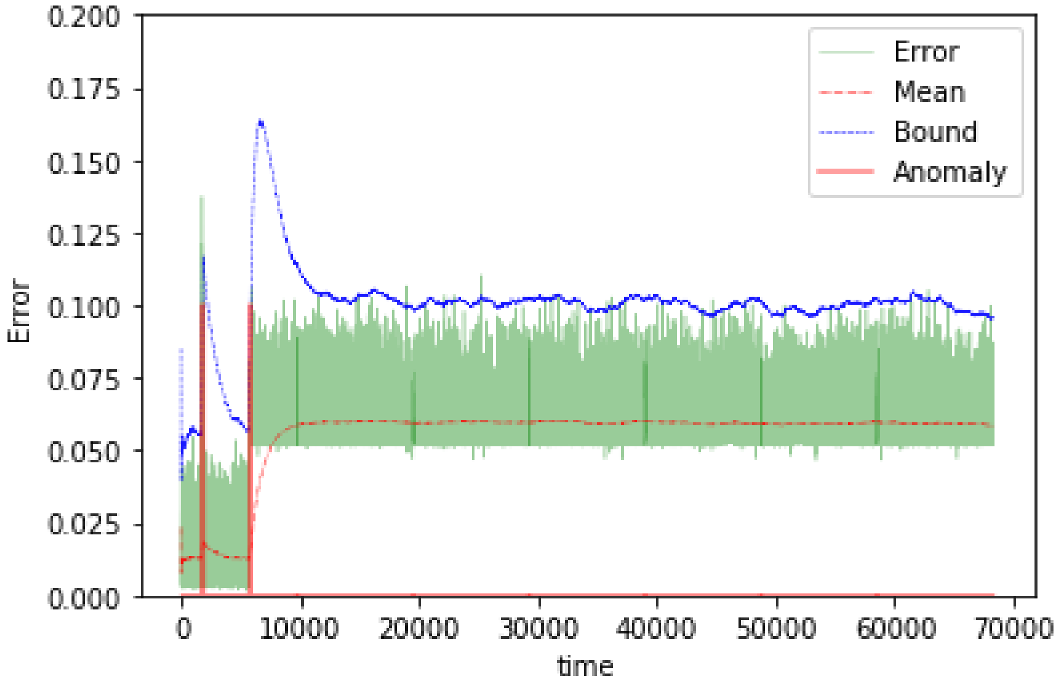

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

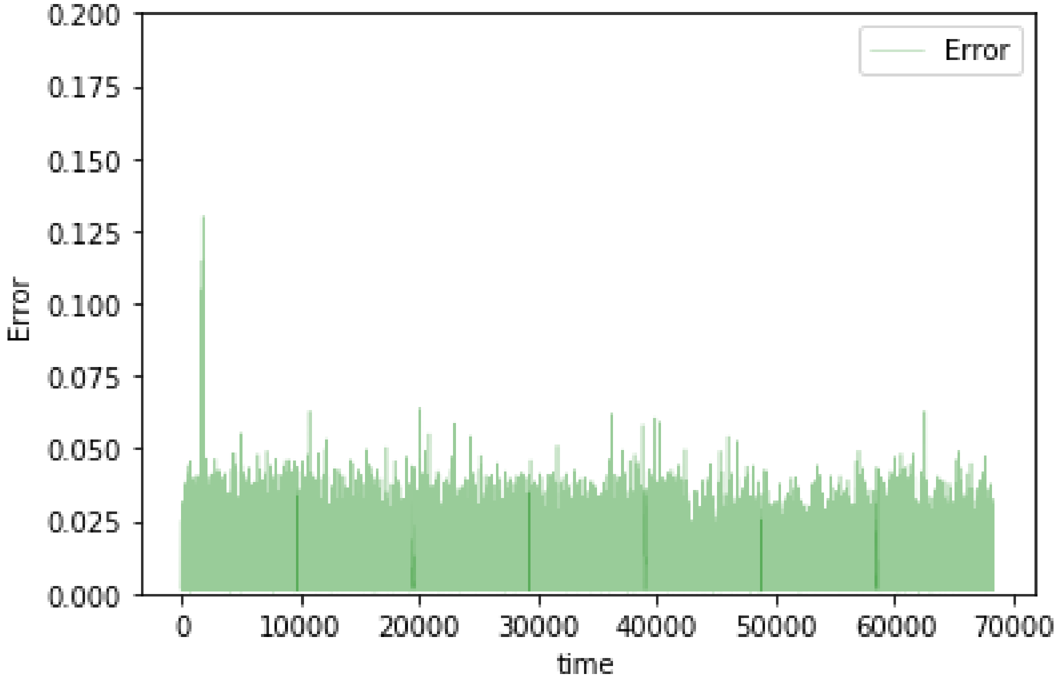
```
The mean error for gru2_5_avg_load_ is 0.05787720236168917 for length 68294
Testing on app change early data.
```

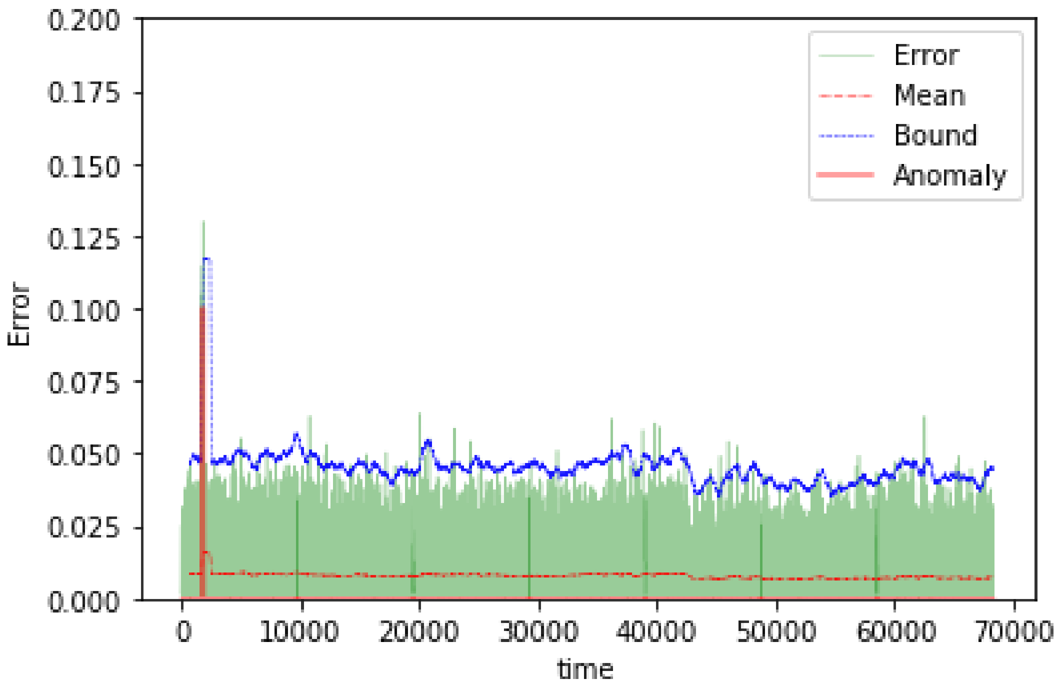


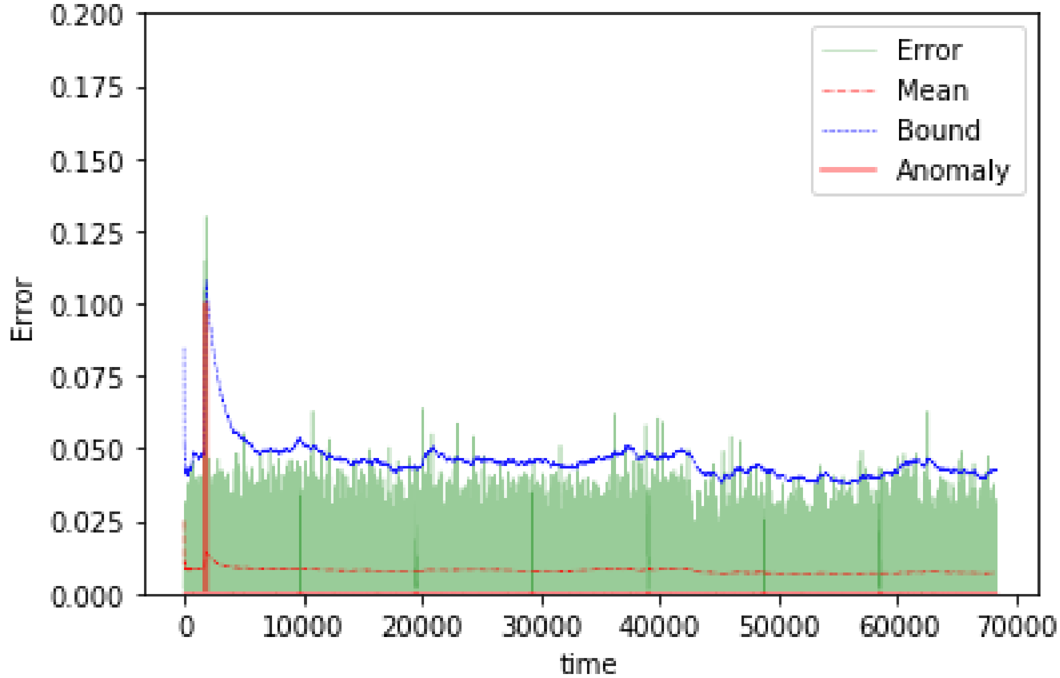





```
The mean error for gru2_5_app_change_early_ is 0.05561973547054804 for length 68294
Testing on Normal data.
```


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
Error
0
10000
20000
30000
40000
50000
60000
70000
time

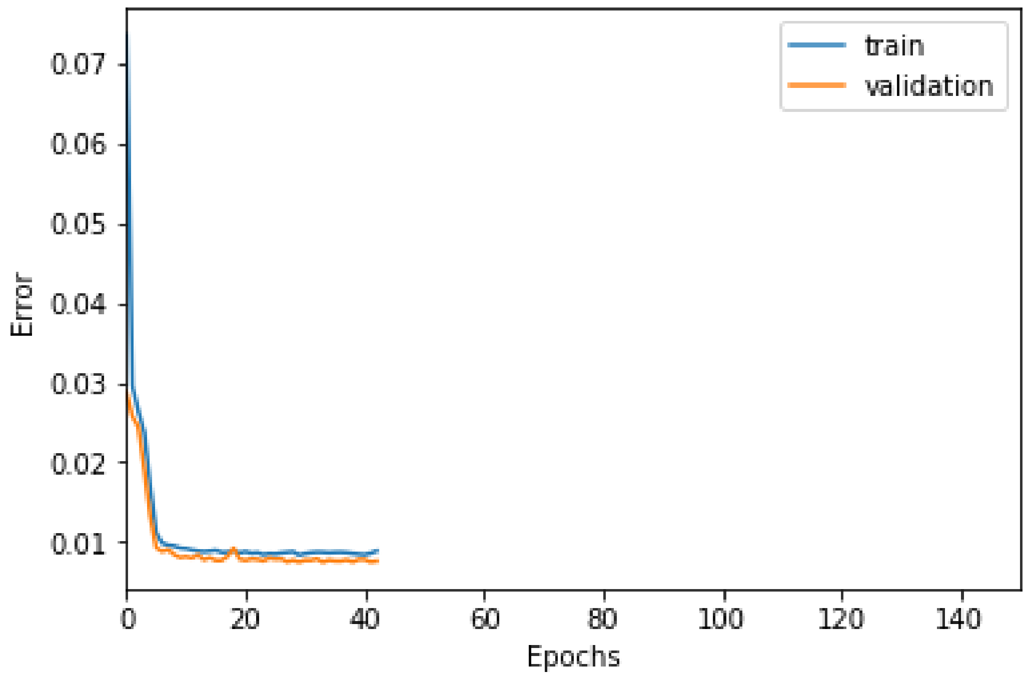

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
Error
Mean
Bound
Anomaly
0
10000
20000
30000
40000
50000
60000
70000
time



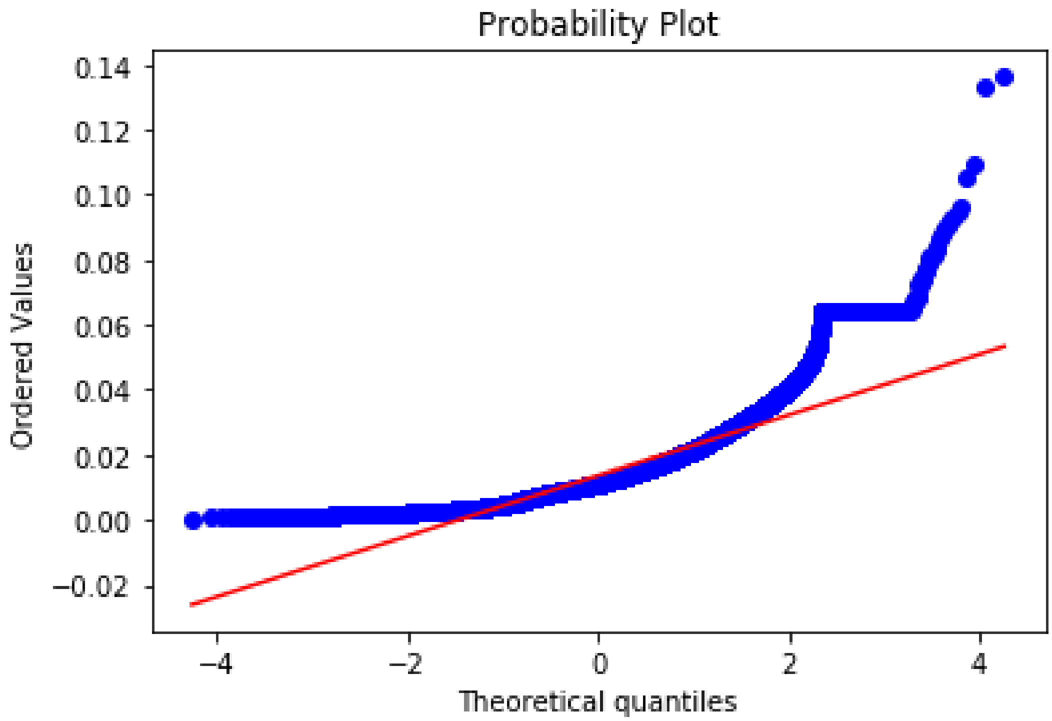

```
The mean error for gru2_5_normal_ is 0.007843673315430828 for length 68294
====================
```

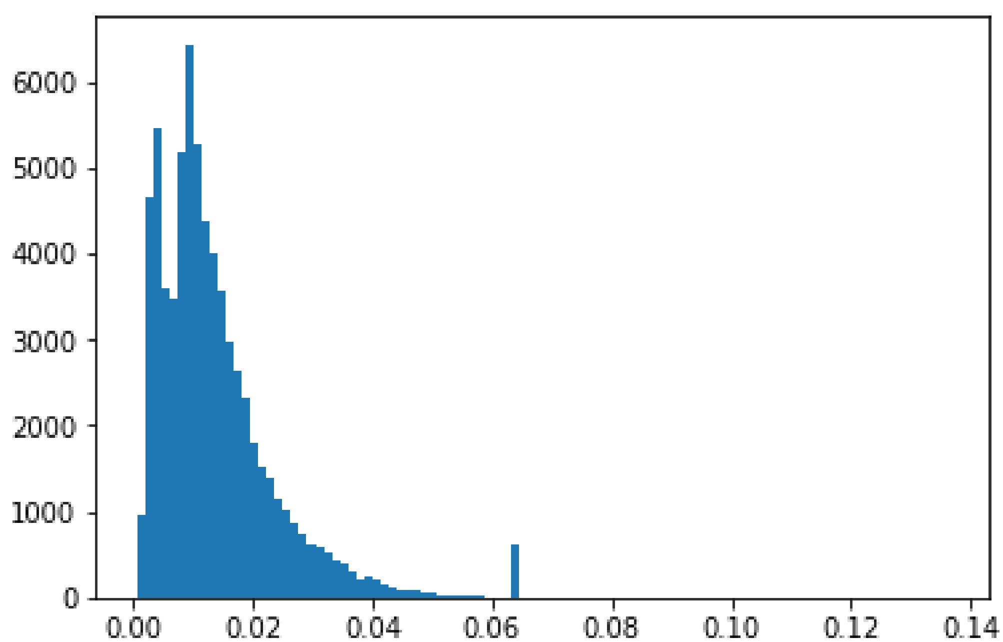
10 steps

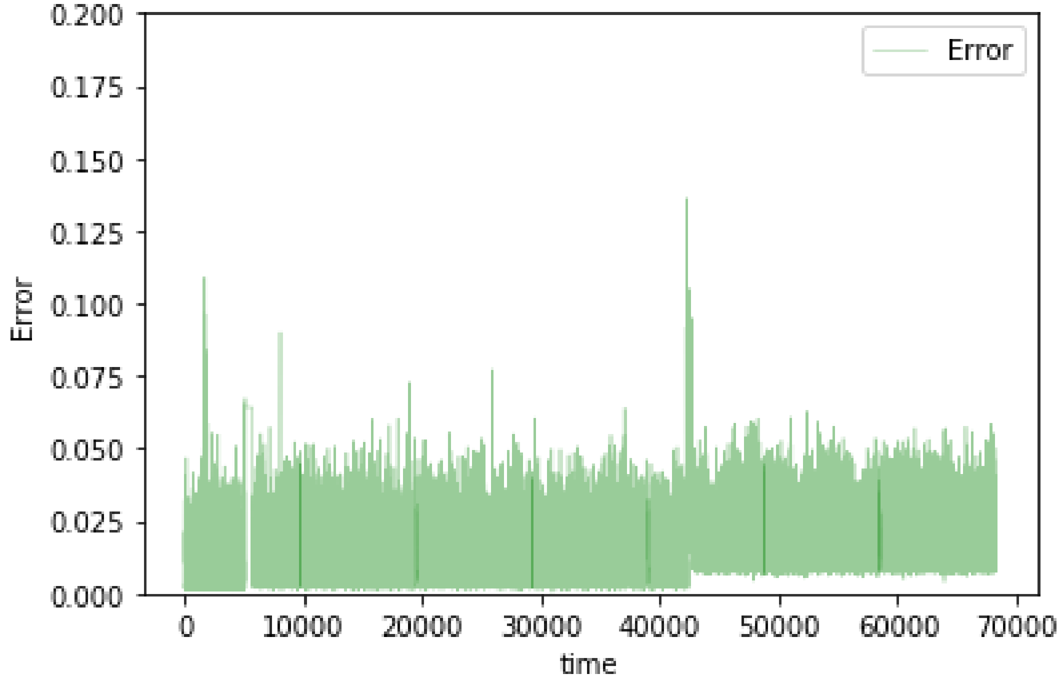
```
In [187]: TIMESTEPS = 10
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS, 0)
          vgen = flat_generator(val_X, TIMESTEPS, 0)
          name = "gru2_10"

In [188]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(10, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [189]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [190]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```



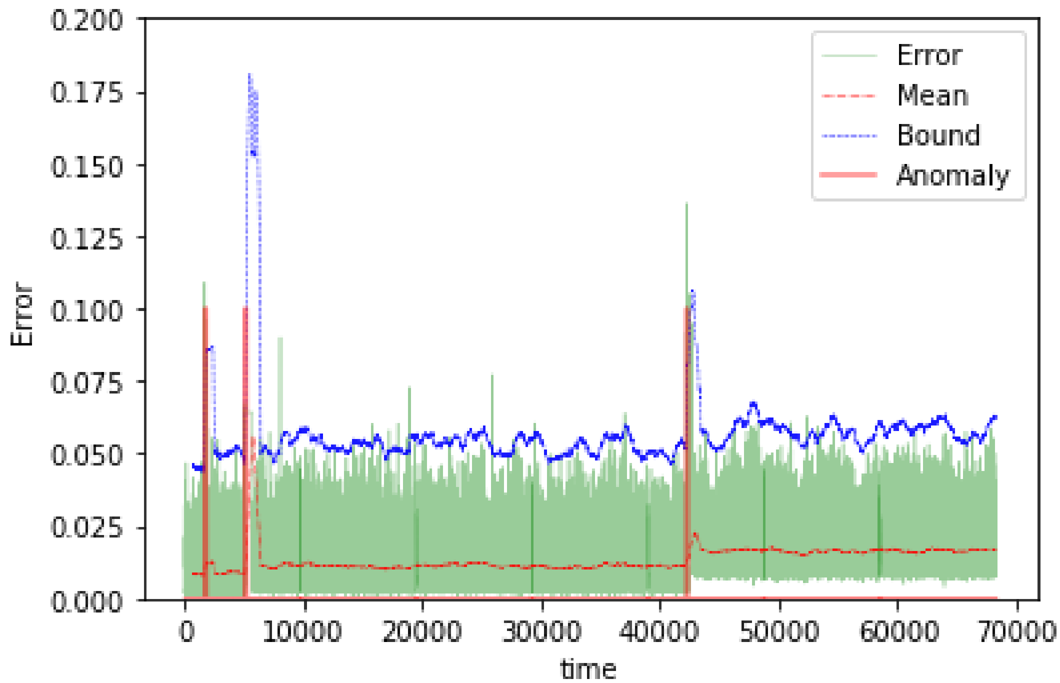

```
Training loss for final epoch is 0.008814556191908195
Validation loss for final epoch is 0.0075454373116372155
---------- Beginning tests for gru2_10 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
−0.02
−4
−2
0
2
4
Theoretical quantiles

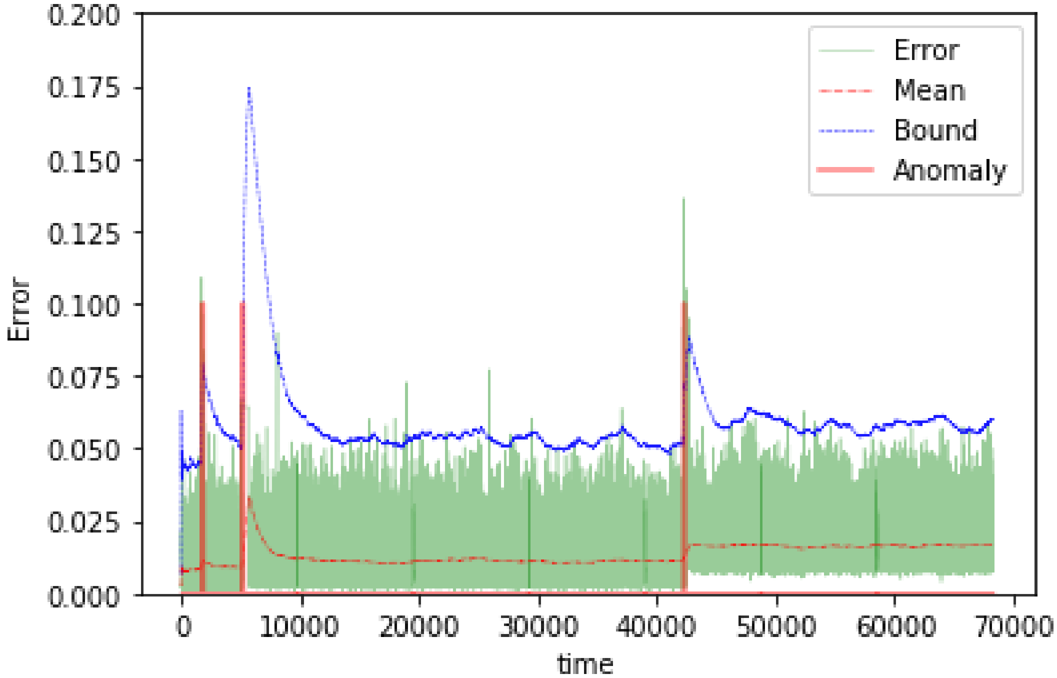

6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14



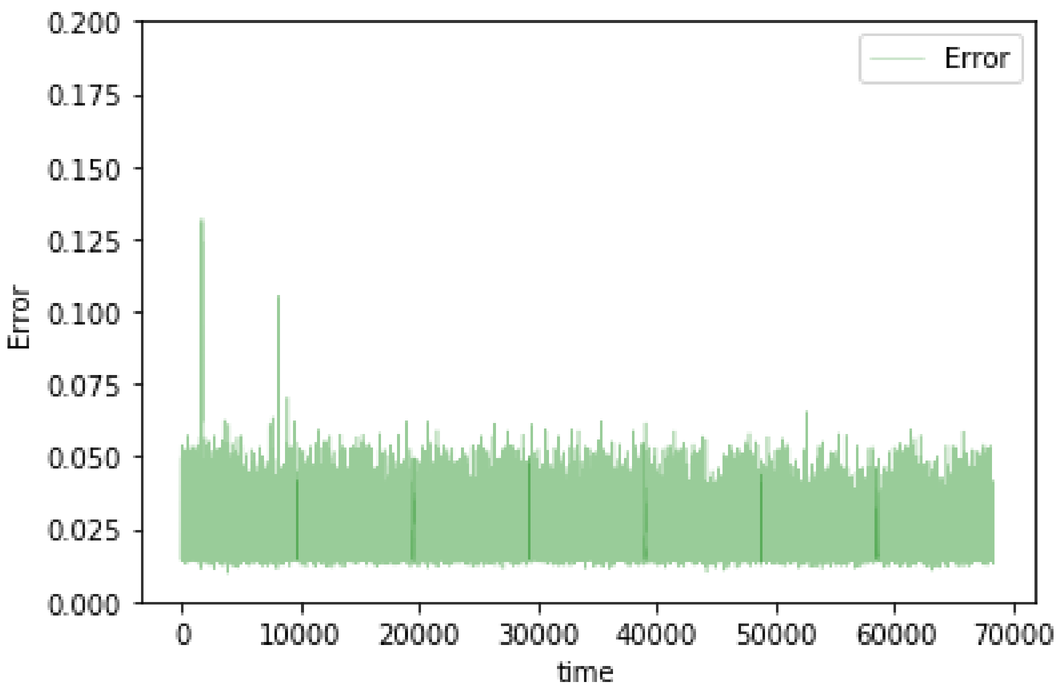
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error

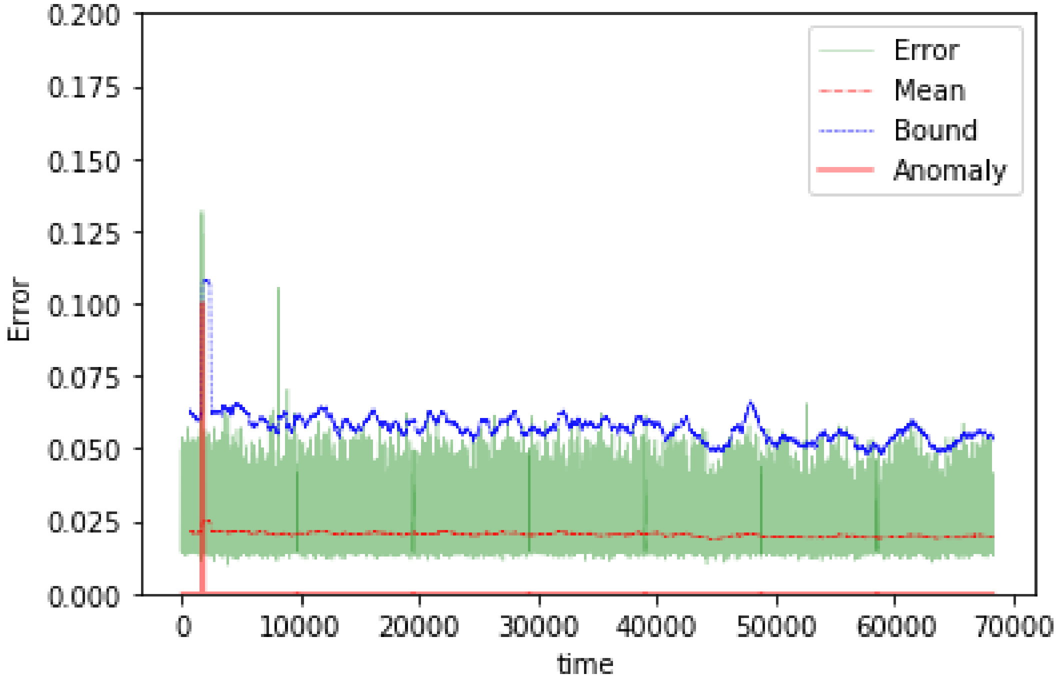


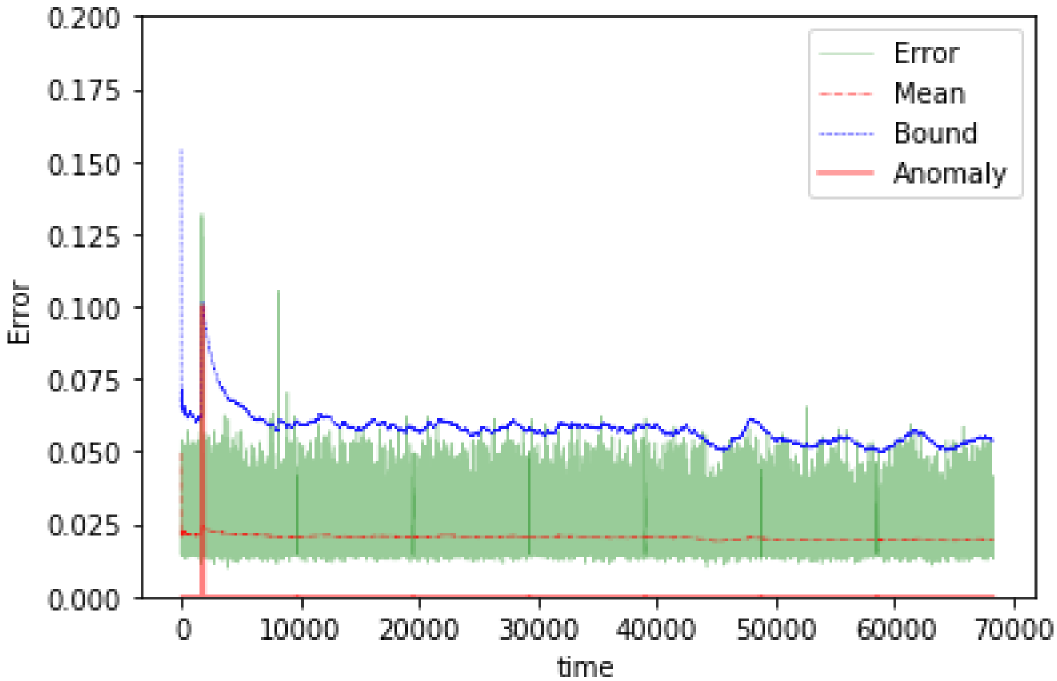
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly



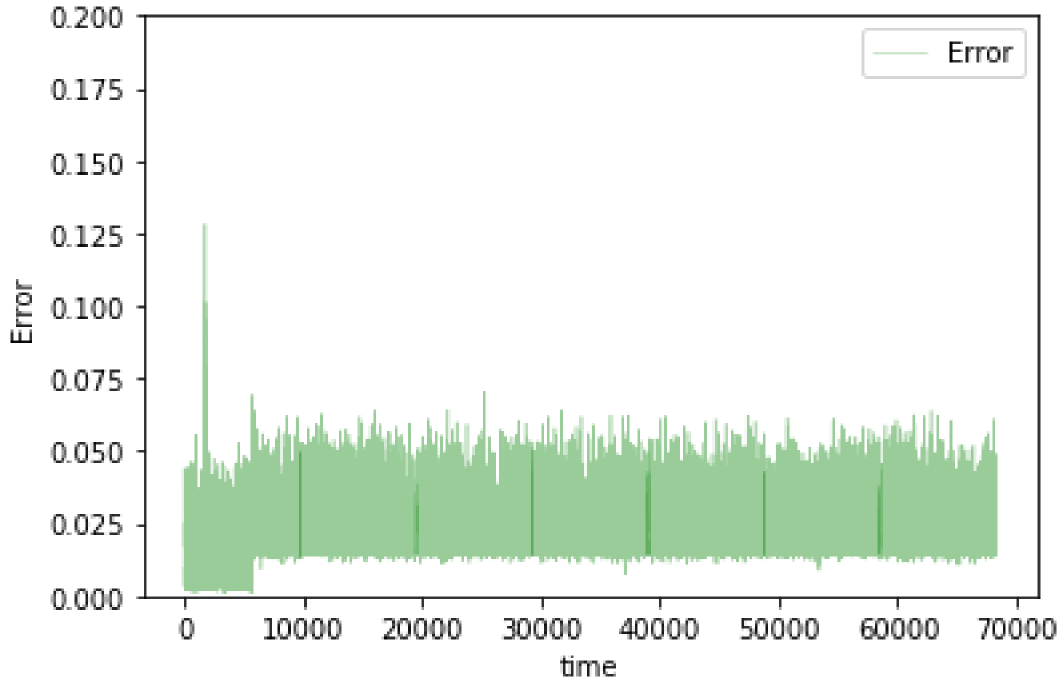

```
The mean error for gru2_10_disk_IO_start_ is 0.013594947602921835 for length 68289
Testing on Avg. load data.
```

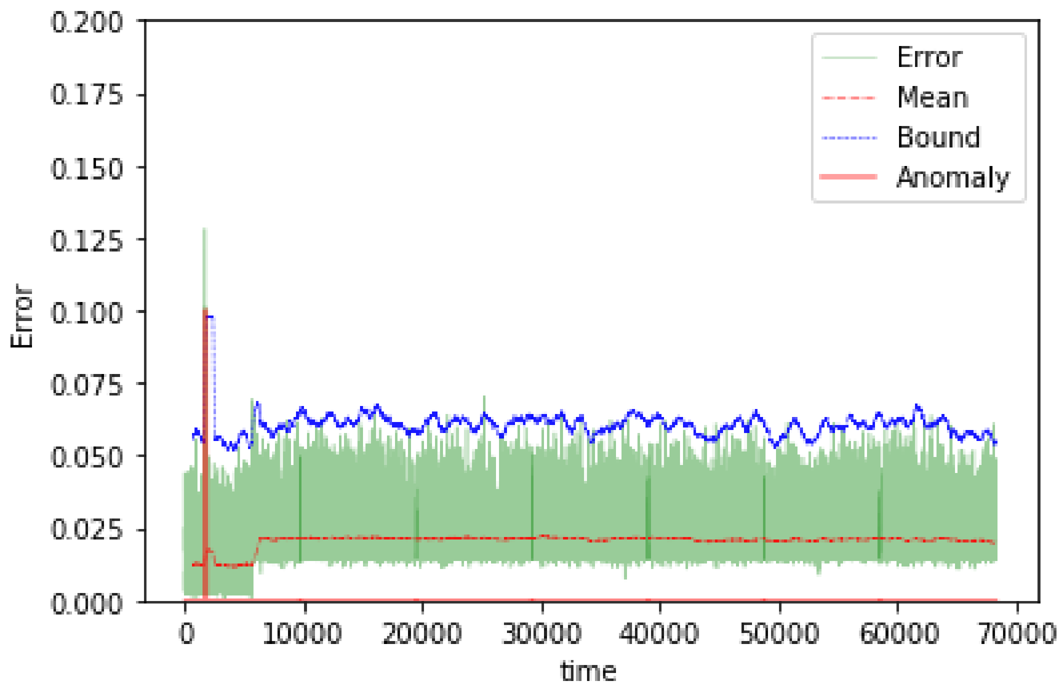


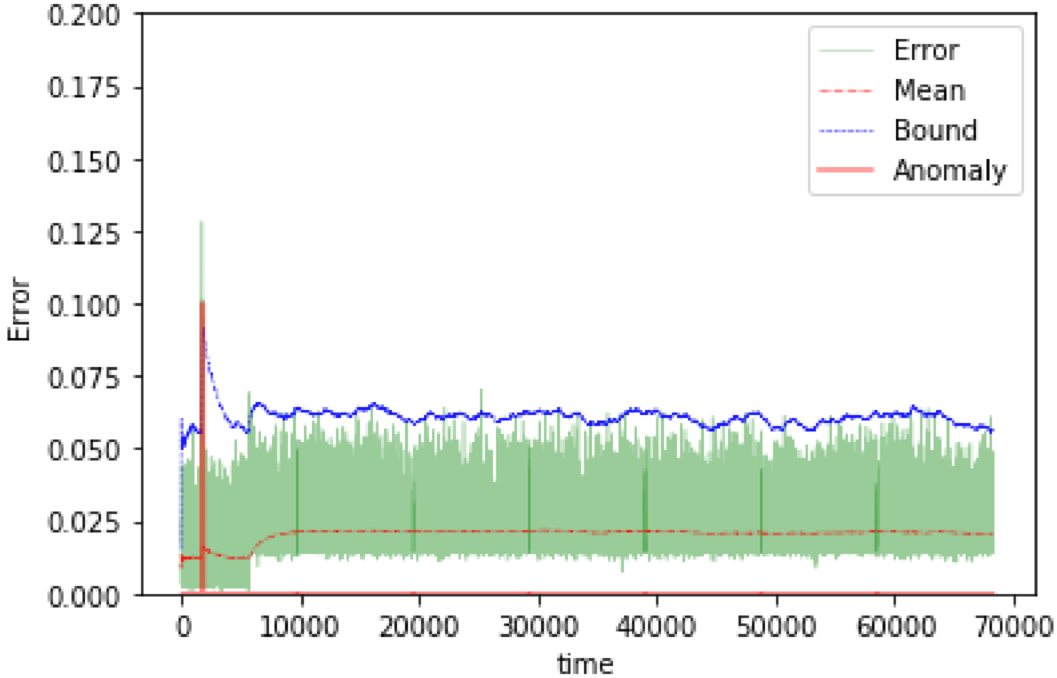

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

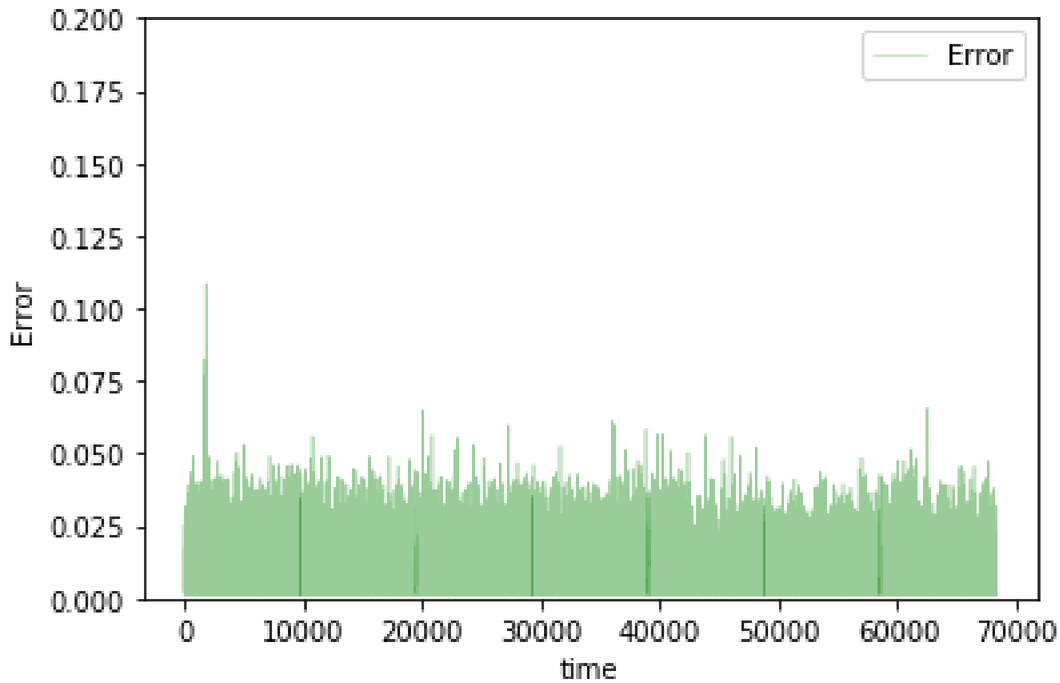
```
The mean error for gru2_10_avg_load_ is 0.020341845001067404 for length 68289
Testing on app change early data.
```



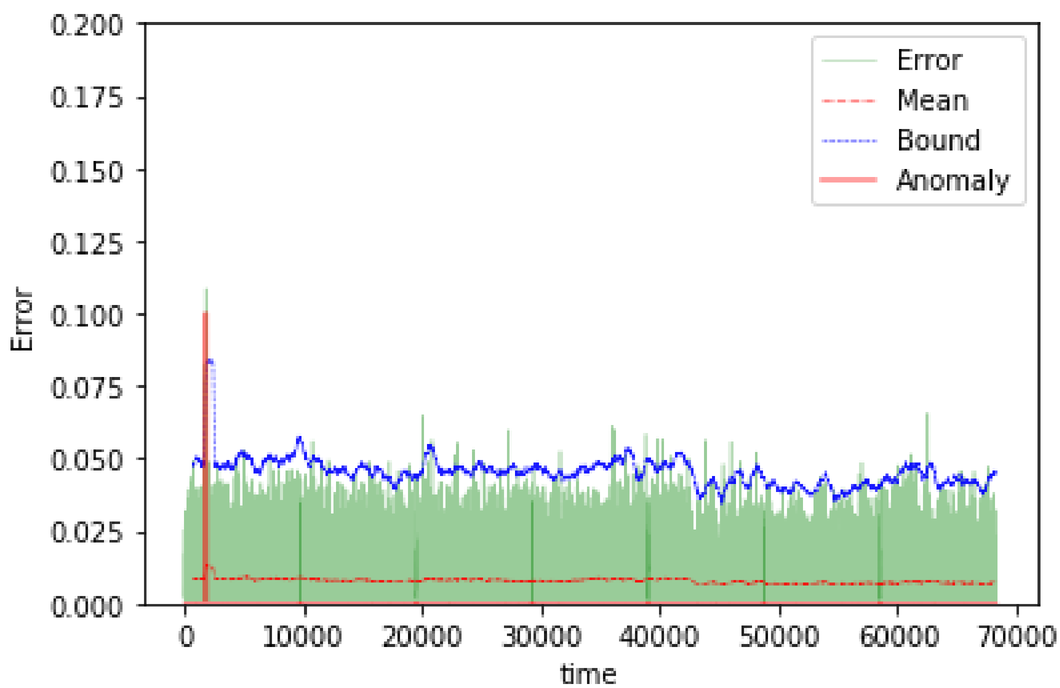




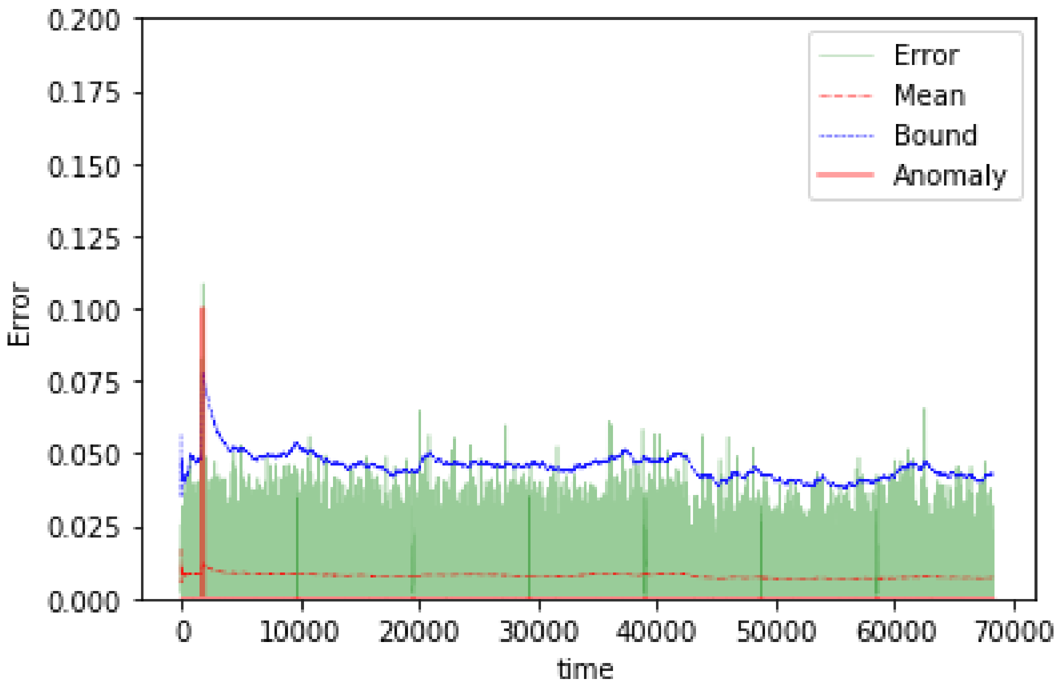

```
The mean error for gru2_10_app_change_early_ is 0.020503378121962672 for length 68289
Testing on Normal data.
```



Error
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000



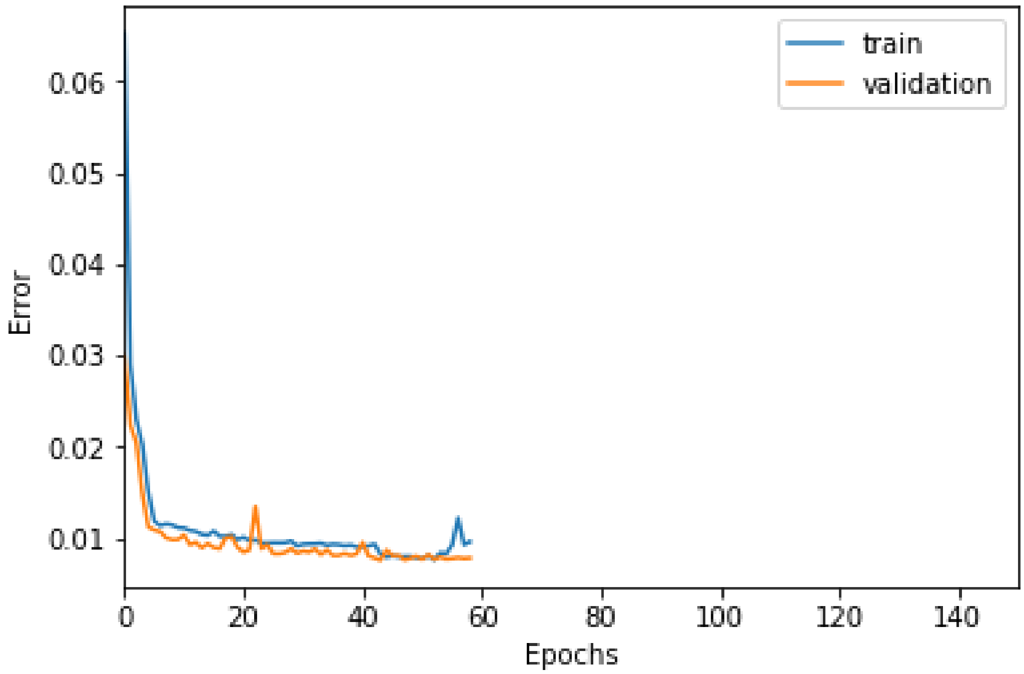
Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000



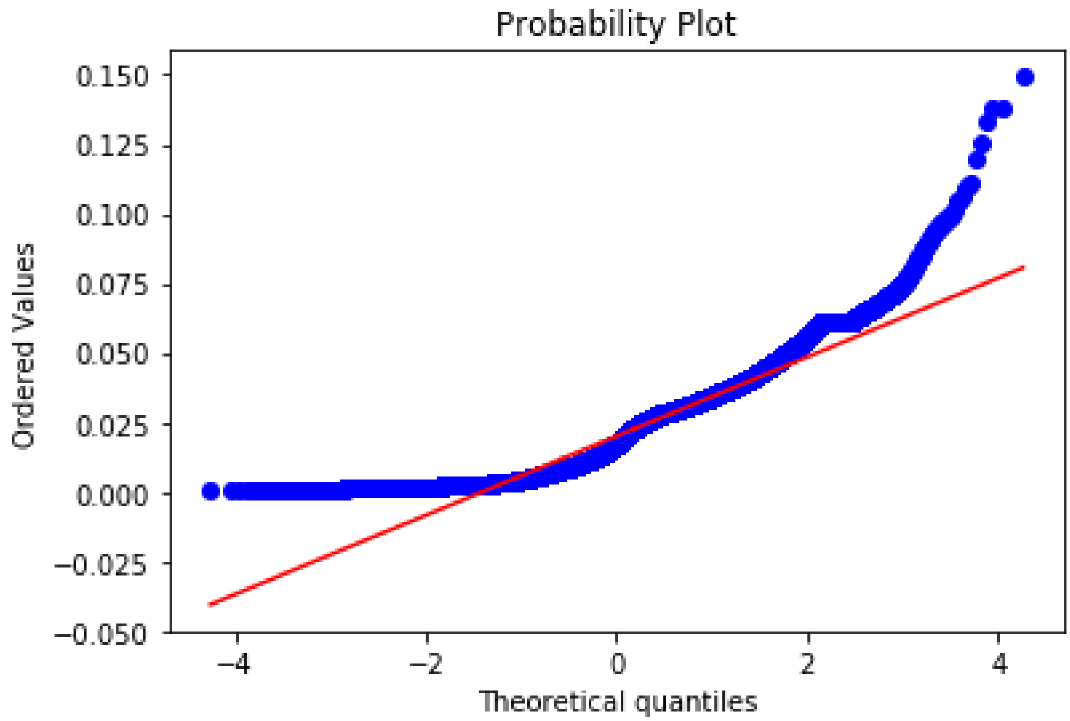

```
The mean error for gru2_10_normal_ is 0.007825121130787977 for length 68289
====================
```

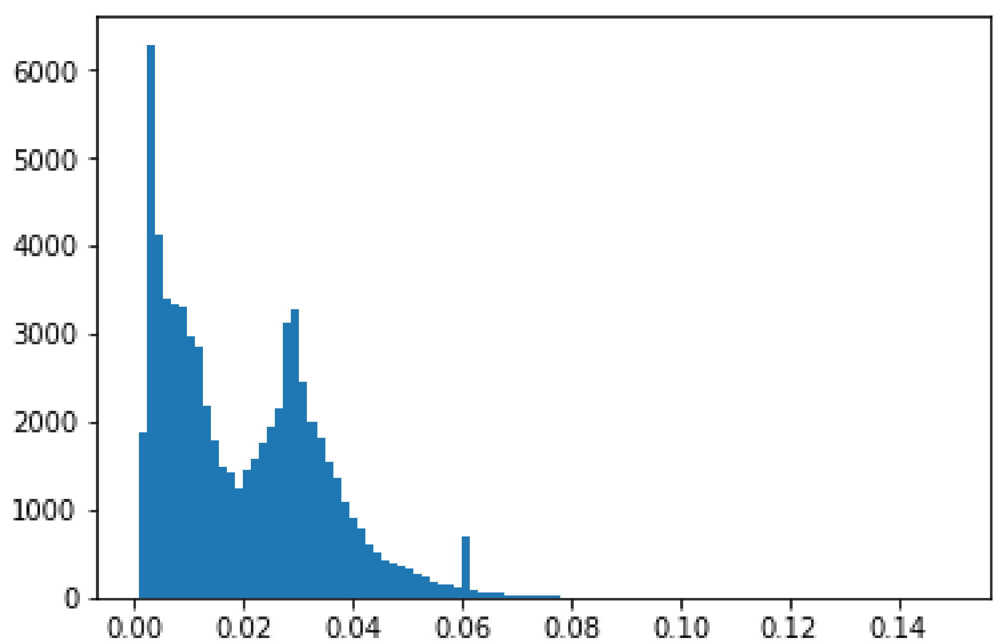
20 steps

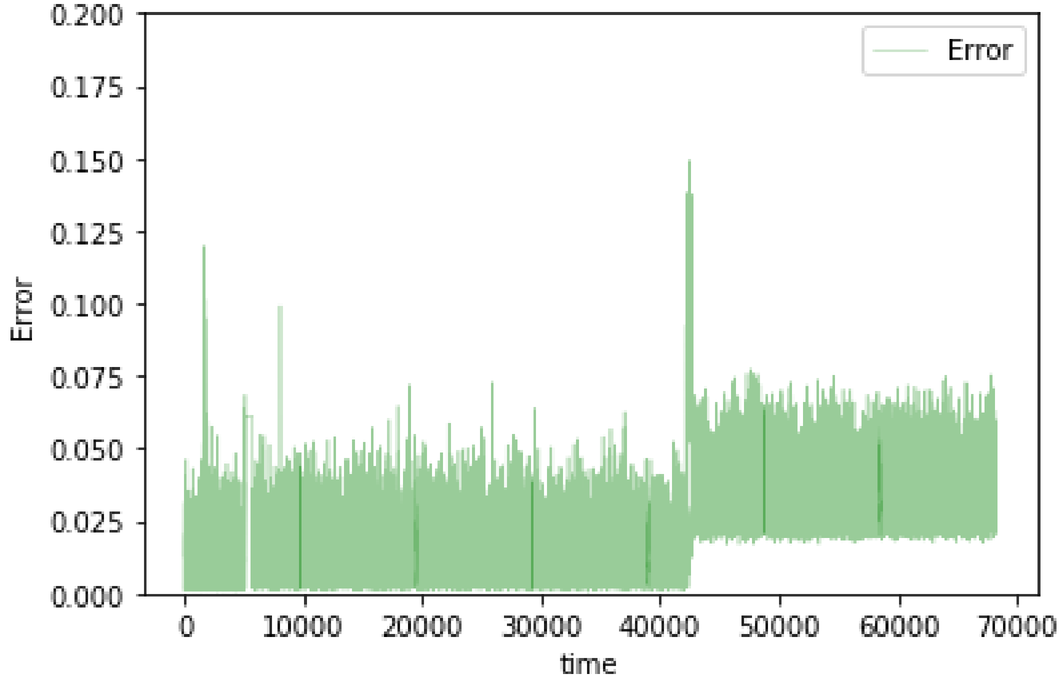
```
In [191]: TIMESTEPS = 20
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru2_20"

In [192]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(10, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [193]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [194]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

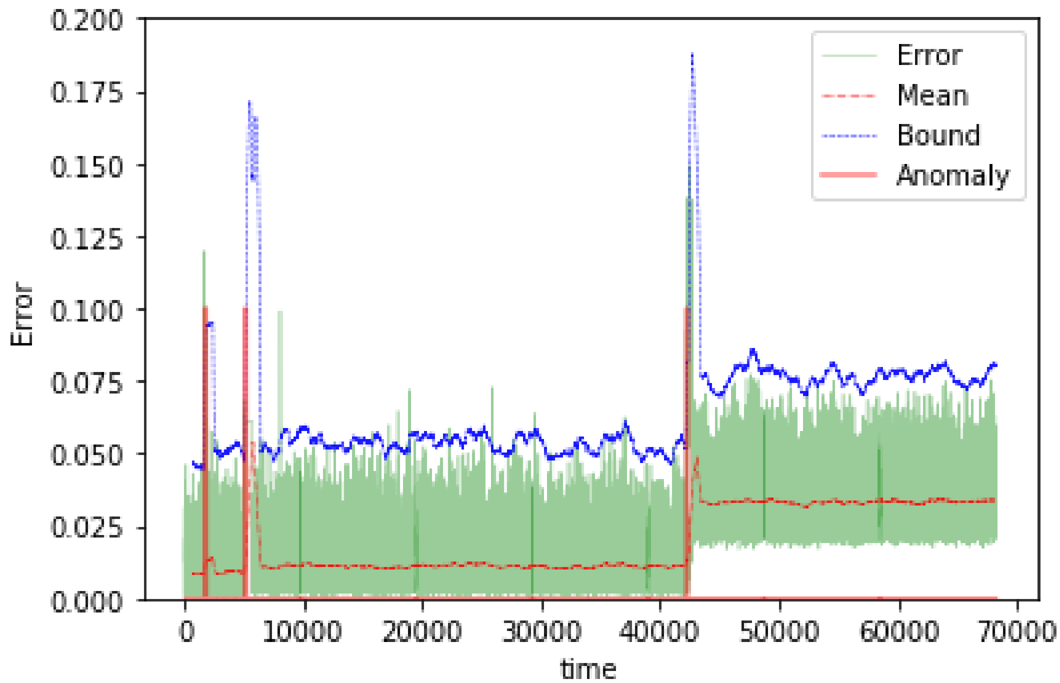


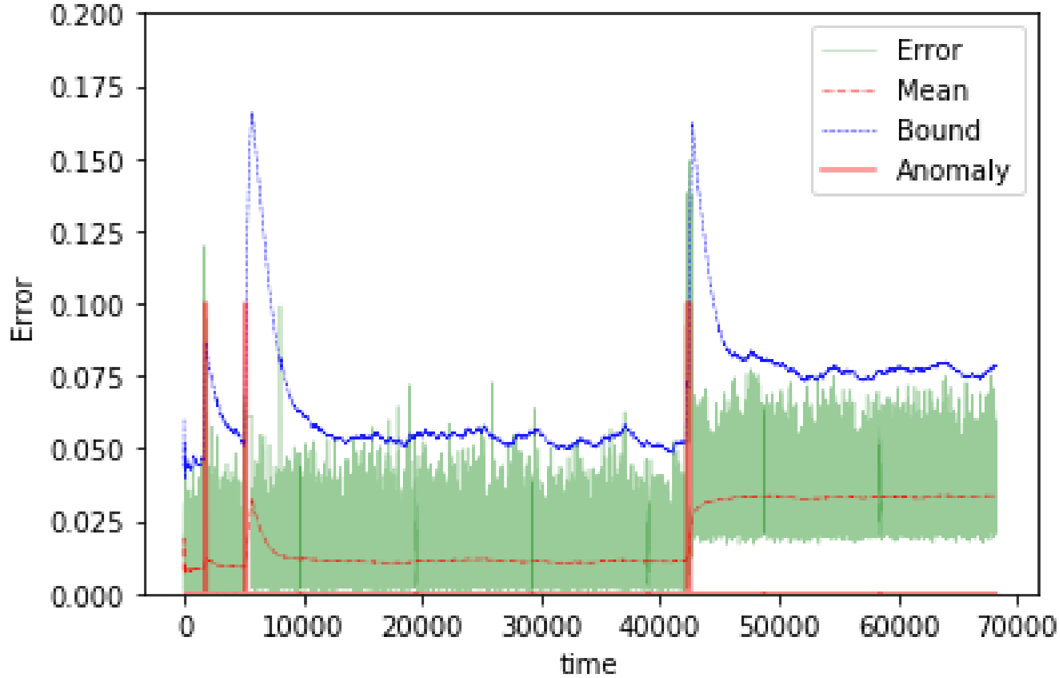

```
Training loss for final epoch is 0.009587624242063612
Validation loss for final epoch is 0.007797279035439714
---------- Beginning tests for gru2_20 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−4
−2
0
2
4

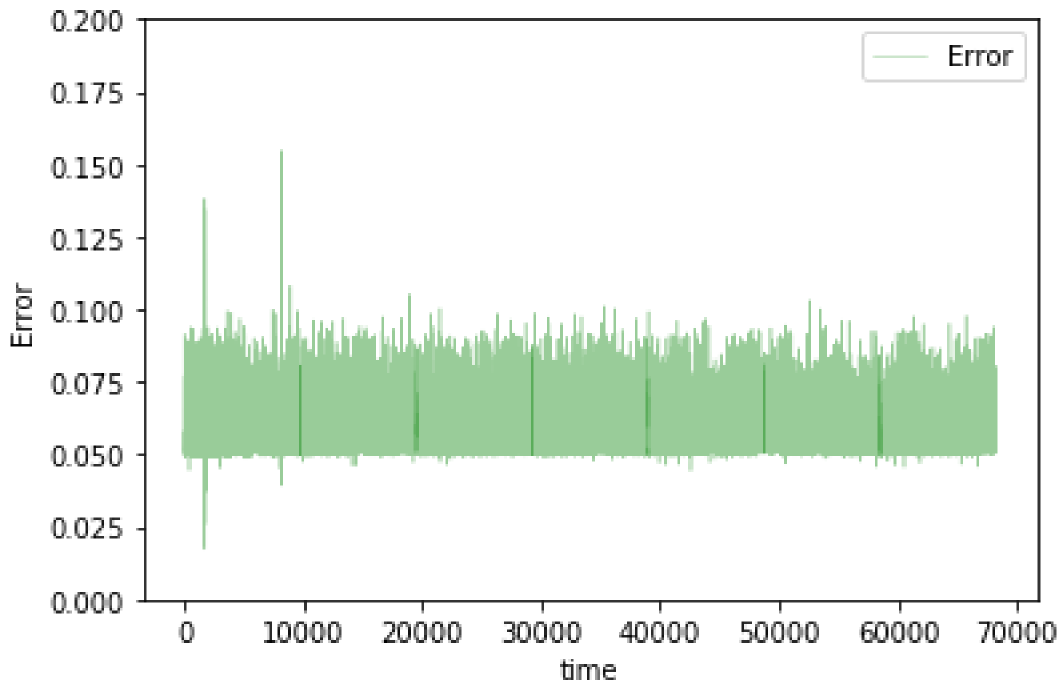

6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

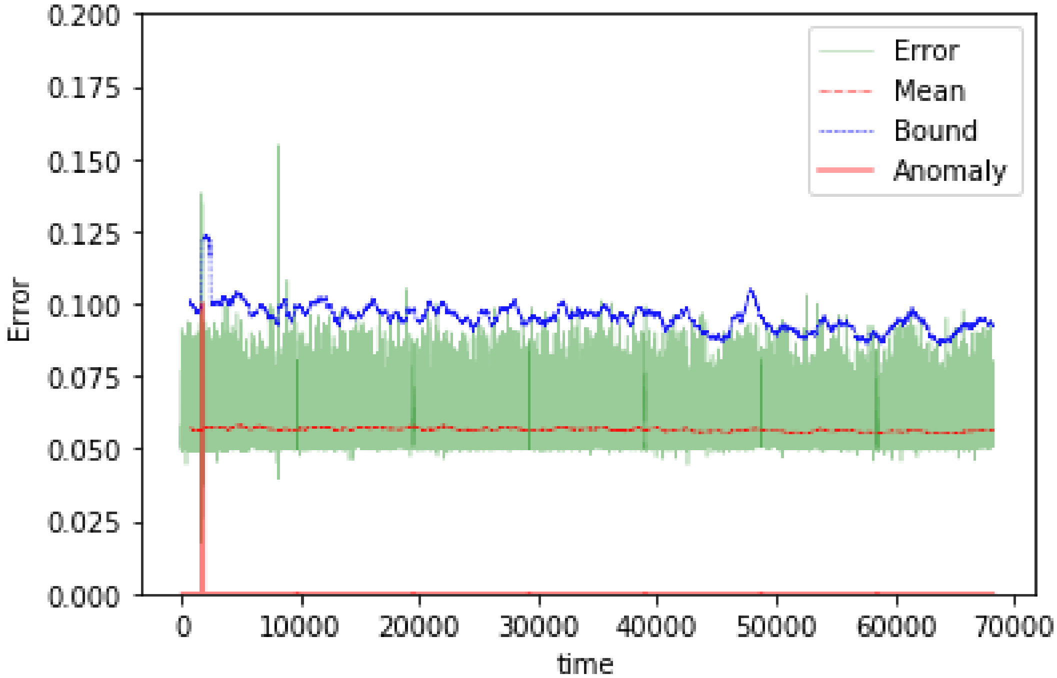

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

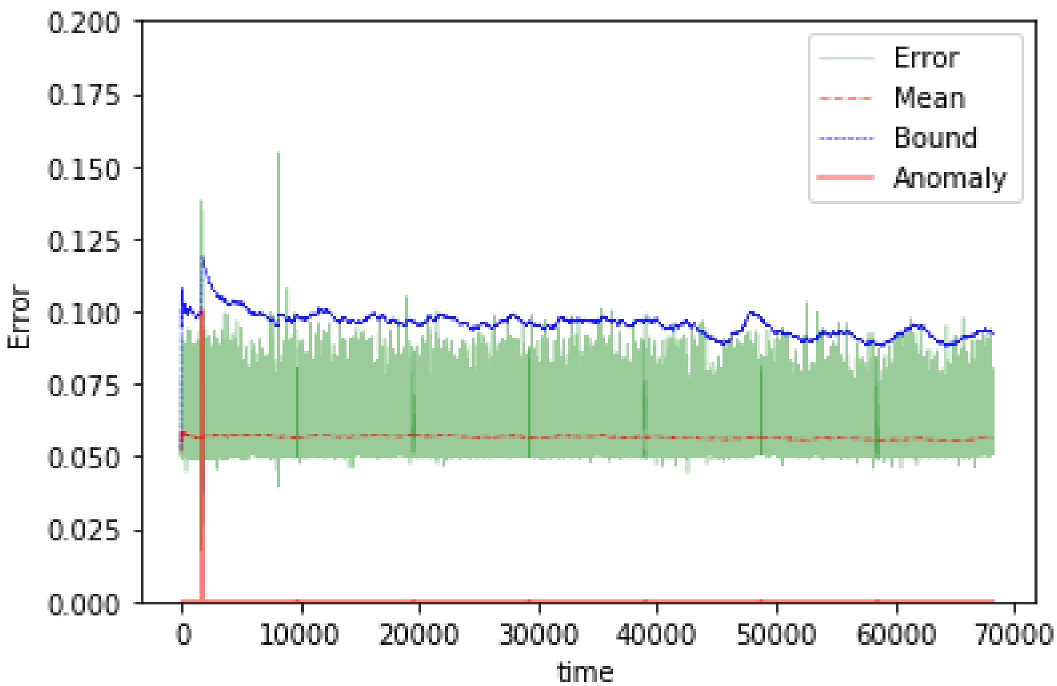



```
The mean error for gru2_20_disk_IO_start_ is 0.02004085323599448 for length 68279
Testing on Avg. load data.
```



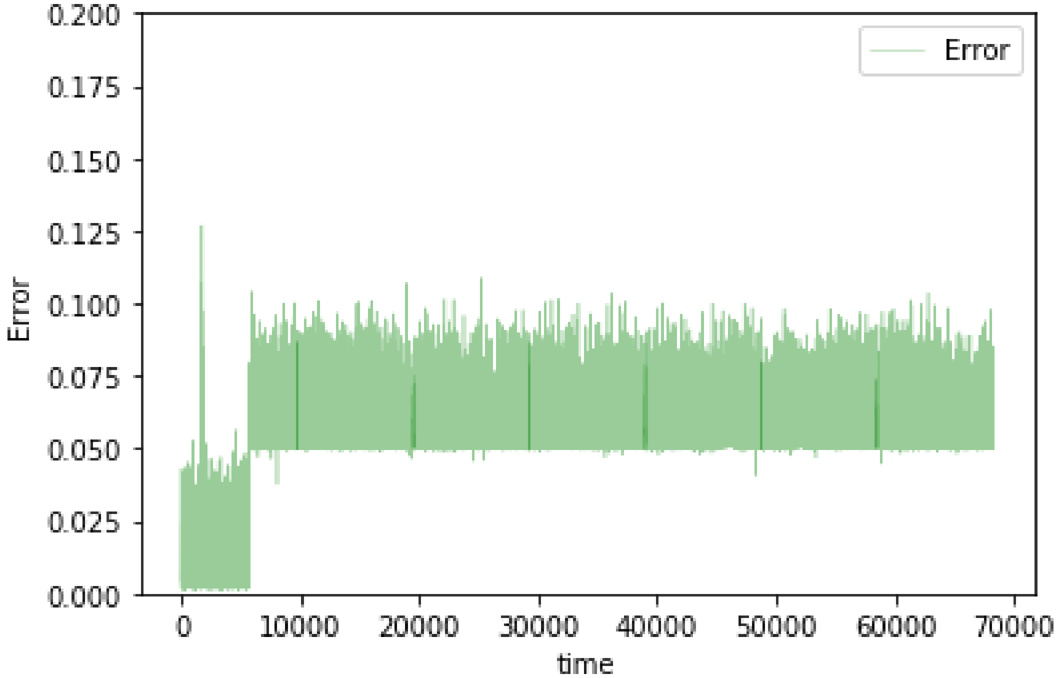


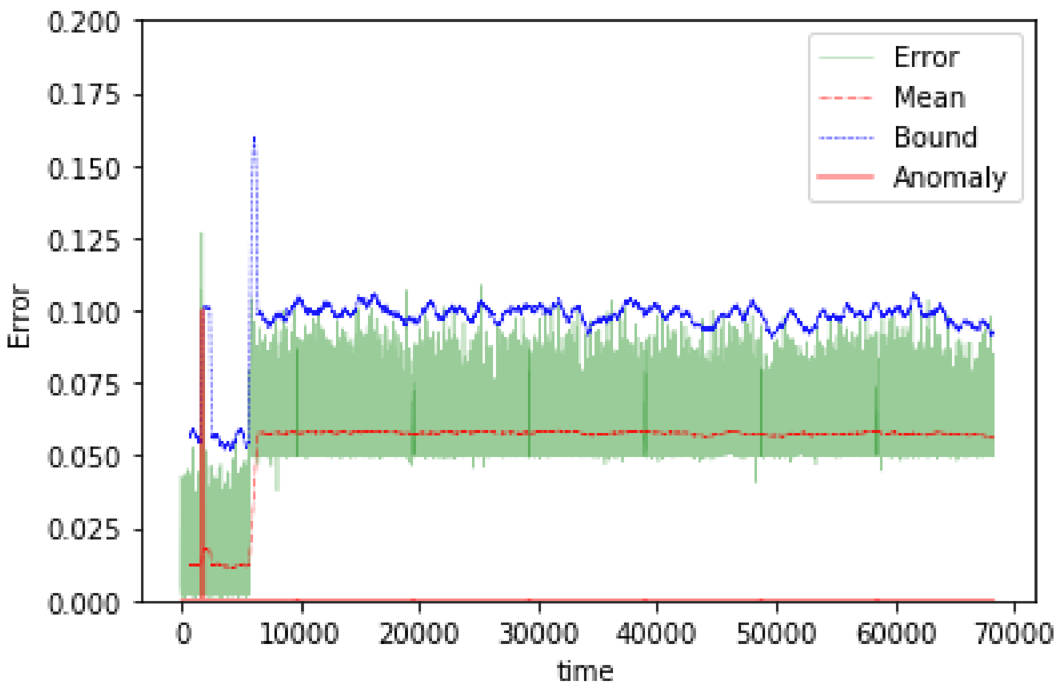
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

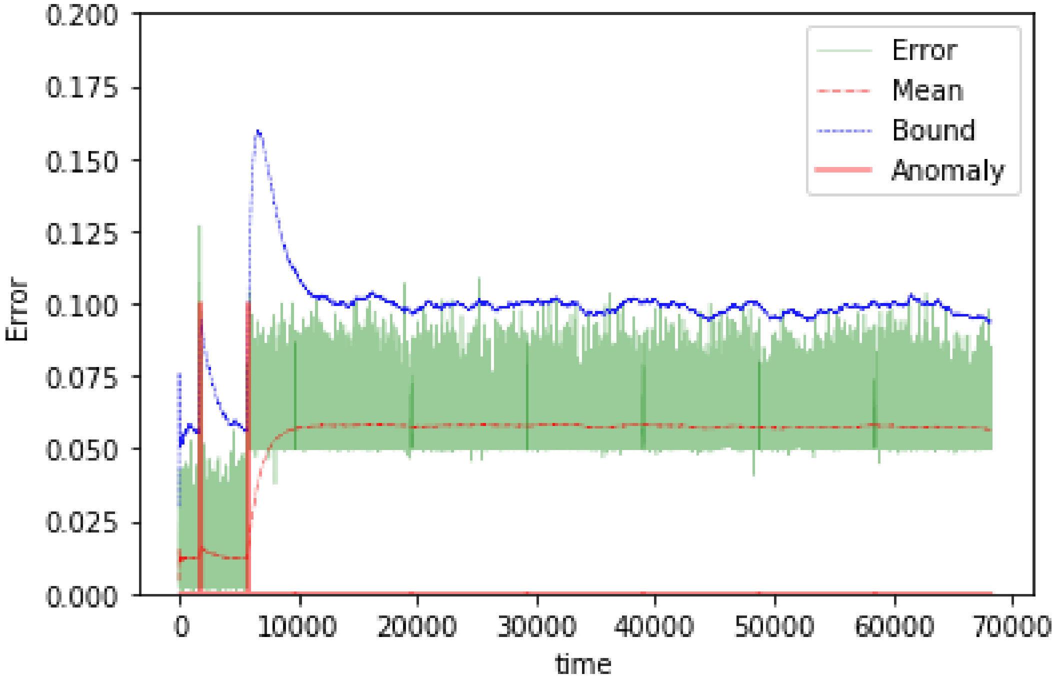


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

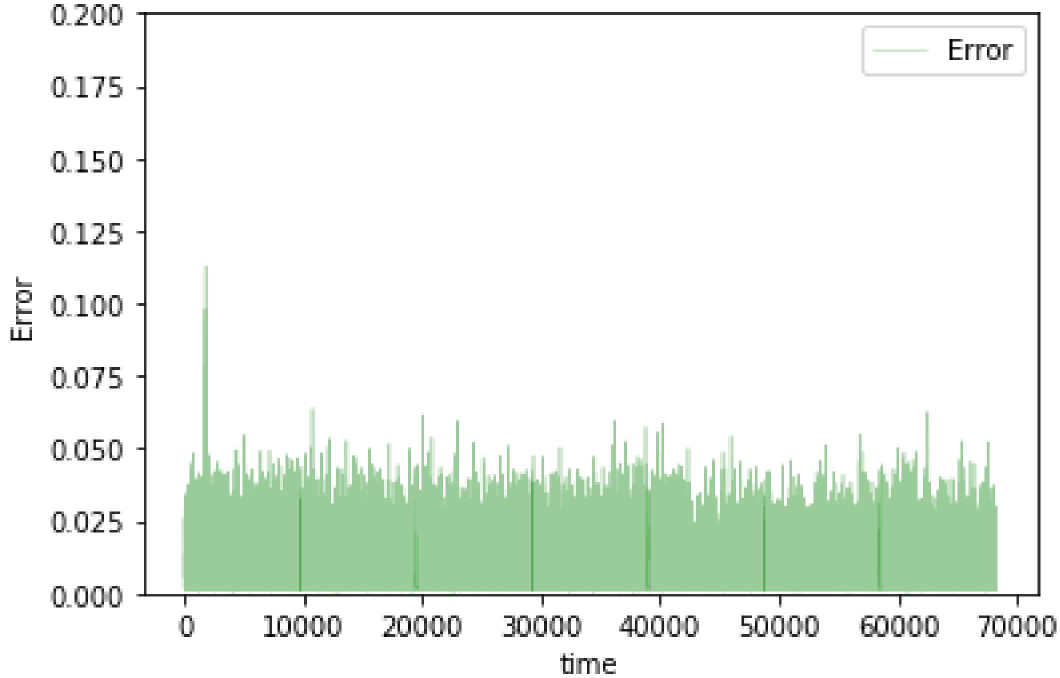
```
The mean error for gru2_20_avg_load_ is 0.05653988270451578 for length 68279
Testing on app change early data.
```

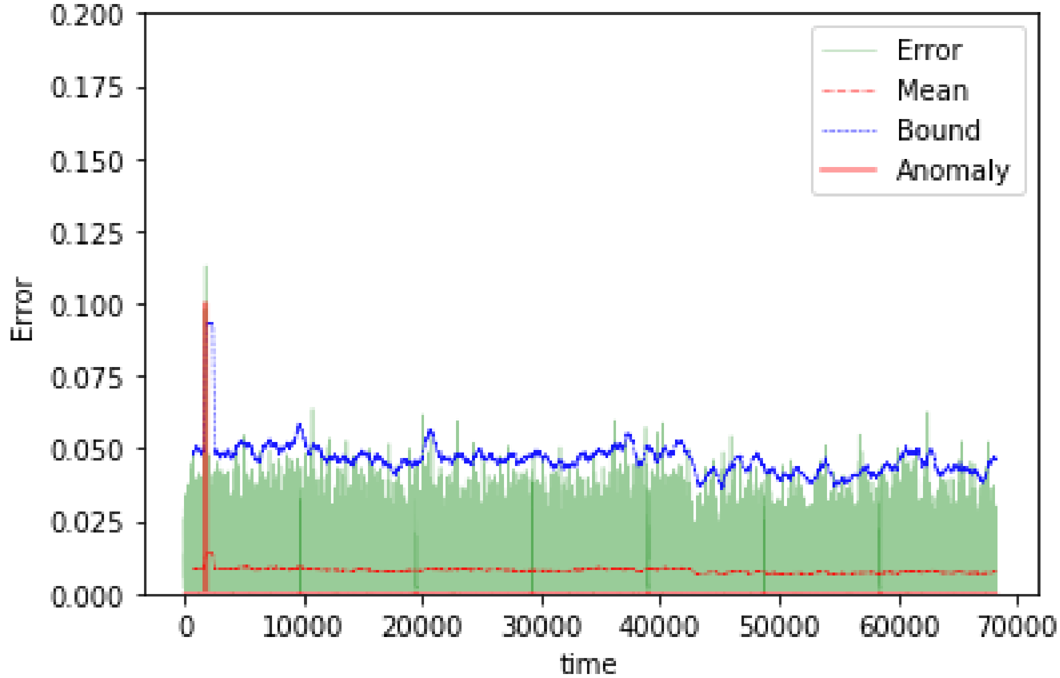


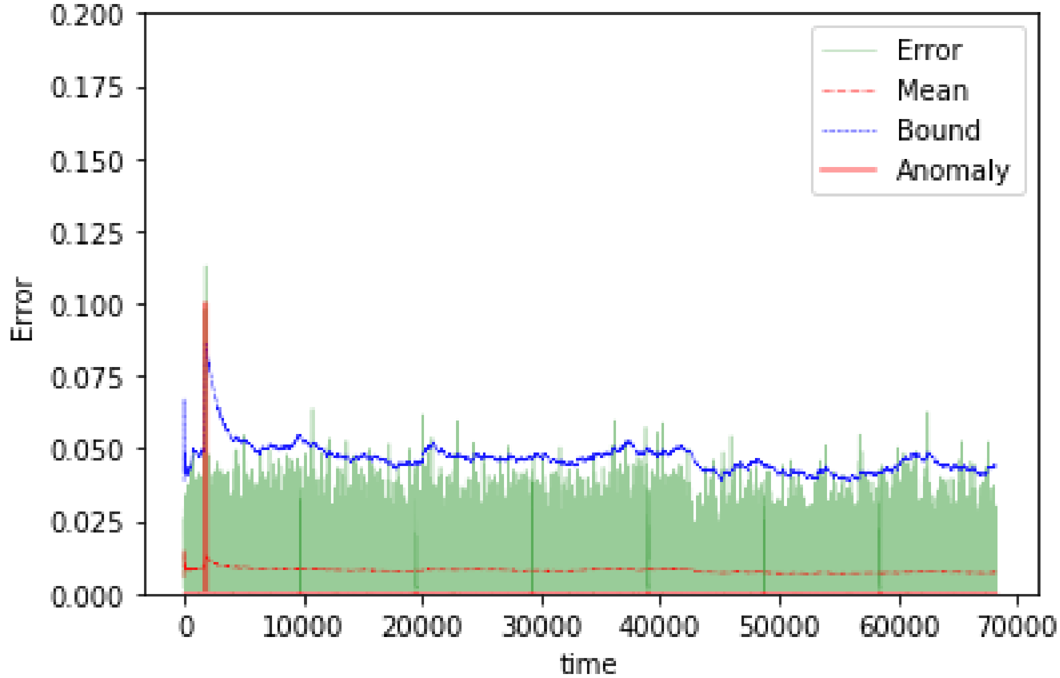





```
The mean error for gru2_20_app_change_early_ is 0.053859122054241454 for length 68279
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

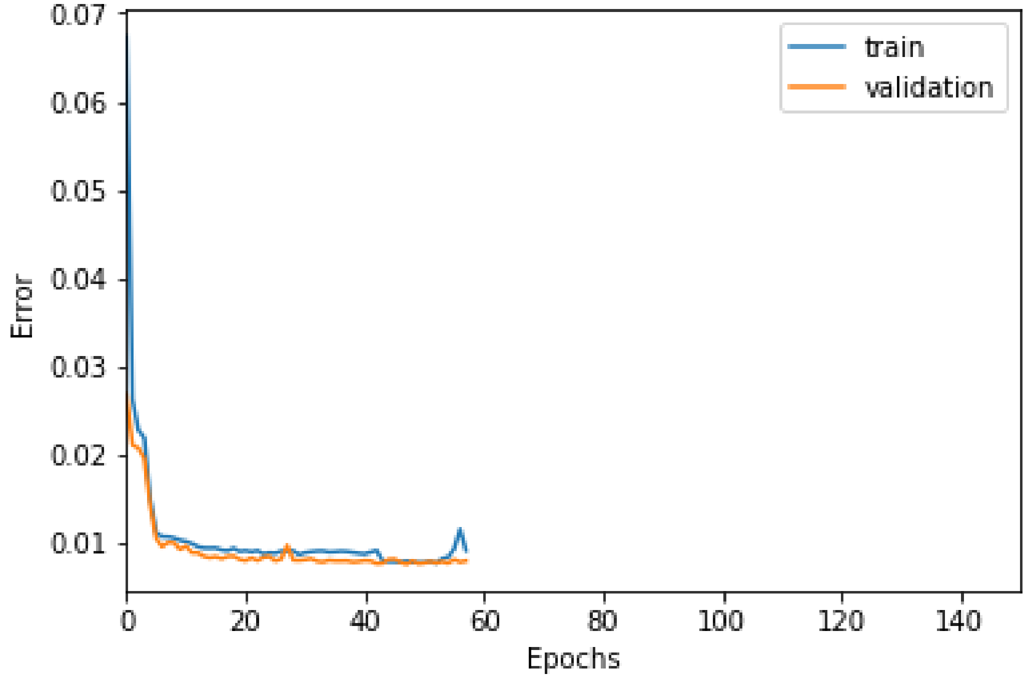

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



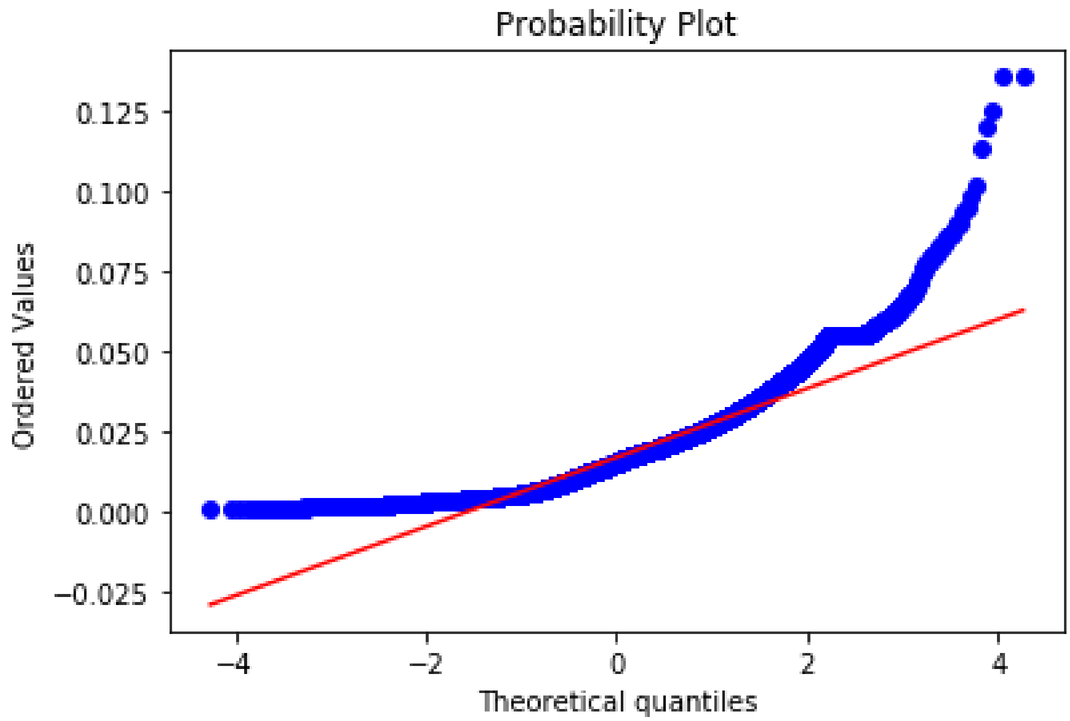

```
The mean error for gru2_20_normal_ is 0.007935573595791647 for length 68279
====================
```

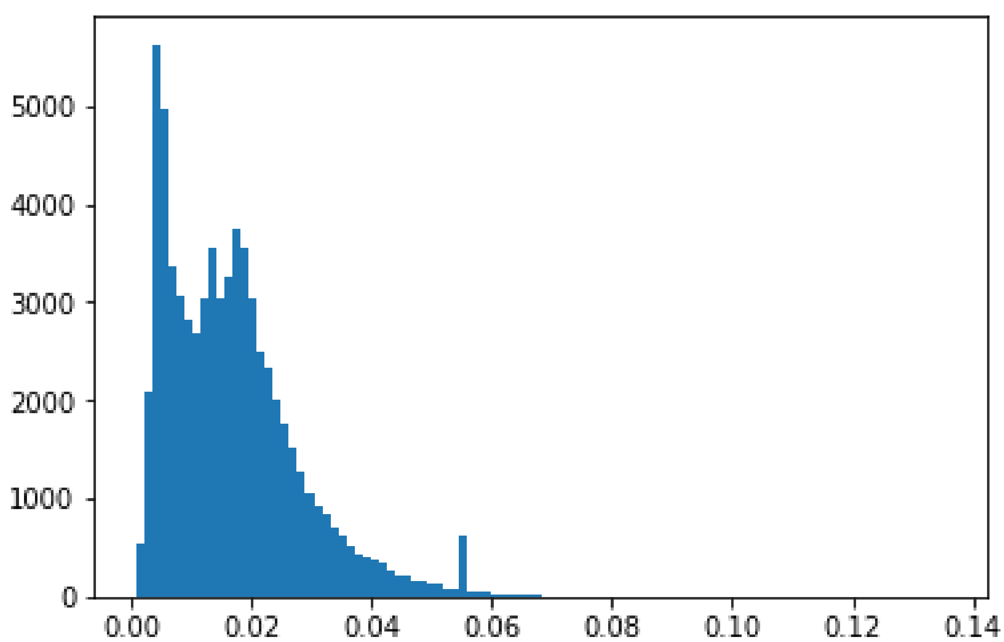
50 steps

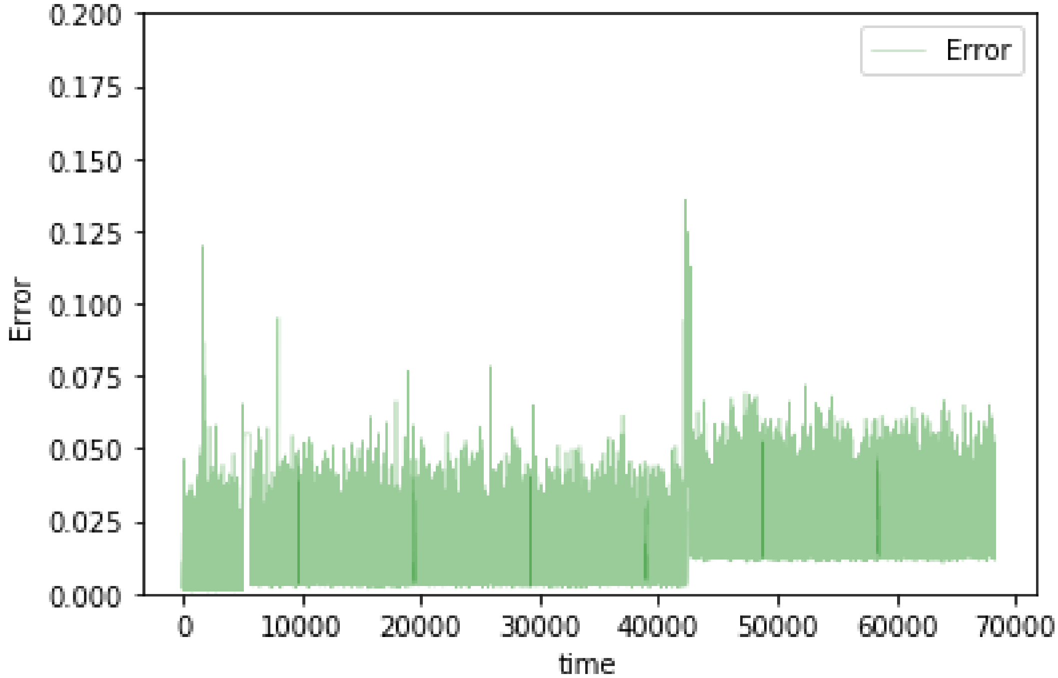
```
In [195]: TIMESTEPS = 50
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru2_50"

In [196]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(10, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [197]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [198]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

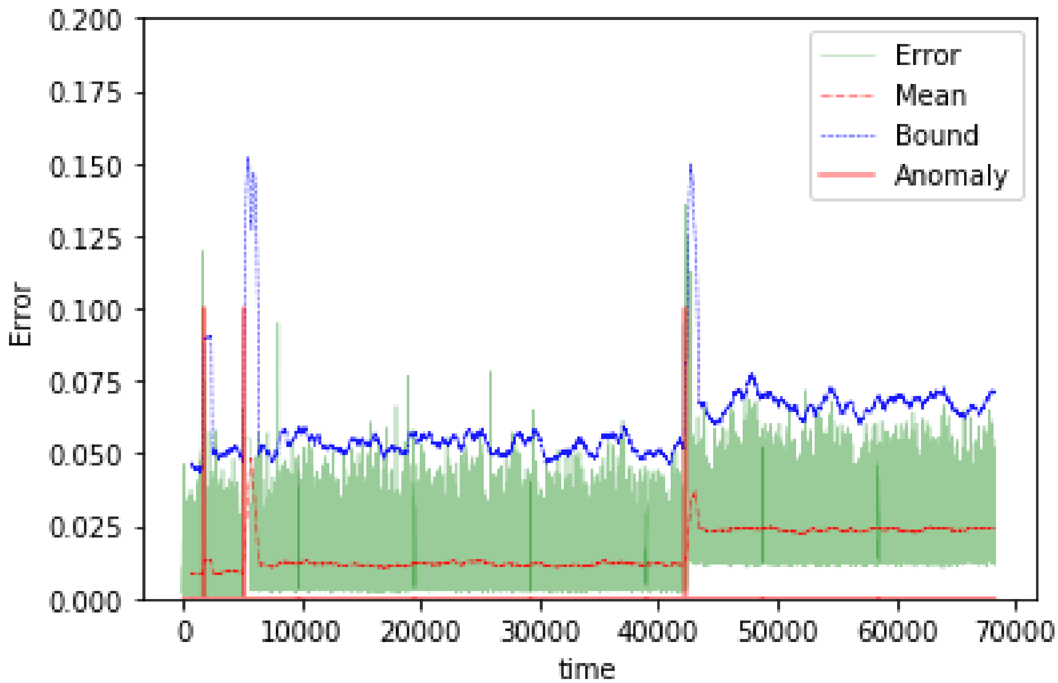


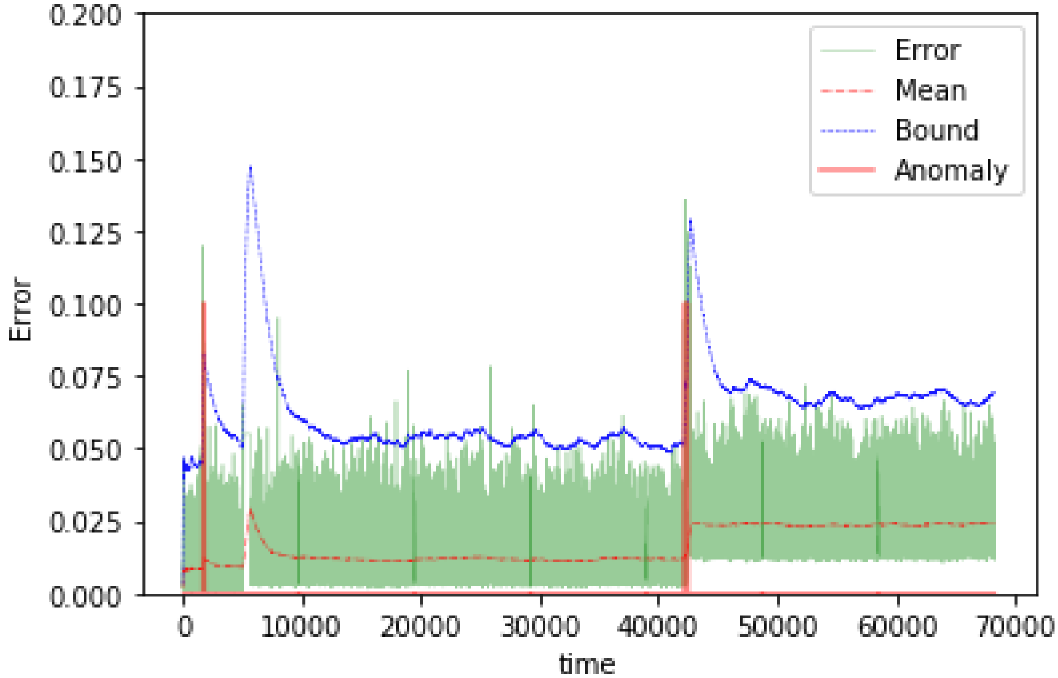

```
Training loss for final epoch is 0.009130591926863416
Validation loss for final epoch is 0.007964523024507798
---------- Beginning tests for gru2_50 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−4
−2
0
2
4

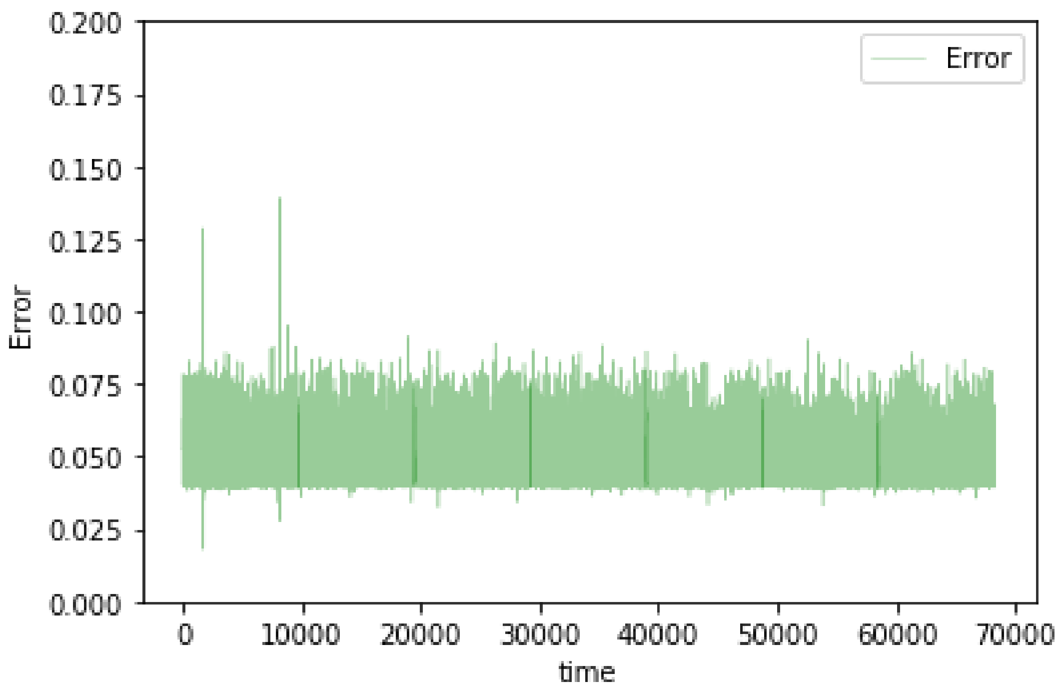

5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

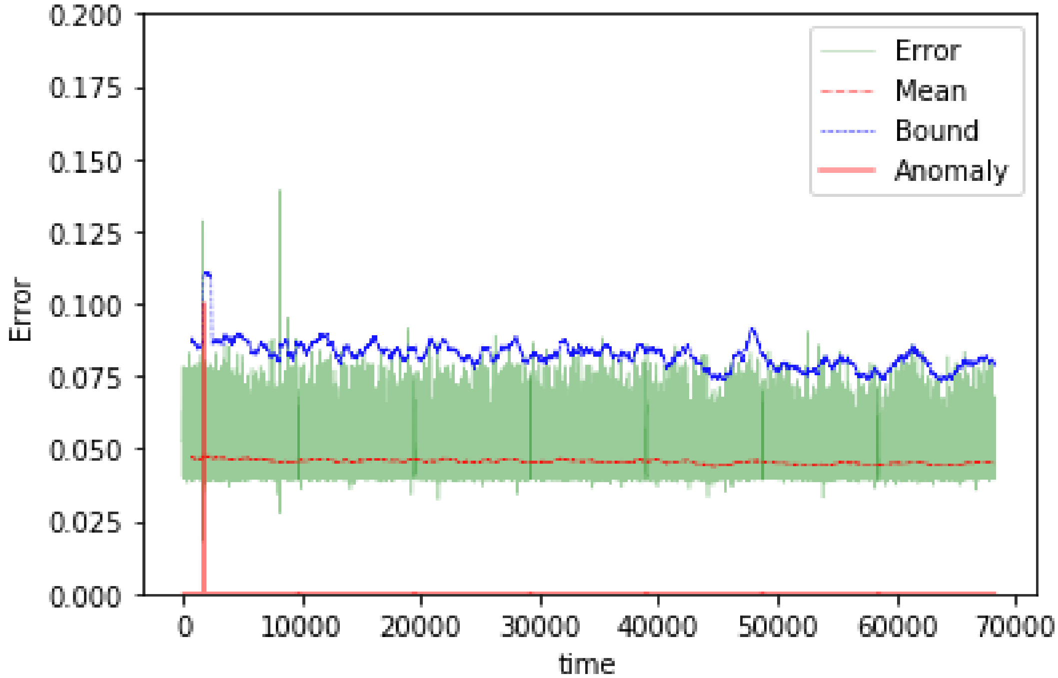

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

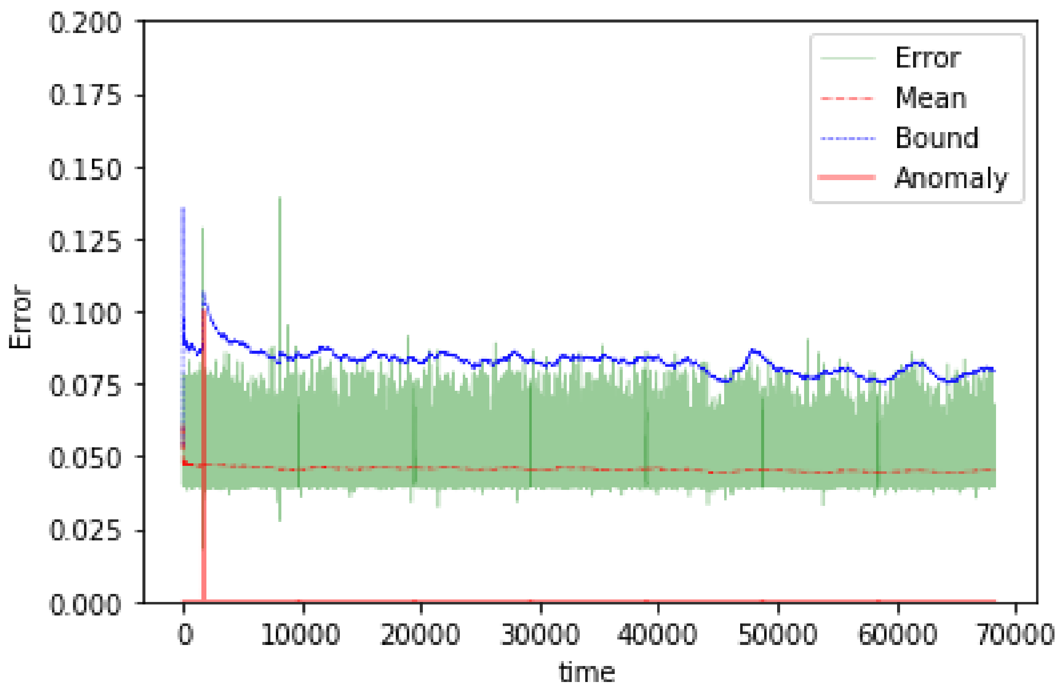

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



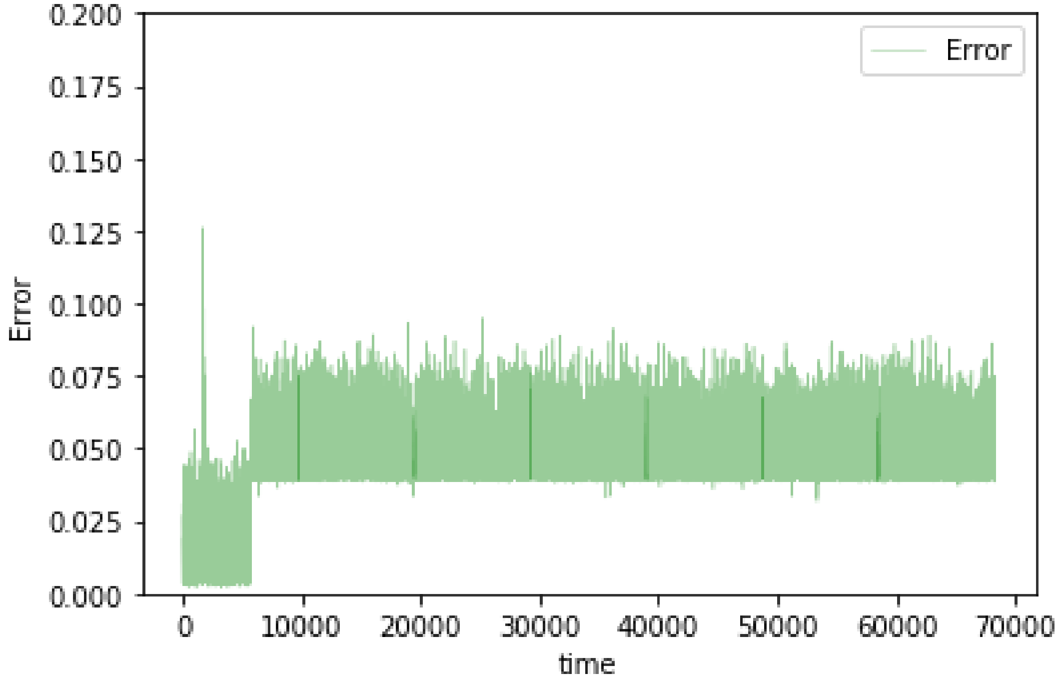

```
The mean error for gru2_50_disk_IO_start_ is 0.016716387265298484 for length 68249
Testing on Avg. load data.
```

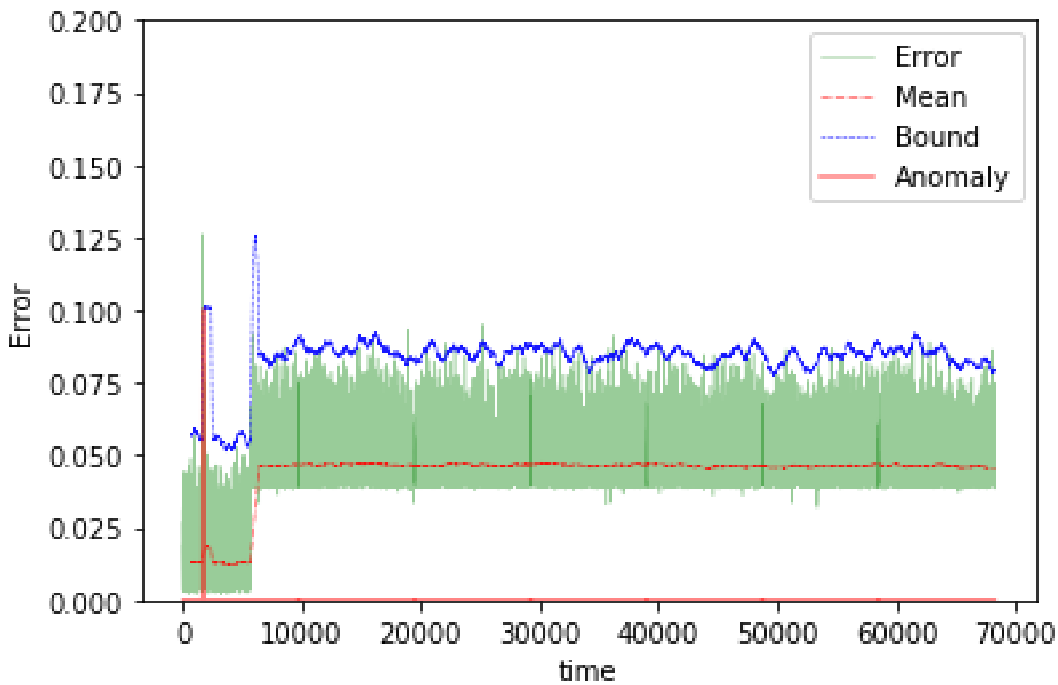


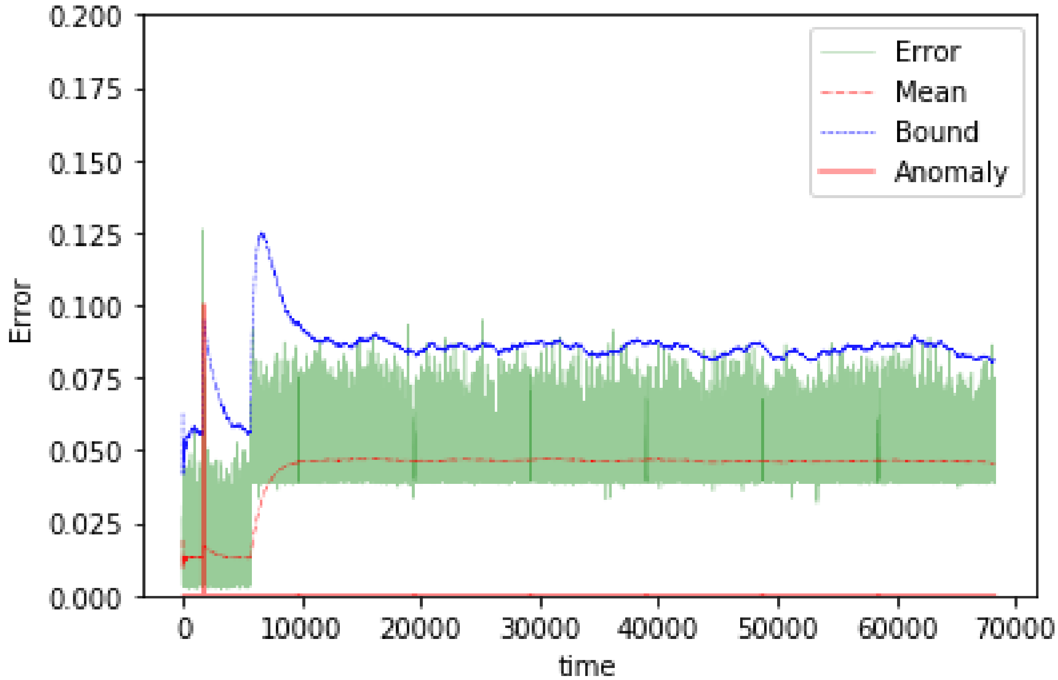

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

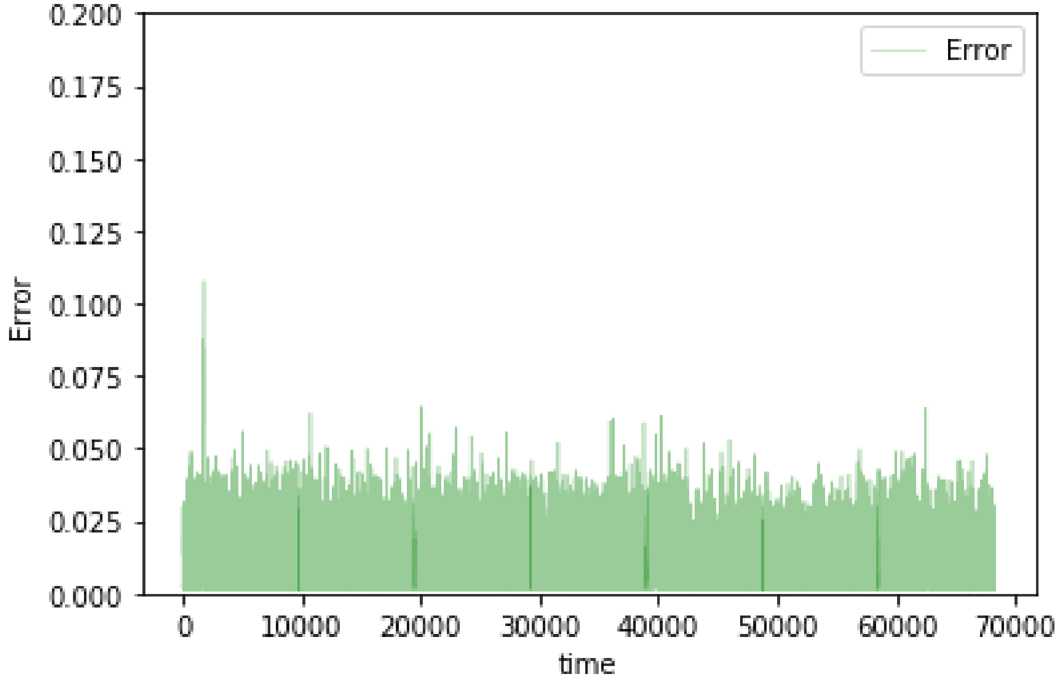
```
The mean error for gru2_50_avg_load_ is 0.045516460465489524 for length 68249
Testing on app change early data.
```

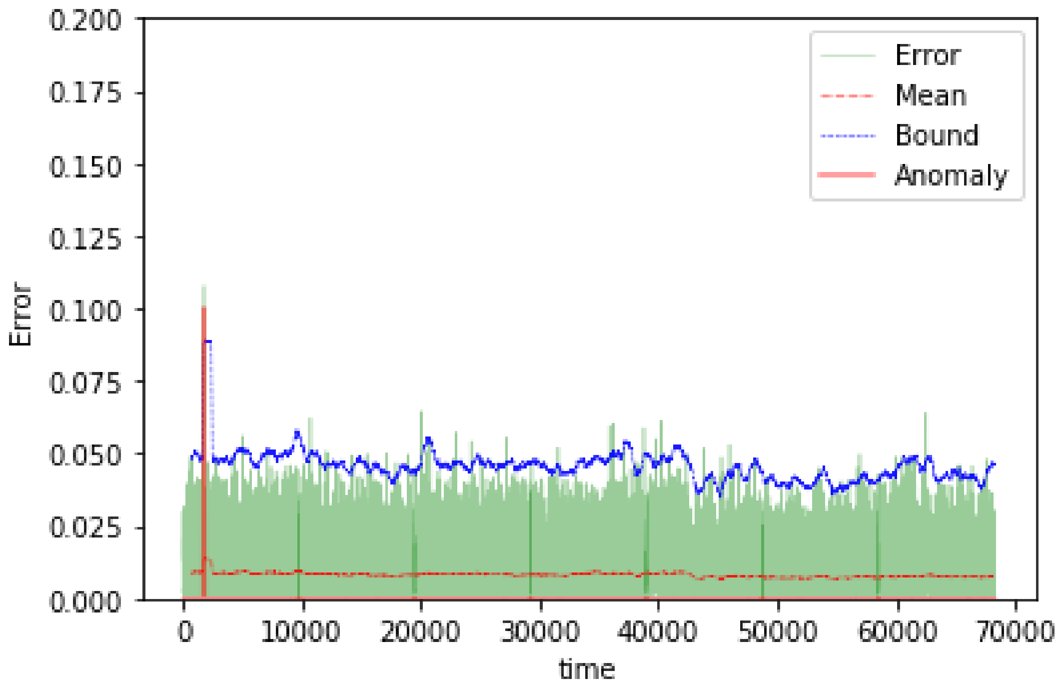


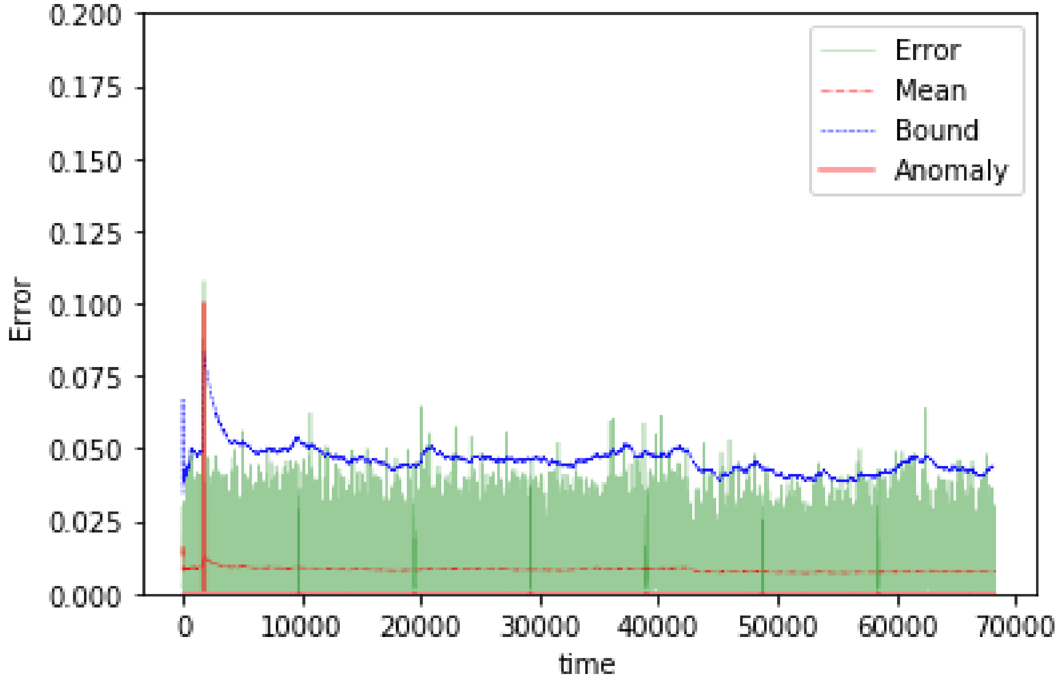





```
The mean error for gru2_50_app_change_early_ is 0.04368597515974404 for length 68249
Testing on Normal data.
```


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error

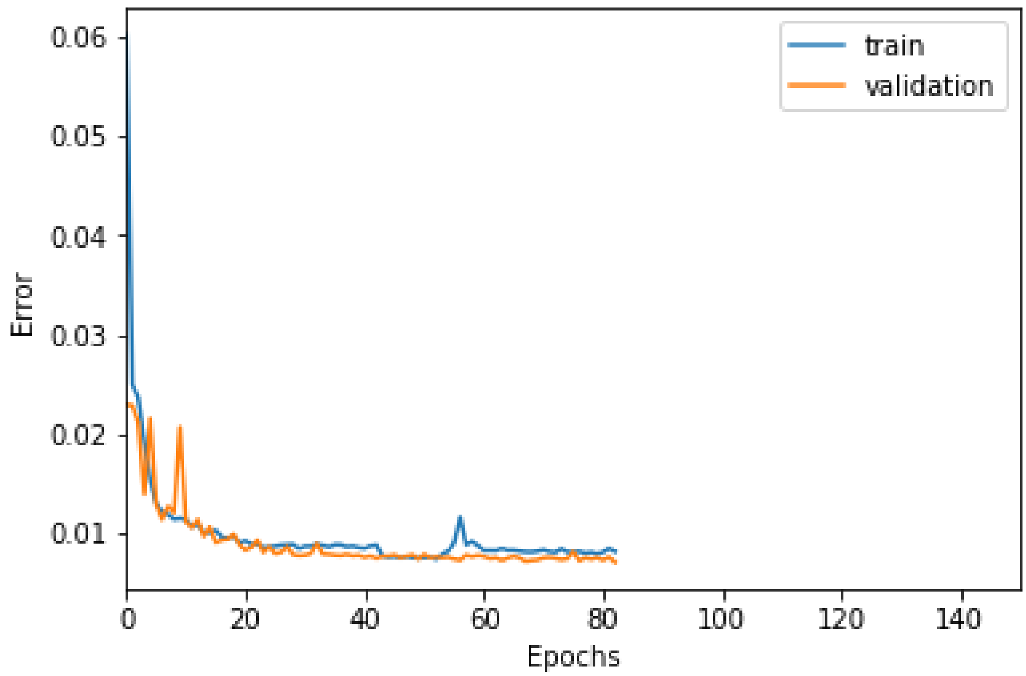

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly



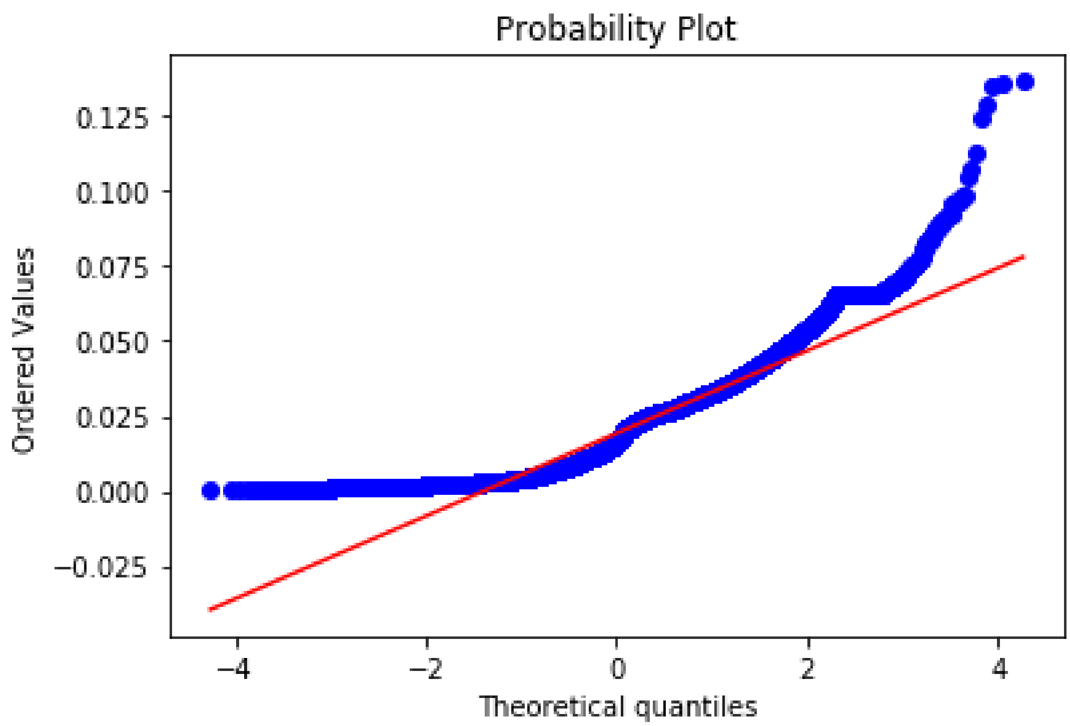

```
The mean error for gru2_50_normal_ is 0.008164221759625143 for length 68249
====================
```

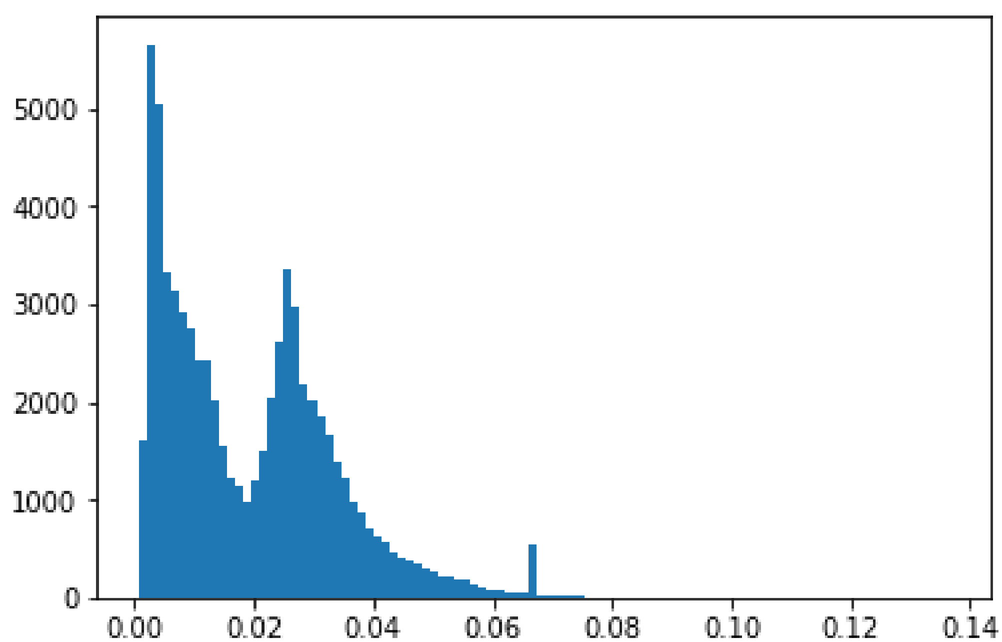
100 steps

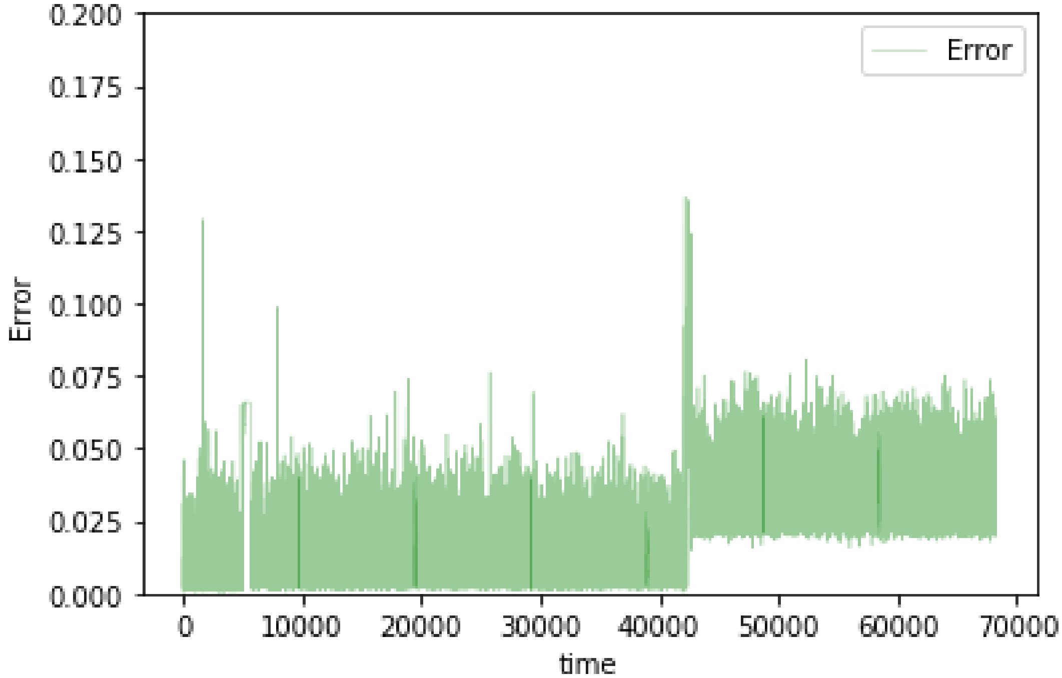
```
In [199]: TIMESTEPS = 100
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru2_100"

In [200]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(10, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [201]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [202]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

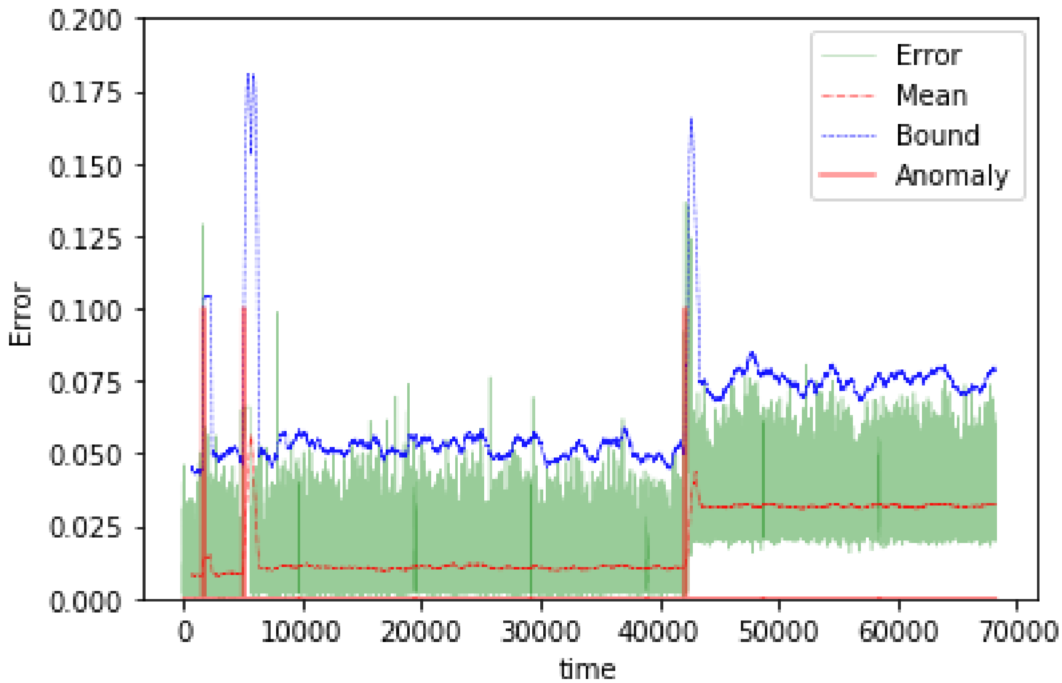


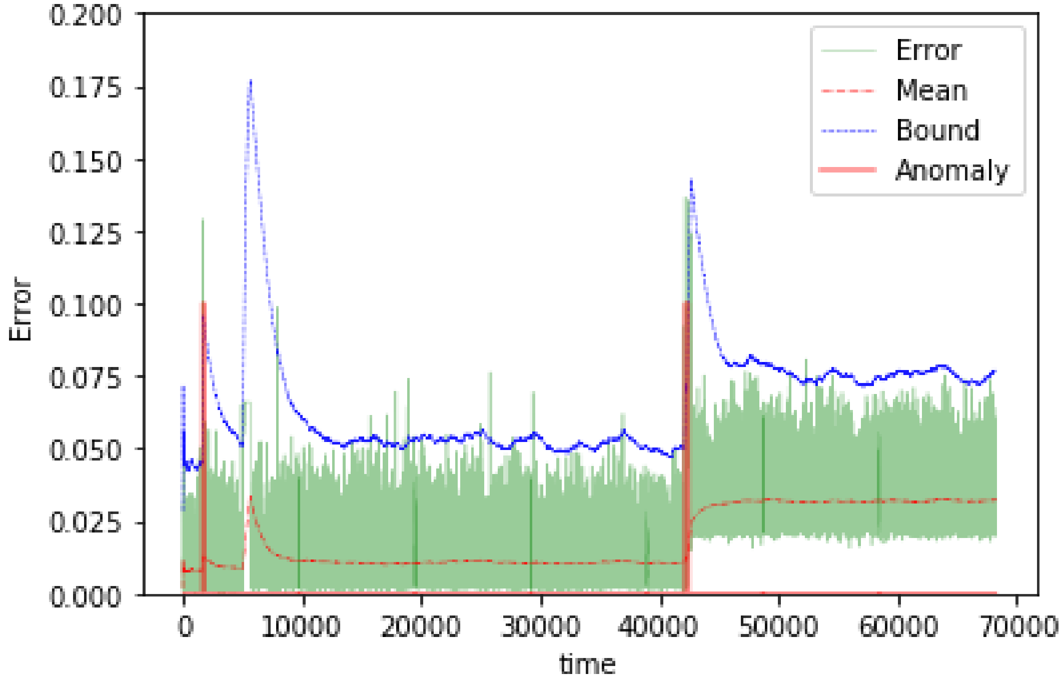

```
Training loss for final epoch is 0.008192111559328624
Validation loss for final epoch is 0.007102832564502023
---------- Beginning tests for gru2_100 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−4
−2
0
2
4

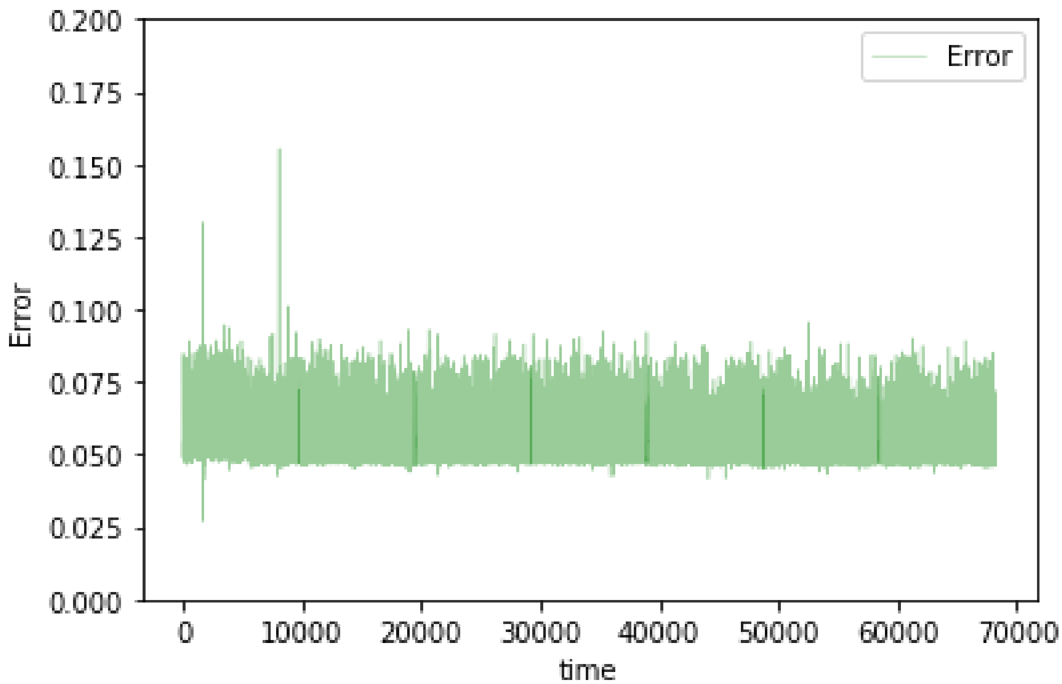

5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

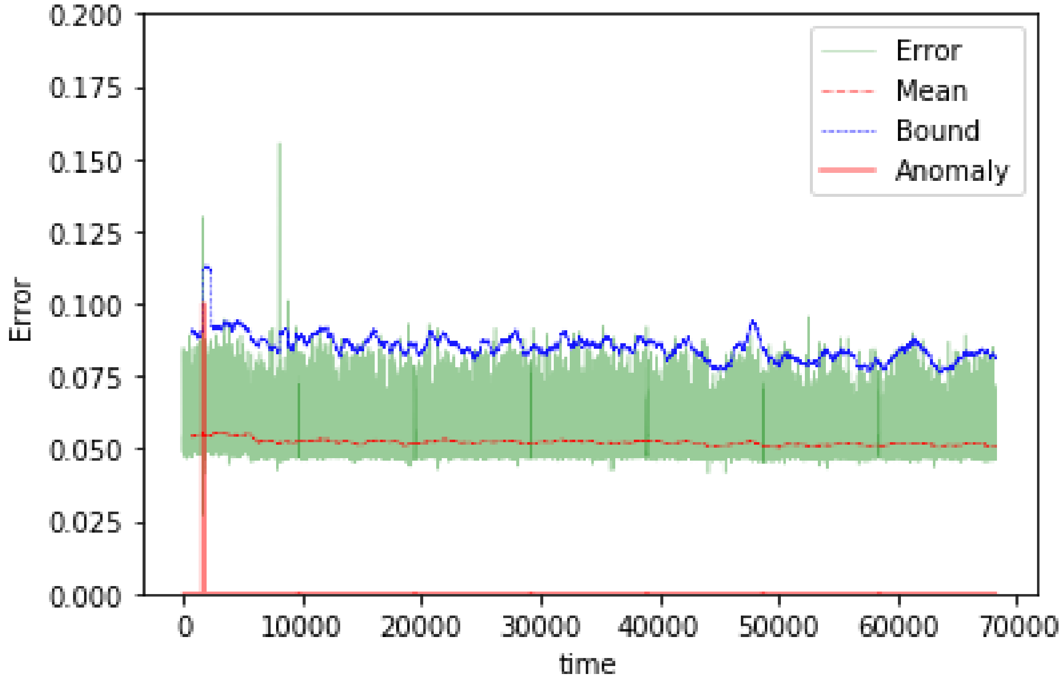

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

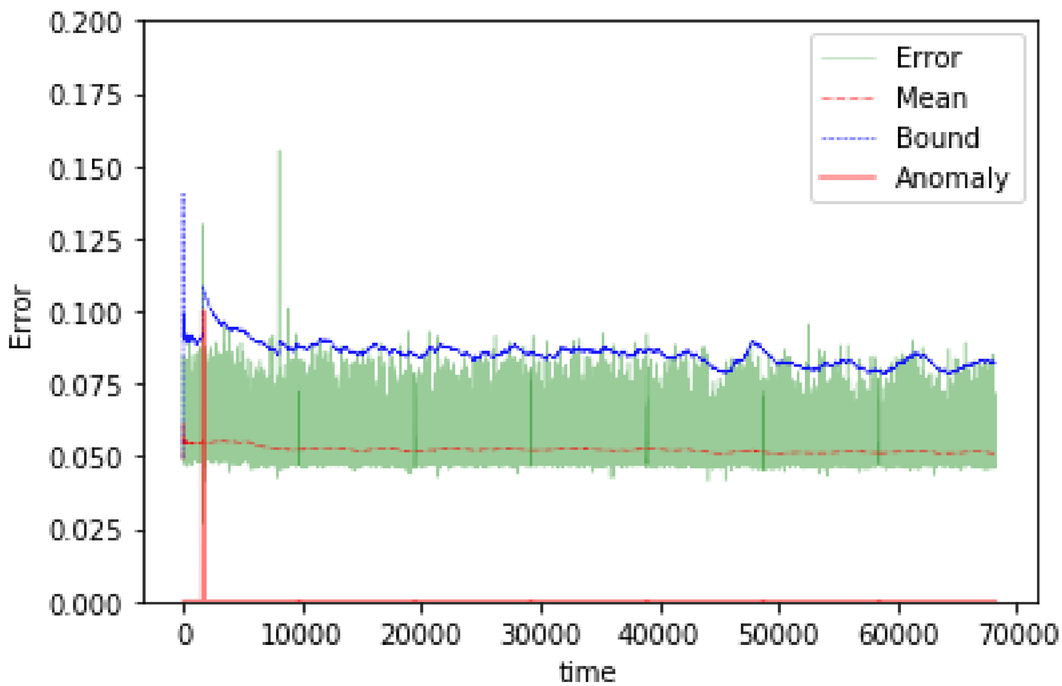

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



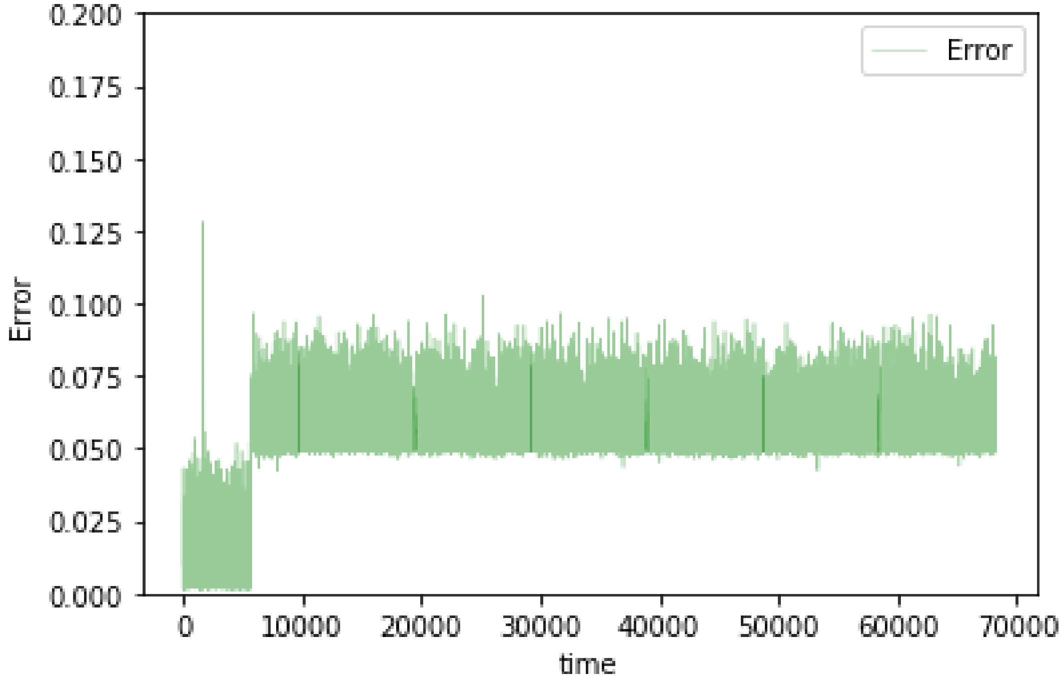

```
The mean error for gru2_100_disk_IO_start_ is 0.0192512137736545 for length 68199
Testing on Avg. load data.
```

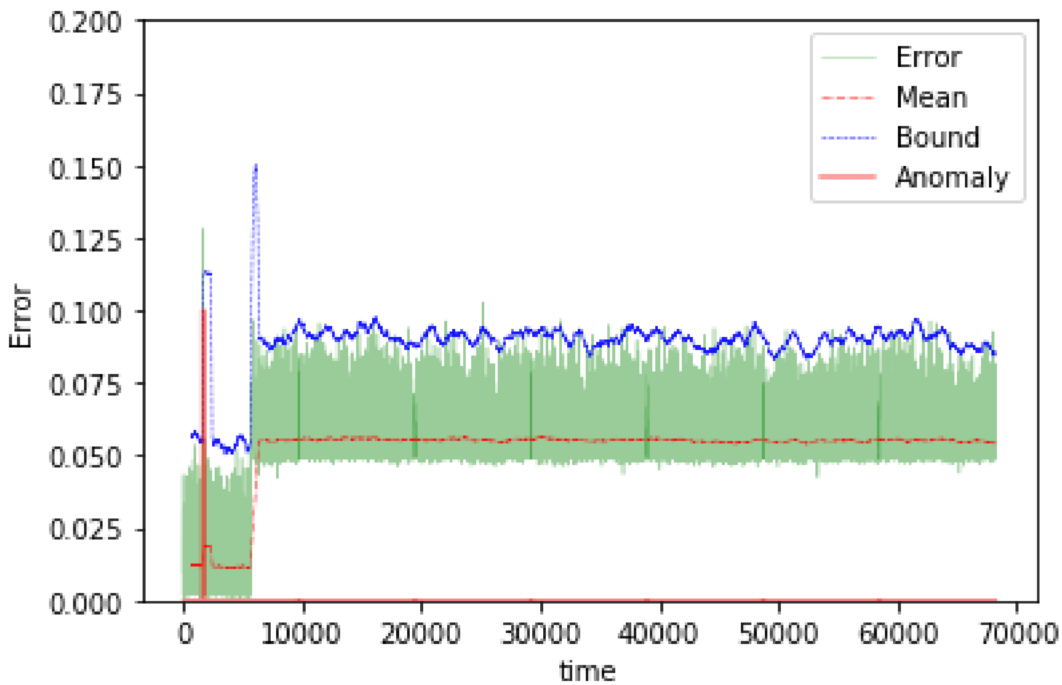


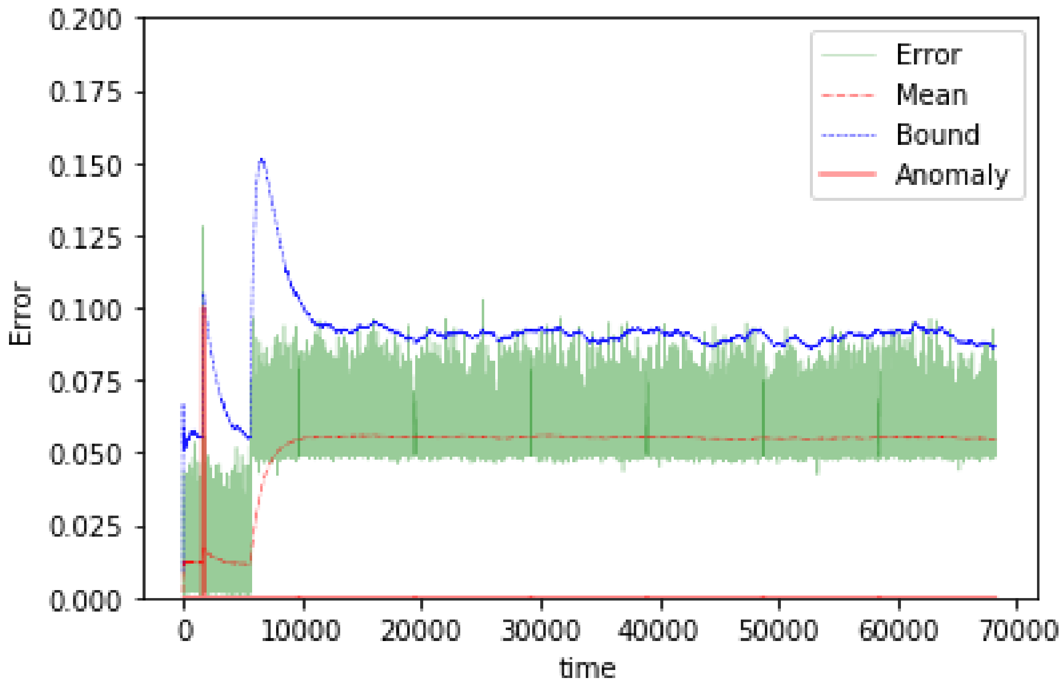

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

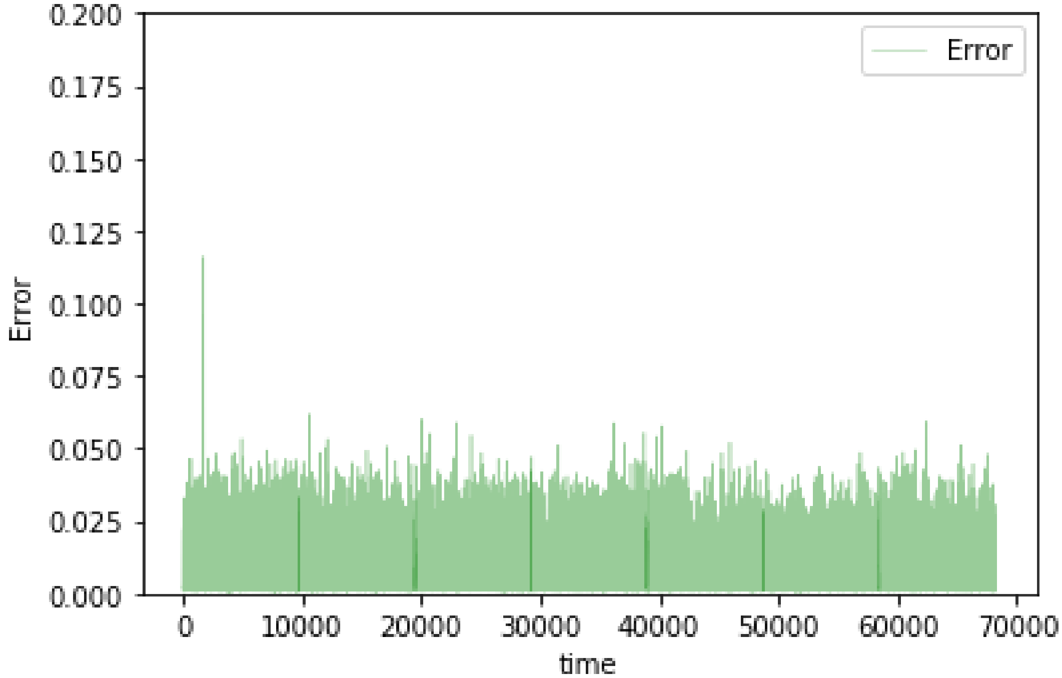
```
The mean error for gru2_100_avg_load_ is 0.0522086721614817 for length 68199
Testing on app change early data.
```

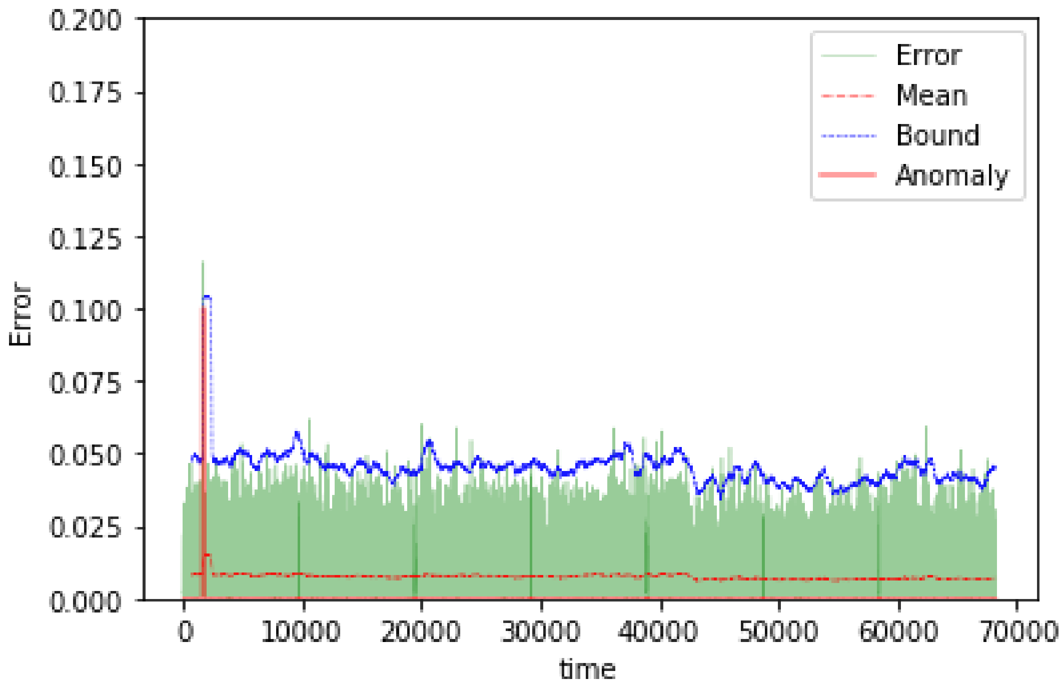


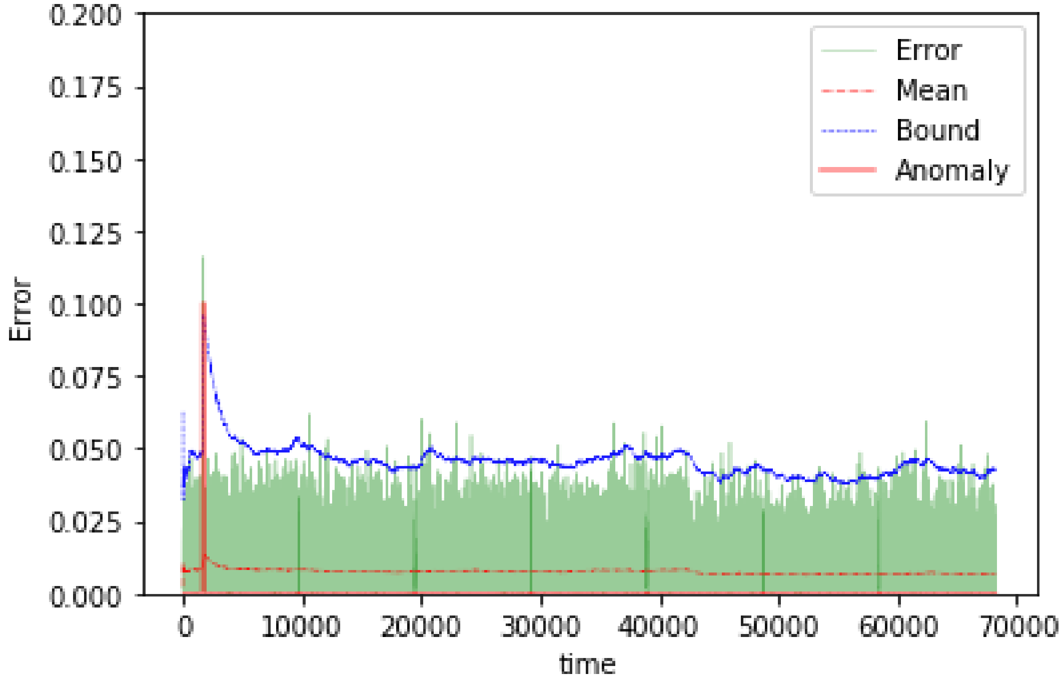





```
The mean error for gru2_100_app_change_early_ is 0.051718380438013294 for length 68199
Testing on Normal data.
```


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error

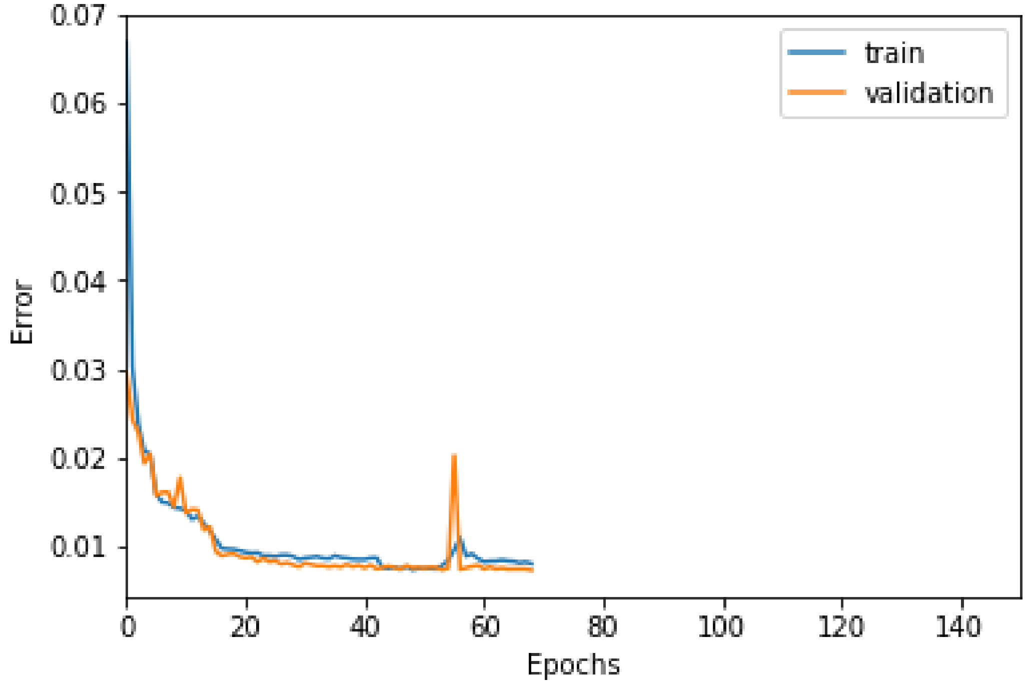

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly



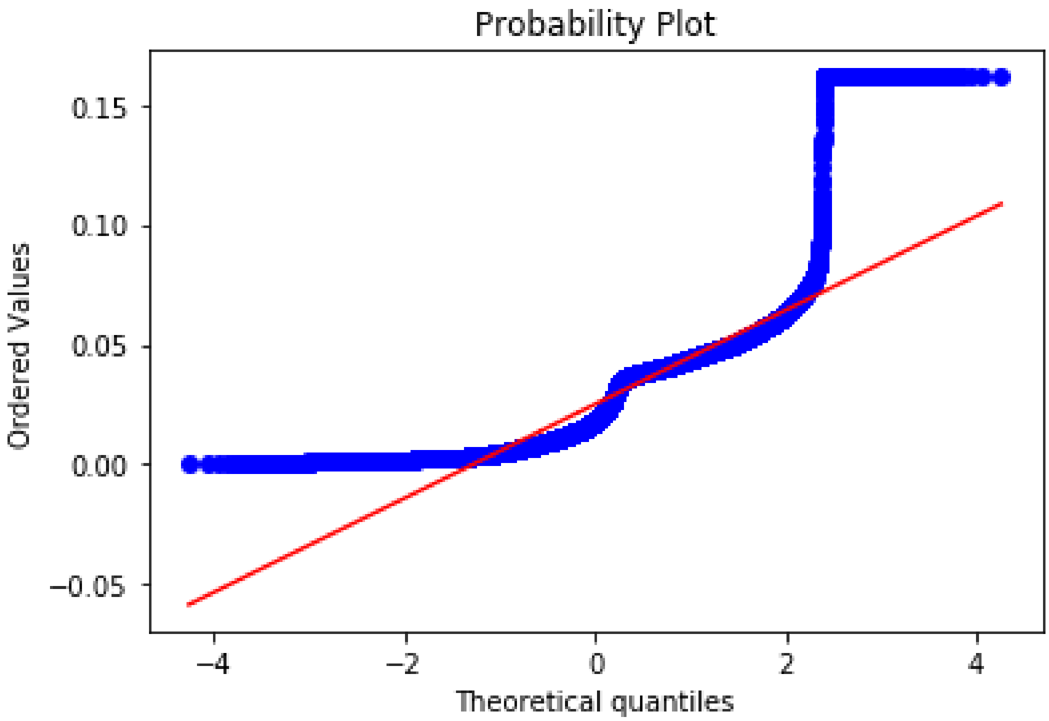

```
The mean error for gru2_100_normal_ is 0.007519294945700586 for length 68199
====================
```

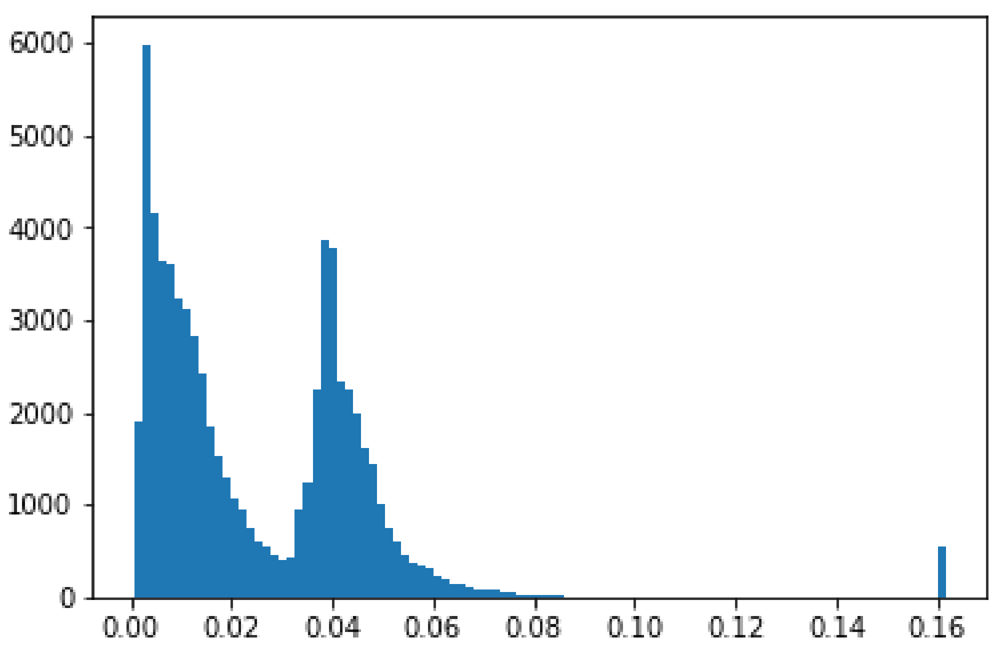
200 steps

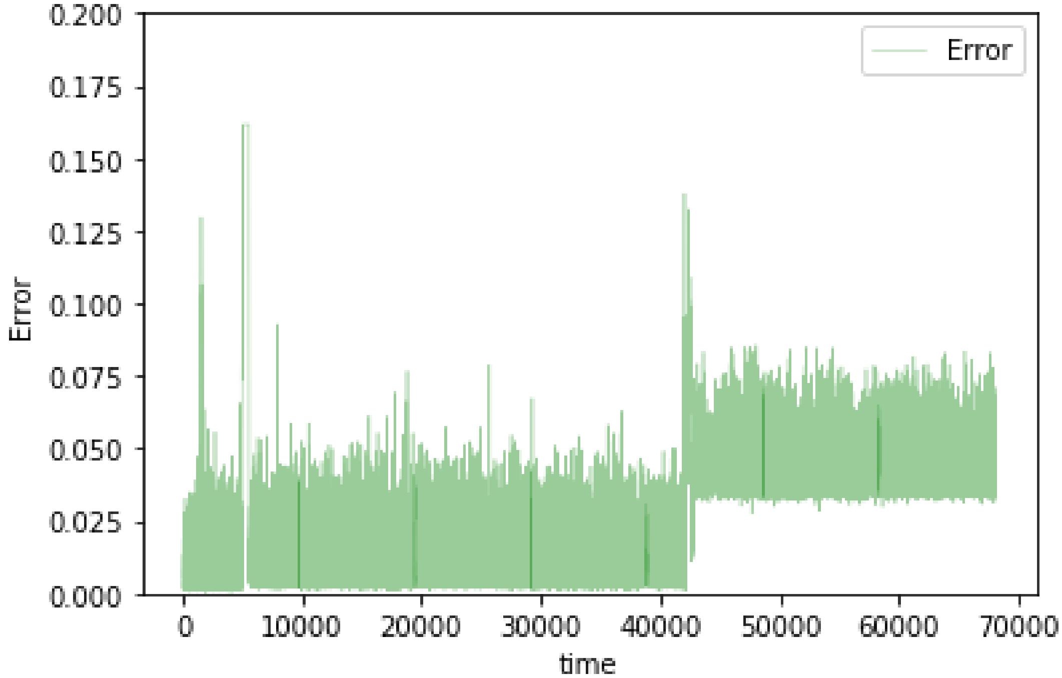
```
In [203]: TIMESTEPS = 200
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru2_200"

In [204]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(10, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [205]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [206]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

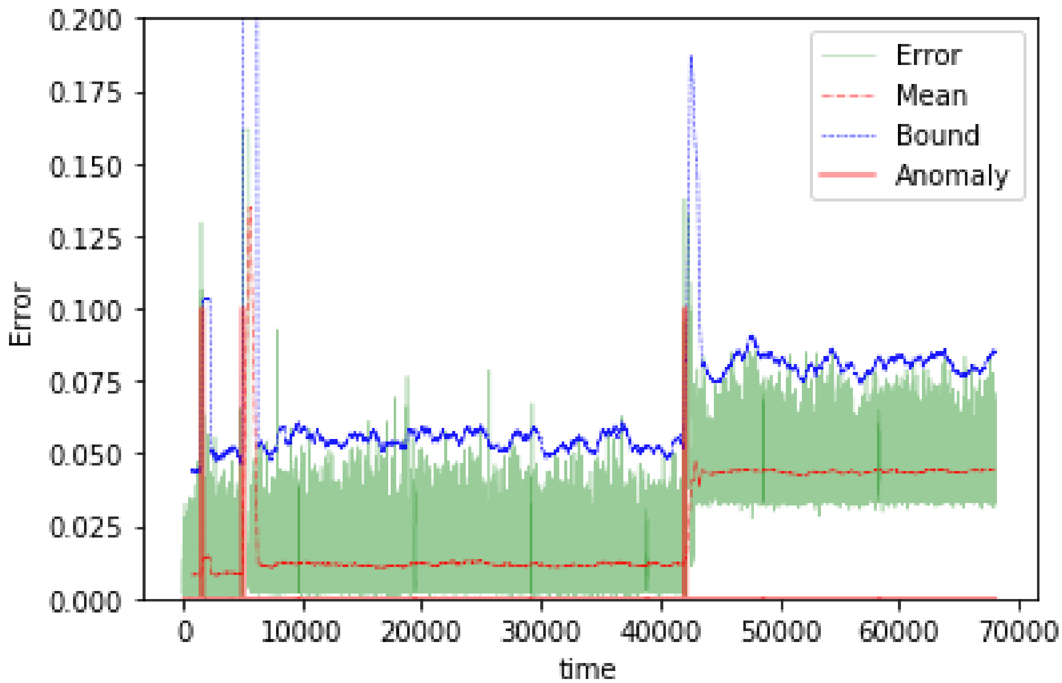


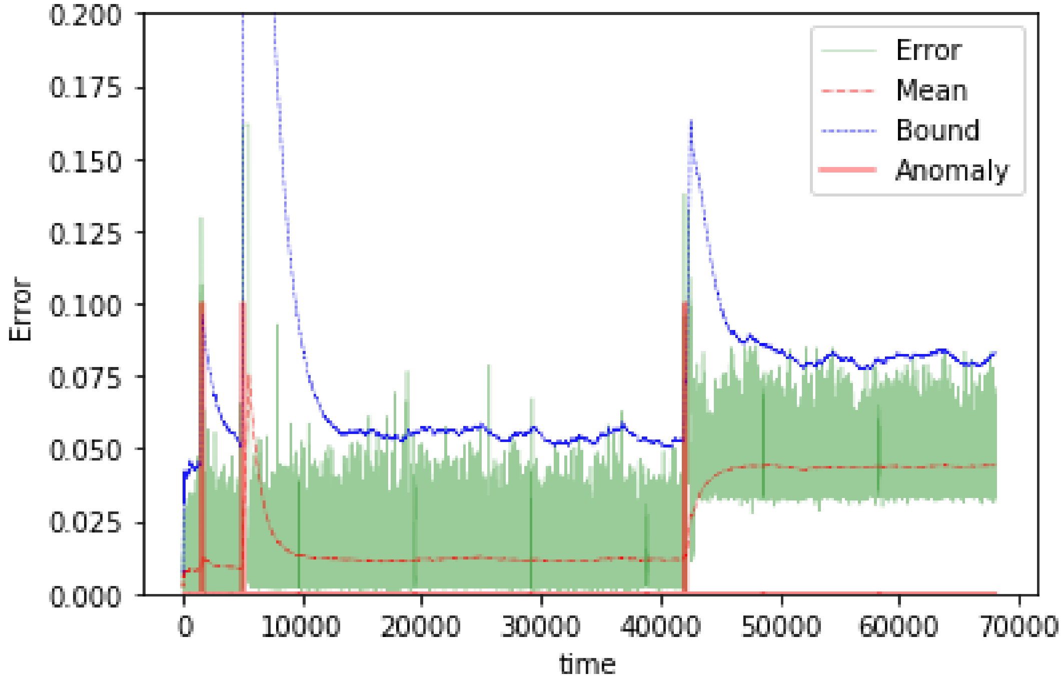

```
Training loss for final epoch is 0.00800835409516003
Validation loss for final epoch is 0.007270382409566082
---------- Beginning tests for gru2_200 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.15
0.10
0.05
0.00
−0.05
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

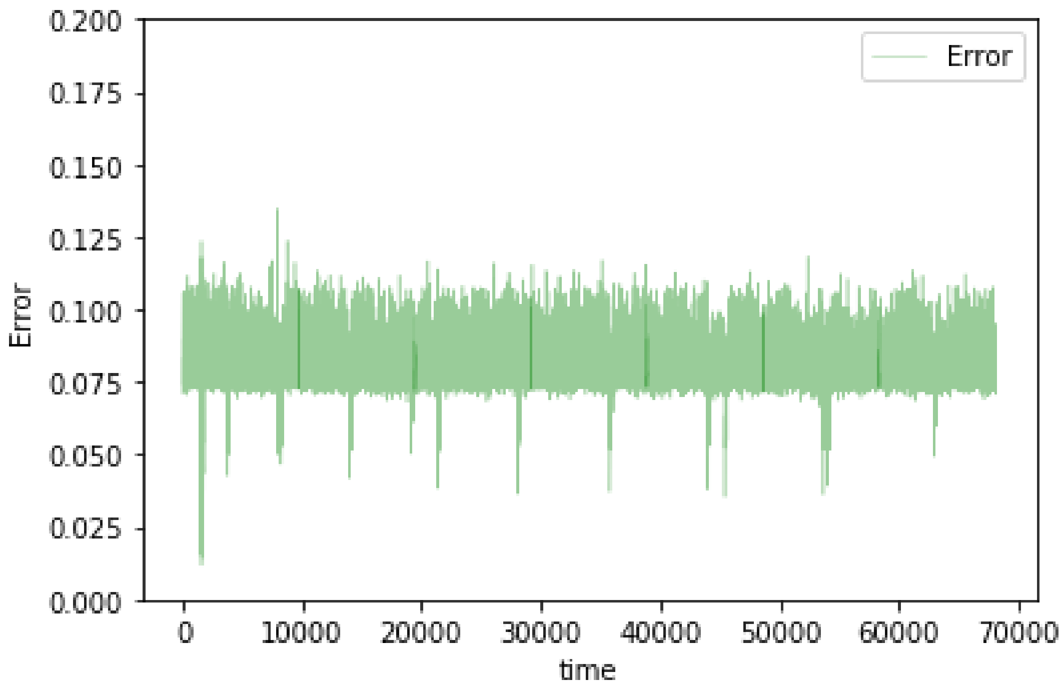

6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16

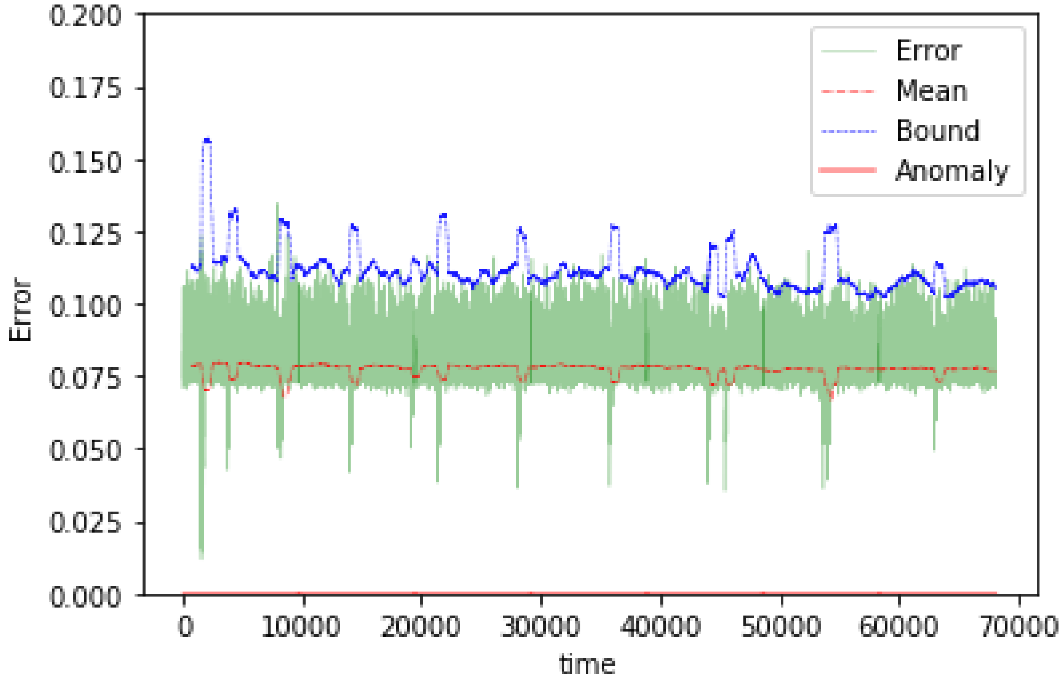

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

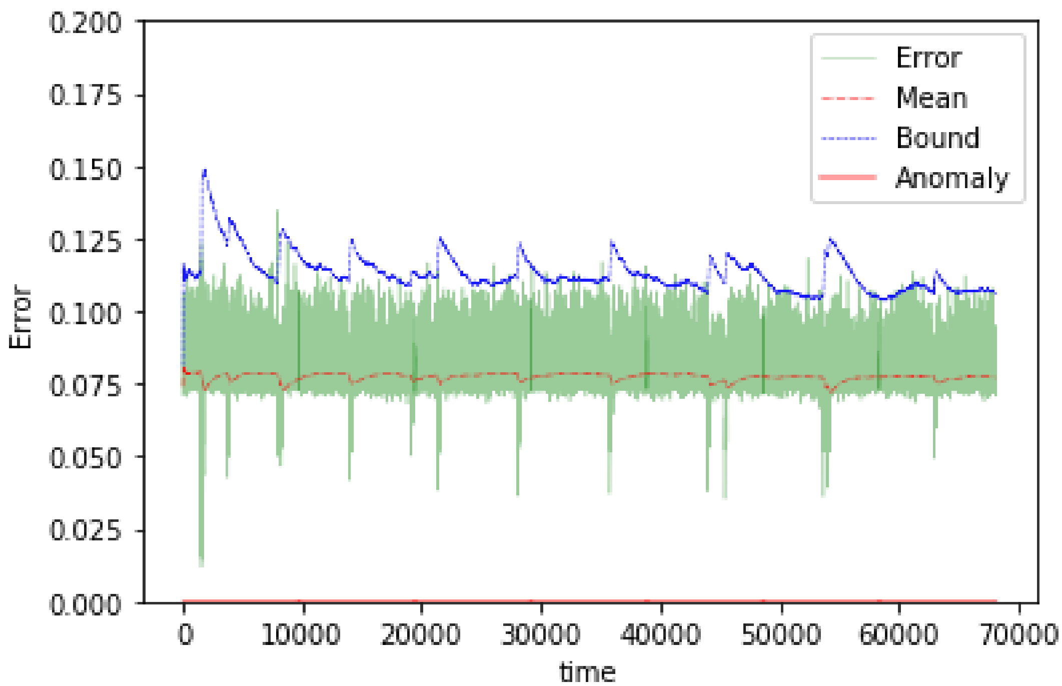

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



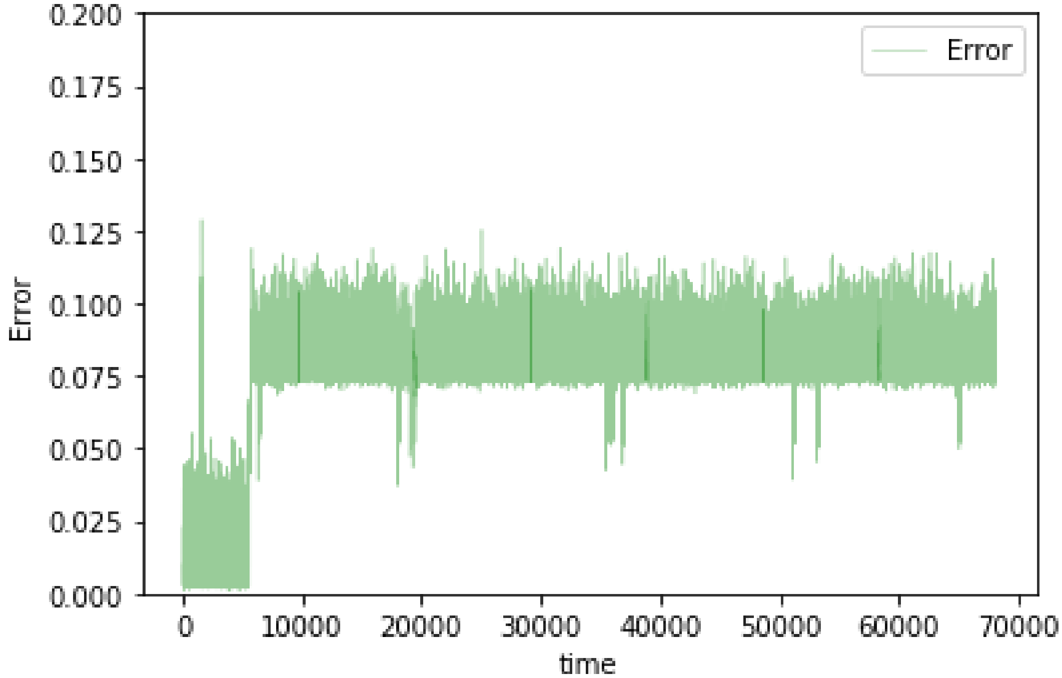

```
The mean error for gru2_200_disk_IO_start_ is 0.02507422095506679 for length 68099
Testing on Avg. load data.
```

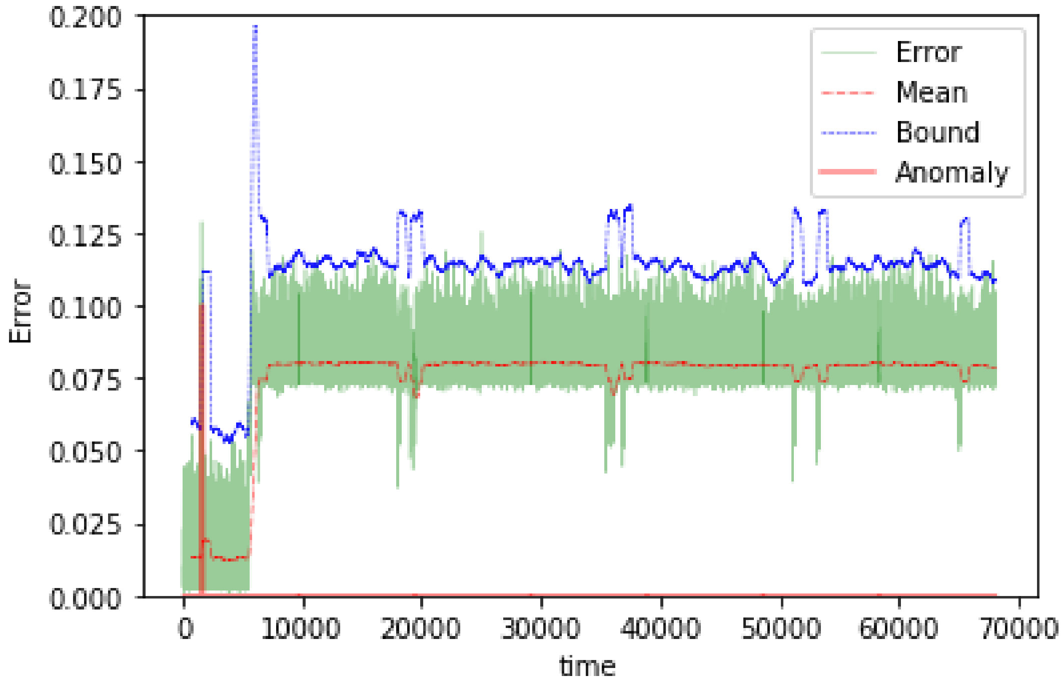


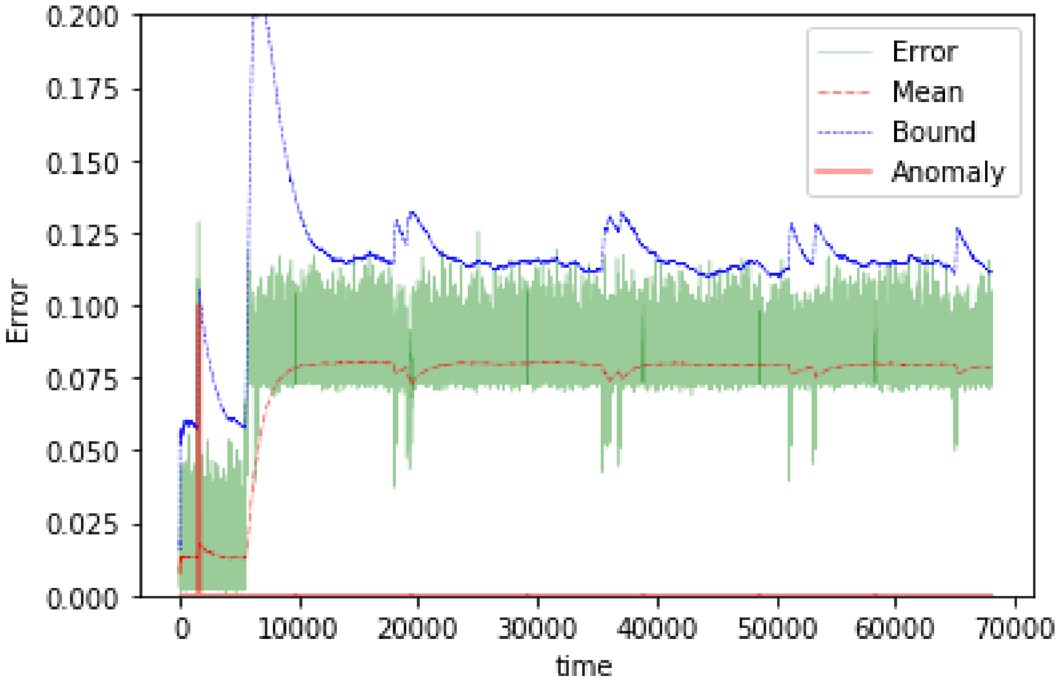

Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000


Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

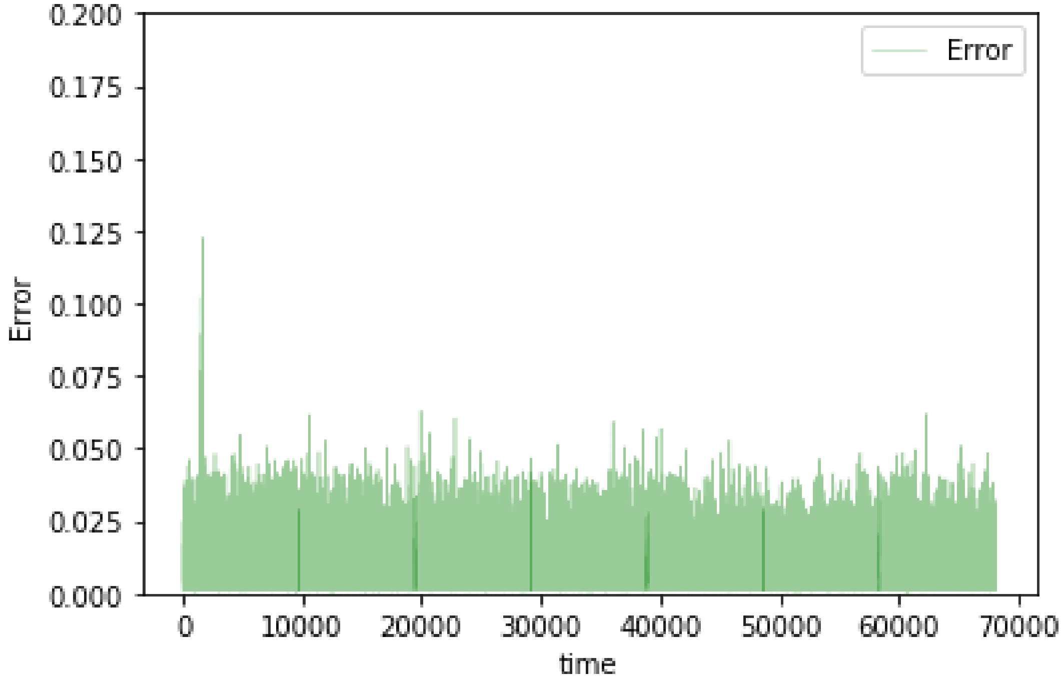
```
The mean error for gru2_200_avg_load_ is 0.07730145500721493 for length 68099
Testing on app change early data.
```

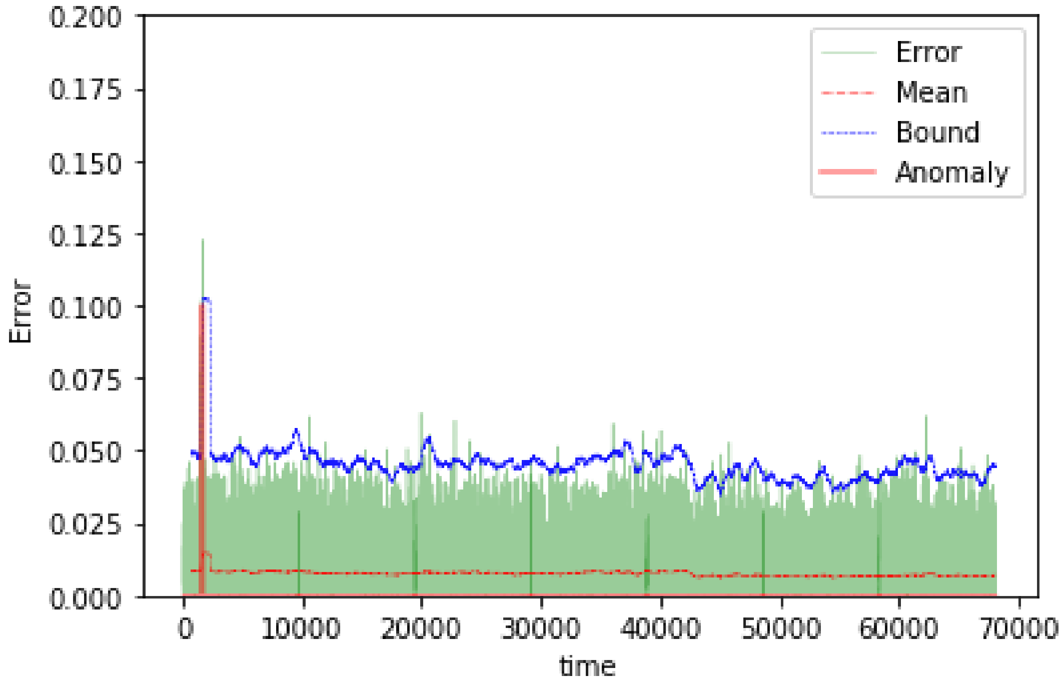


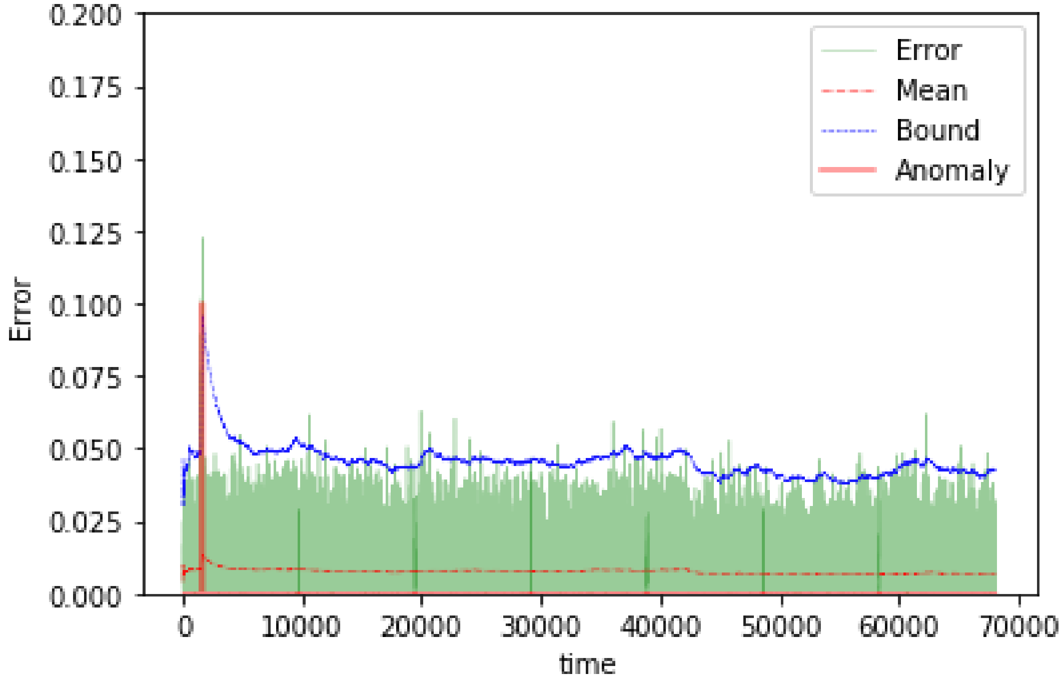





```
The mean error for gru2_200_app_change_early_ is 0.07364077019867381 for length 68099
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error




```
The mean error for gru2_200_normal_ is 0.007551722548531012 for length 68099
====================
```

1.11.7 RNN with 3 GRU layers

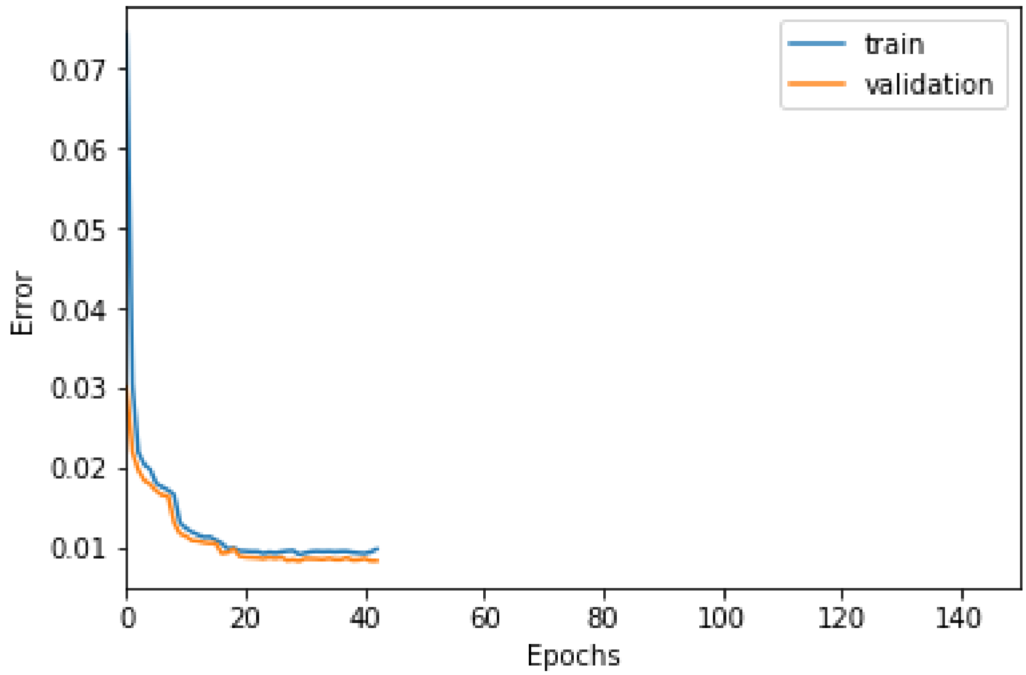
2 steps

```
In [207]: TIMESTEPS = 2
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru3_2"
```

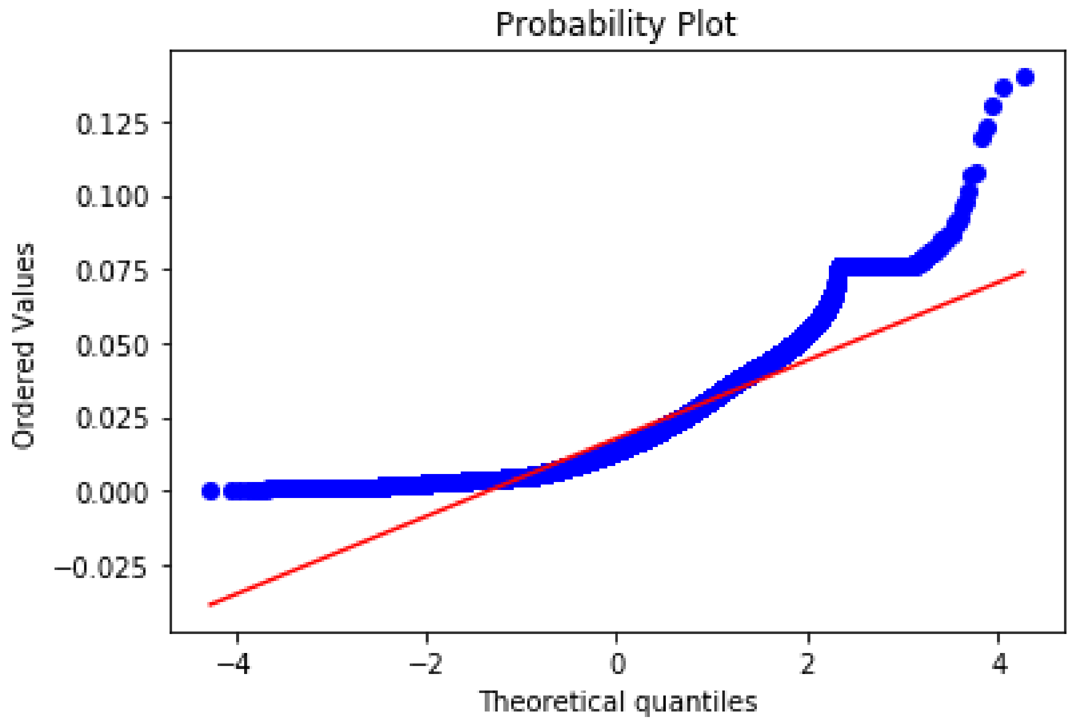
```
In [208]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(10, activation='relu', return_sequences=True)(hidden)
          hidden = GRU(10, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)
```

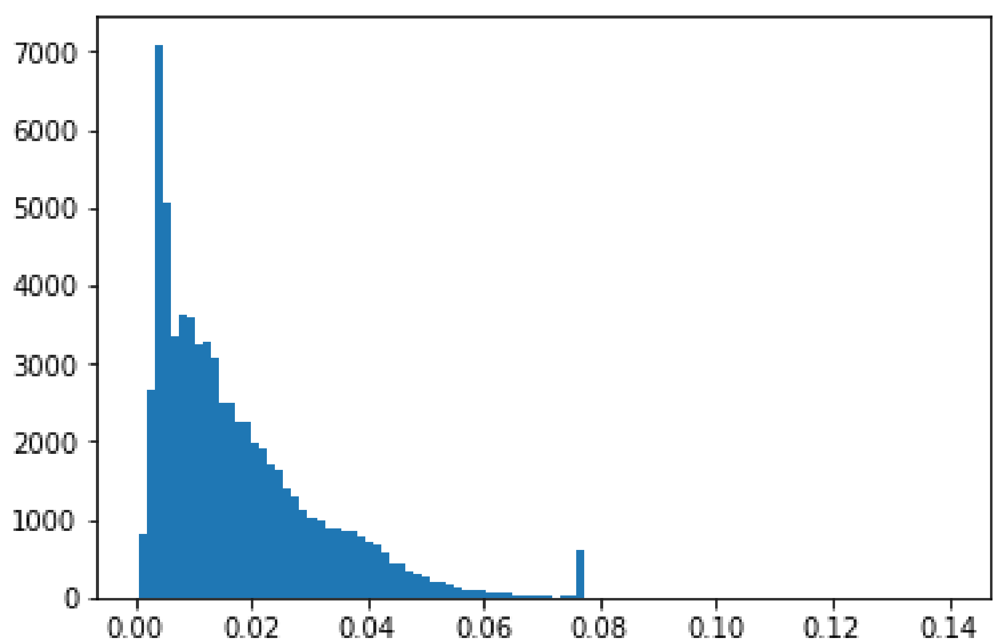
```
In [209]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])
```

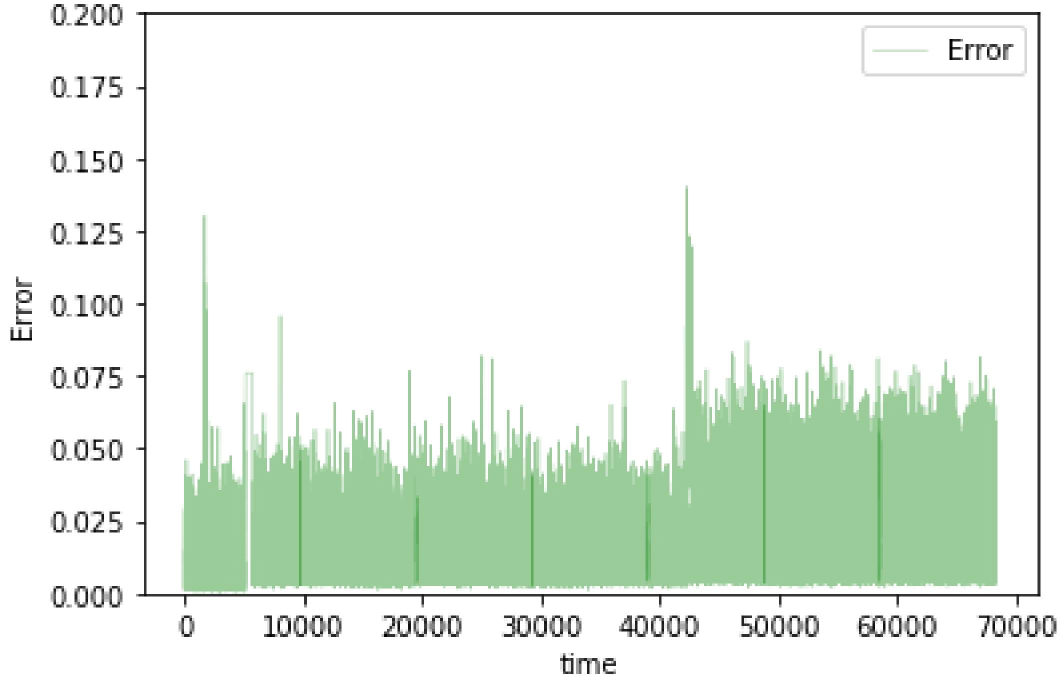
```
In [210]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

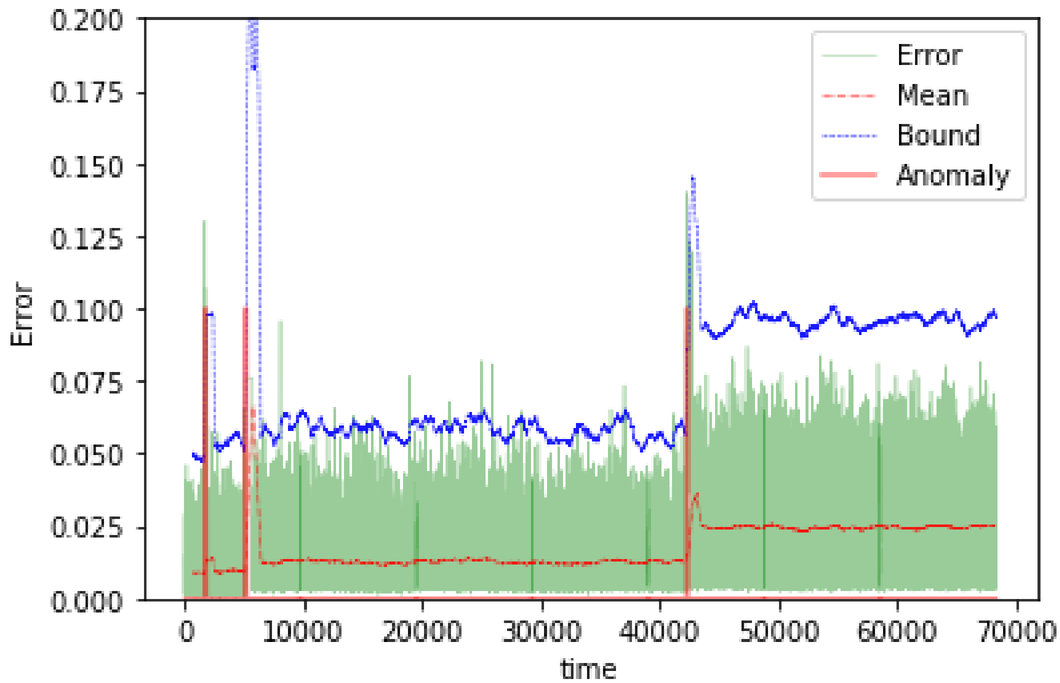


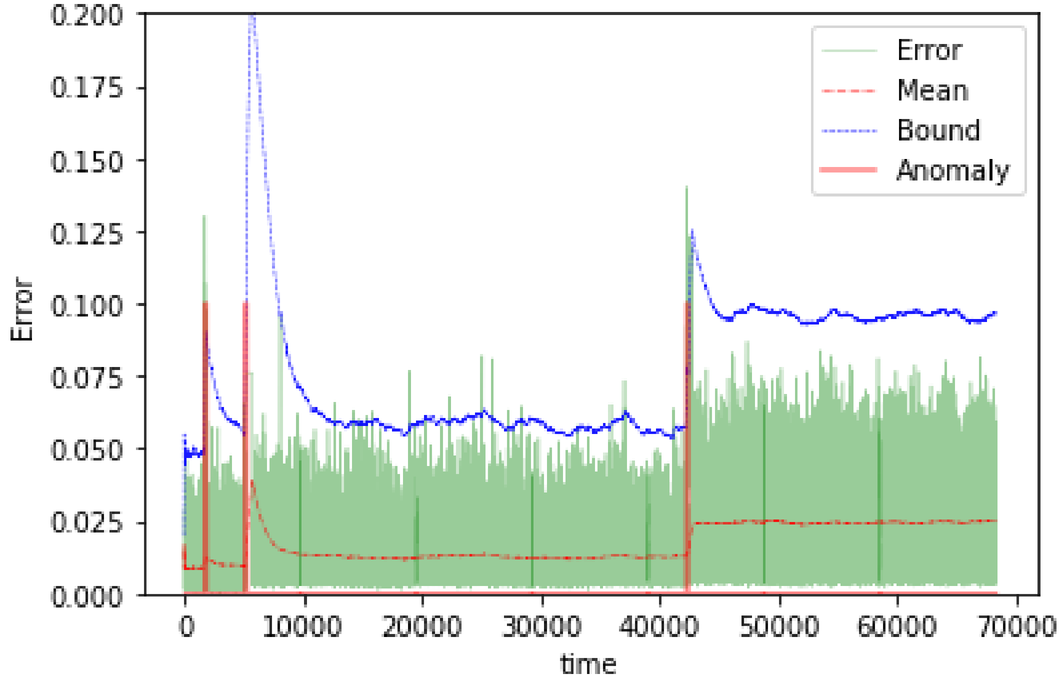

```
Training loss for final epoch is 0.009872144119581207
Validation loss for final epoch is 0.00841051997244358
---------- Beginning tests for gru3_2 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

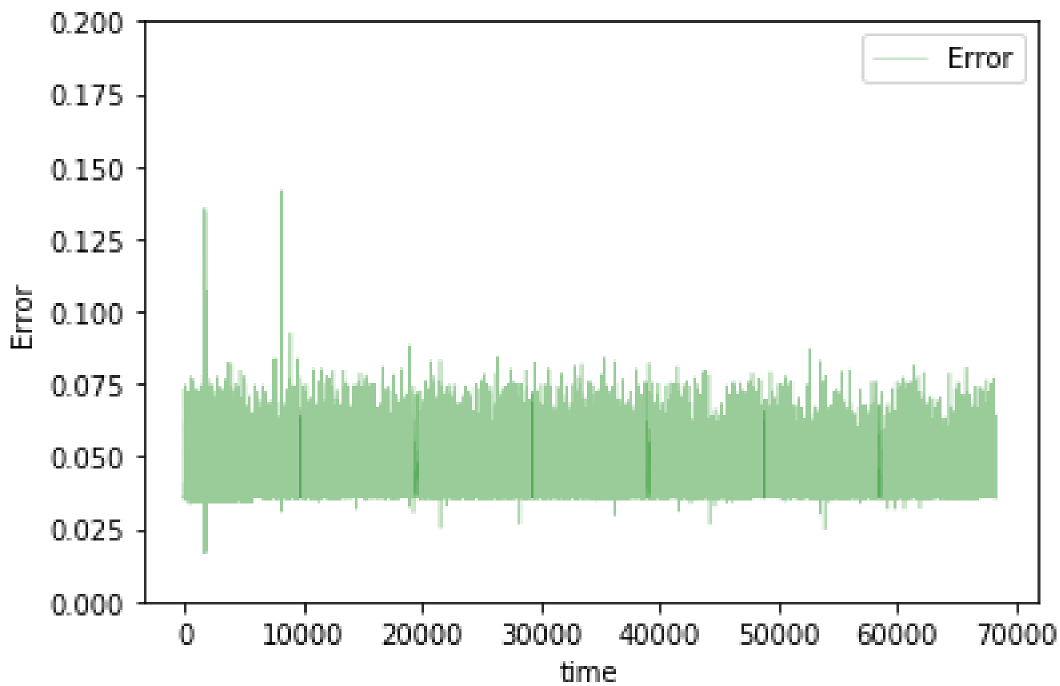

7000
6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

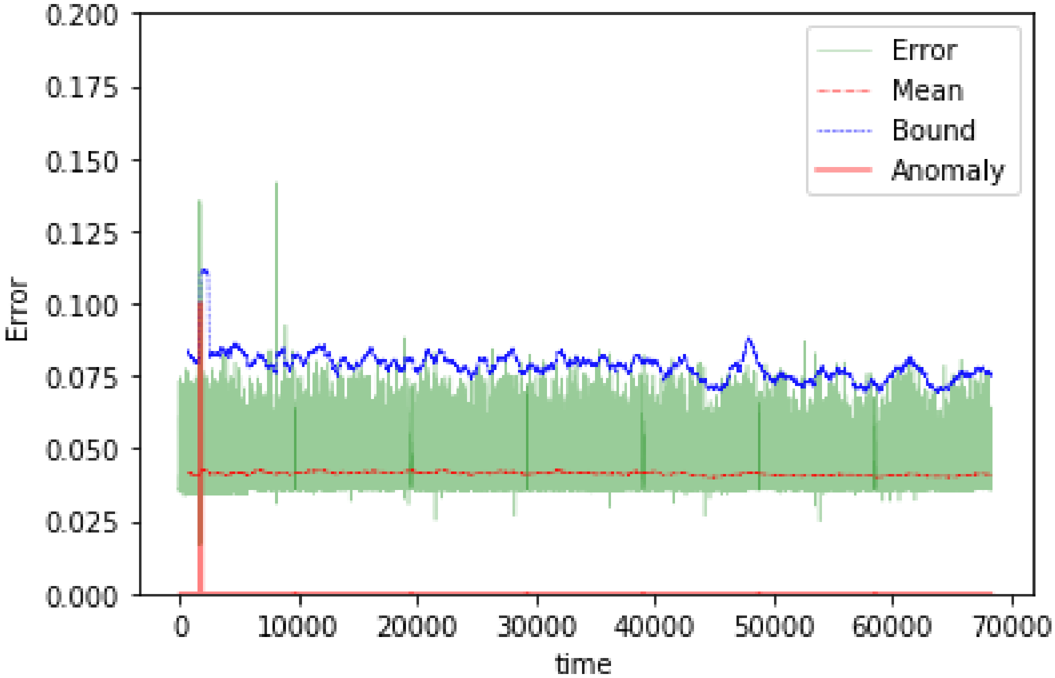

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error

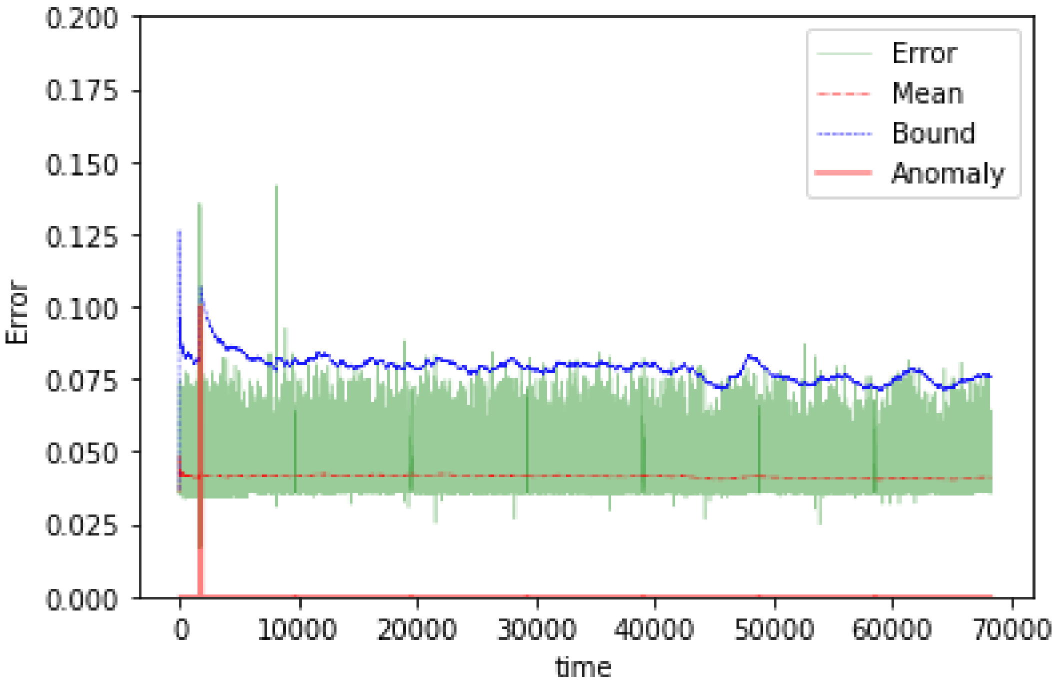

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly



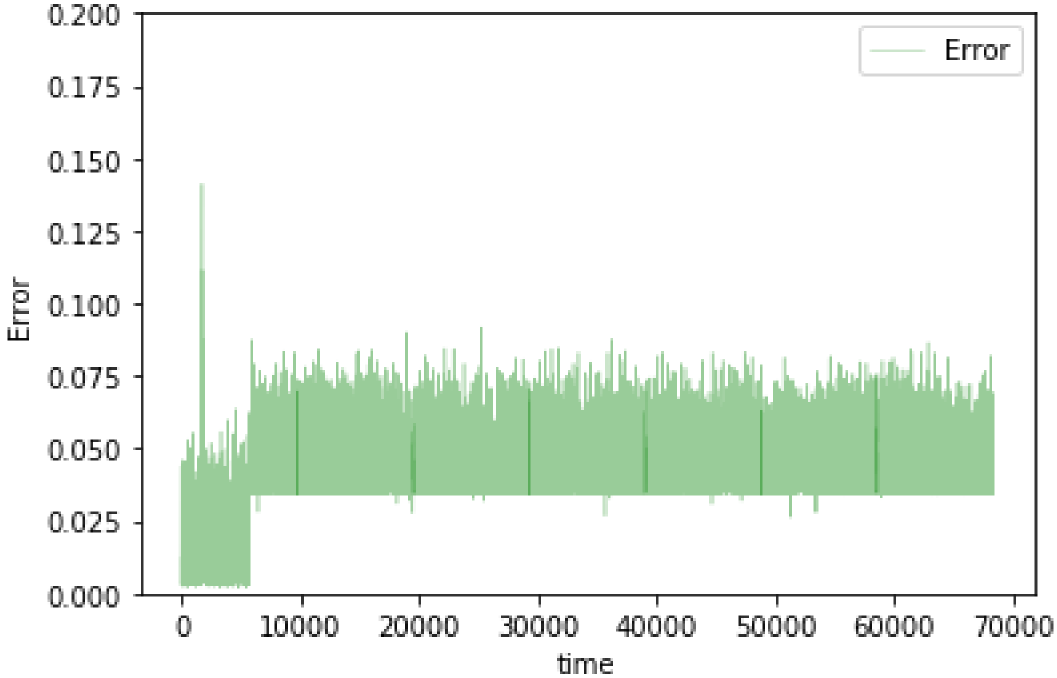

```
The mean error for gru3_2_disk_IO_start_ is 0.017630883880487 84 for length 68297
Testing on Avg. load data.
```

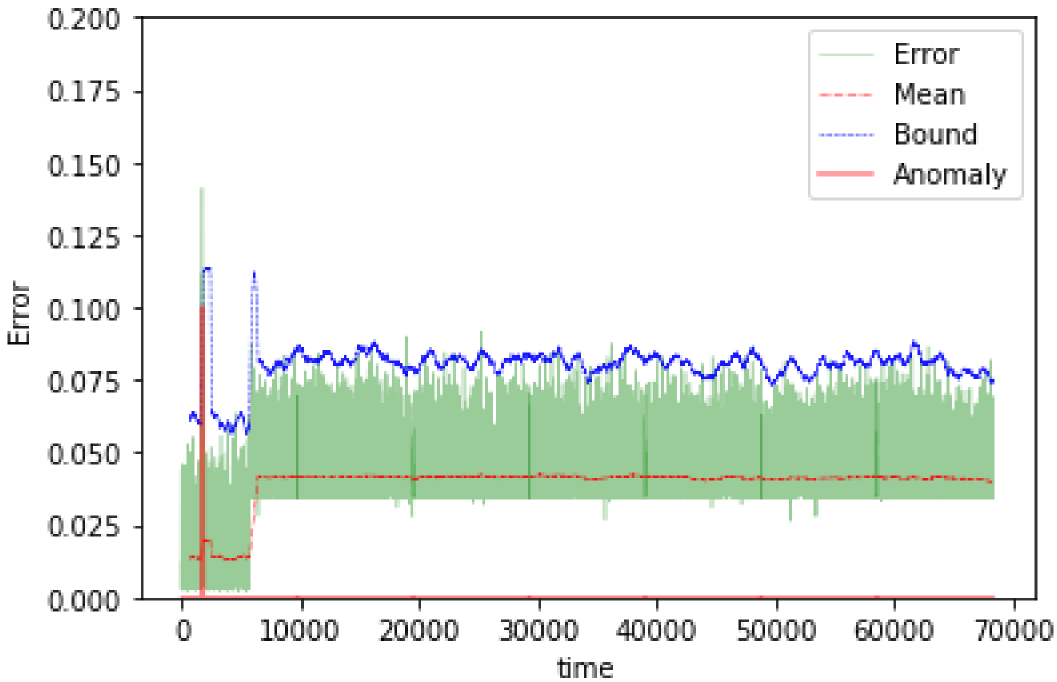


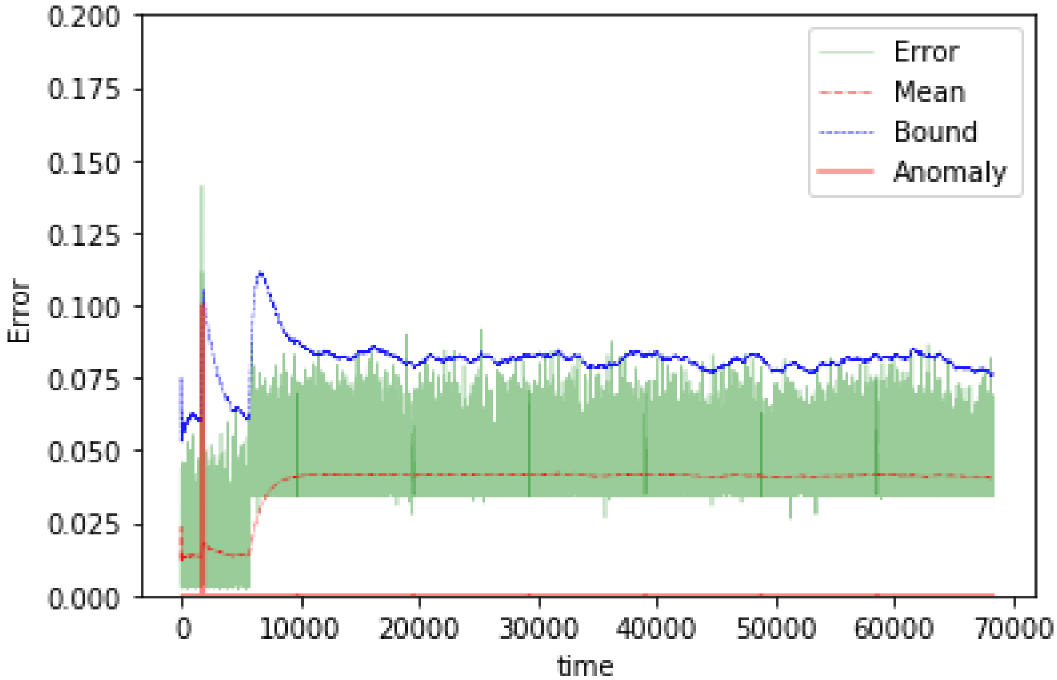

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

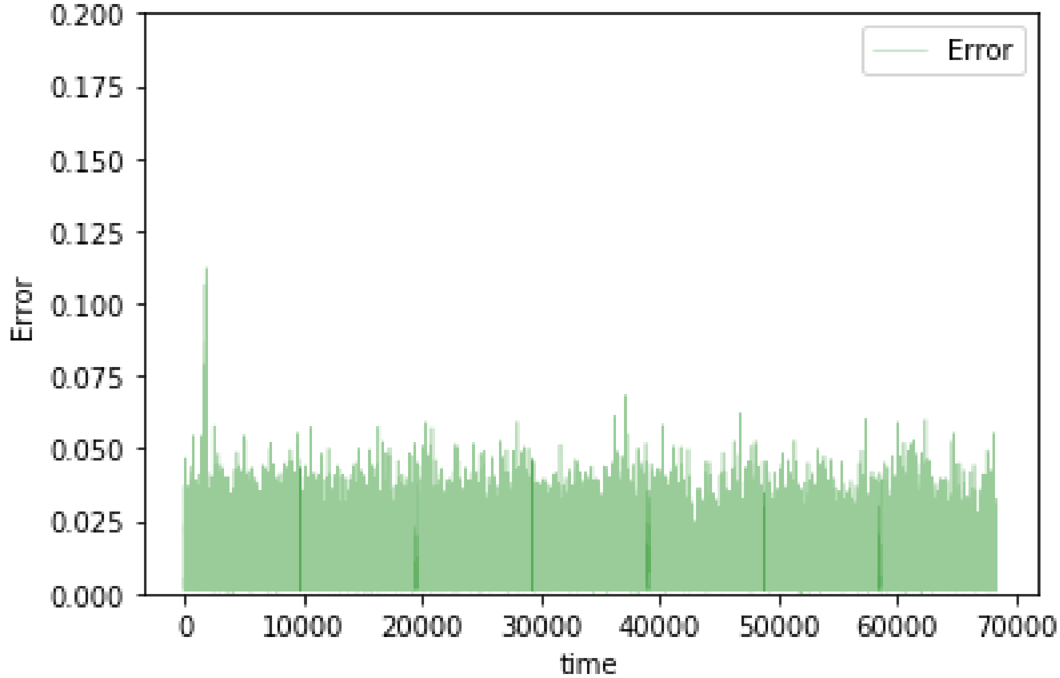
```
The mean error for gru3_2_avg_load_ is 0.041336440357052064 for length 68297
Testing on app change early data.
```

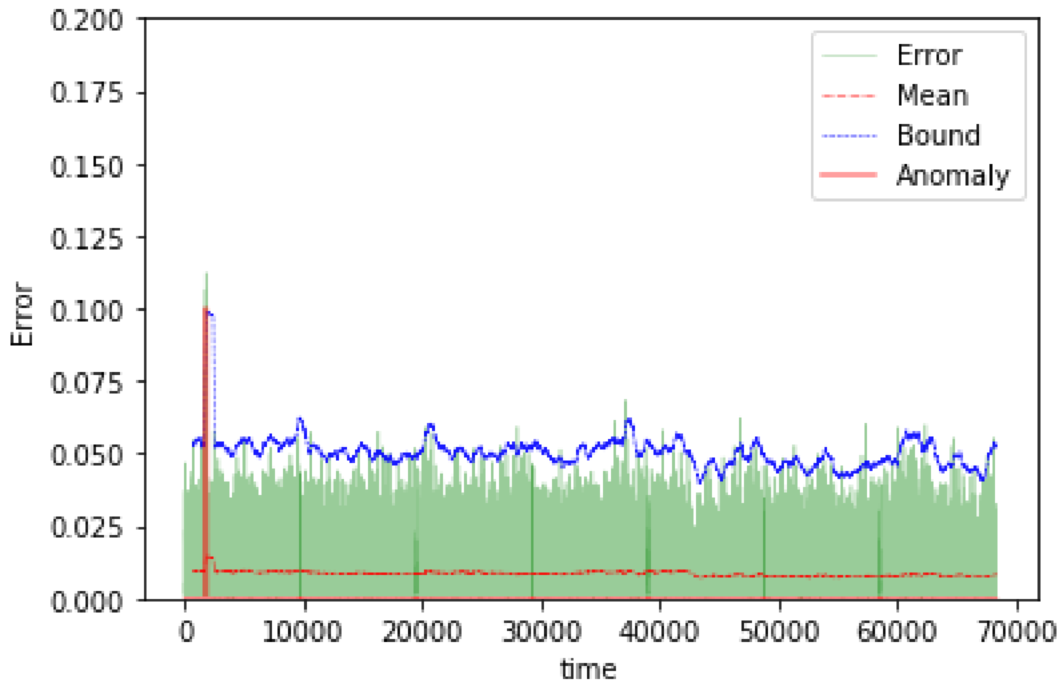


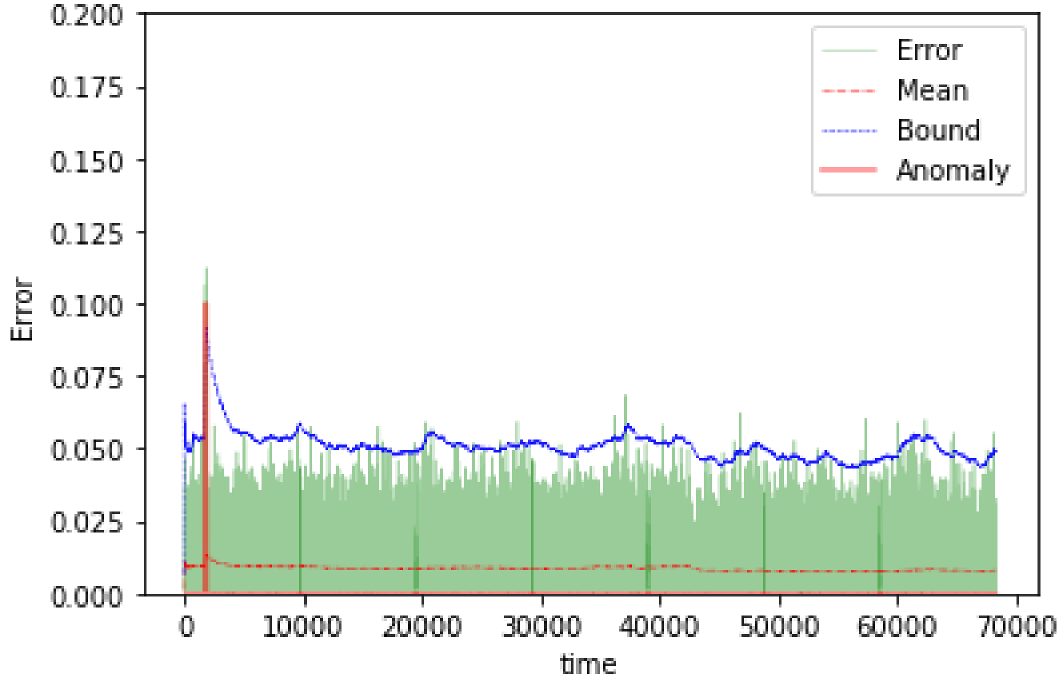





```
The mean error for gru3_2_app_change_early_ is 0.039152300760763634 for length 68297
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

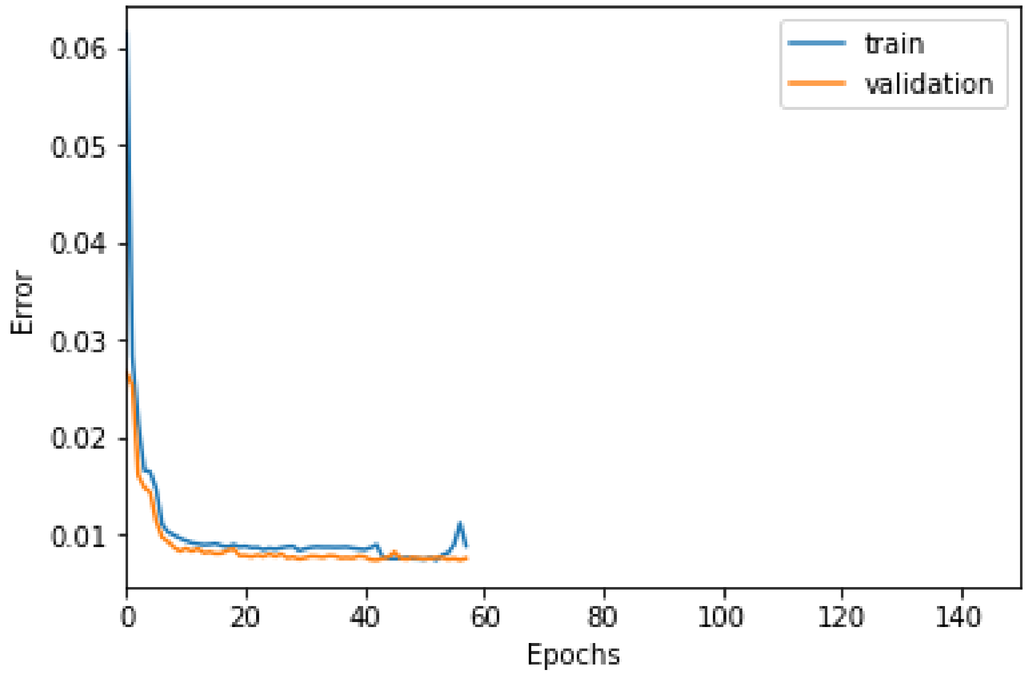

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



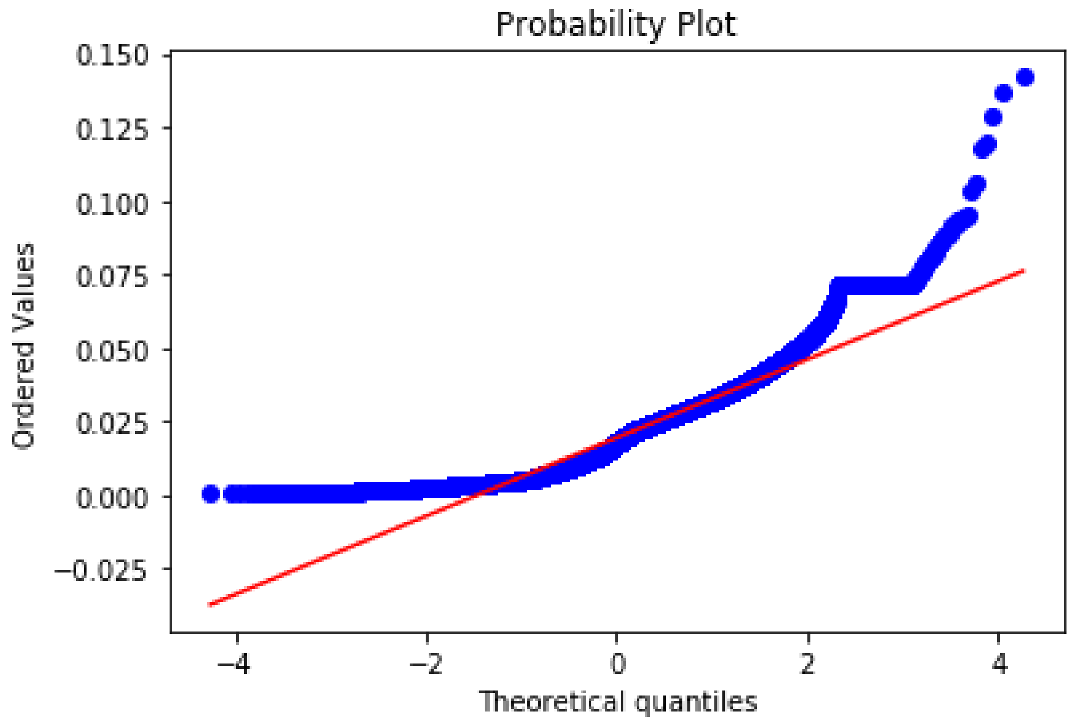

```
The mean error for gru3_2_normal_ is 0.008578787117894181 for length 68297
====================
```

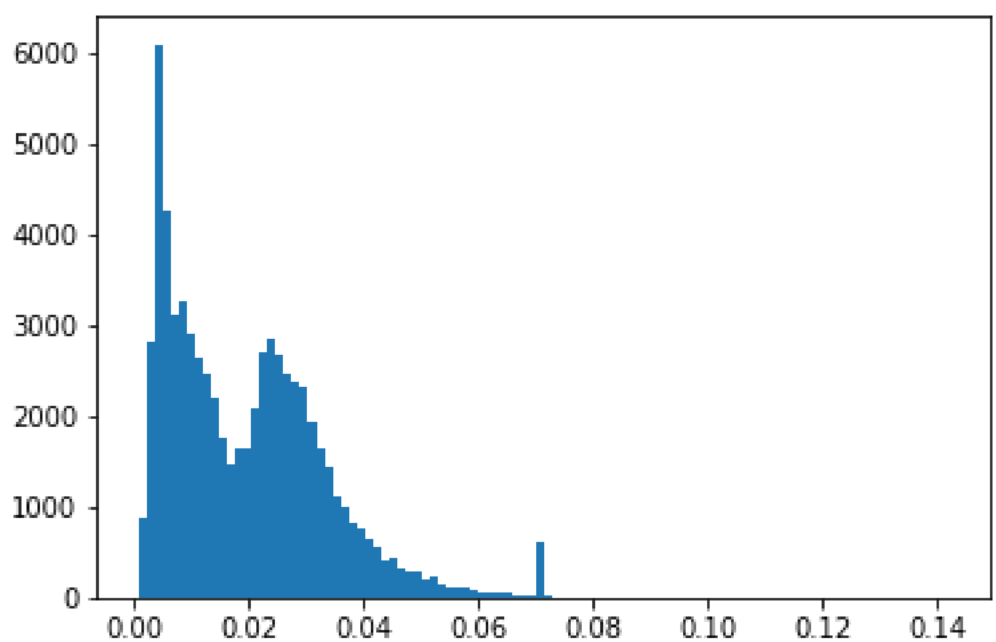
5 steps

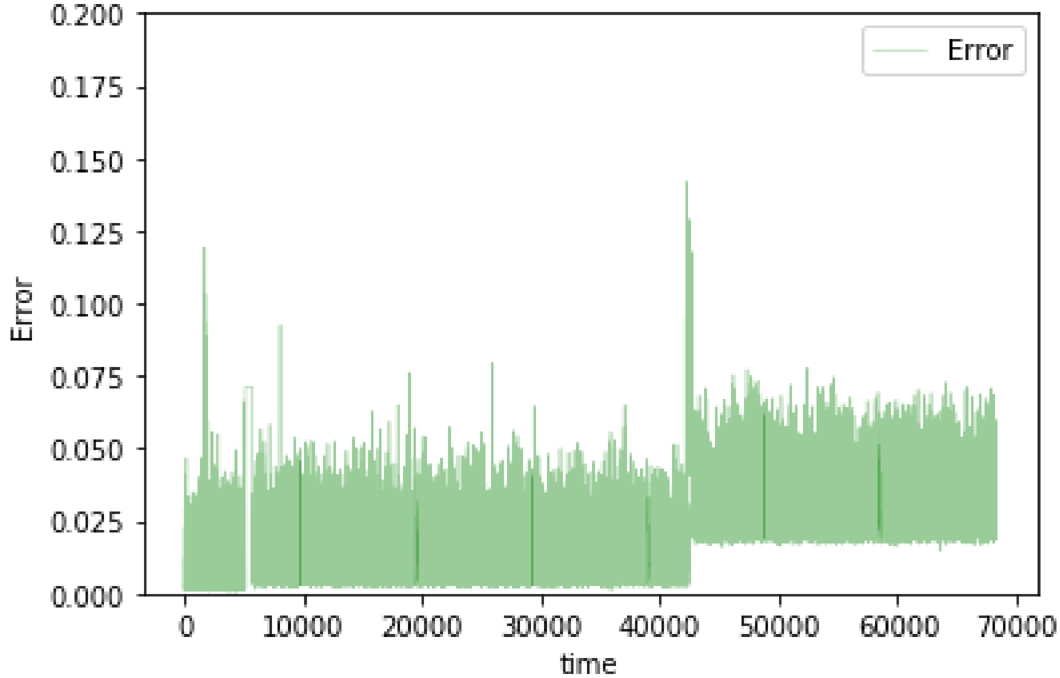
```
In [211]: TIMESTEPS = 5
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS, 0)
          vgen = flat_generator(val_X, TIMESTEPS, 0)
          name = "gru3_5"

In [212]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(10, activation='relu', return_sequences=True)(hidden)
          hidden = GRU(10, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [213]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [214]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

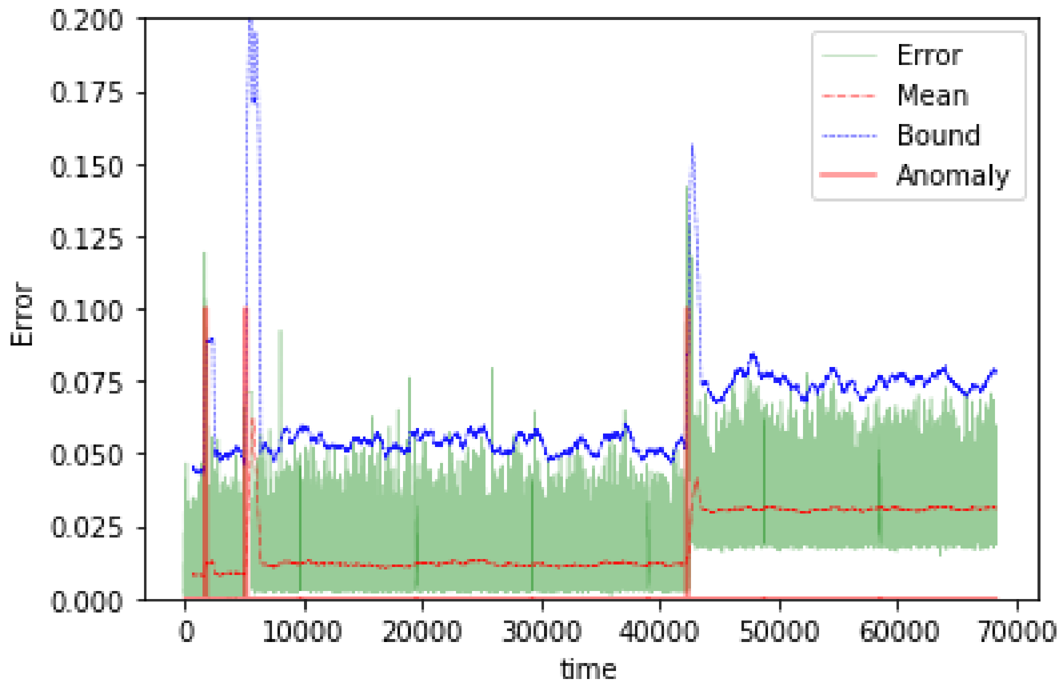


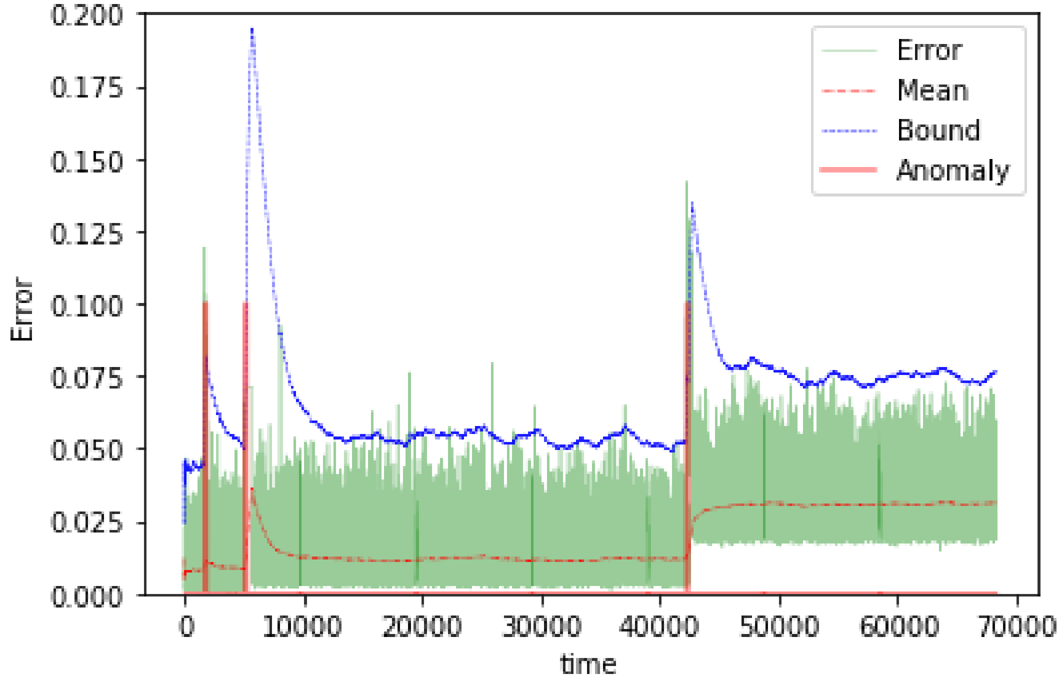

```
Training loss for final epoch is 0.008854601583327166
Validation loss for final epoch is 0.007585983700933866
---------- Beginning tests for gru3_5 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

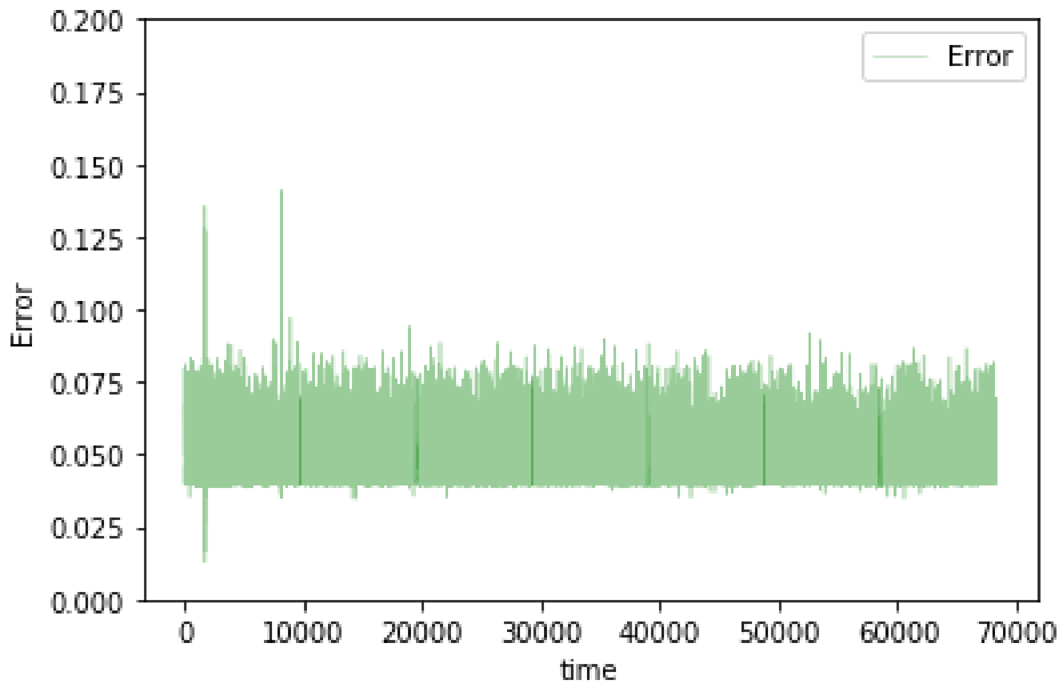

6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

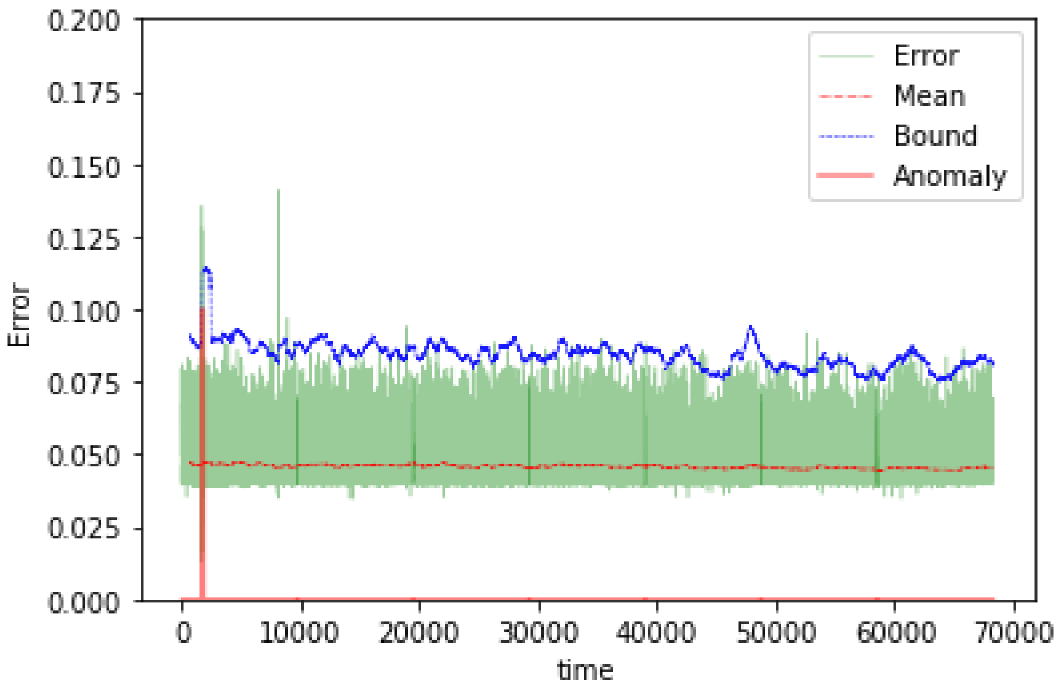

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

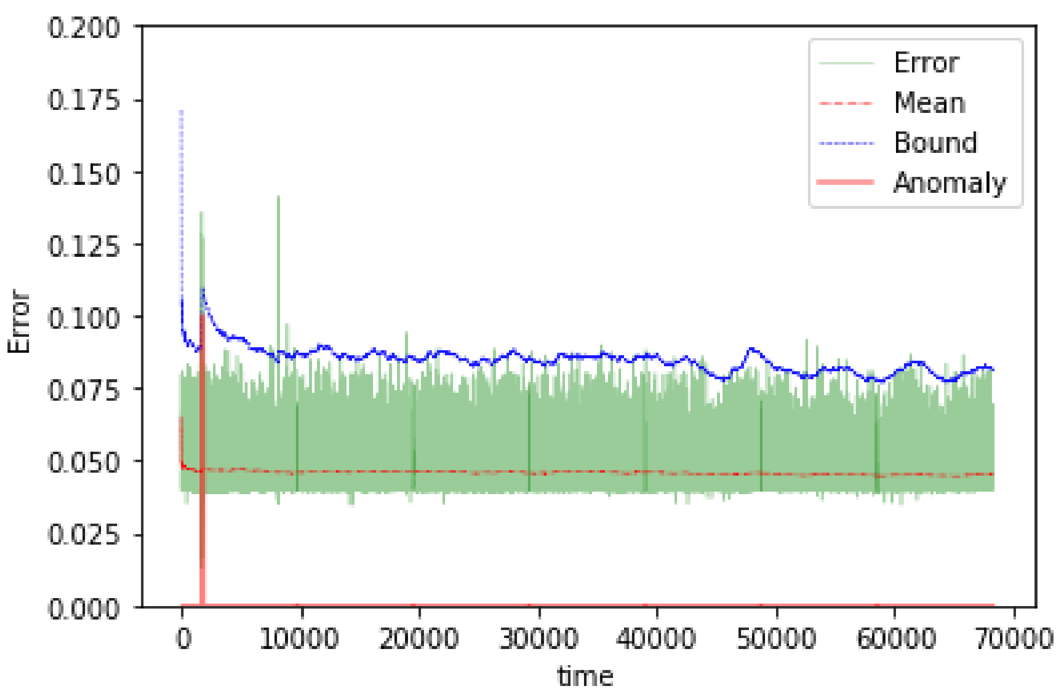

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



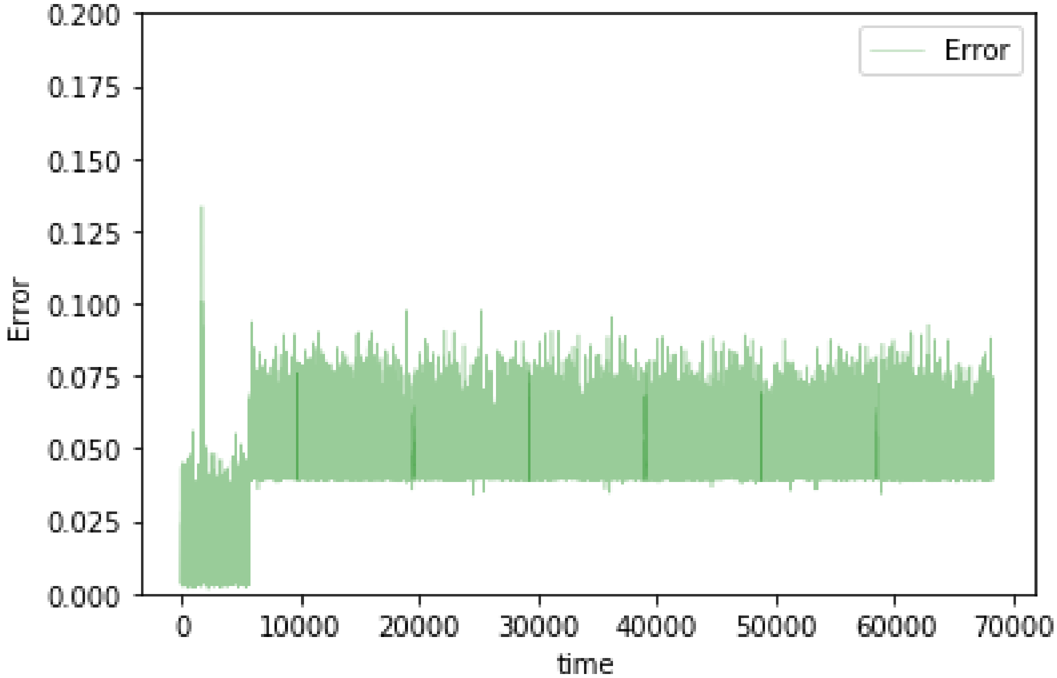

```
The mean error for gru3_5_disk_IO_start_ is 0.019459468447570835 for length 68294
Testing on Avg. load data.
```

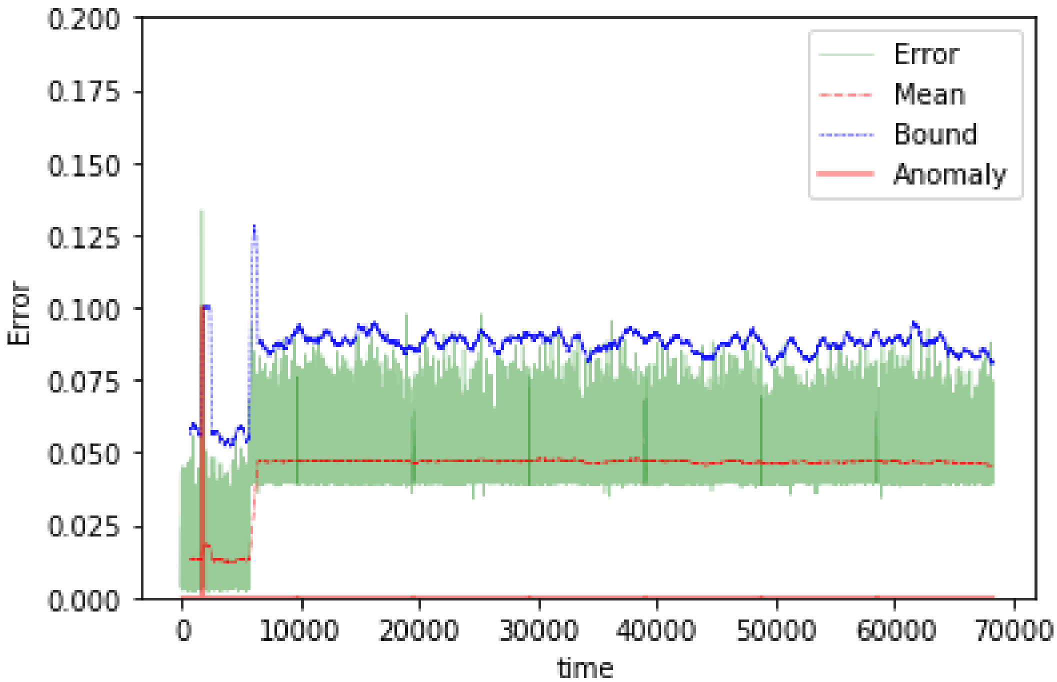


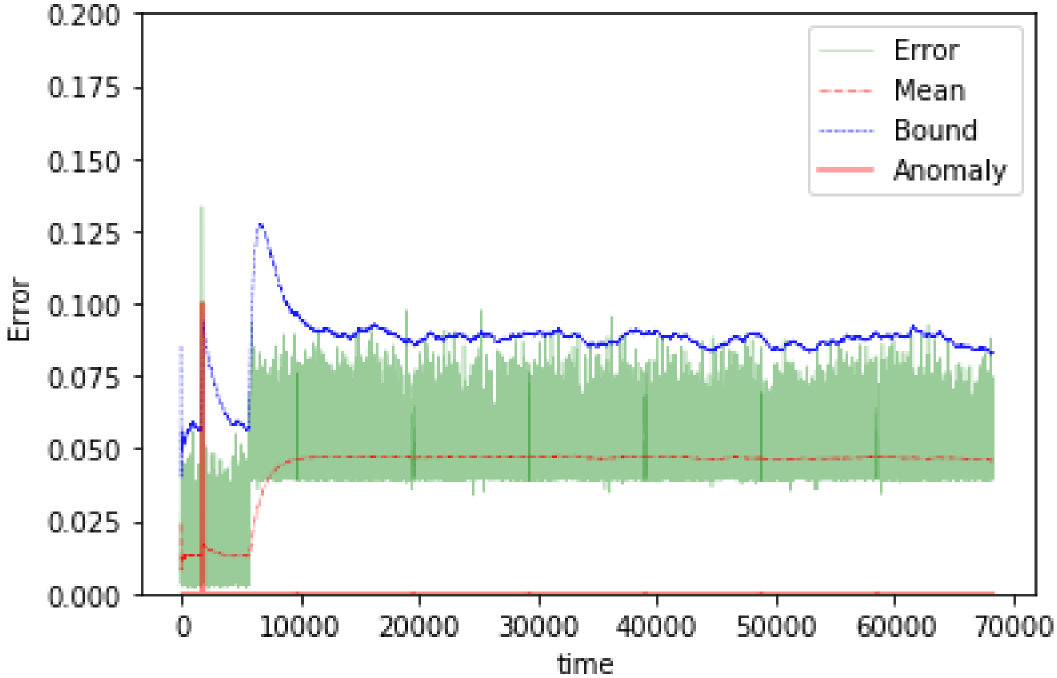

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

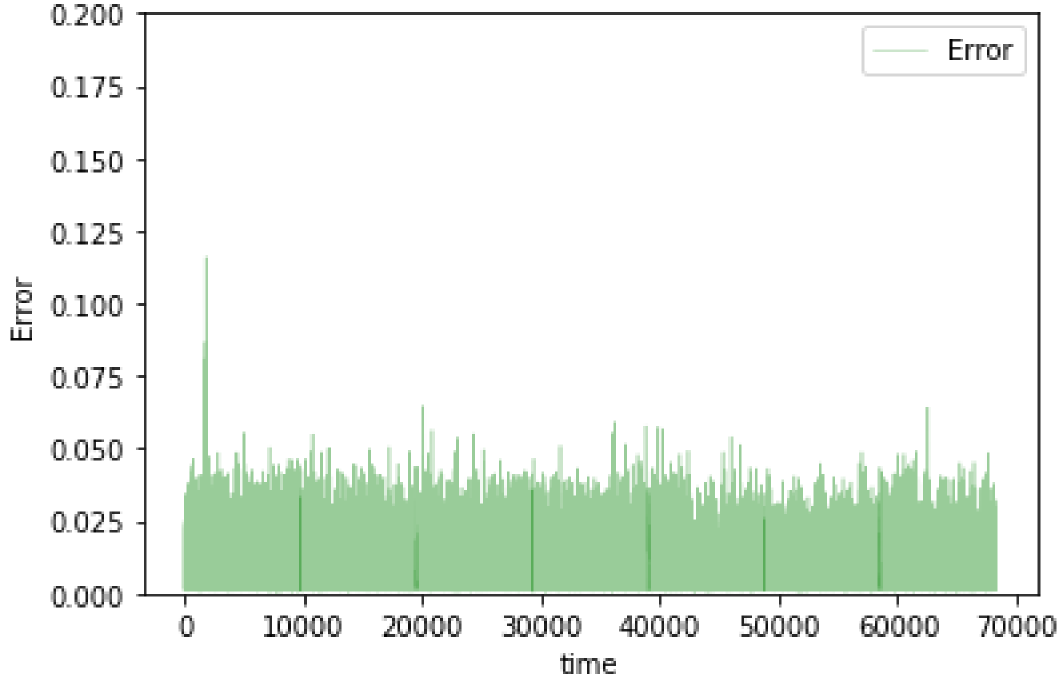
```
The mean error for gru3_5_avg_load_ is 0.045797722926479656 for length 68294
Testing on app change early data.
```

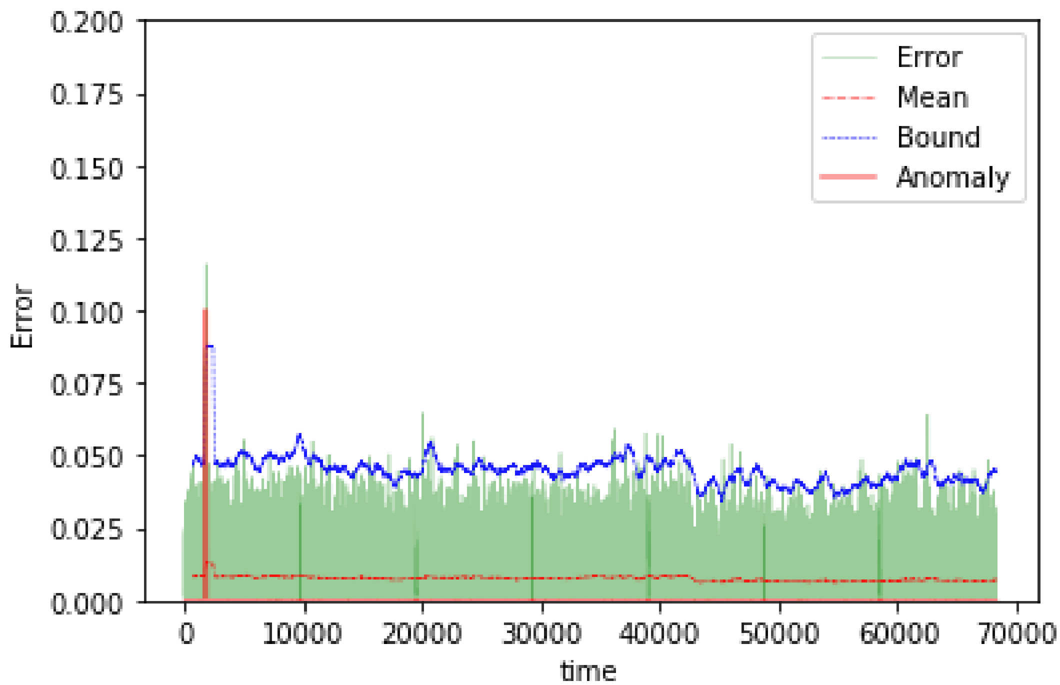


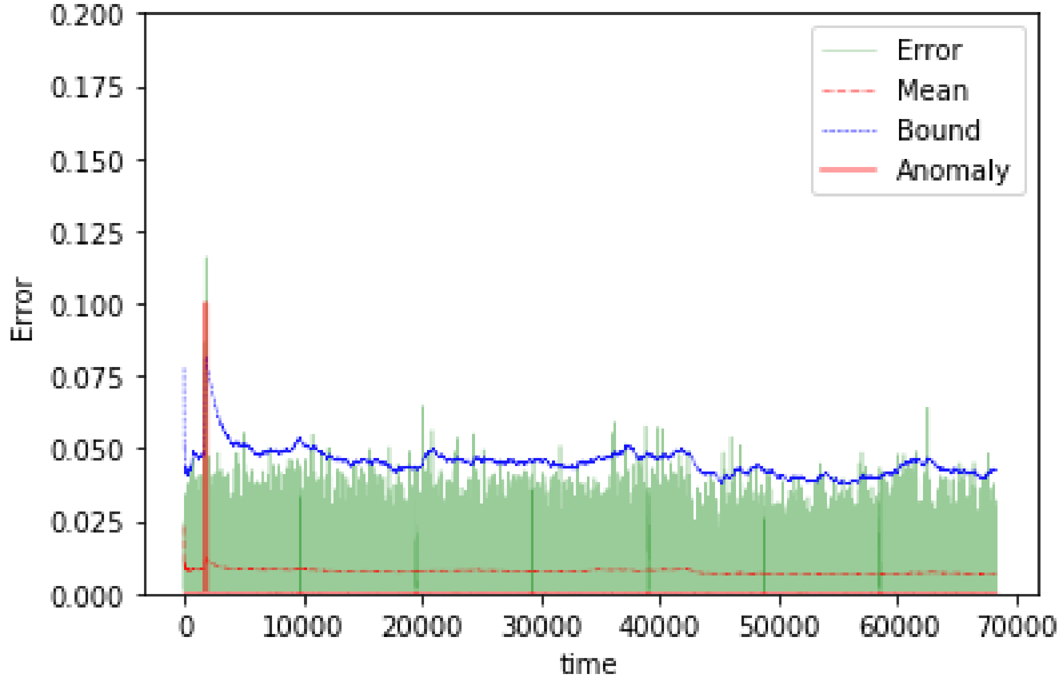





```
The mean error for gru3_5_app_change_early_ is 0.044086462937778244 for length 68294
Testing on Normal data.
```


Error
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

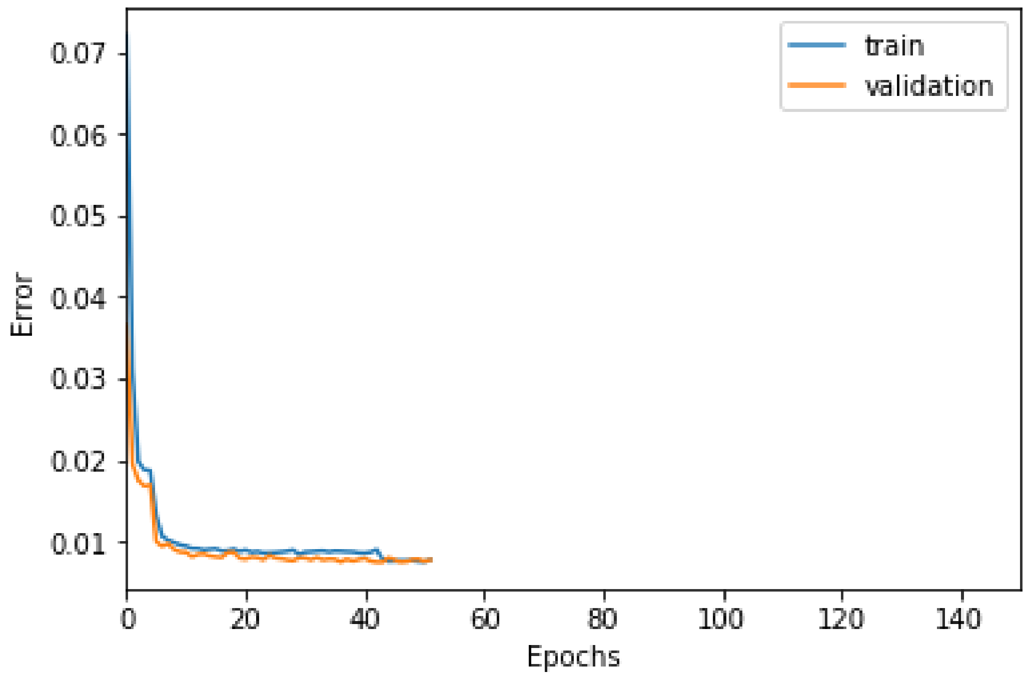

Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000



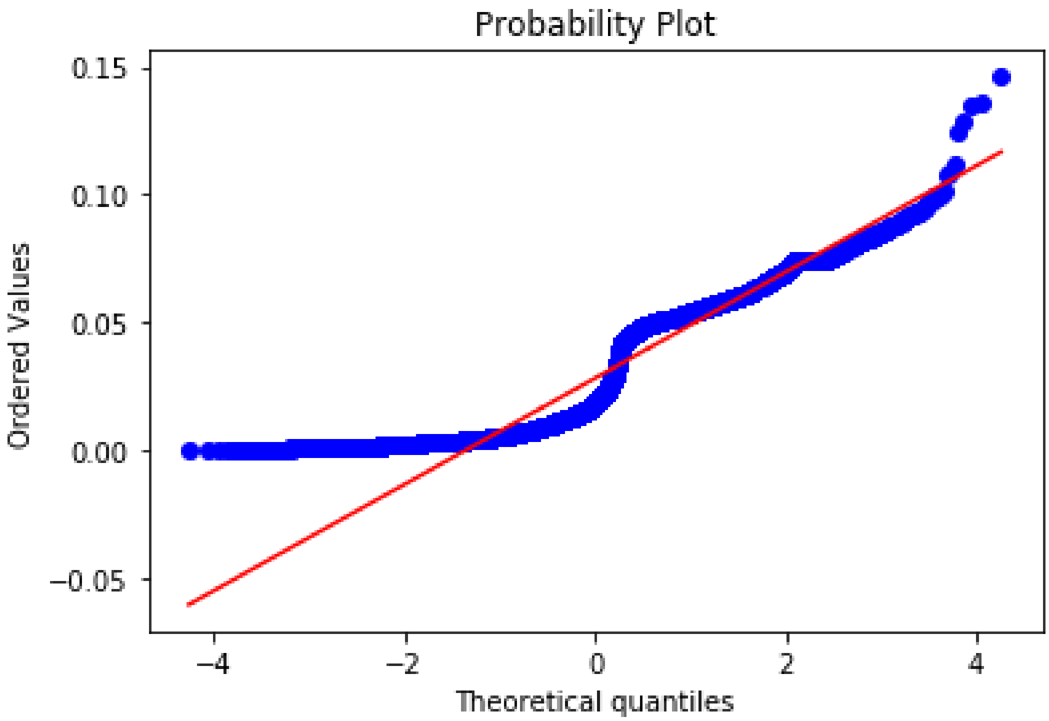

```
The mean error for gru3_5_normal_ is 0.007593853277649745 for length 68294
====================
```

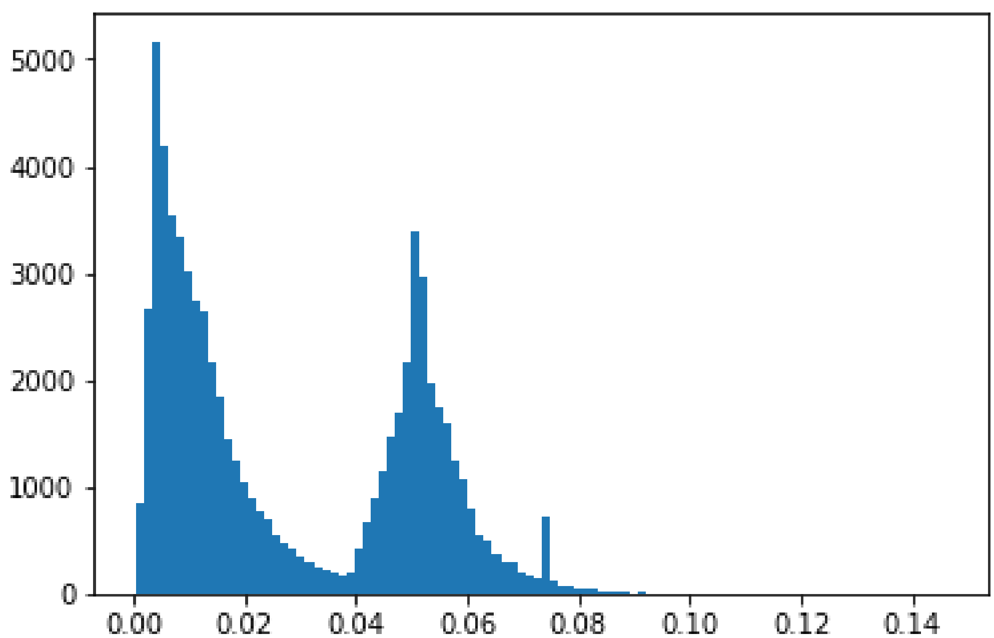
10 steps

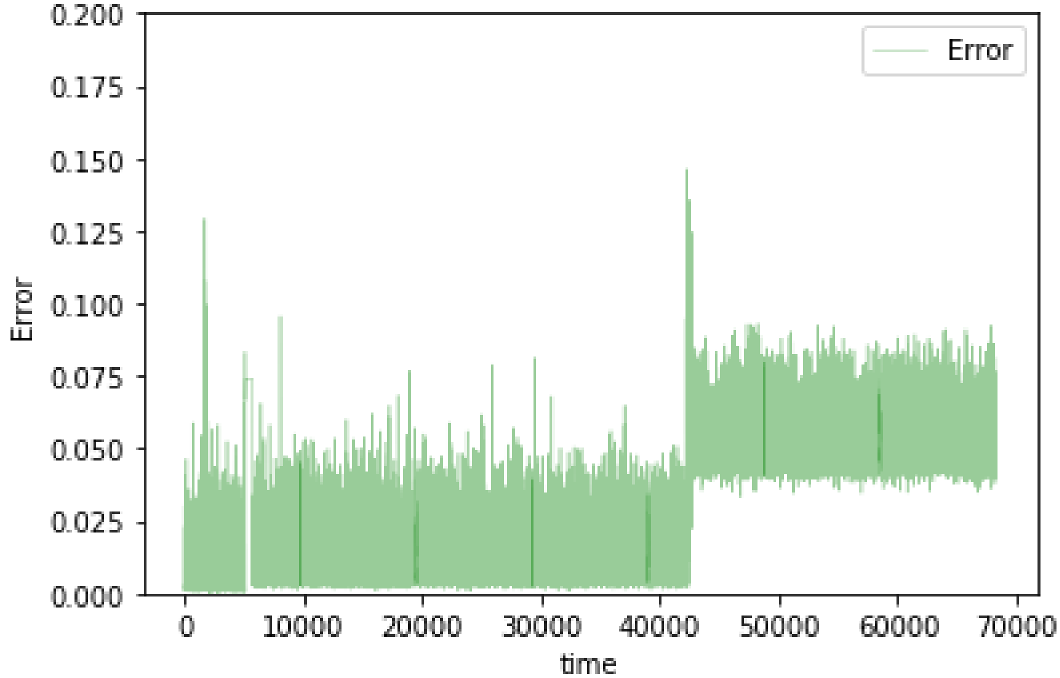
```
In [215]: TIMESTEPS = 10
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS, 0)
          vgen = flat_generator(val_X, TIMESTEPS, 0)
          name = "gru3_10"

In [216]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(10, activation='relu', return_sequences=True)(hidden)
          hidden = GRU(10, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [217]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [218]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

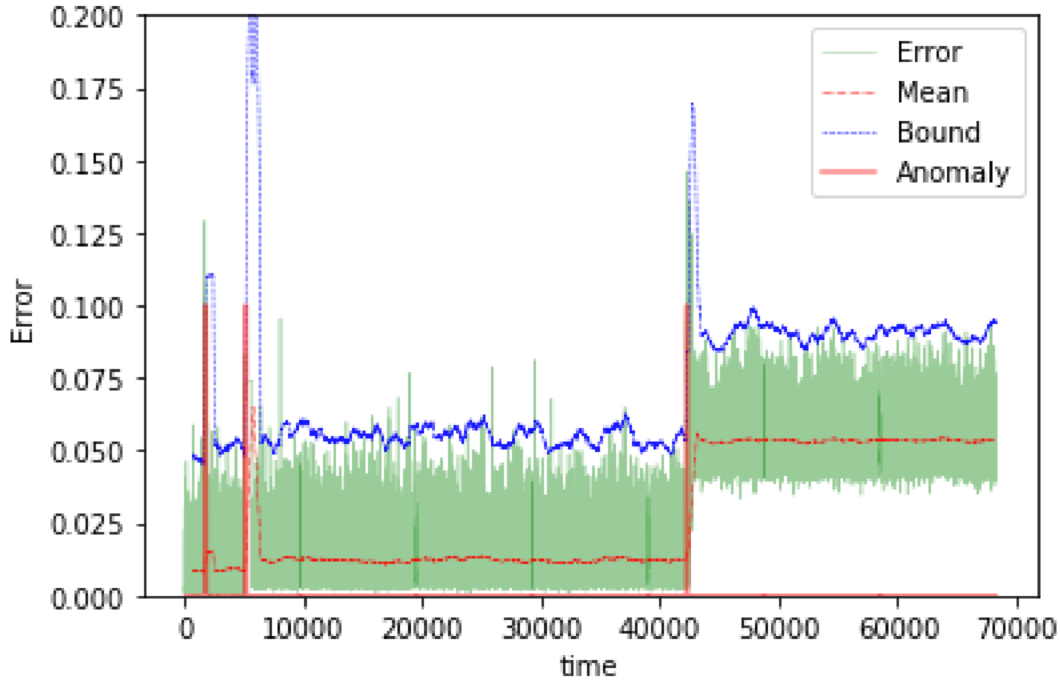


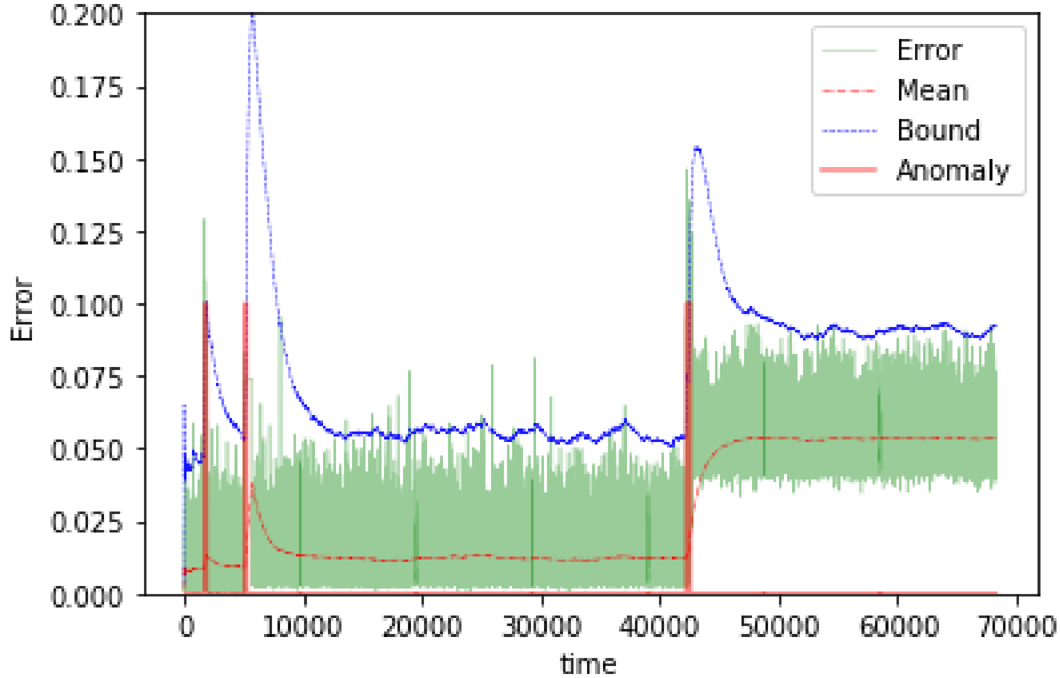

```
Training loss for final epoch is 0.007820138113107533
Validation loss for final epoch is 0.007798917698091827
---------- Beginning tests for gru3_10 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.15
0.10
0.05
0.00
−0.05
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

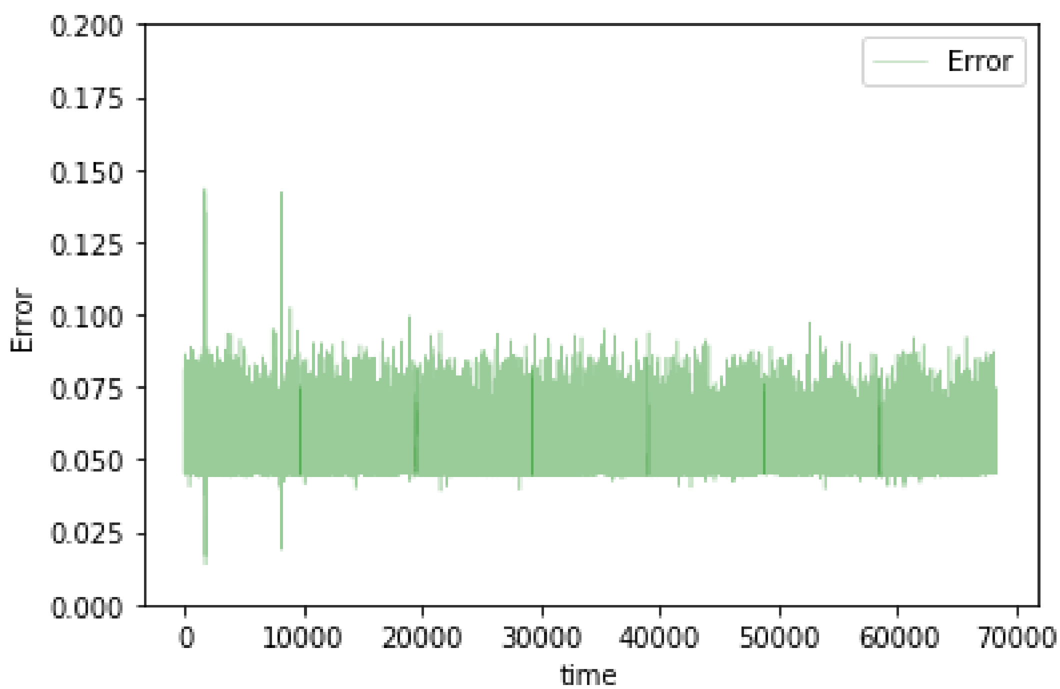

5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

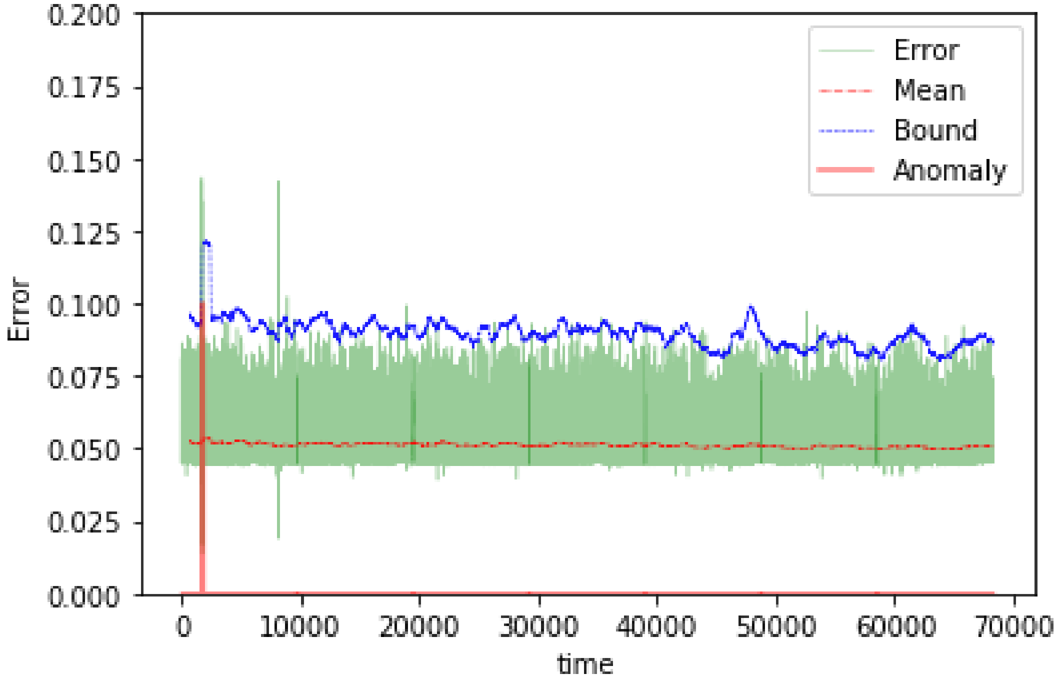

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

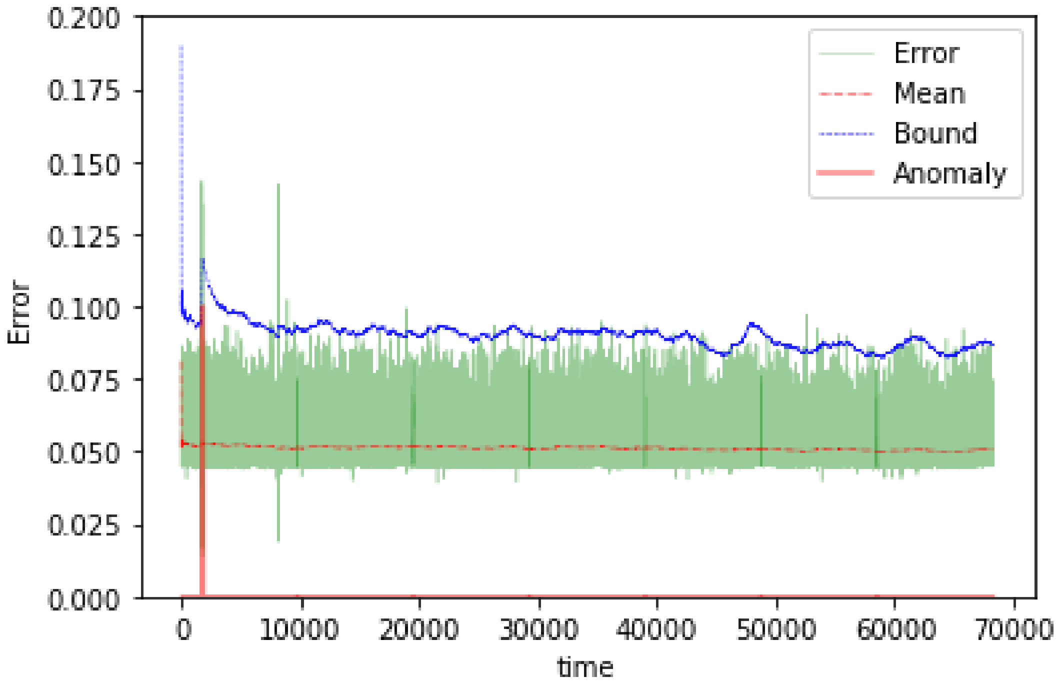

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time



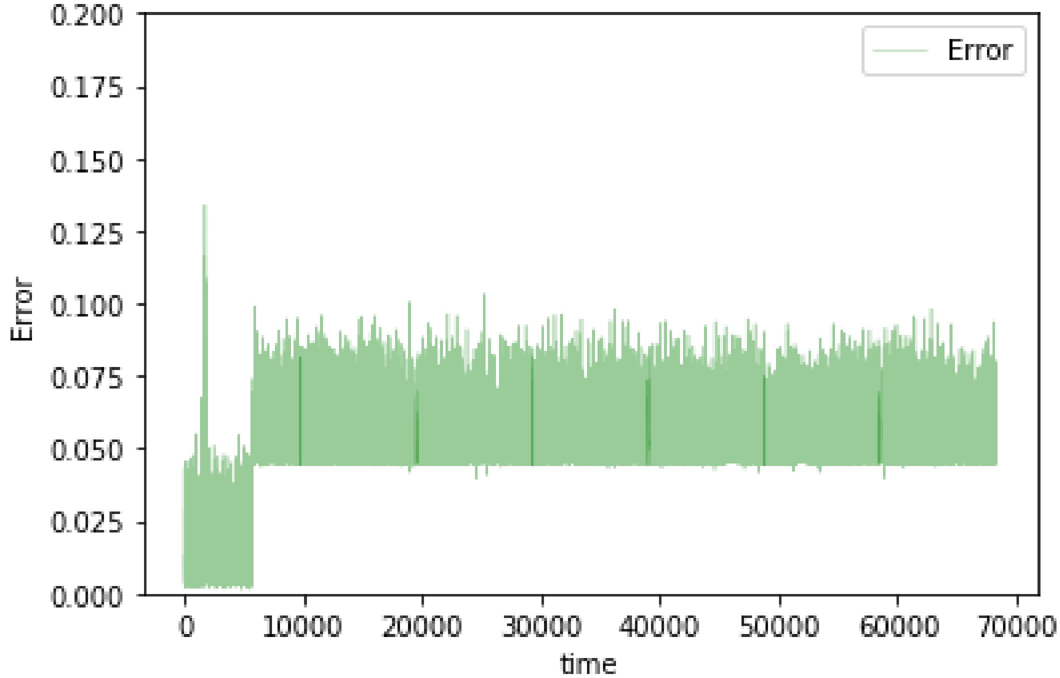

```
The mean error for gru3_10_disk_IO_start_ is 0.02821544123695163 for length 68289
Testing on Avg. load data.
```

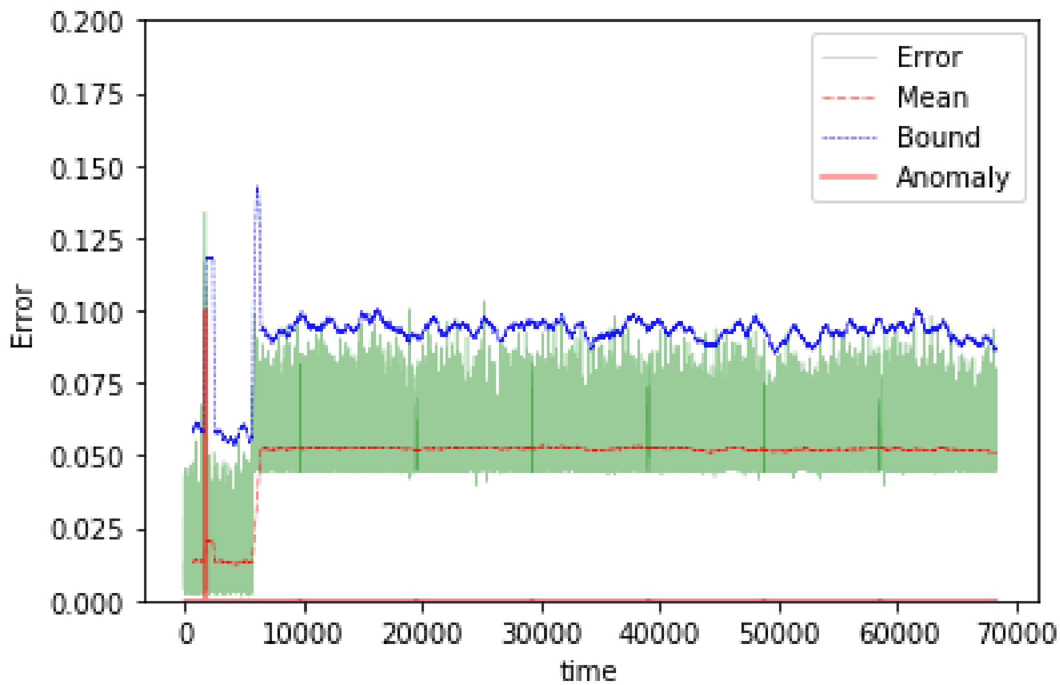


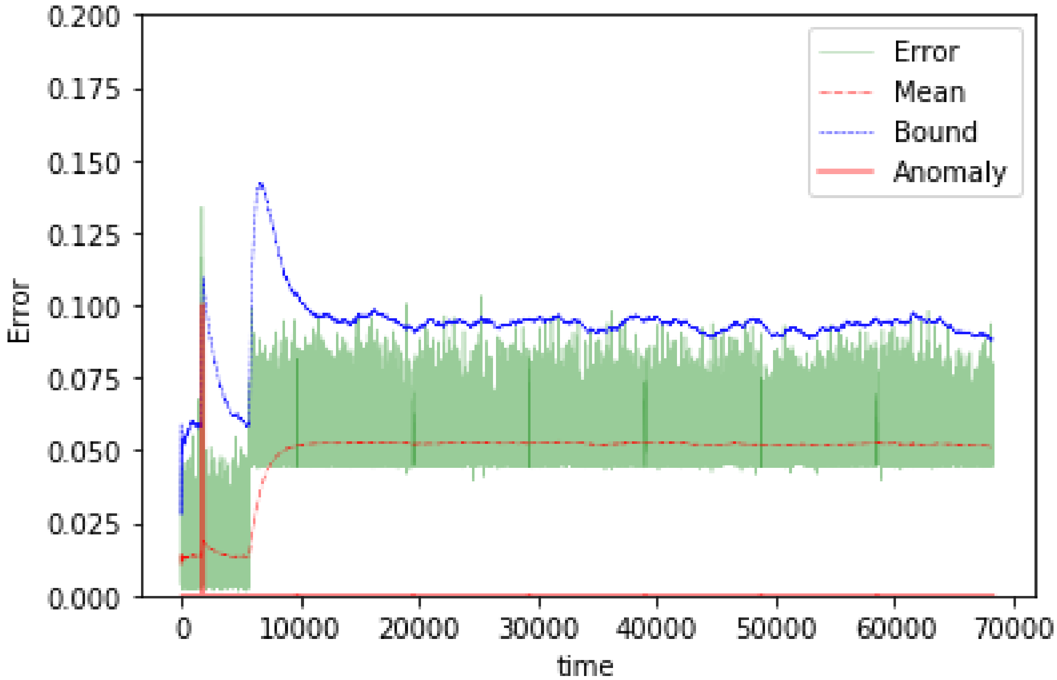

Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000


Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

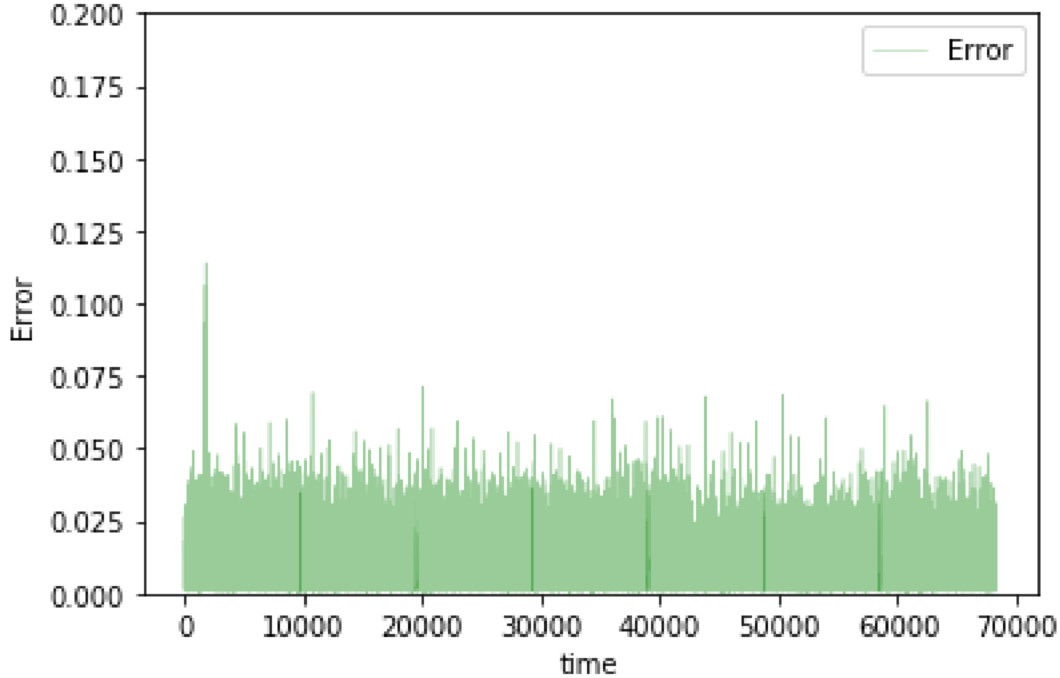
```
The mean error for gru3_10_avg_load_ is 0.05112914289314304 for length 68289
Testing on app change early data.
```



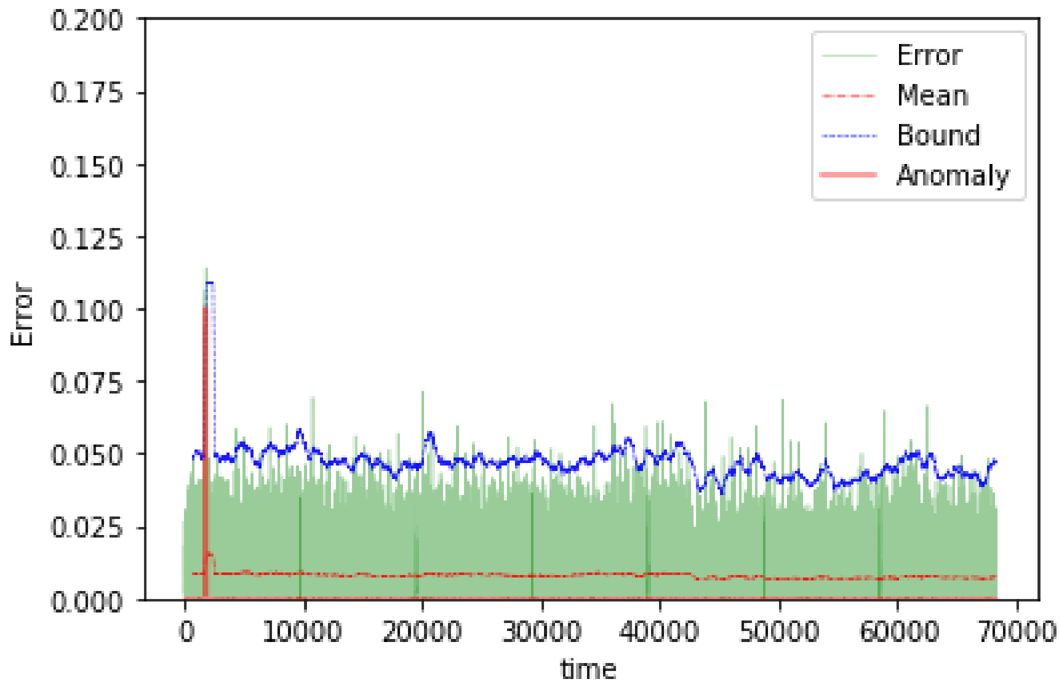




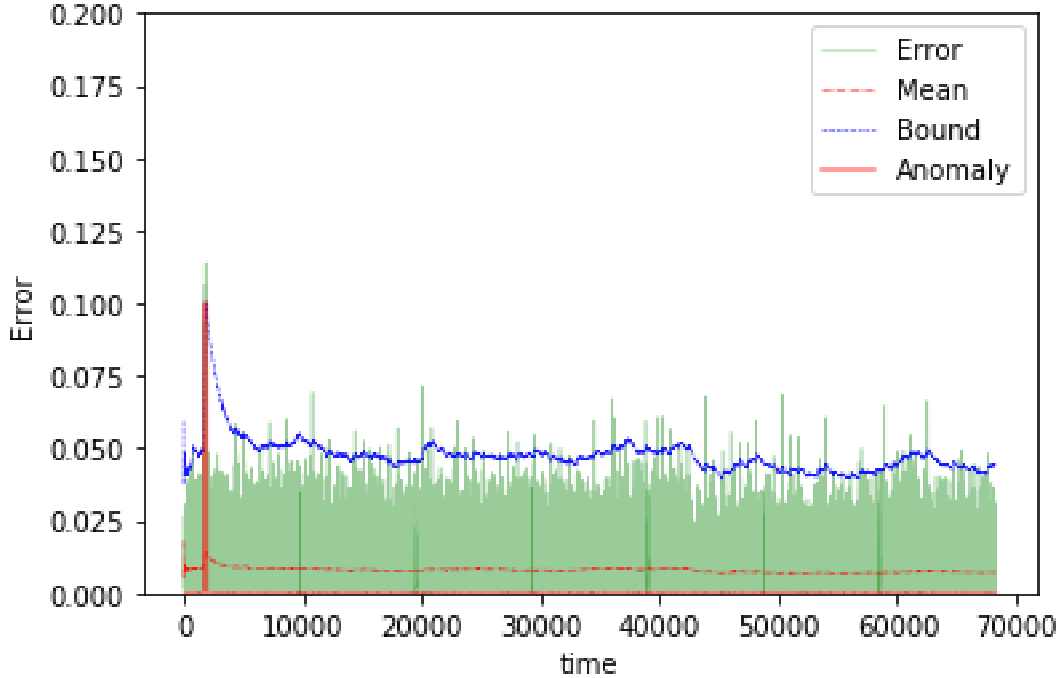

```
The mean error for gru3_10_app_change_early_ is 0.04908746926779183 for length 68289
Testing on Normal data.
```



0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
Error
0
10000
20000
30000
40000
50000
60000
70000
time



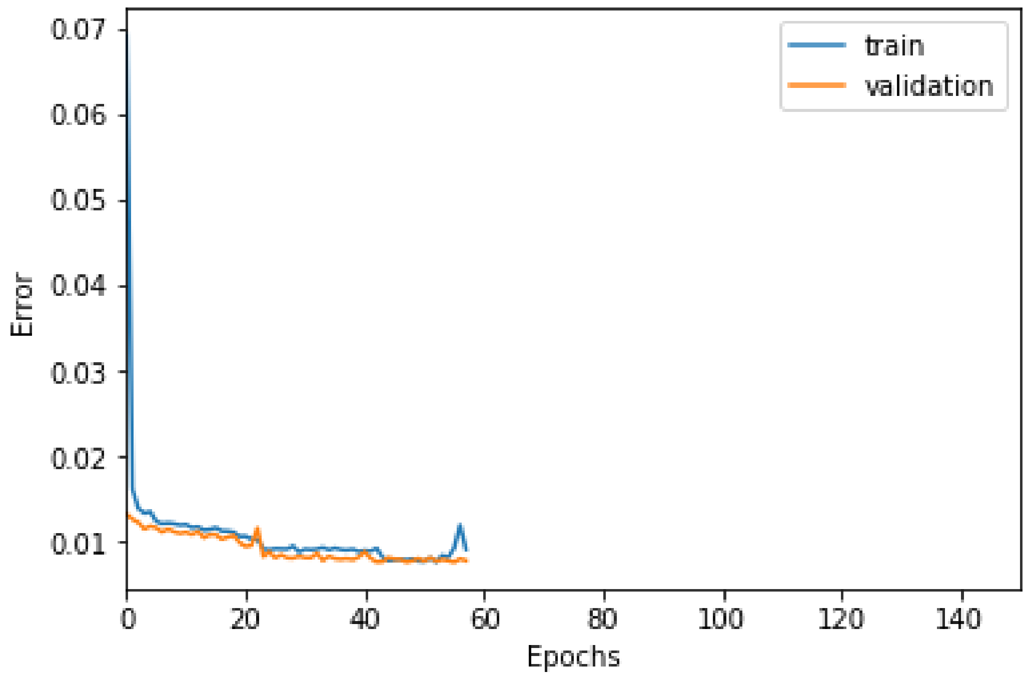
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
Error
Mean
Bound
Anomaly
0
10000
20000
30000
40000
50000
60000
70000
time



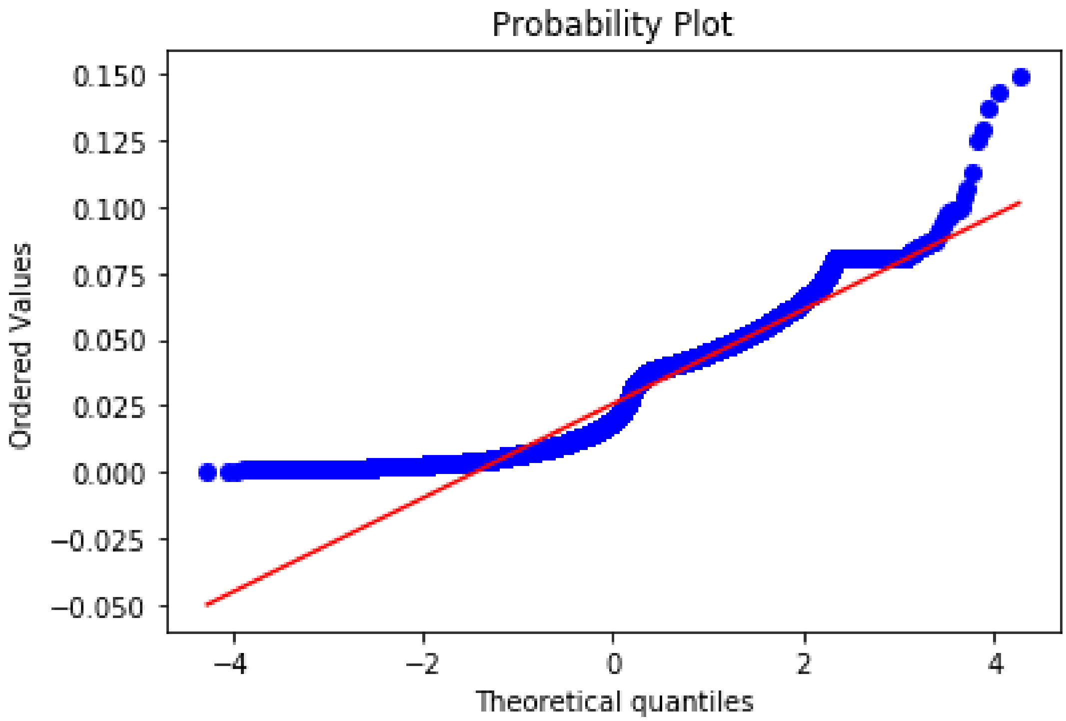

```
The mean error for gru3_10_normal_ is 0.007901852744683486 for length 68289
====================
```

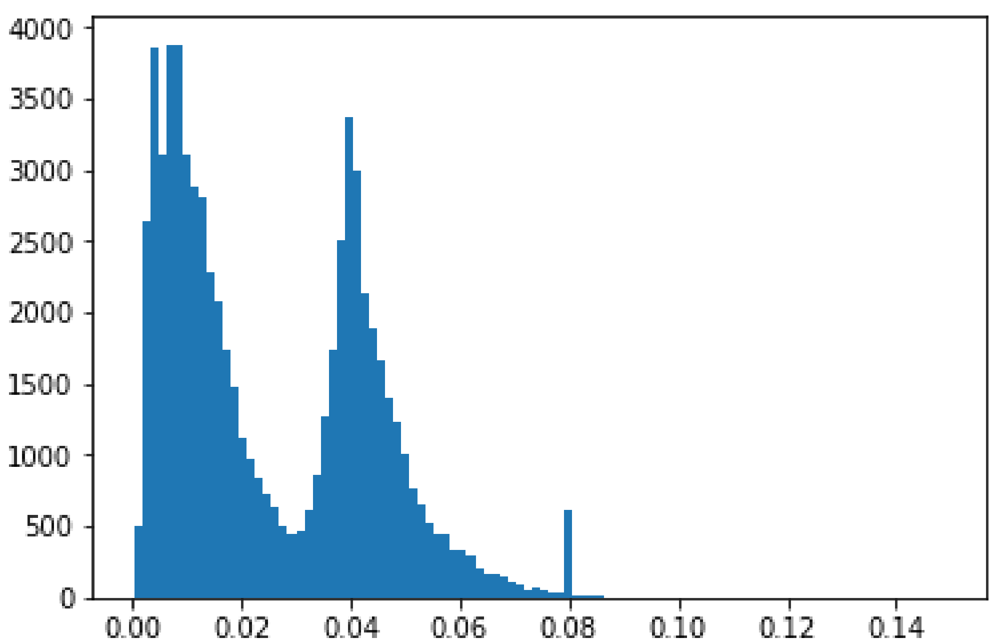
20 steps

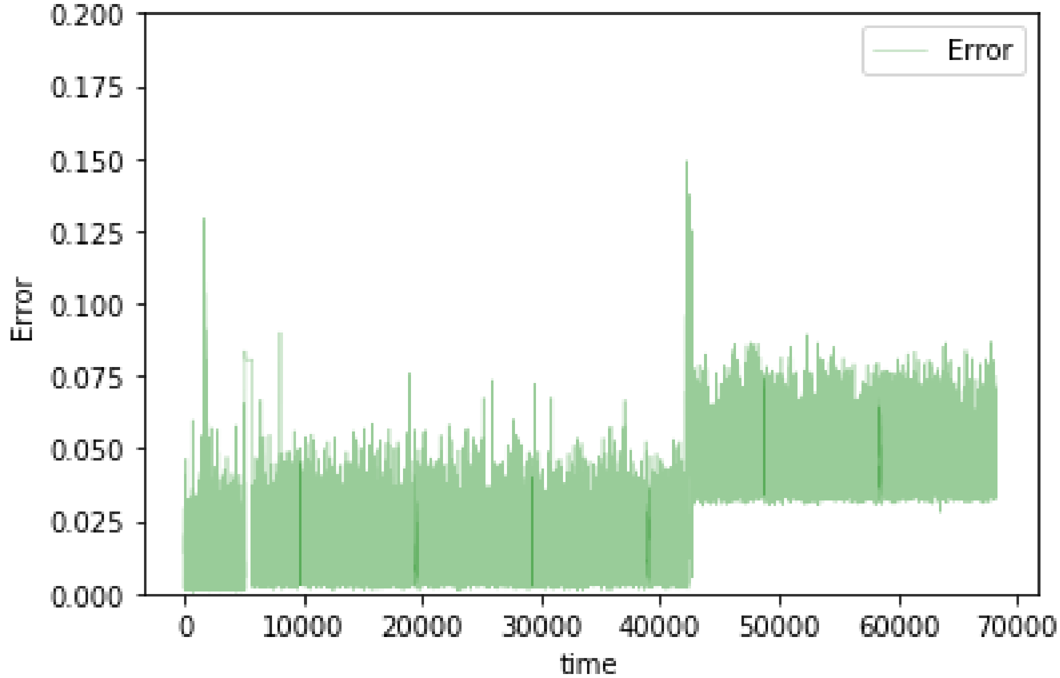
```
In [219]: TIMESTEPS = 20
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru3_20"

In [220]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(10, activation='relu', return_sequences=True)(hidden)
          hidden = GRU(10, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [221]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [222]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

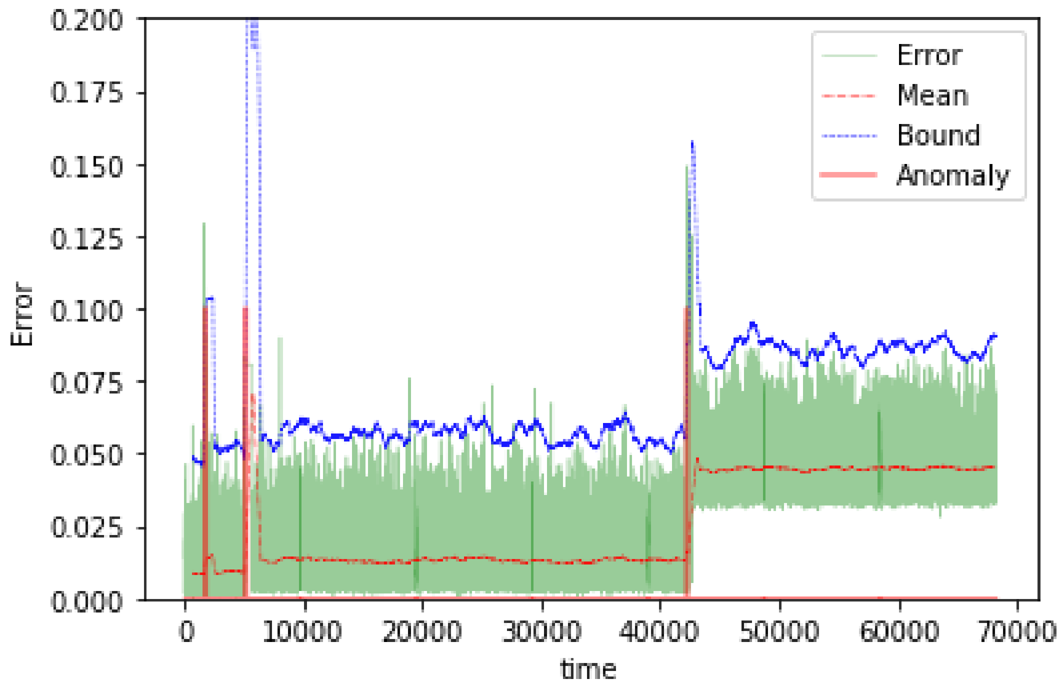


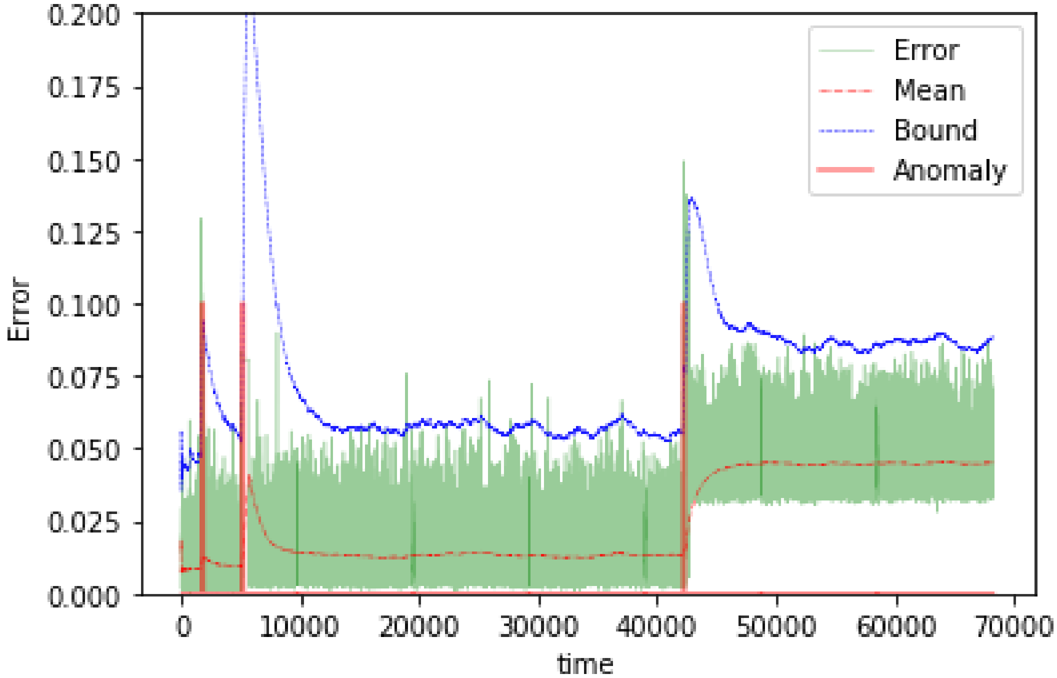

```
Training loss for final epoch is 0.009122088901116513
Validation loss for final epoch is 0.007820557567640208
---------- Beginning tests for gru3_20 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

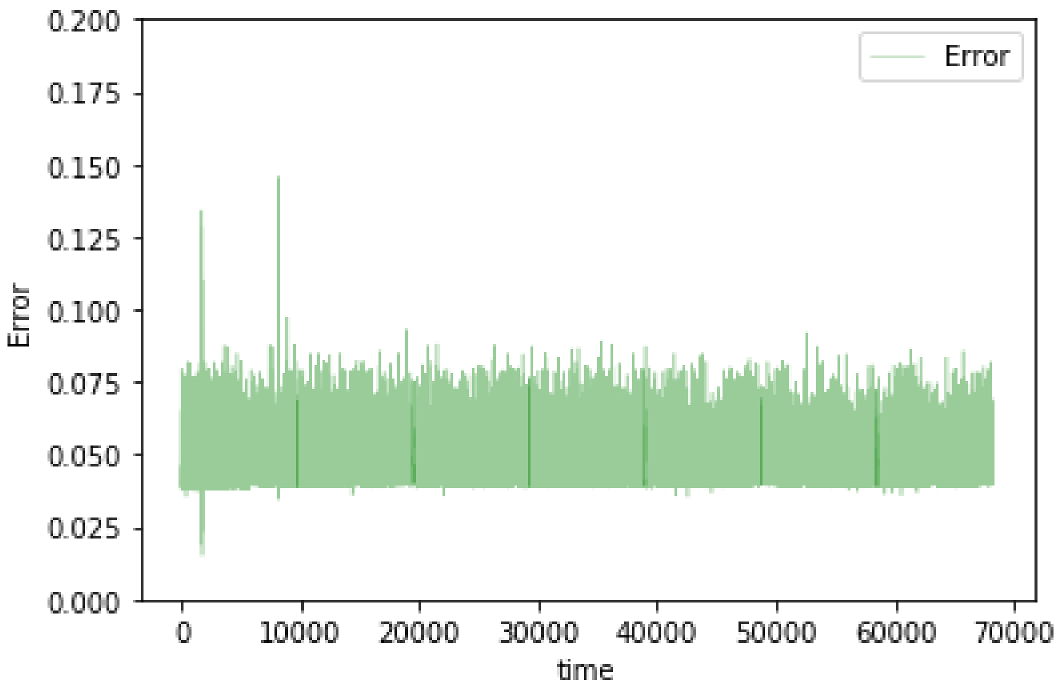

4000
3500
3000
2500
2000
1500
1000
500
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

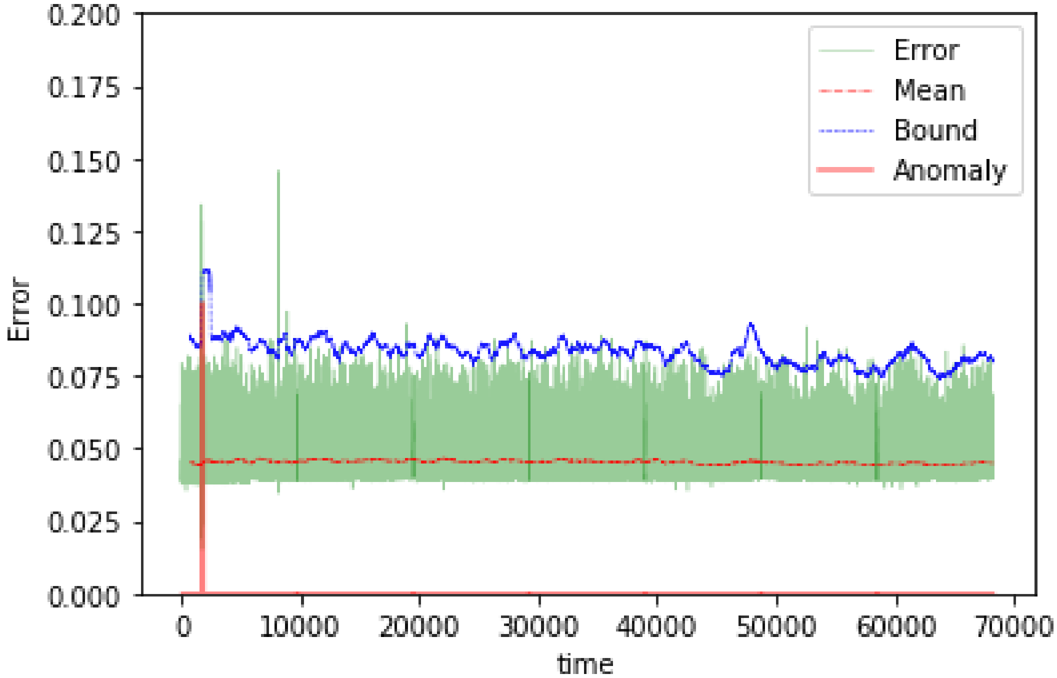

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

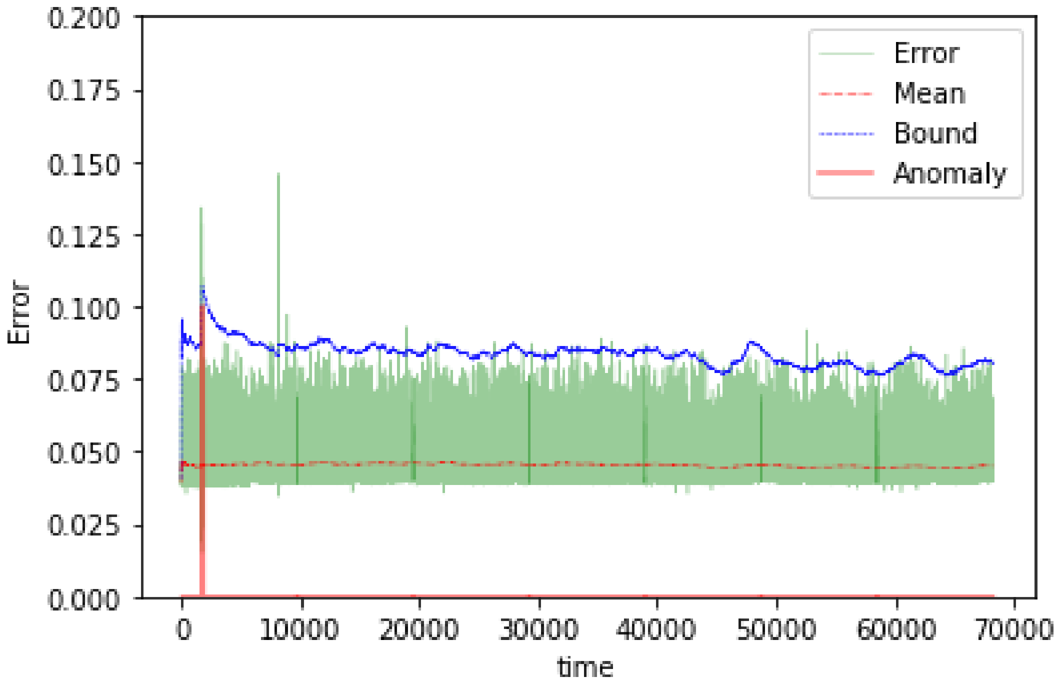



```
The mean error for gru3_20_disk_IO_start_ is 0.02557159436569083 for length 68279
Testing on Avg. load data.
```



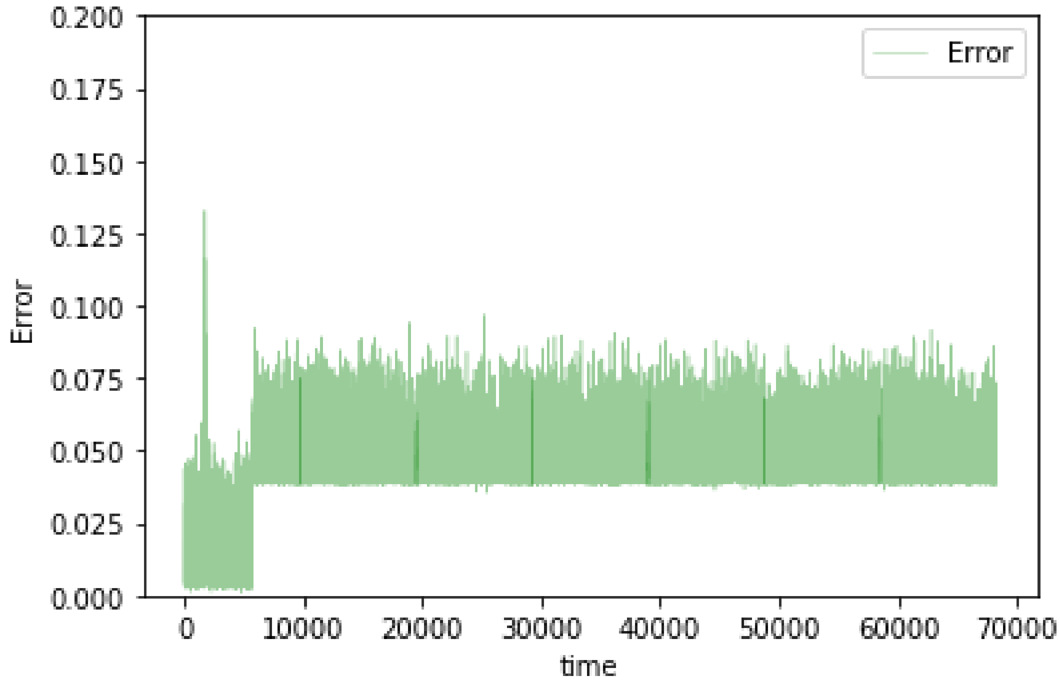


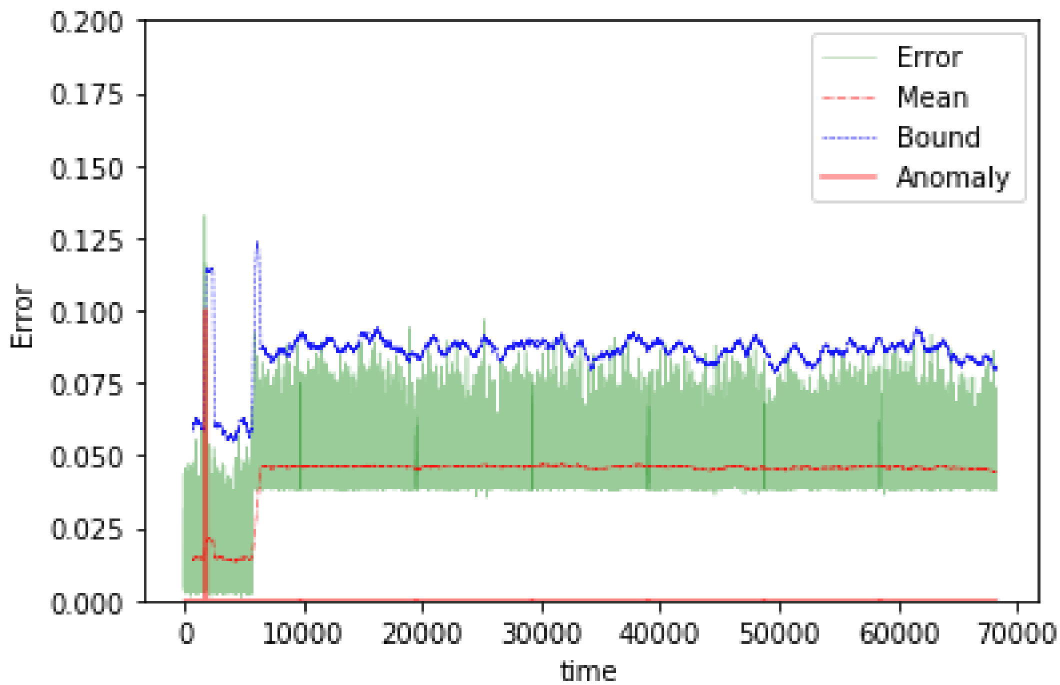
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

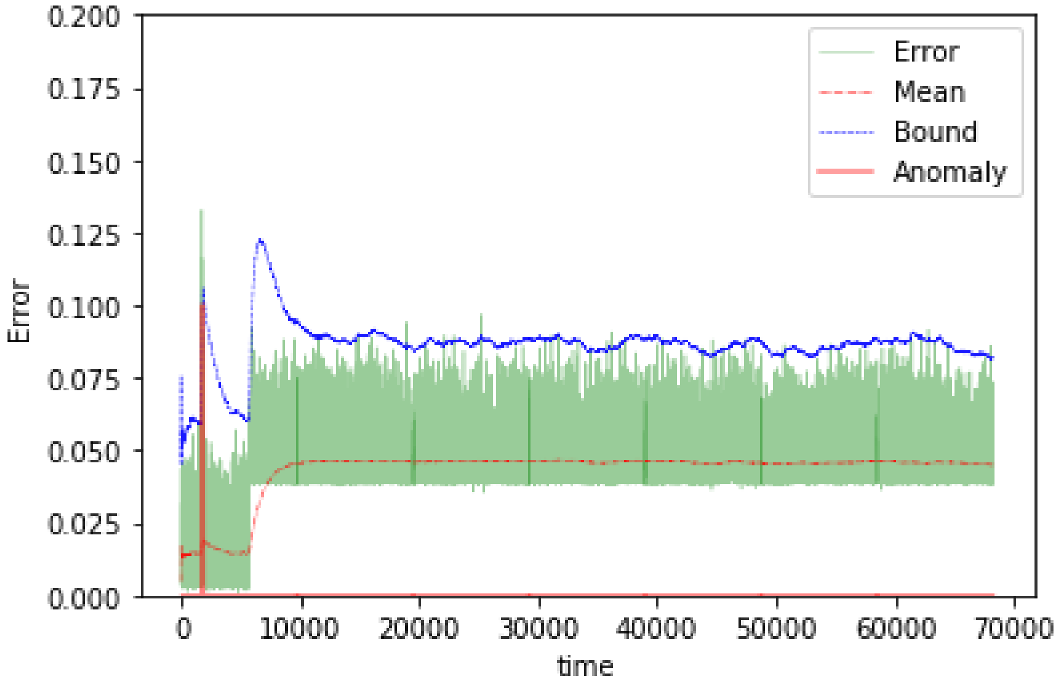


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

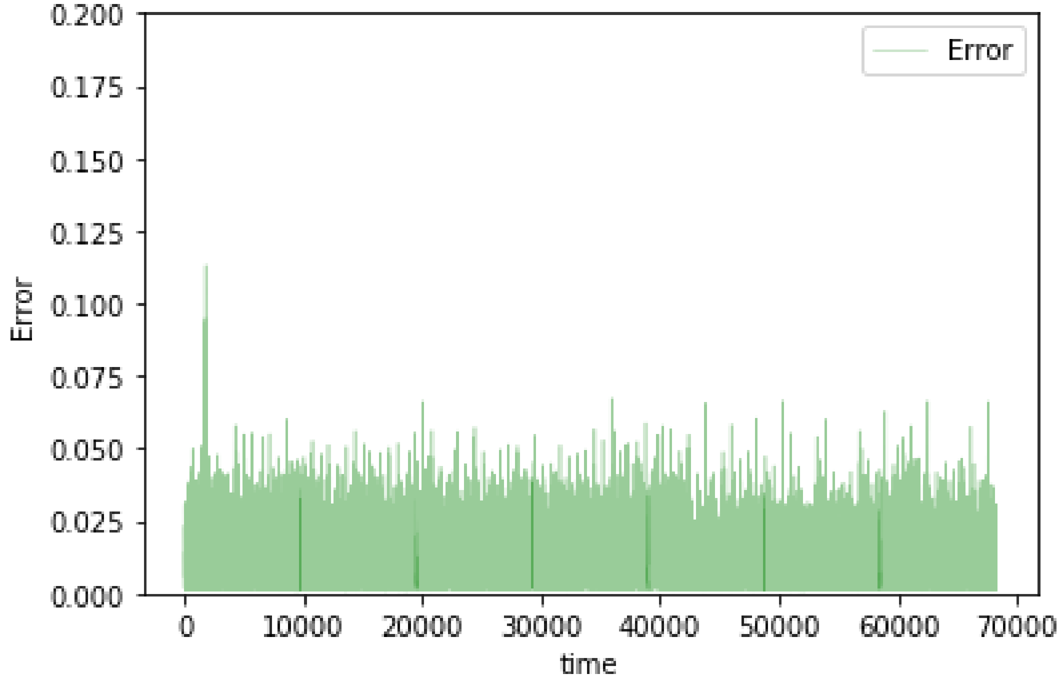
```
The mean error for gru3_20_avg_load_ is 0.04534195943203466 for length 68279
Testing on app change early data.
```

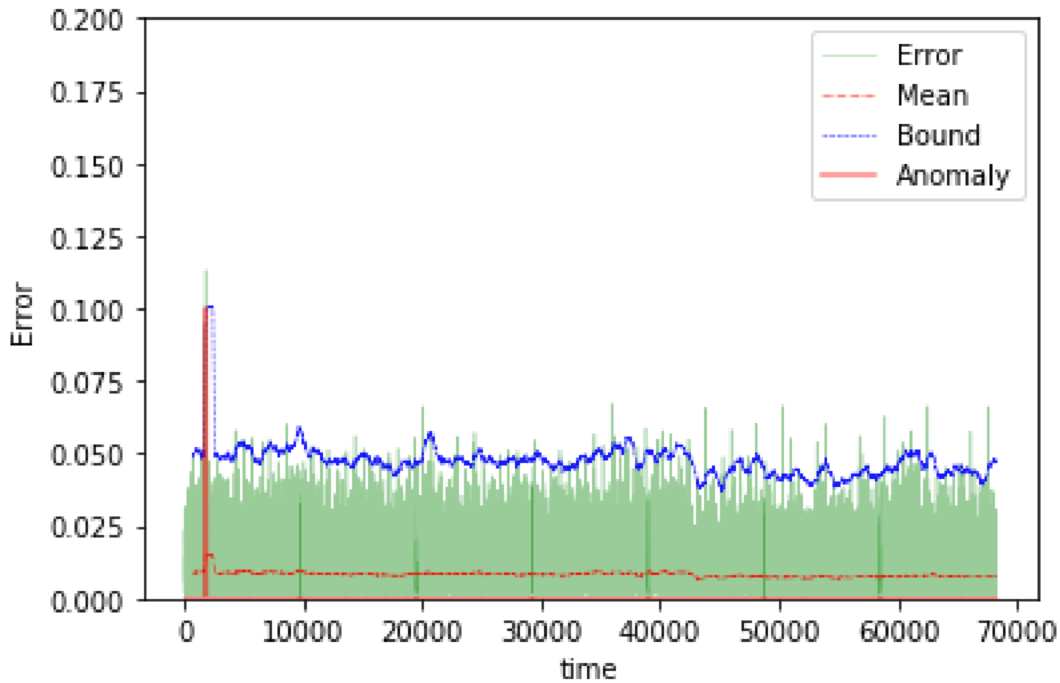


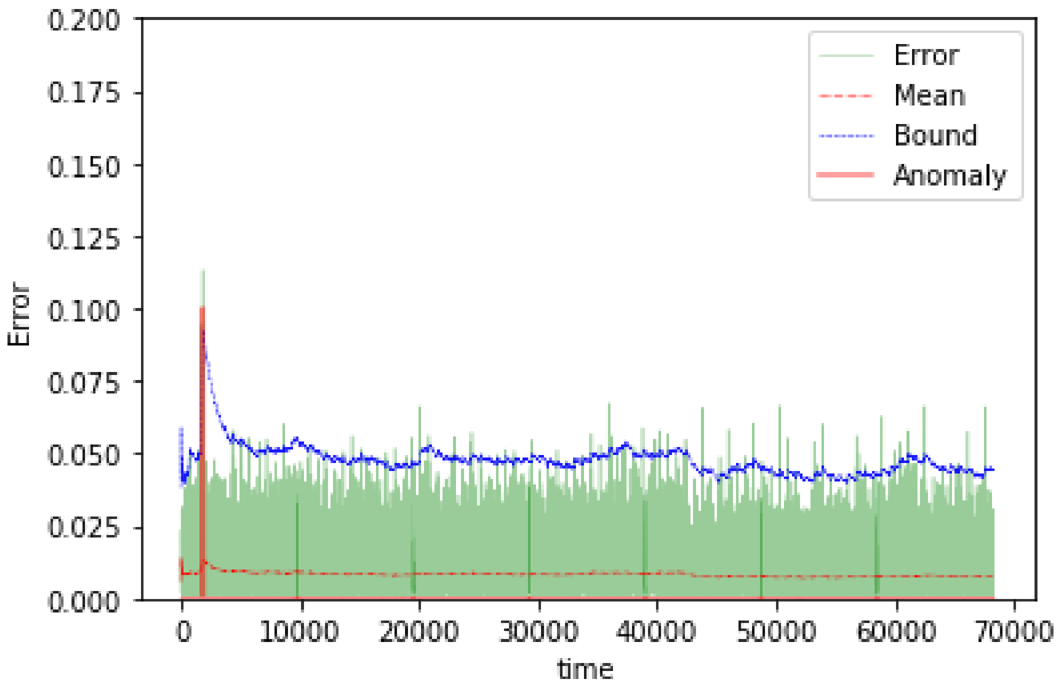





```
The mean error for gru3_20_app_change_early_ is 0.043406805525565084 for length 68279
Testing on Normal data.
```


Error
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

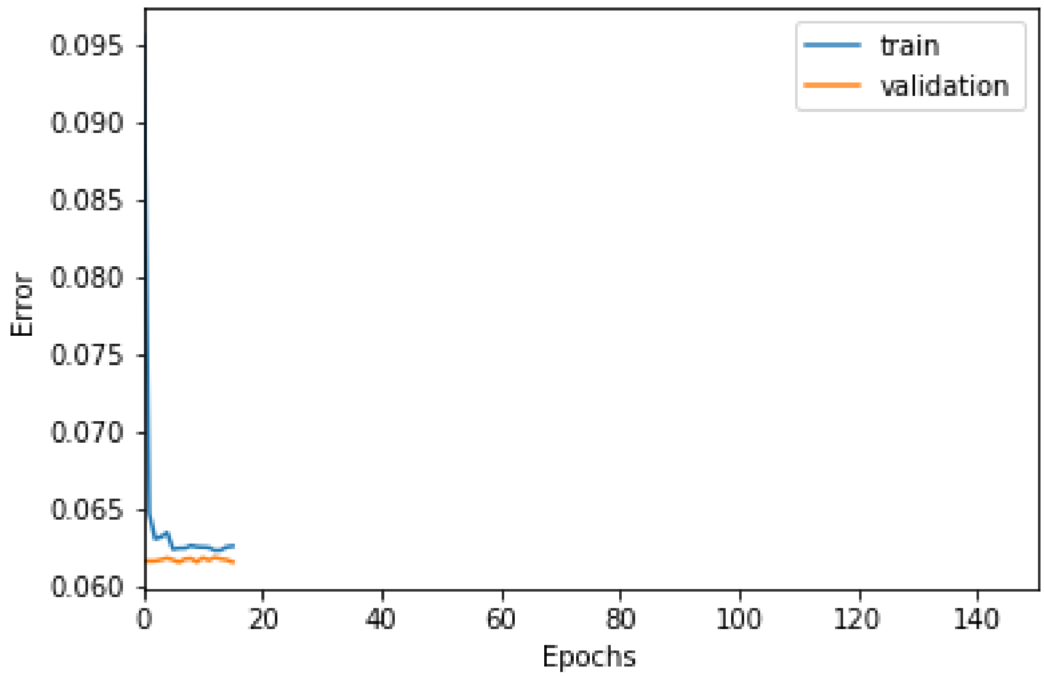

Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000



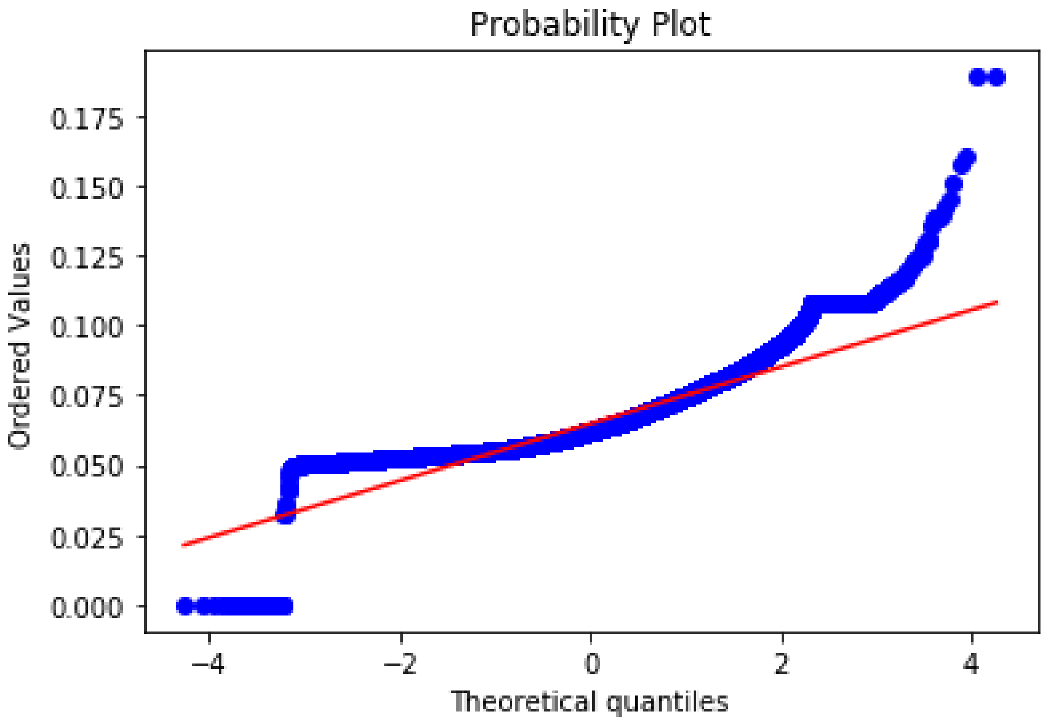

```
The mean error for gru3_20_normal_ is 0.00826633060353806 for length 68279
====================
```

50 steps

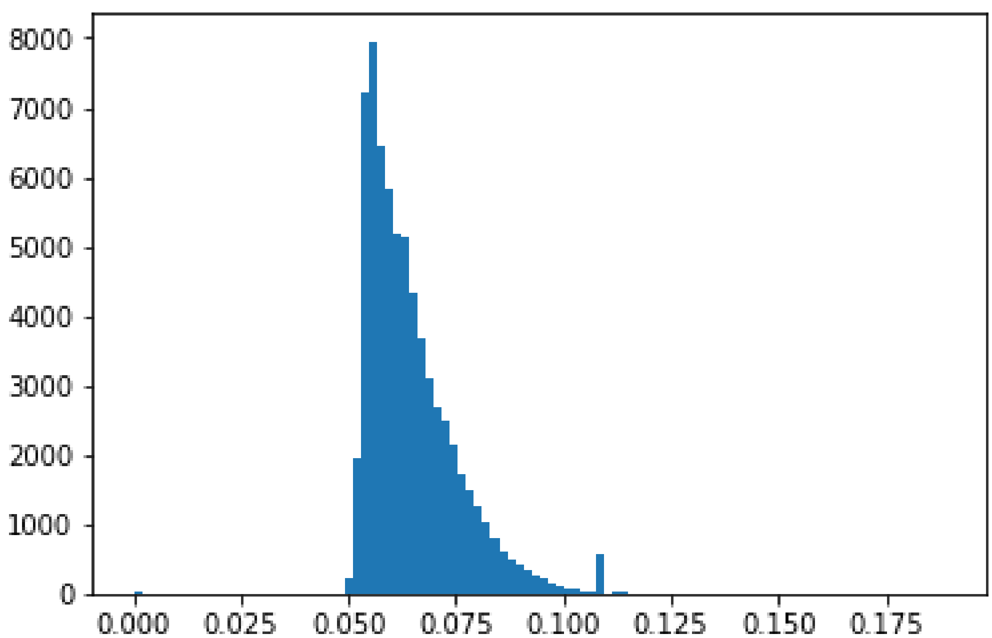
```
In [223]: TIMESTEPS = 50
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru3_50"

In [224]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(10, activation='relu', return_sequences=True)(hidden)
          hidden = GRU(10, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [225]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [226]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```



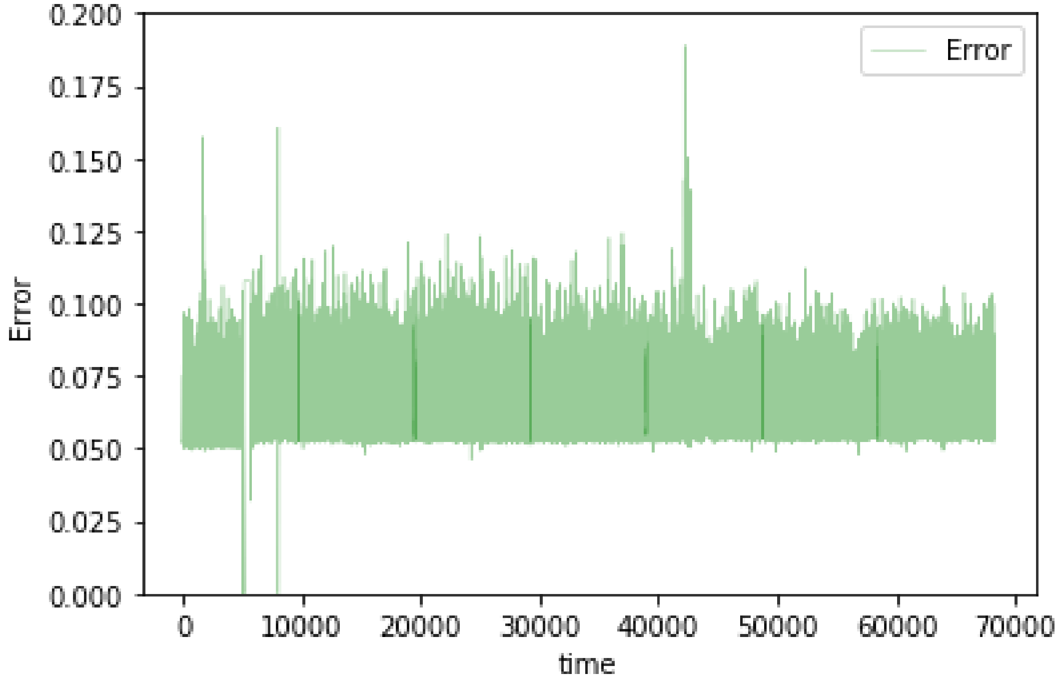

```
Training loss for final epoch is 0.06253467321023345
Validation loss for final epoch is 0.06151152671128511
---------- Beginning tests for gru3_50 ----------
Testing on Disk IO begin data.
```



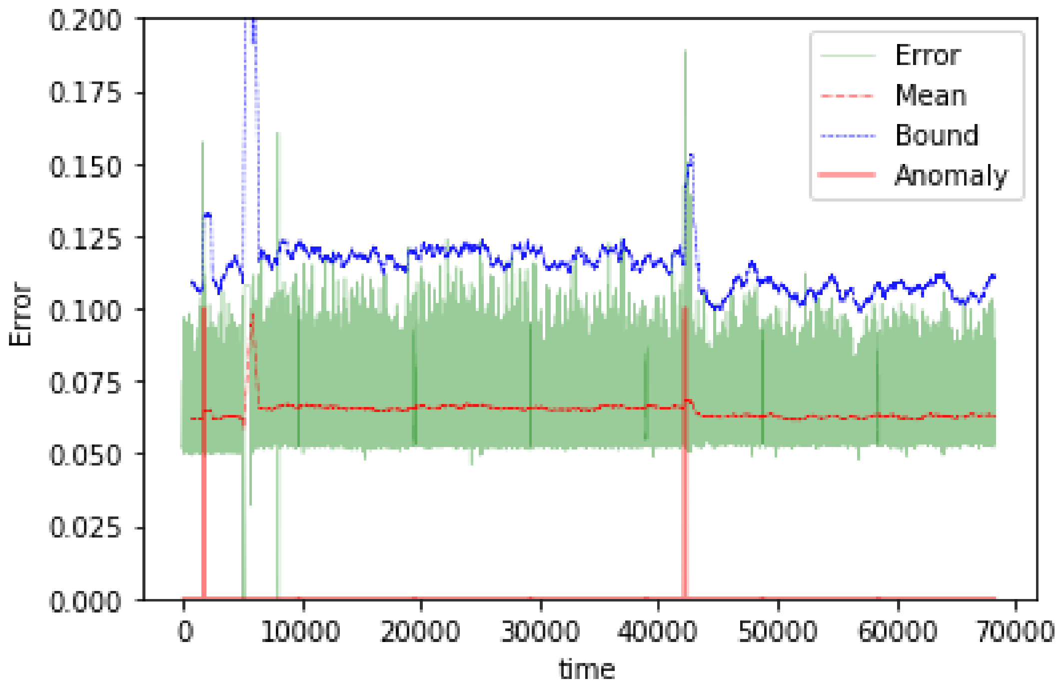
Probability Plot
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Ordered Values
−4
−2
0
2
4
Theoretical quantiles



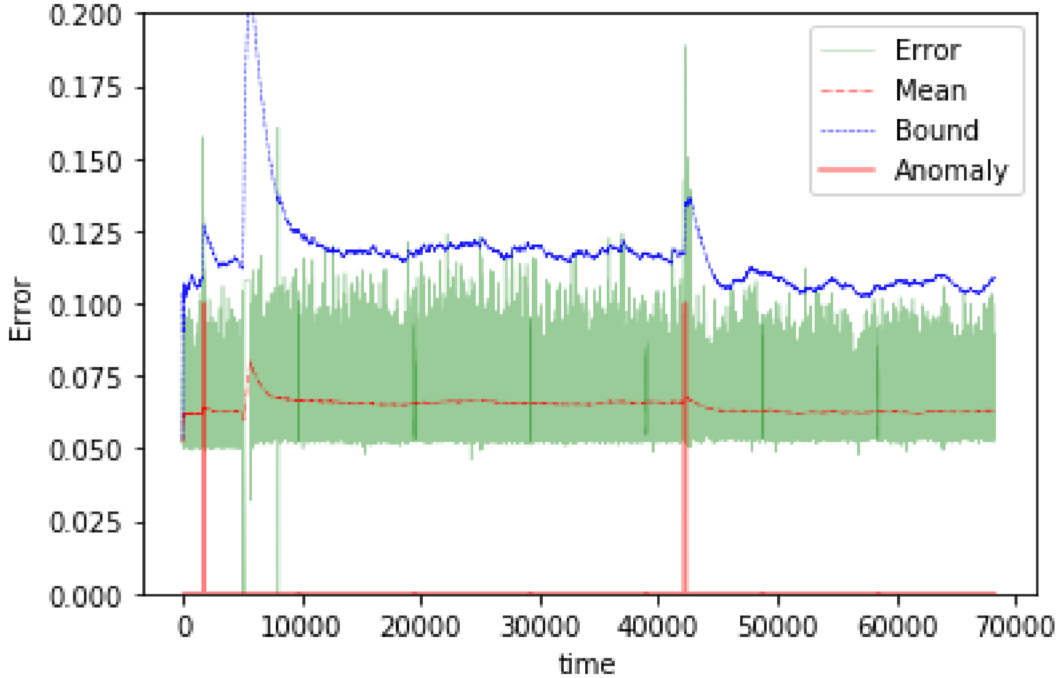
8000
7000
6000
5000
4000
3000
2000
1000
0
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175



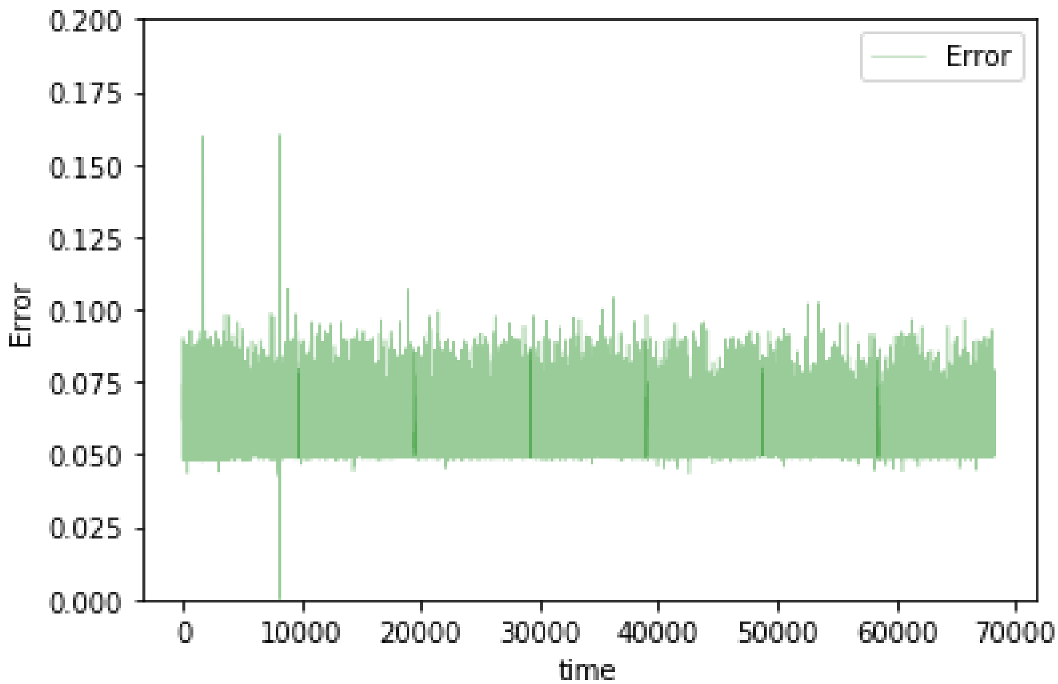
Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

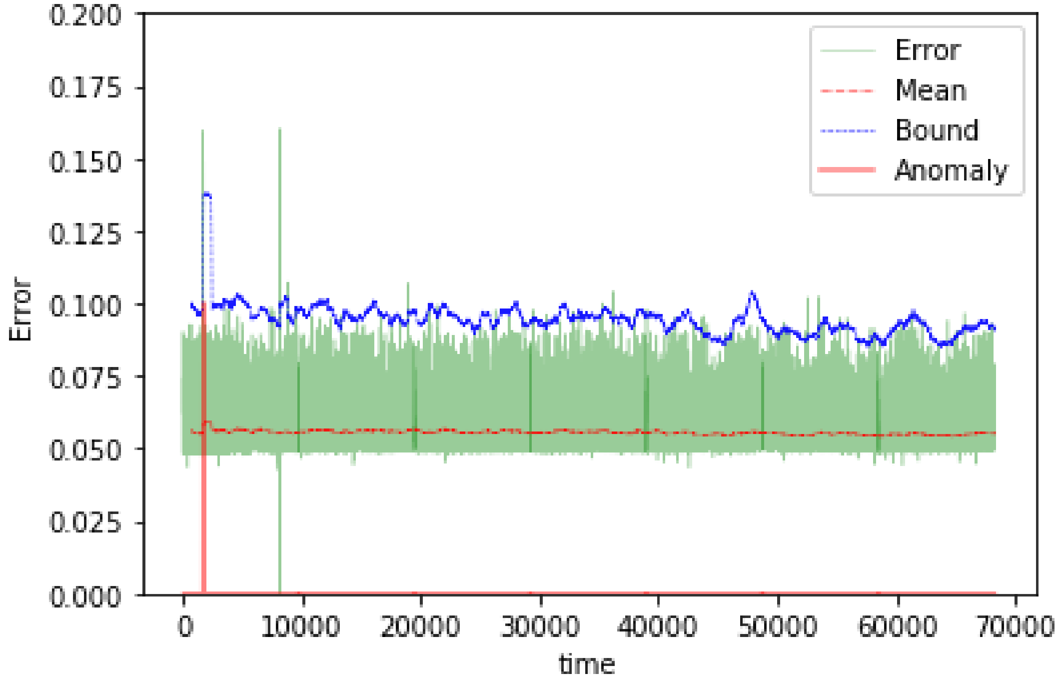


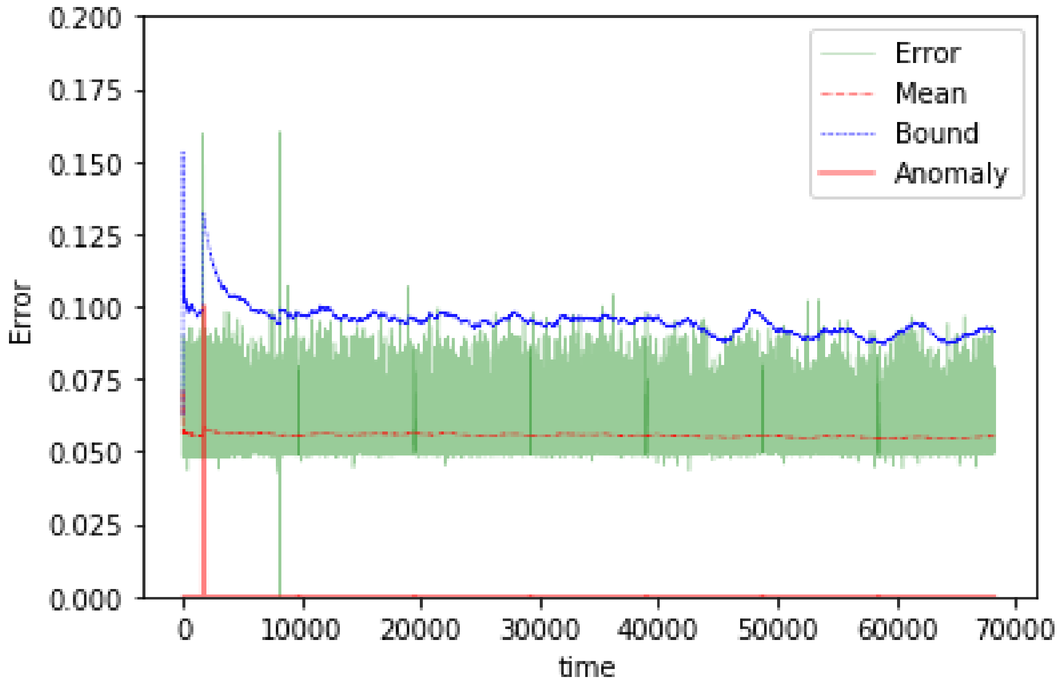
Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



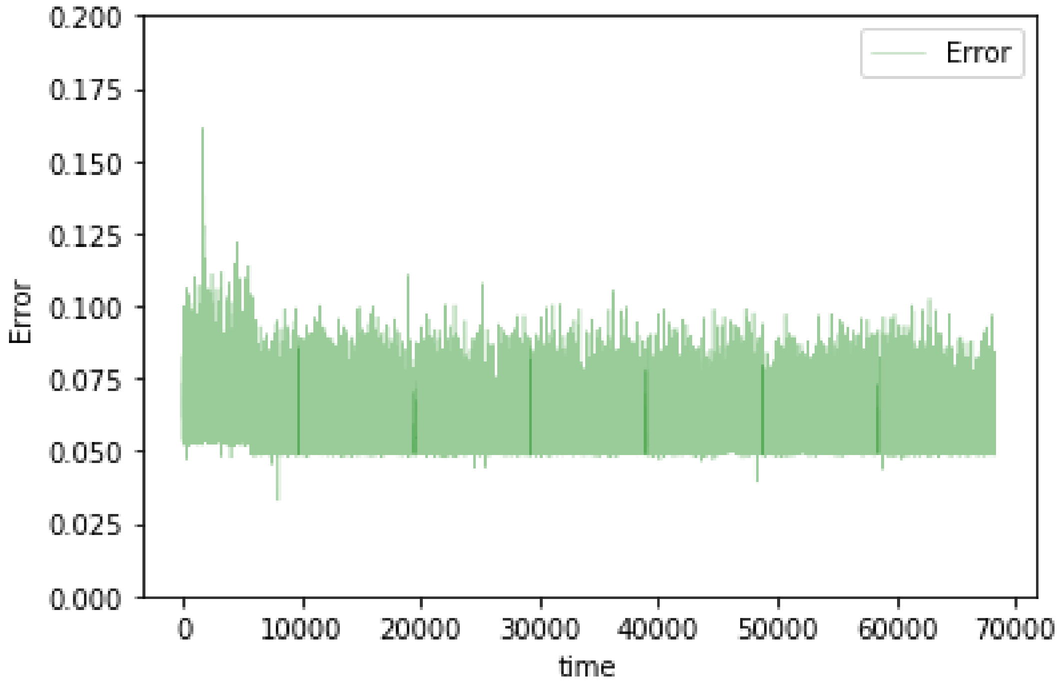
The mean error for gru3_50_disk_IO_start_ is 0.06469625327098957 for length 68249
Testing on Avg. load data.

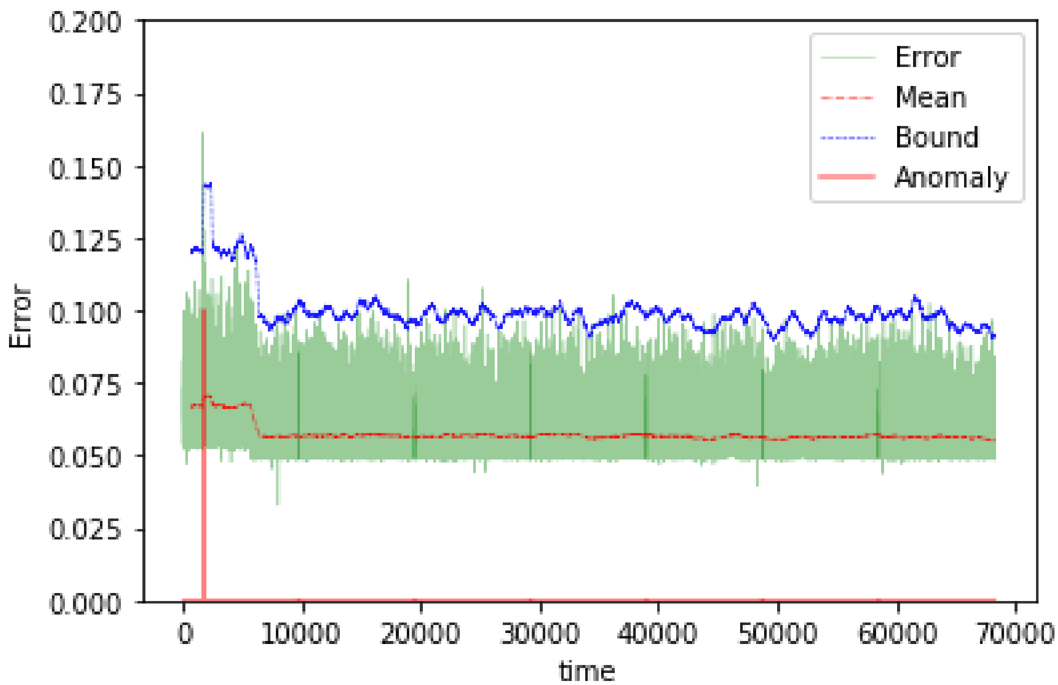


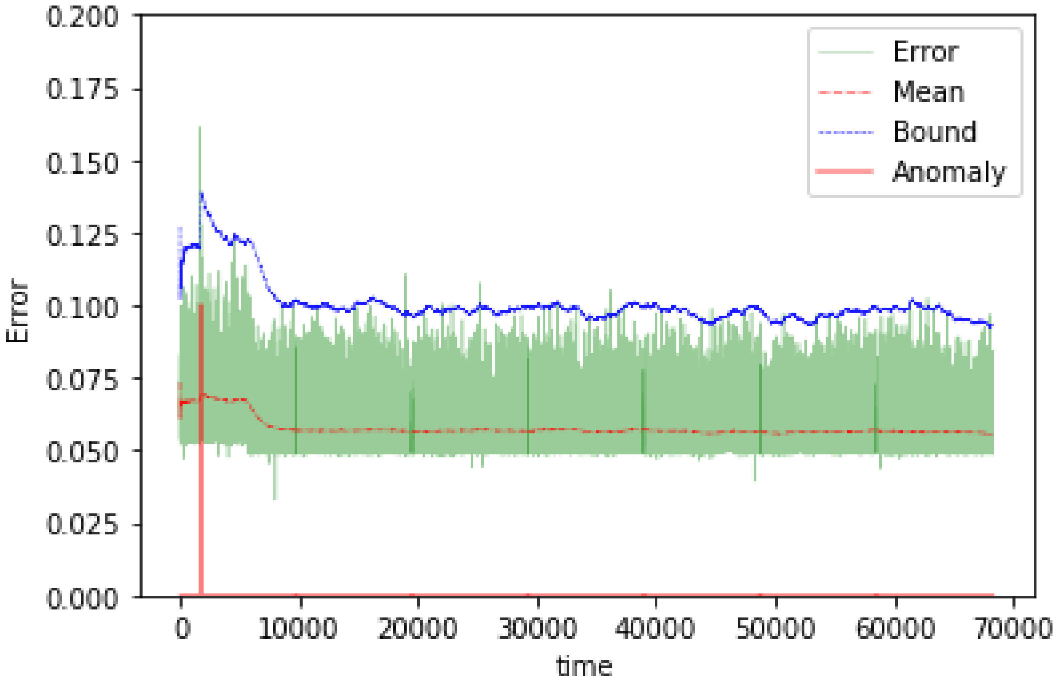

Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000


Error
Mean
Bound
Anomaly
Error
time
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000

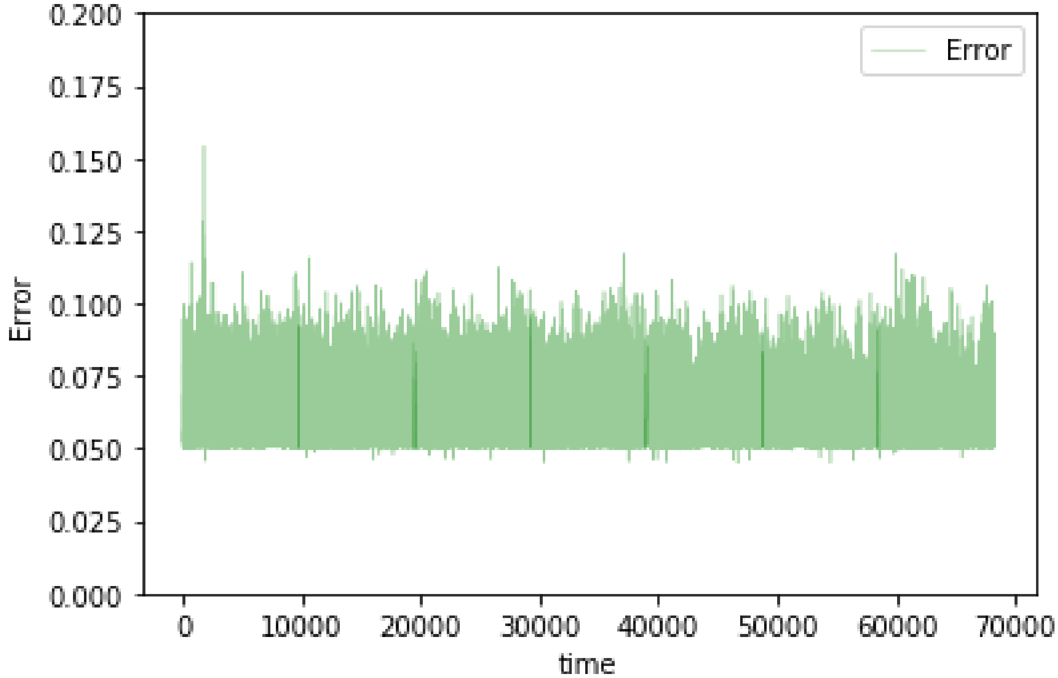
```
The mean error for gru3_50_avg_load_ is 0.05554627695952979 for length 68249
Testing on app change early data.
```

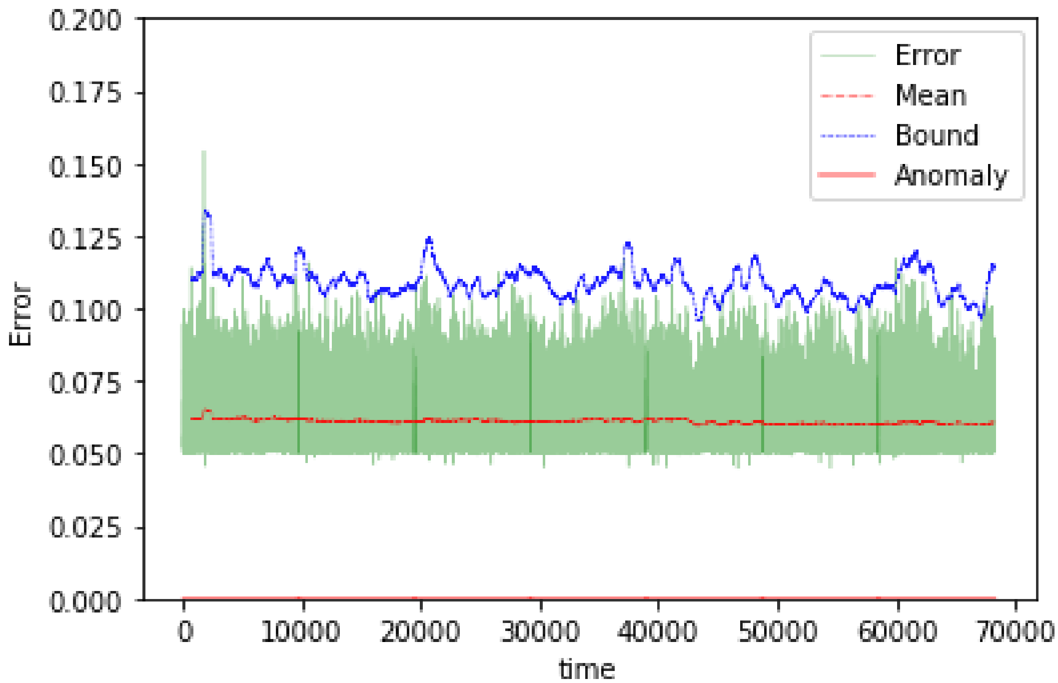


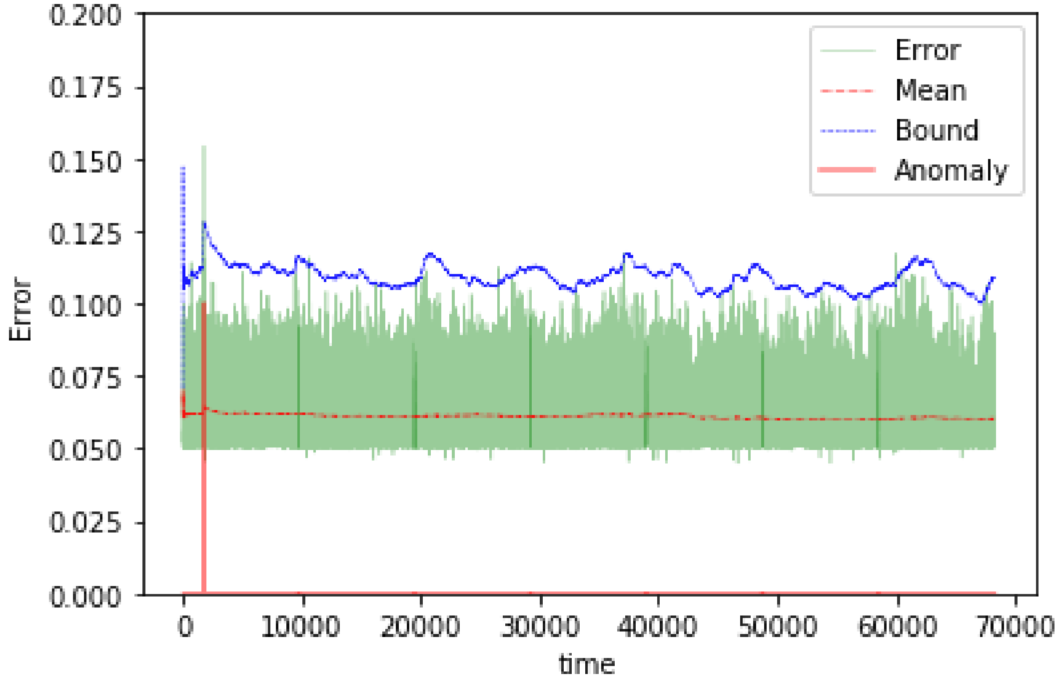





```
The mean error for gru3_50_app_change_early_ is 0.05747549622224253 for length 68249
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

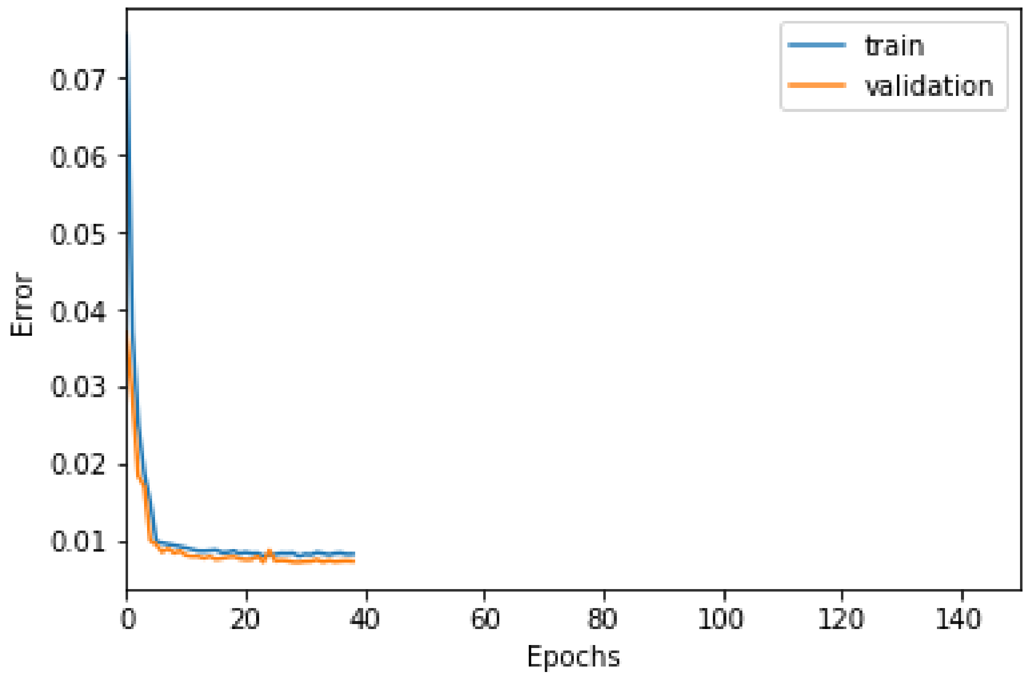

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



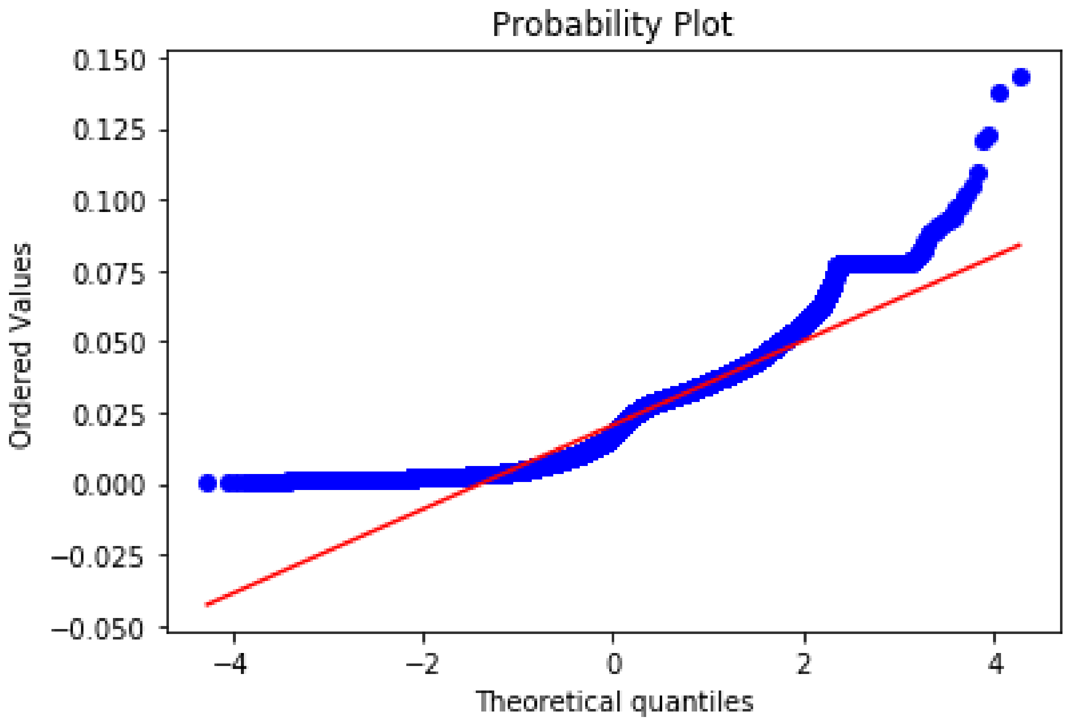

```
The mean error for gru3_50_normal_ is 0.06102623760382589 for length 68249
====================
```

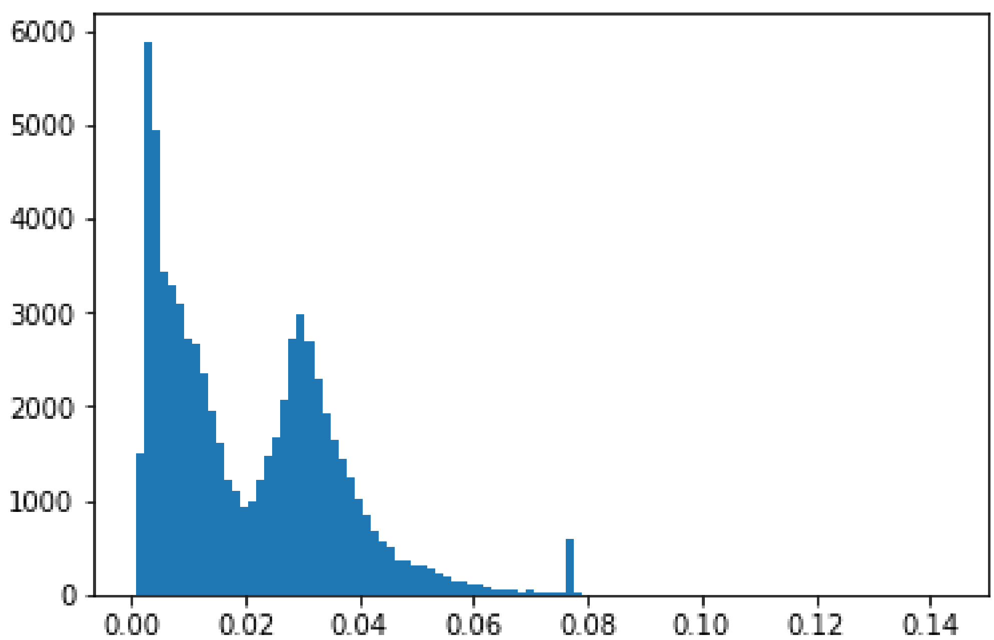
100 steps

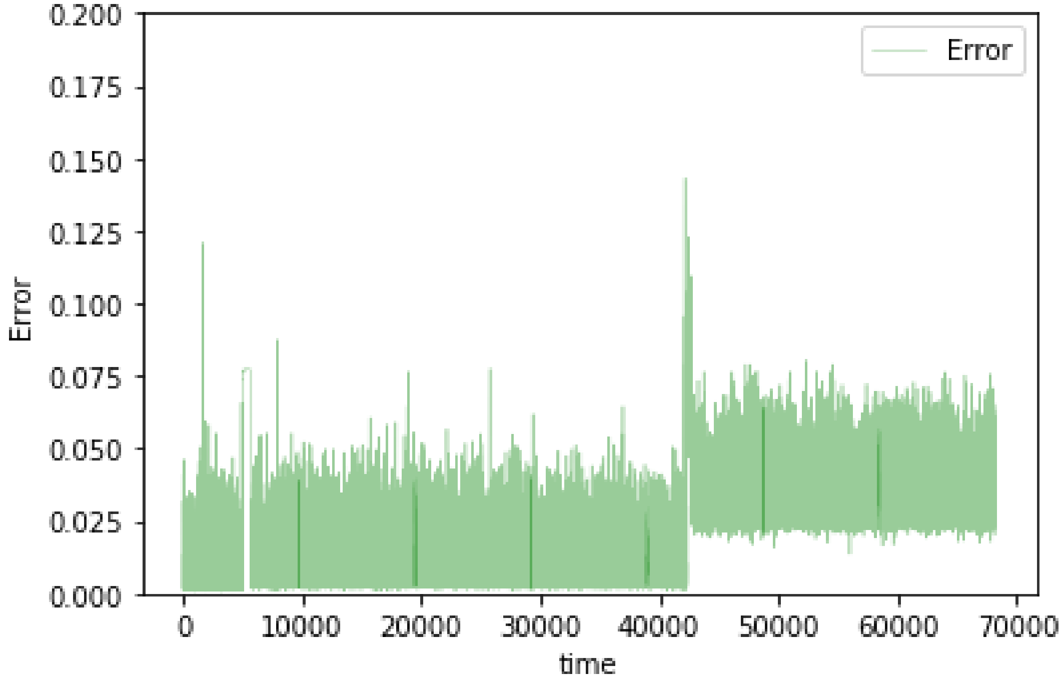
```
In [227]: TIMESTEPS = 100
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru3_100"

In [228]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(10, activation='relu', return_sequences=True)(hidden)
          hidden = GRU(10, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [229]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [230]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

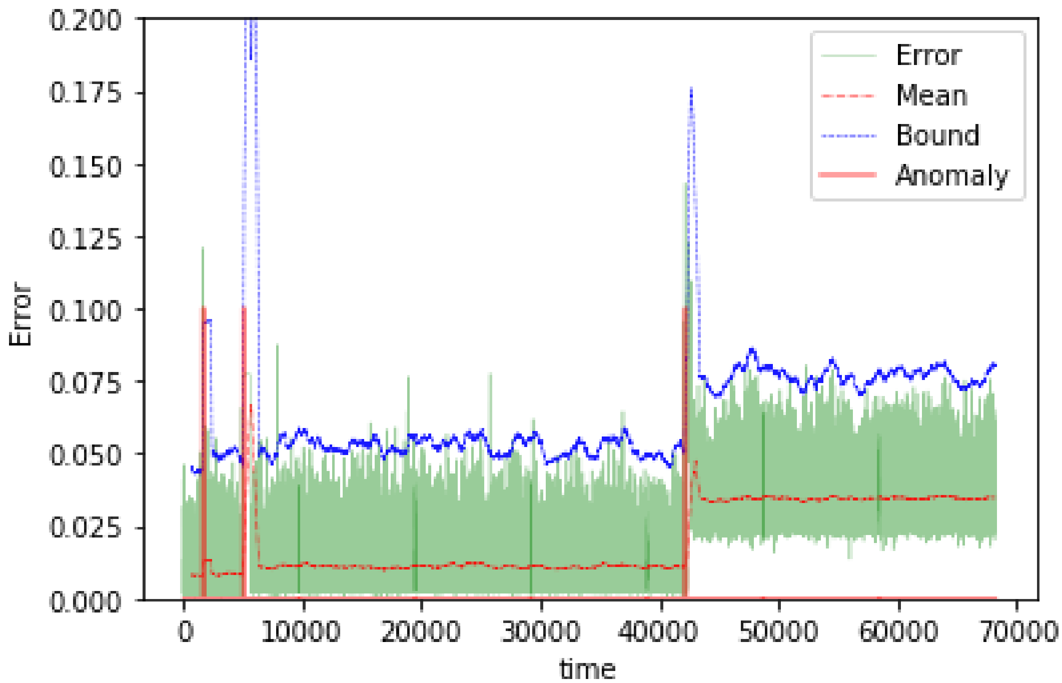


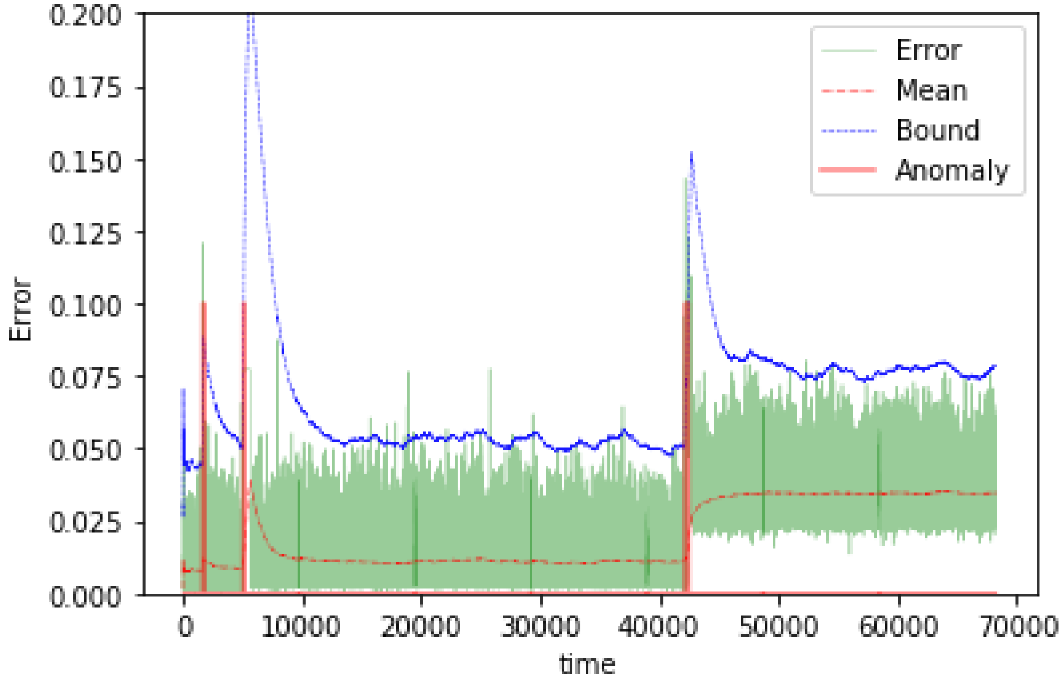

```
Training loss for final epoch is 0.008317155483306238
Validation loss for final epoch is 0.007386366069898941
---------- Beginning tests for gru3_100 ----------
Testing on Disk IO begin data.
```


Probability Plot
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
Ordered Values
−4
−2
0
2
4
Theoretical quantiles

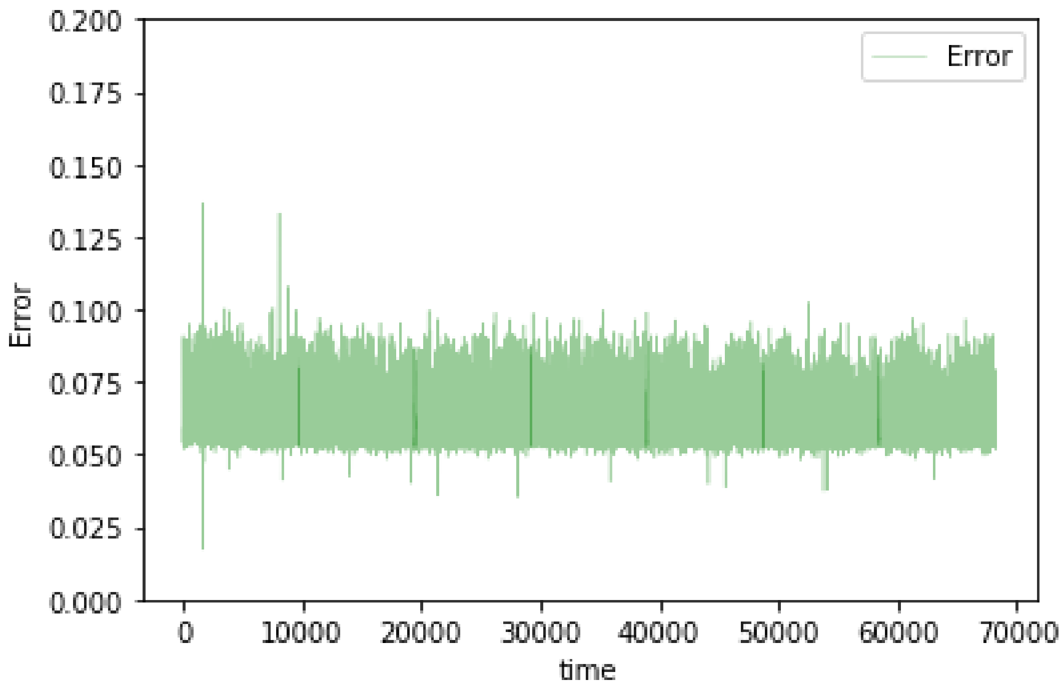

6000
5000
4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

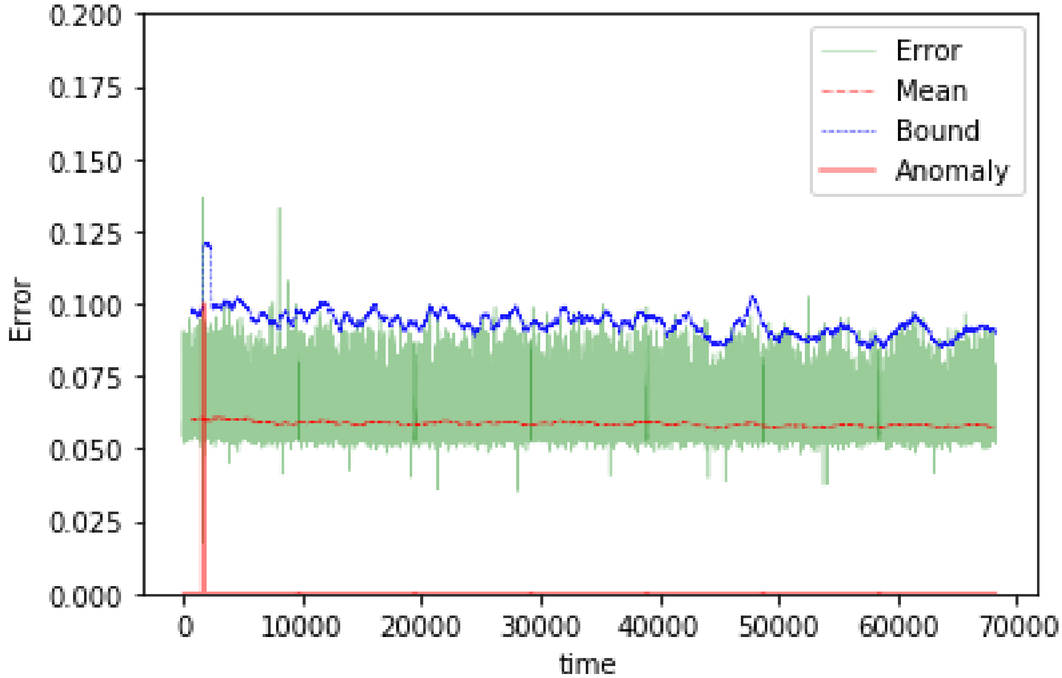

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

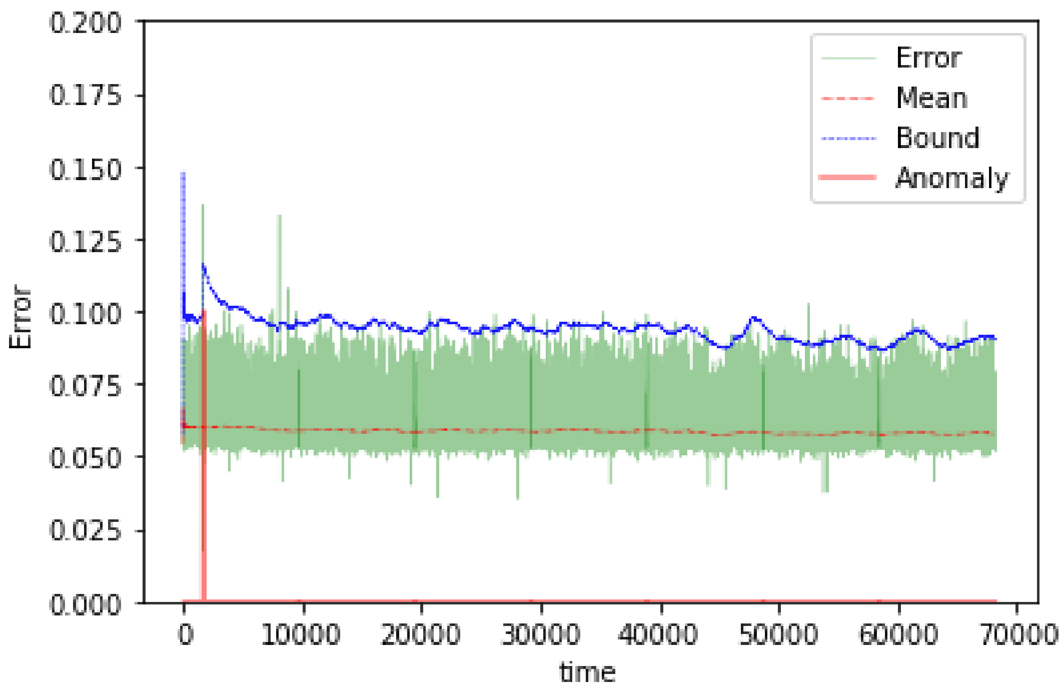

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



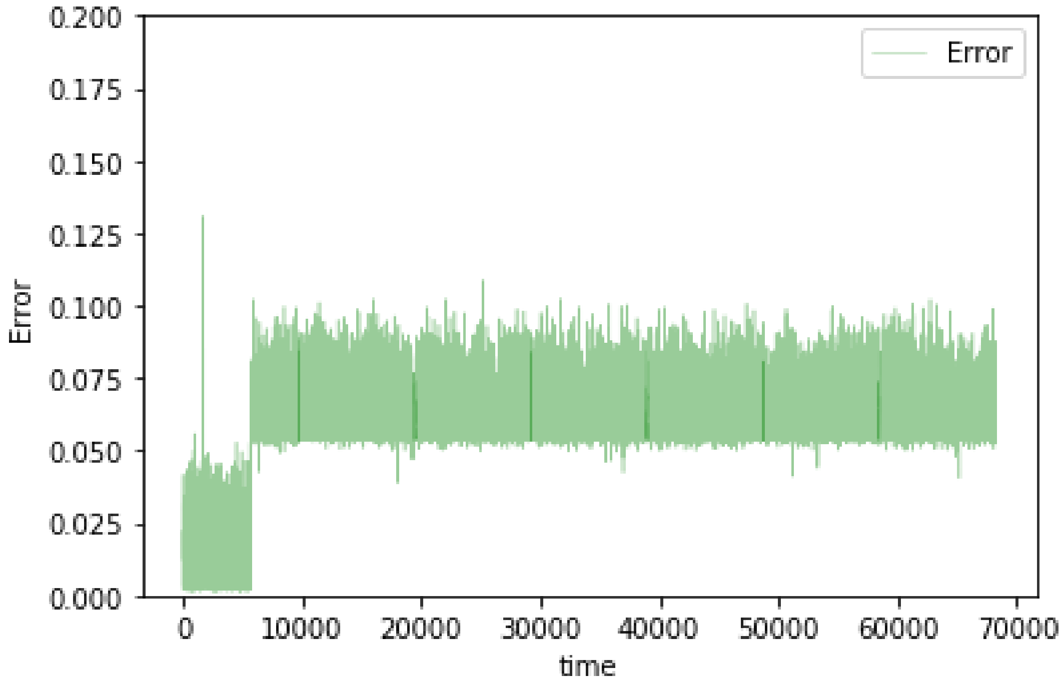

```
The mean error for gru3_100_disk_IO_start_ is 0.02052566829663445 for length 68199
Testing on Avg. load data.
```

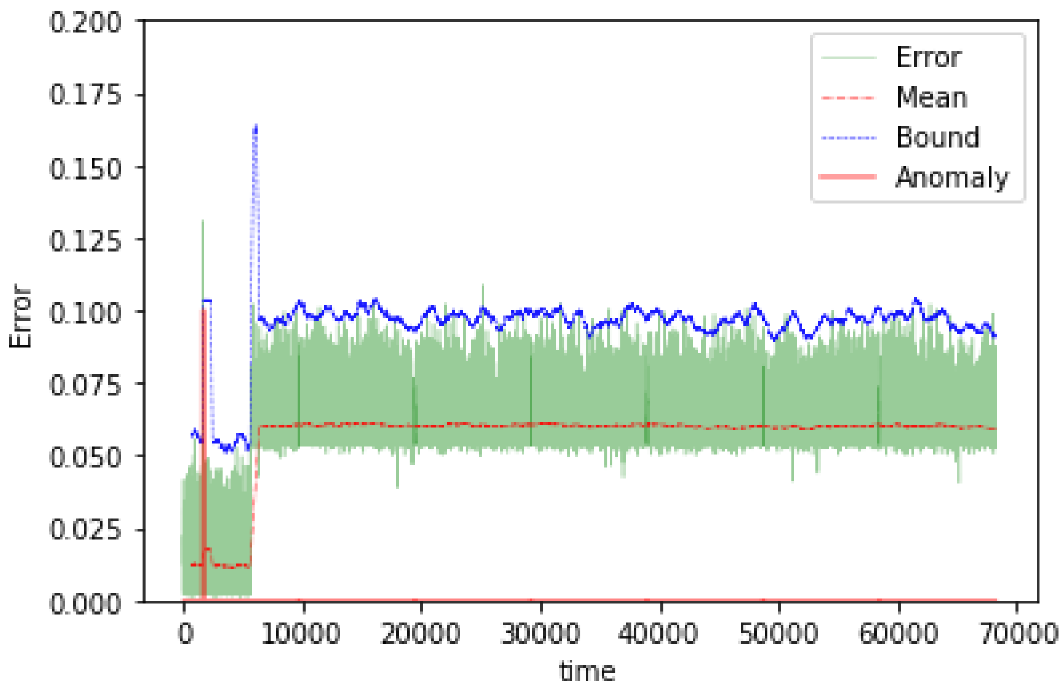


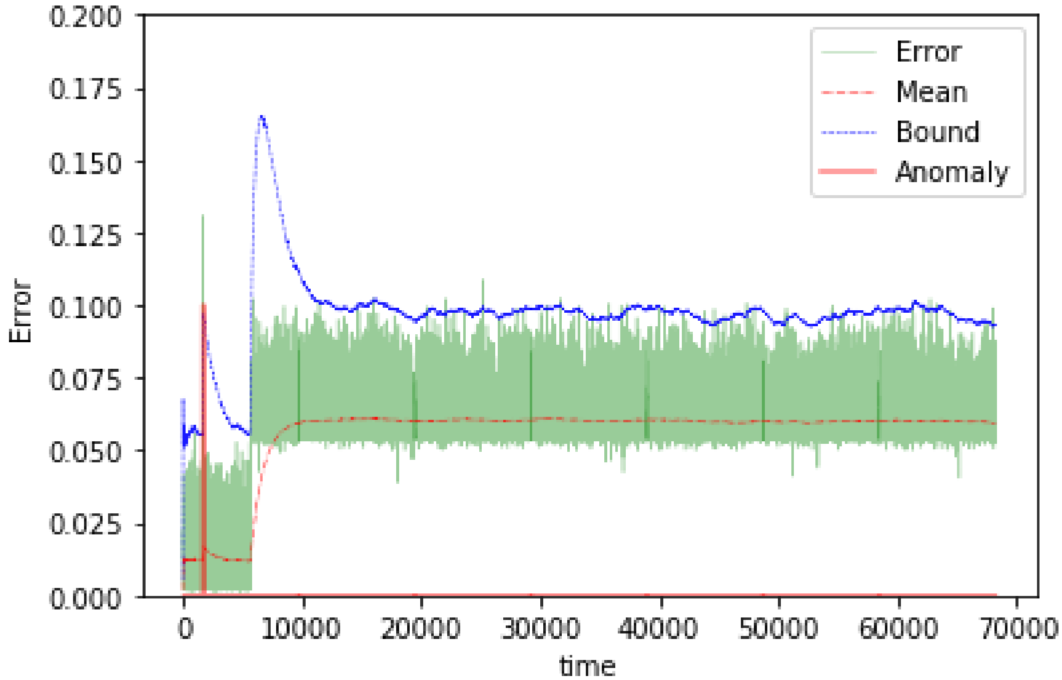

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly

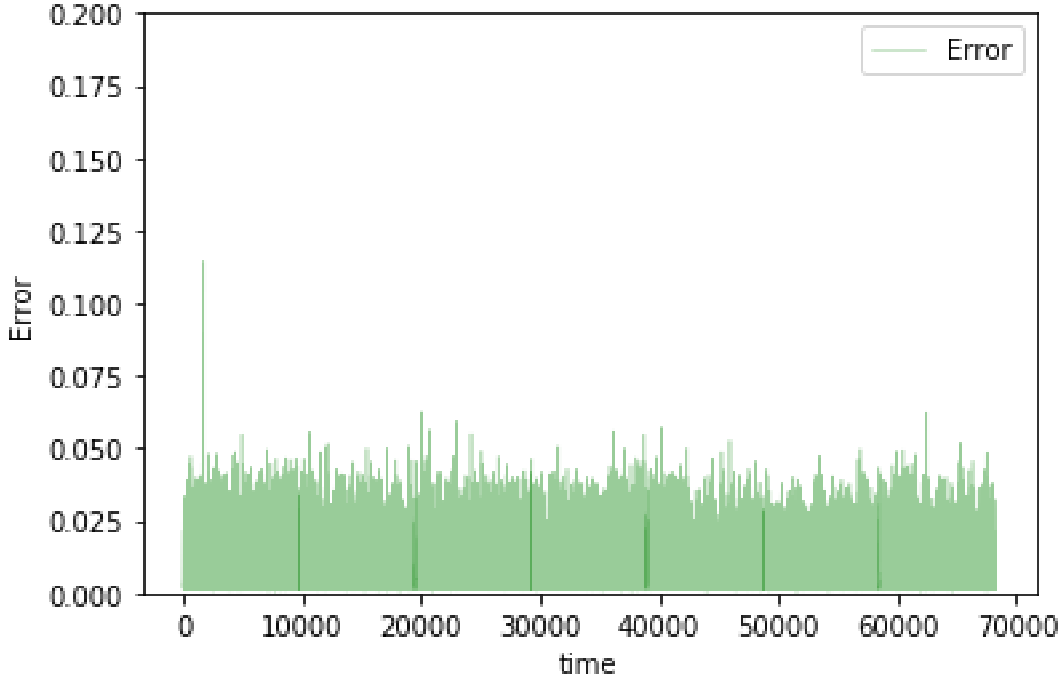
```
The mean error for gru3_100_avg_load_ is 0.058537951625704854 for length 68199
Testing on app change early data.
```

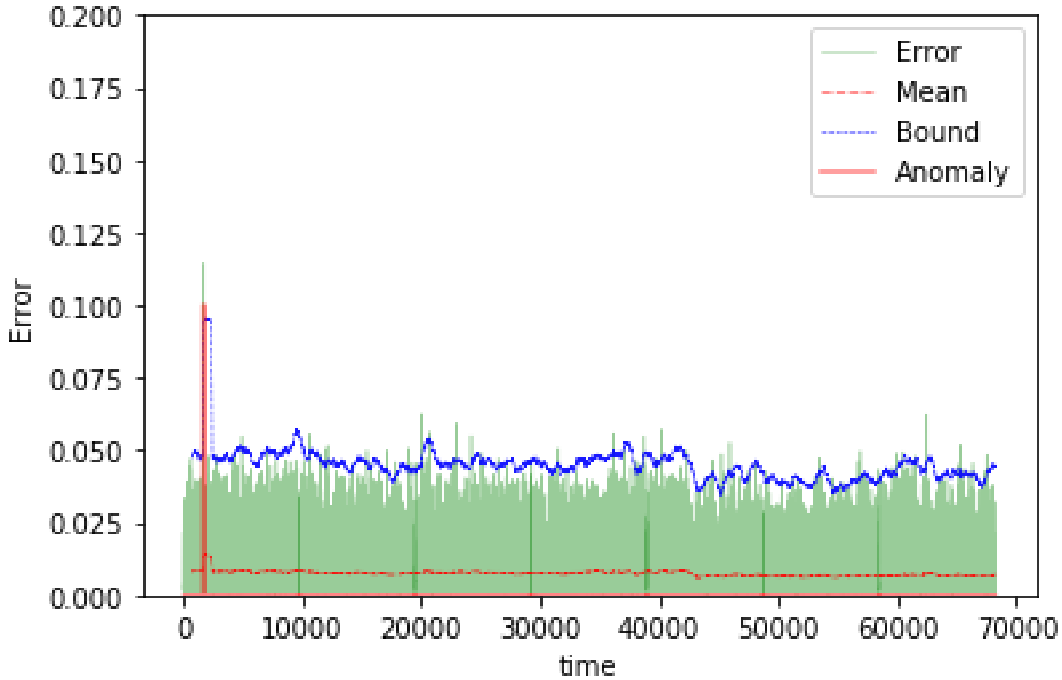


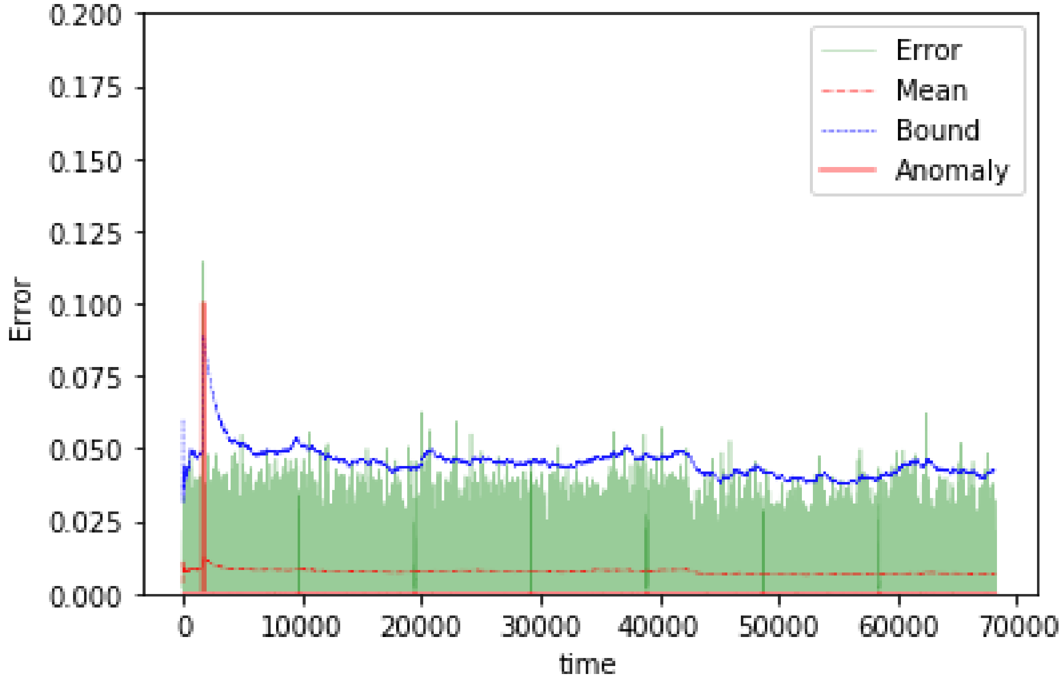





```
The mean error for gru3_100_app_change_early_ is 0.056218945638665356 for length 68199
Testing on Normal data.
```


Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time

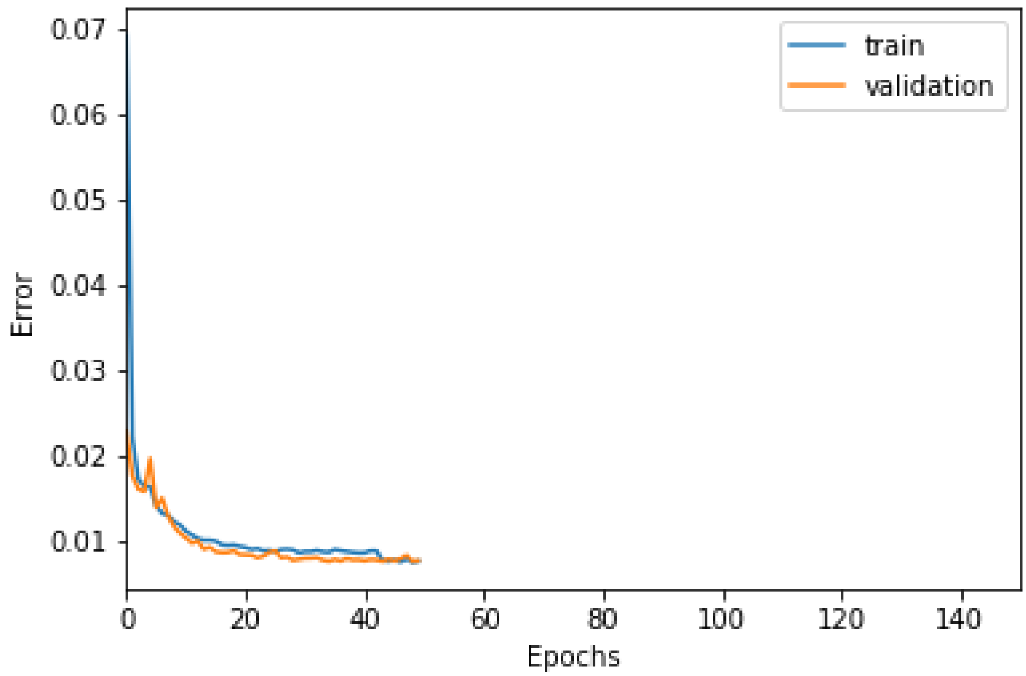

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time



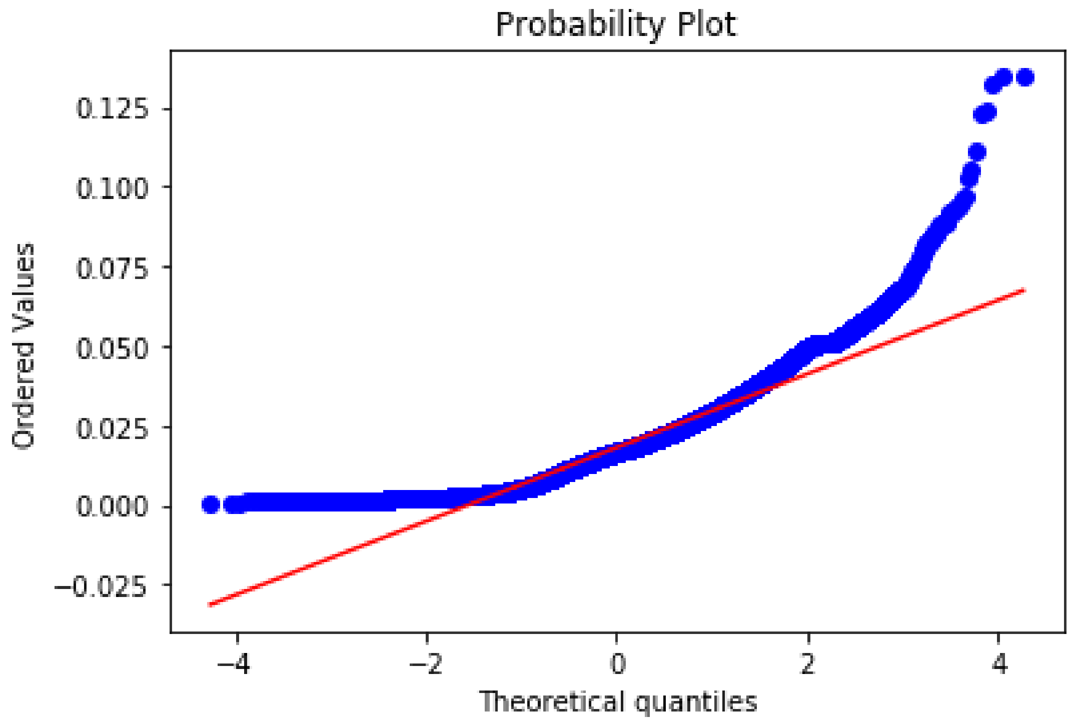

```
The mean error for gru3_100_normal_ is 0.007506866633528395 for length 68199
====================
```

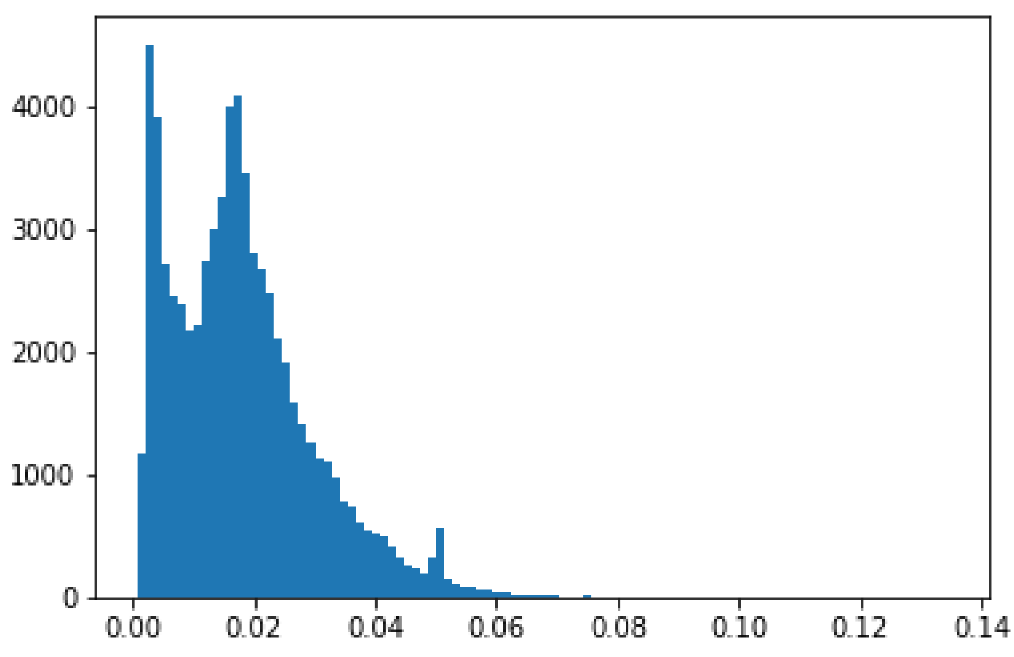
200 steps

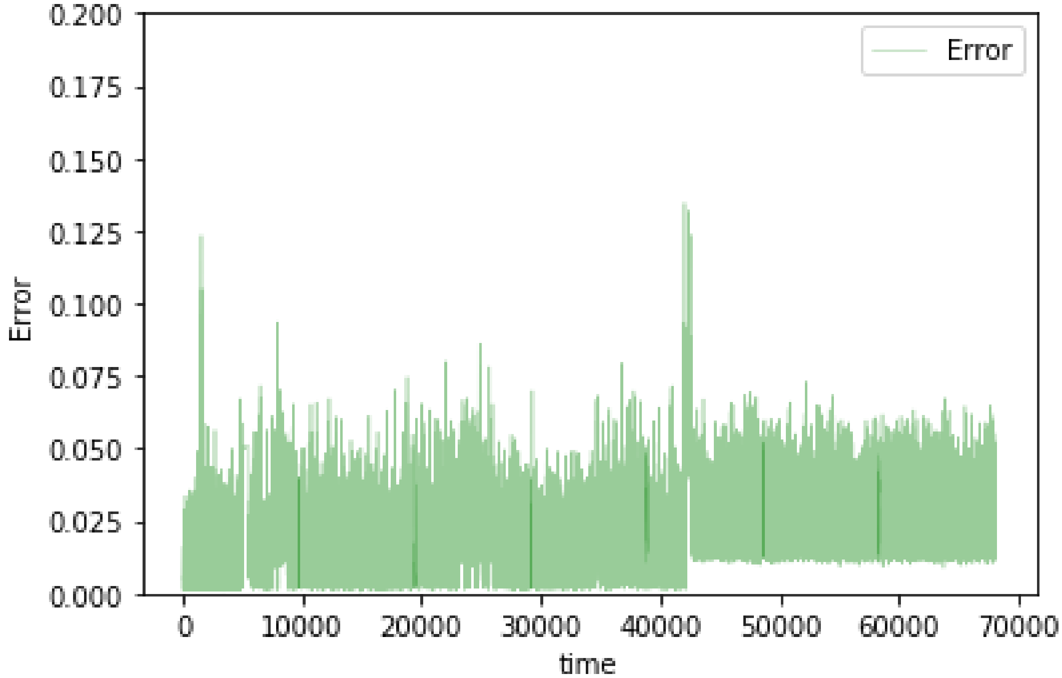
```
In [231]: TIMESTEPS = 200
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru3_200"

In [232]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(10, activation='relu', return_sequences=True)(hidden)
          hidden = GRU(10, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

In [233]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [234]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```

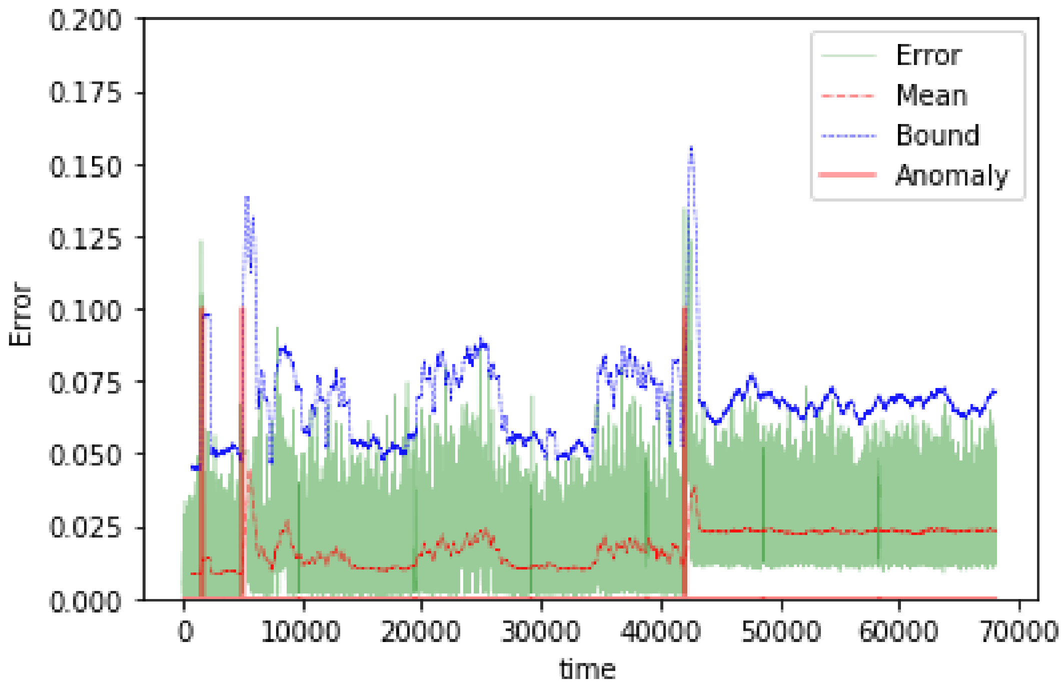


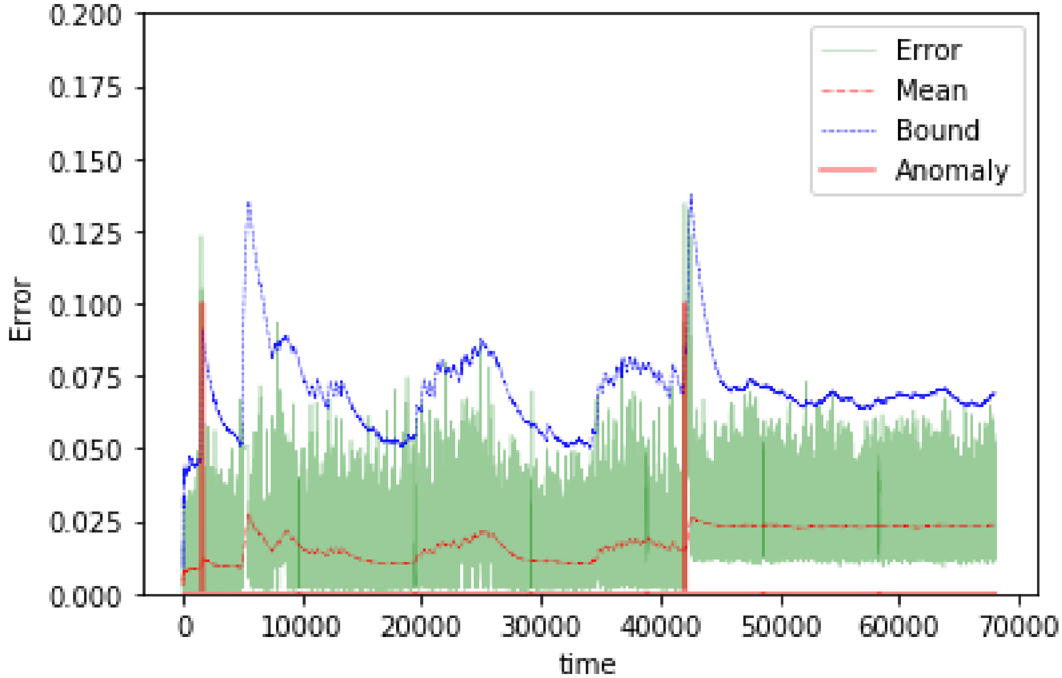

```
Training loss for final epoch is 0.007539048173814081
Validation loss for final epoch is 0.007592364177689888
---------- Beginning tests for gru3_200 ----------
Testing on Disk IO begin data.
```


Probability Plot
Ordered Values
Theoretical quantiles
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−4
−2
0
2
4

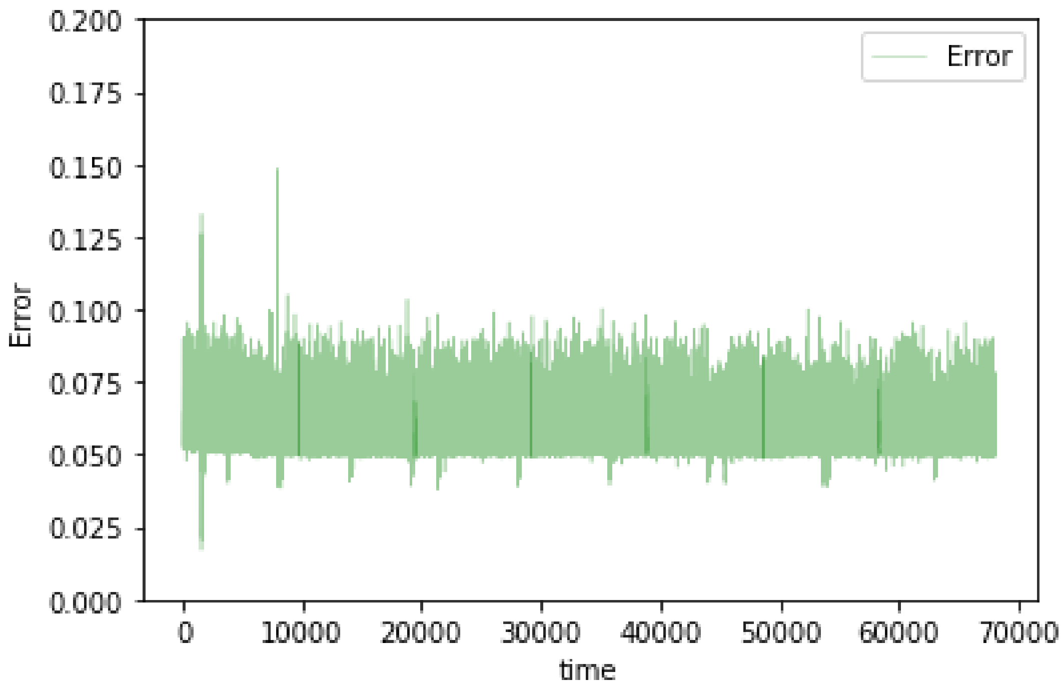

4000
3000
2000
1000
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

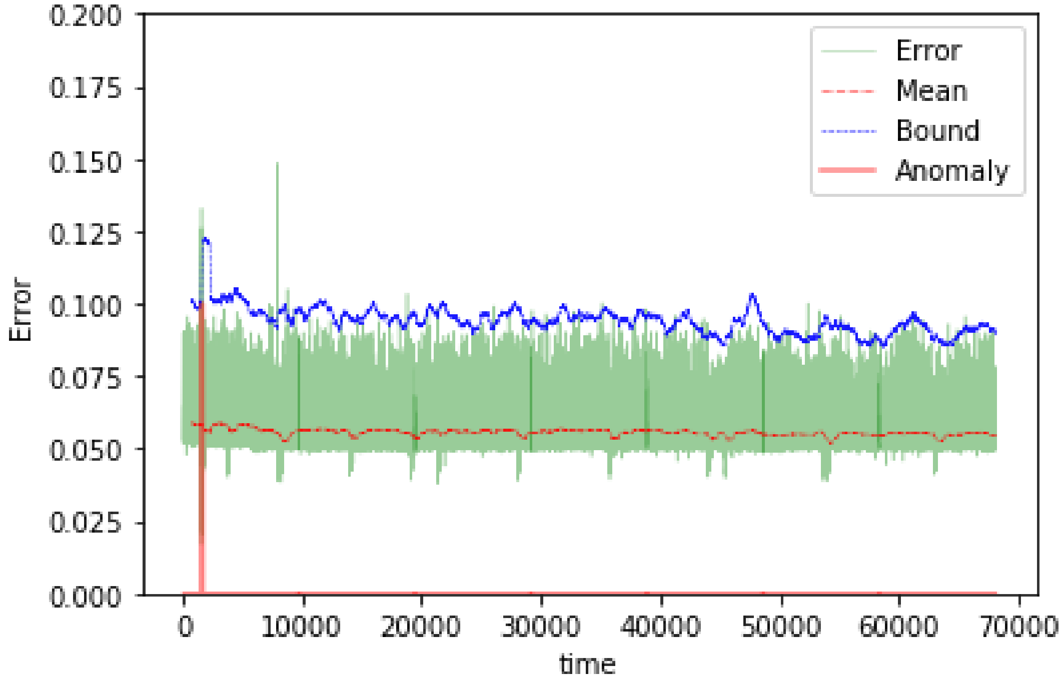

Error
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error

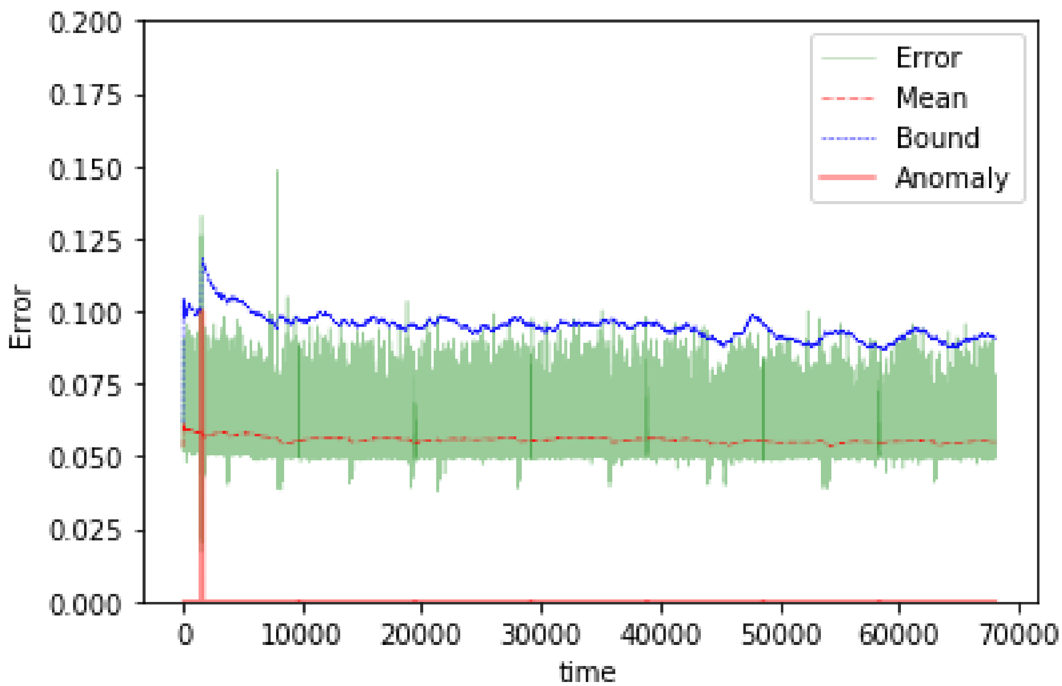

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10000
20000
30000
40000
50000
60000
70000
time
Error



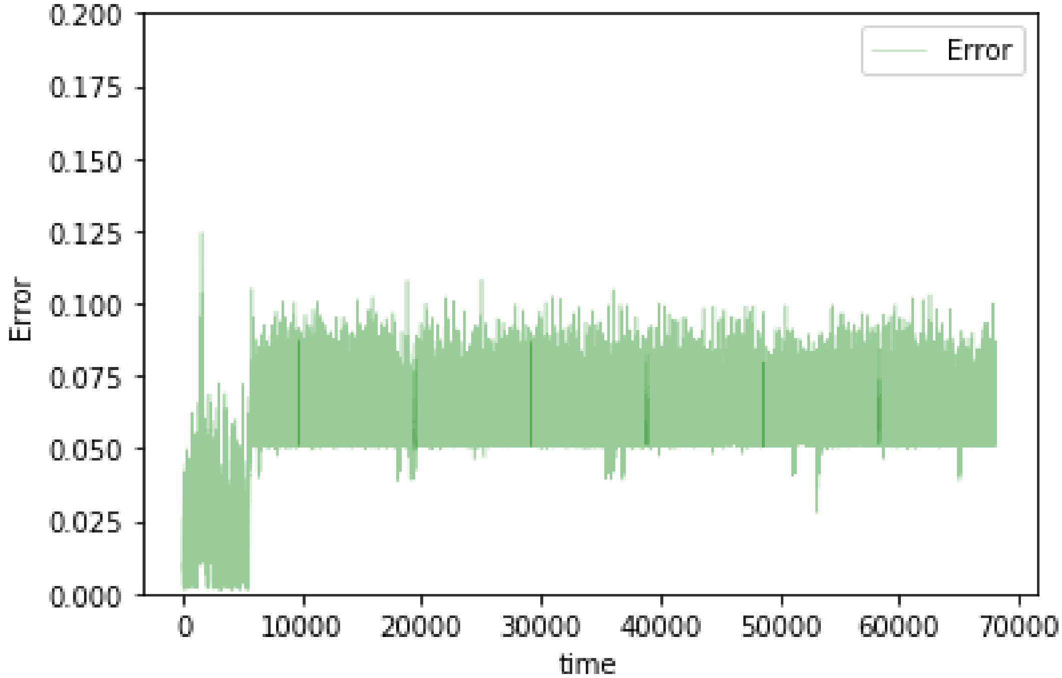

```
The mean error for gru3_200_disk_IO_start_ is 0.01796777453040097 for length 68099
Testing on Avg. load data.
```

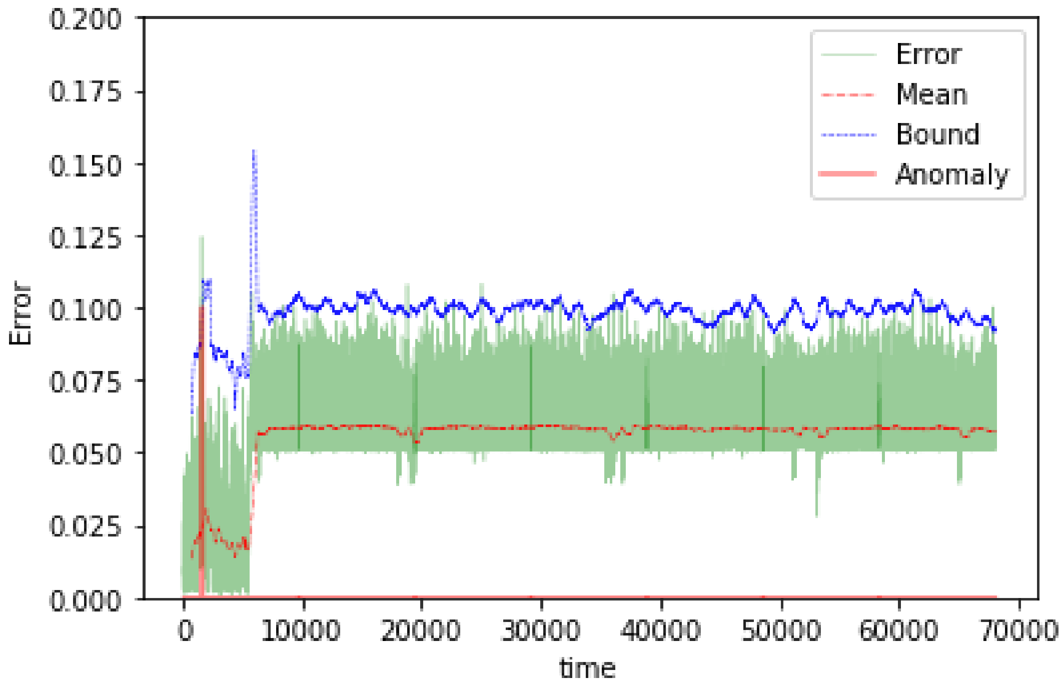


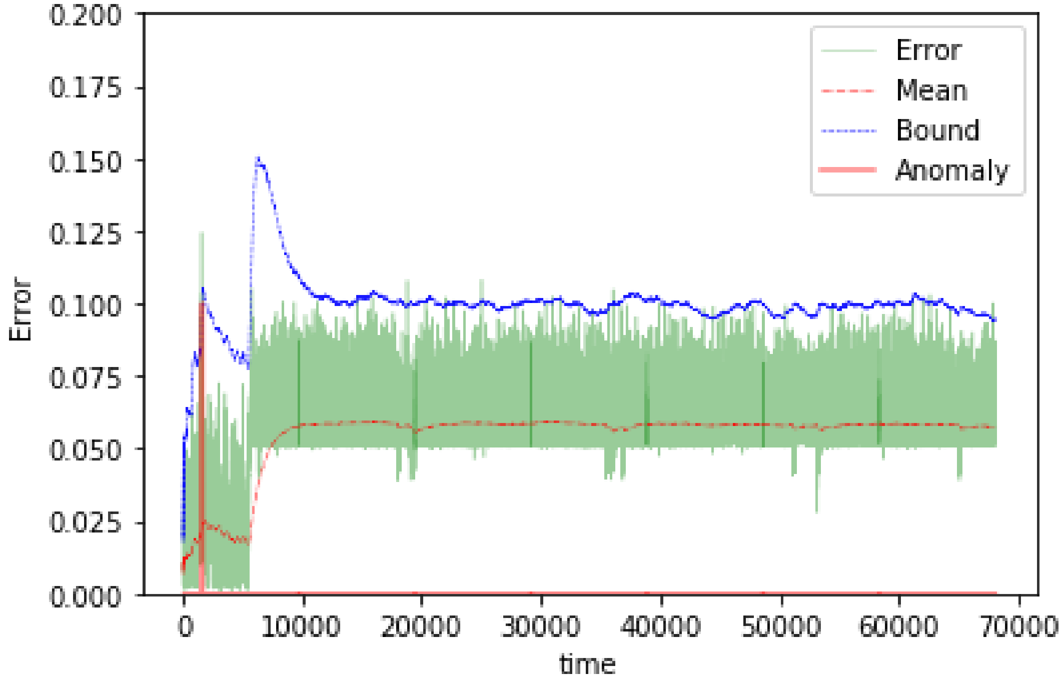

Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time


Error
Mean
Bound
Anomaly
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time

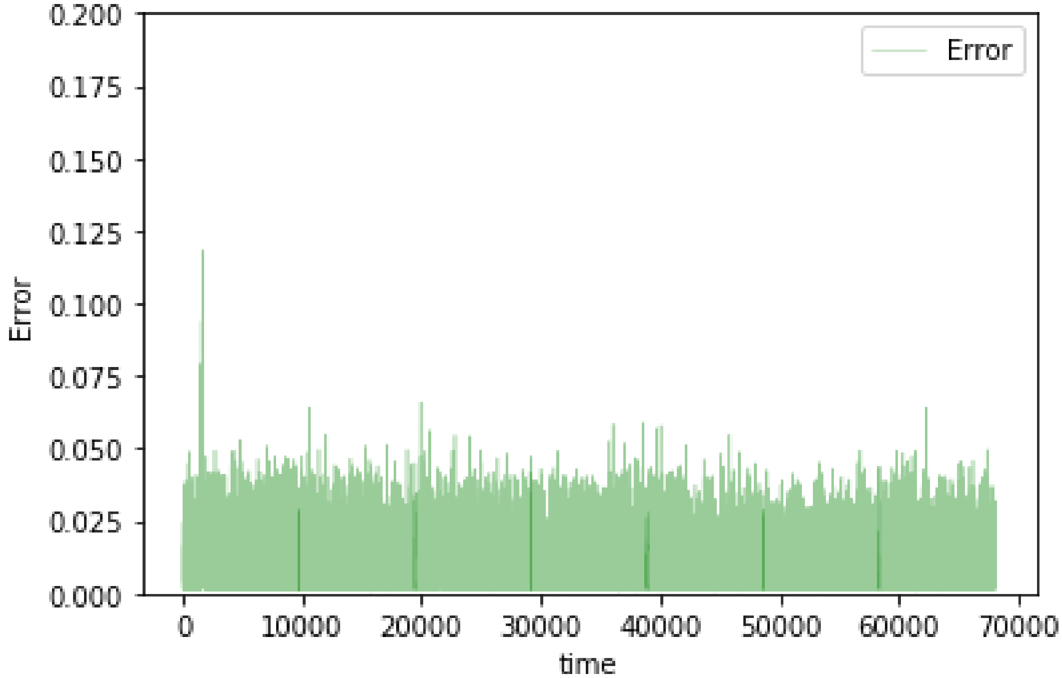
```
The mean error for gru3_200_avg_load_ is 0.055642597350820326 for length 68099
Testing on app change early data.
```

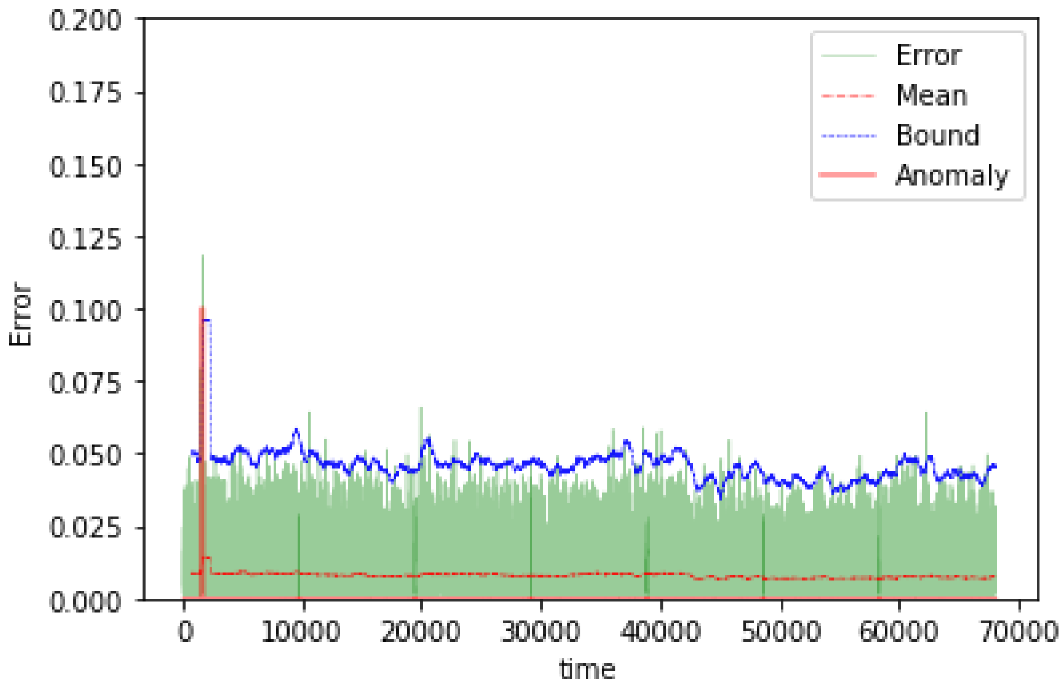


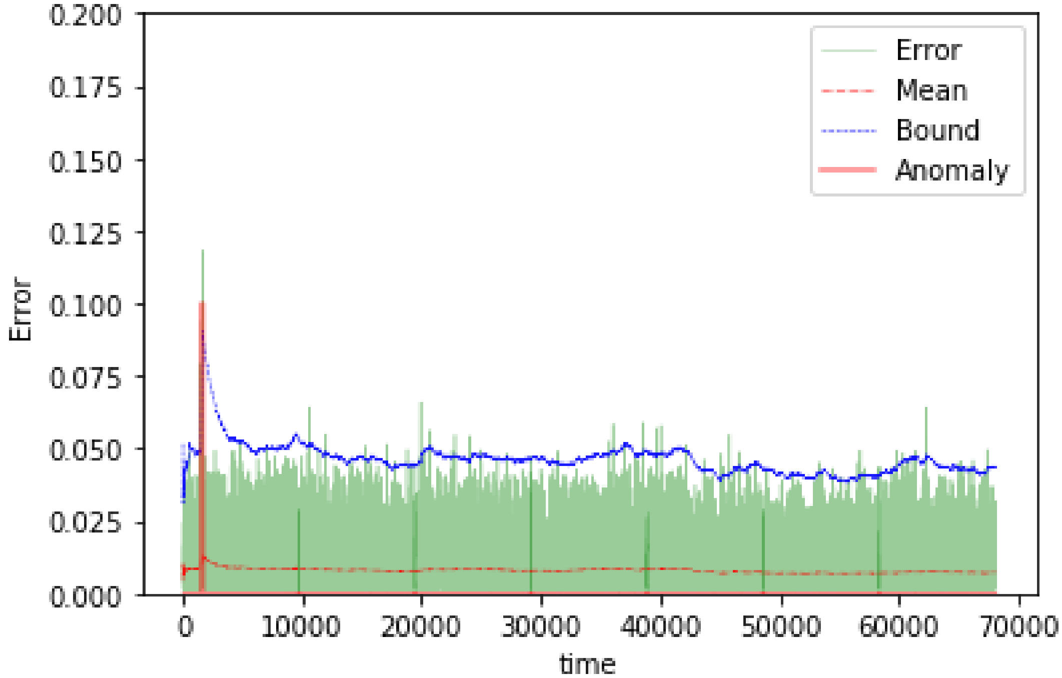





```
The mean error for gru3_200_app_change_early_ is 0.05496298178499459 for length 68099
Testing on Normal data.
```


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Error
0
10000
20000
30000
40000
50000
60000
70000
time
Error
Mean
Bound
Anomaly




```
The mean error for gru3_200_normal_ is 0.007939147554820255 for length 68099
====================
```