

ansi_regression

April 10, 2018

1 ANSI Application analysis

```
In [1]: import numpy
        import pandas
        import matplotlib.pyplot as plotter
        from scipy.stats import pearsonr, probplot, beta
        from sklearn.metrics import mean_squared_error, mean_absolute_error

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

1.1 CPU data

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

In [4]: #cpu_df.columns

In [5]: #view_boxplot(cpu_df)
```

1.2 Network TX

```
In [6]: txnet_df = pandas.read_csv('data/ansi_fake_data/ansi_fake_data_network_tx.csv', index_col='Time')

In [7]: #txnet_df.columns

In [8]: #view_boxplot(txnet_df)
```

1.3 Network RX

```
In [9]: rxnet_df = pandas.read_csv('data/ansi_fake_data/ansi_fake_data_network_rx.csv', index_col='Time')

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_fake_data/ansi_fake_data_disk_io.csv', index_col=0)

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_fake_data/ansi_fake_data_context.csv', index_col=0)

In [16]: #context_df.columns

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

1.6 Separate into proper dataframes for each node

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

for i in range(1,5):
    frames = []

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

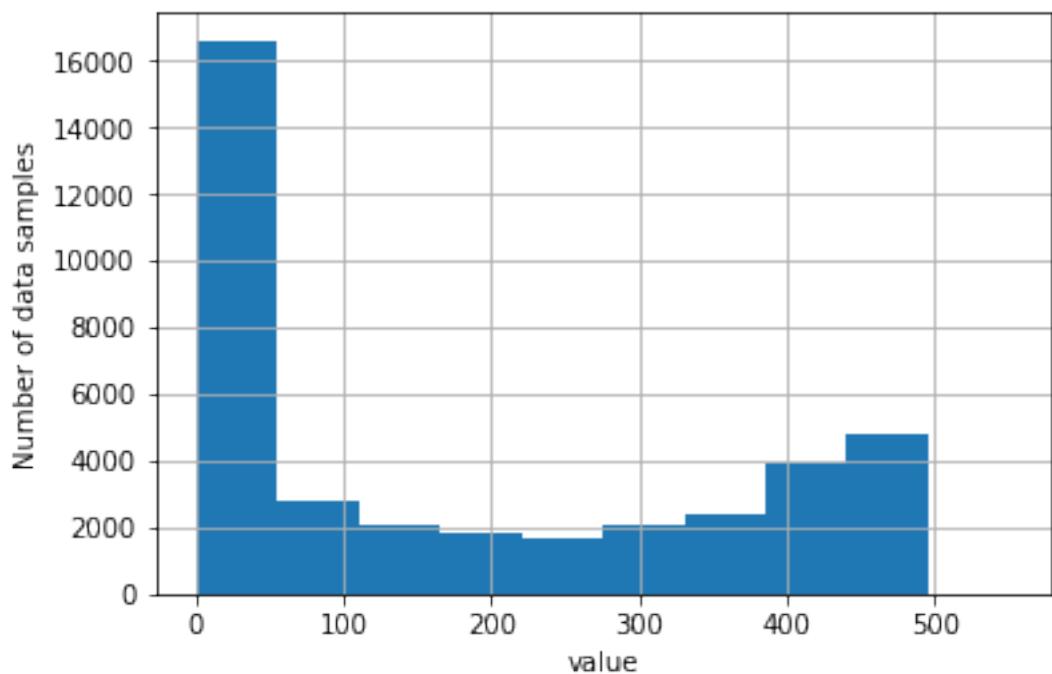
    node[i] = pandas.concat(frames, join='inner', axis=1).fillna(0)[:38200]

In [19]: for i in range(1,5):
    print(node[i].shape)
    print(node[i].columns)
    for column in node[i].columns:
        print(f'Distribution of data for {column}')
        node[i][column].hist()
        plotter.ylabel("Number of data samples")
        plotter.xlabel("value")
        plotter.savefig(f"node{i}_{column}_hist.png")
        plotter.show()
    pass

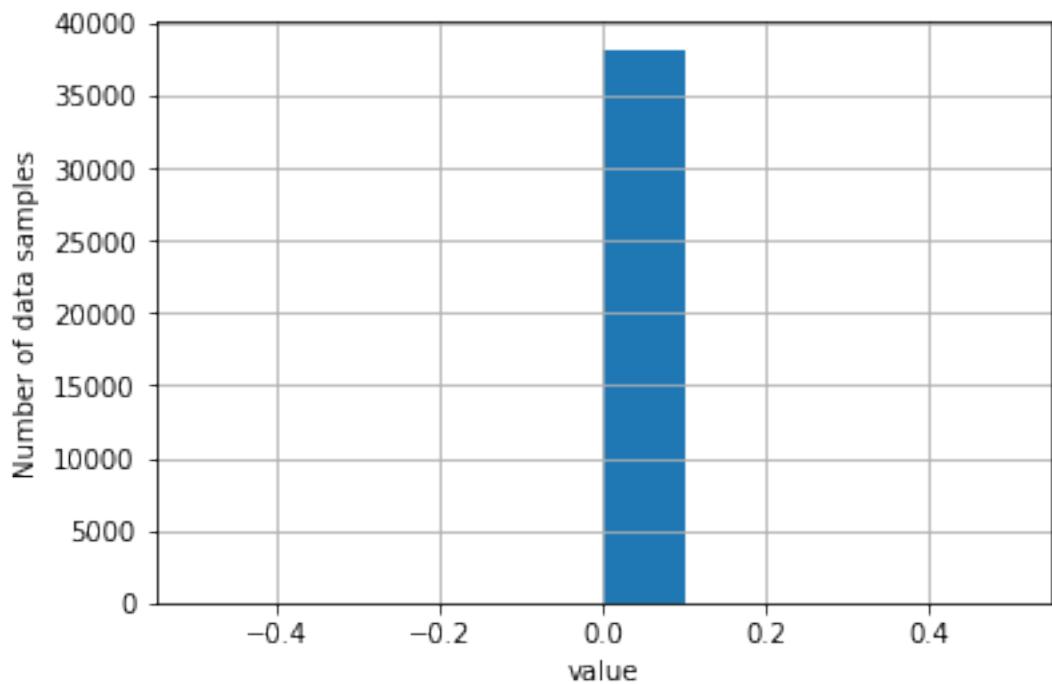
(38200, 29)
Index(['cpu_value host bb1localdomain type_instance idle',
       'cpu_value host bb1localdomain type_instance interrupt',
       'cpu_value host bb1localdomain type_instance nice',
       'cpu_value host bb1localdomain type_instance softirq',
```

```
'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')
```

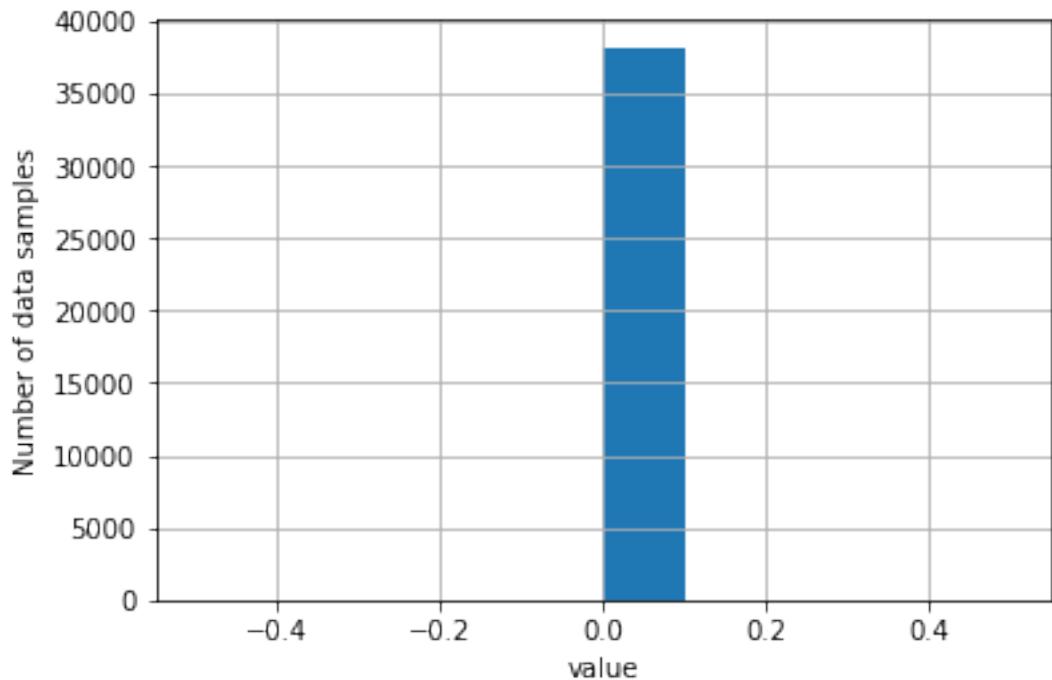
Distribution of data for `cpu_value host bb1localdomain type_instance idle`



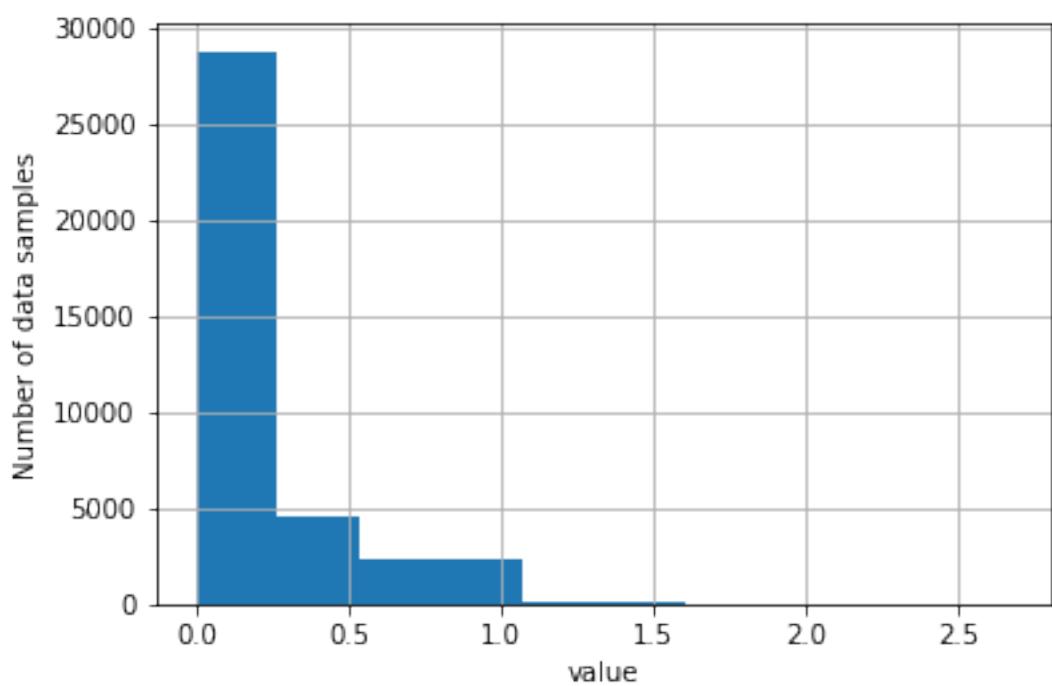
Distribution of data for `cpu_value` host `bb1localdomain` type_`instance` interrupt



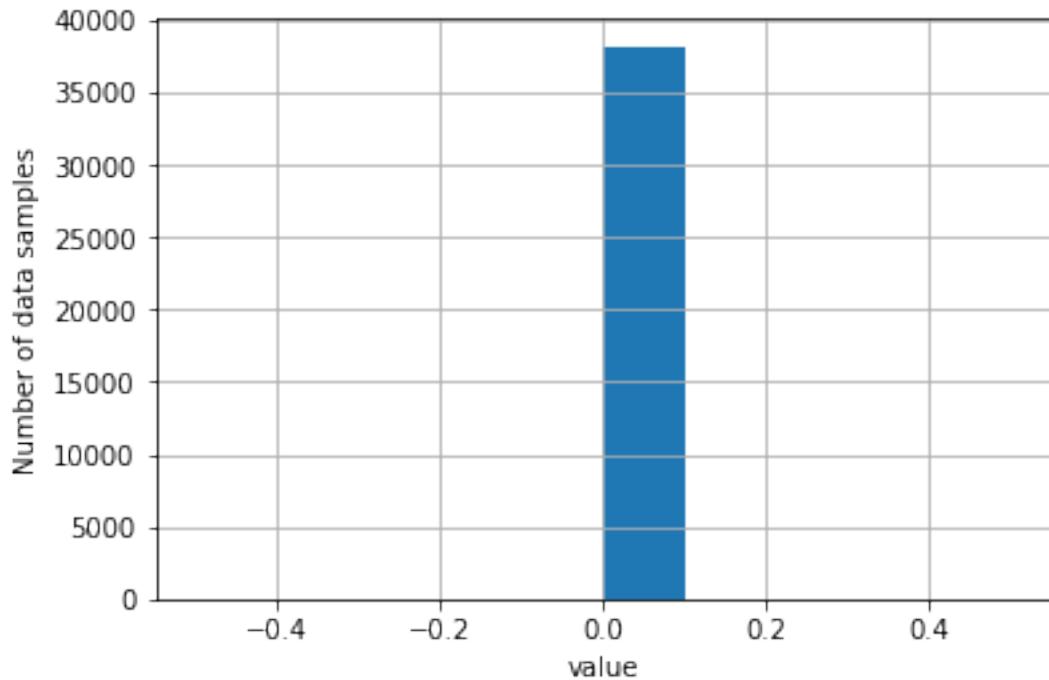
Distribution of data for cpu_value host bb1localdomain type_instance nice



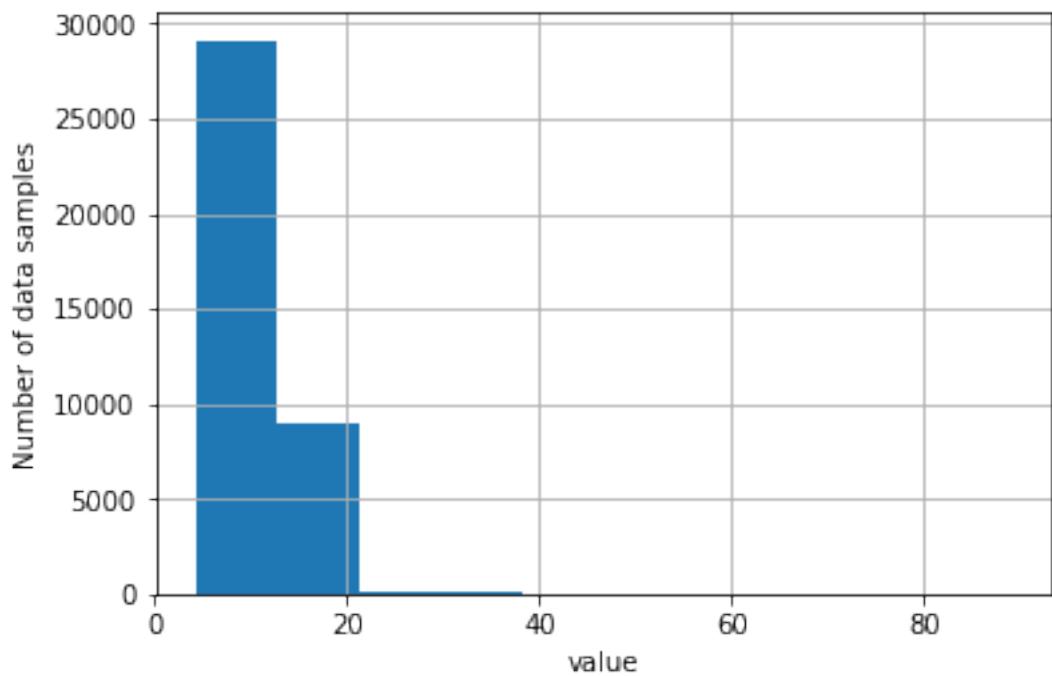
Distribution of data for cpu_value host bb1localdomain type_instance softirq



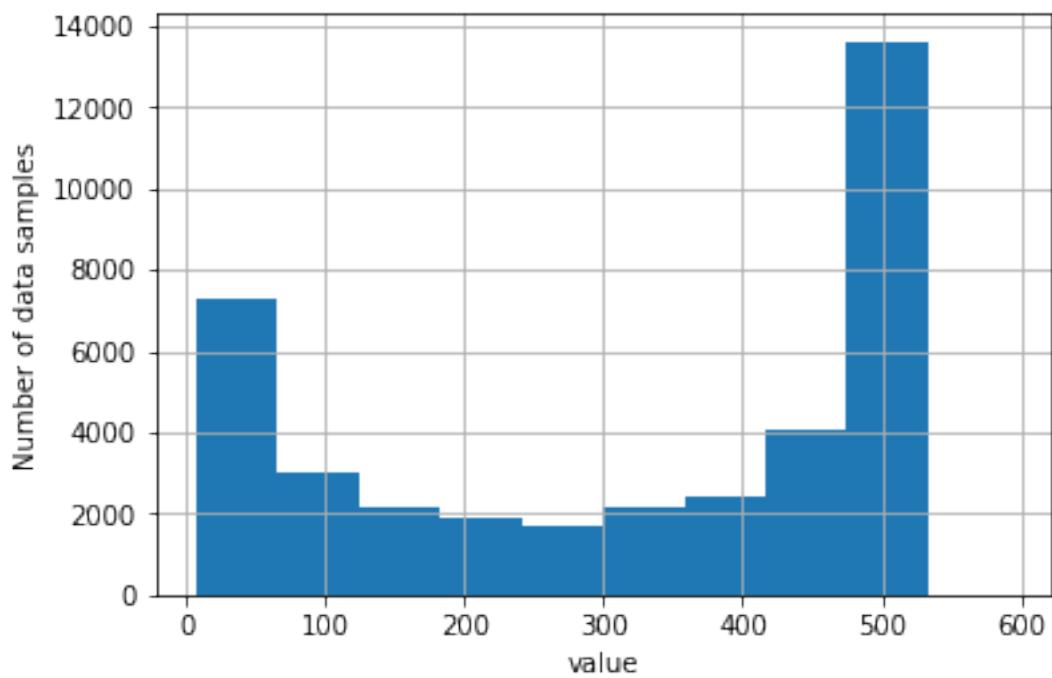
Distribution of data for cpu_value host bb1localdomain type_instance steal



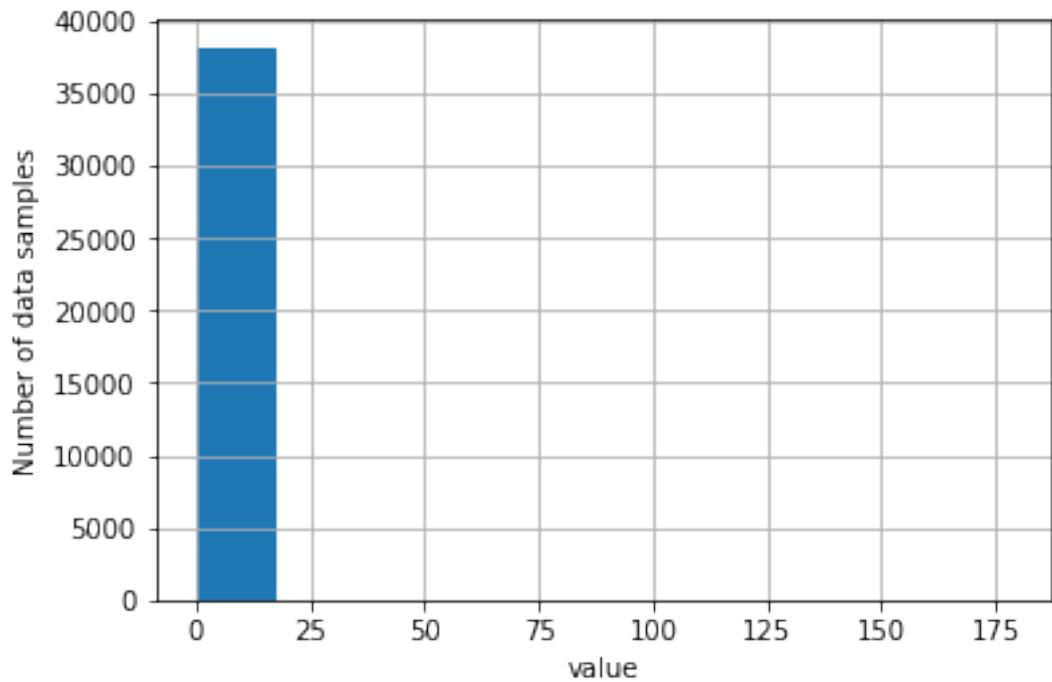
Distribution of data for cpu_value host bb1localdomain type_instance system



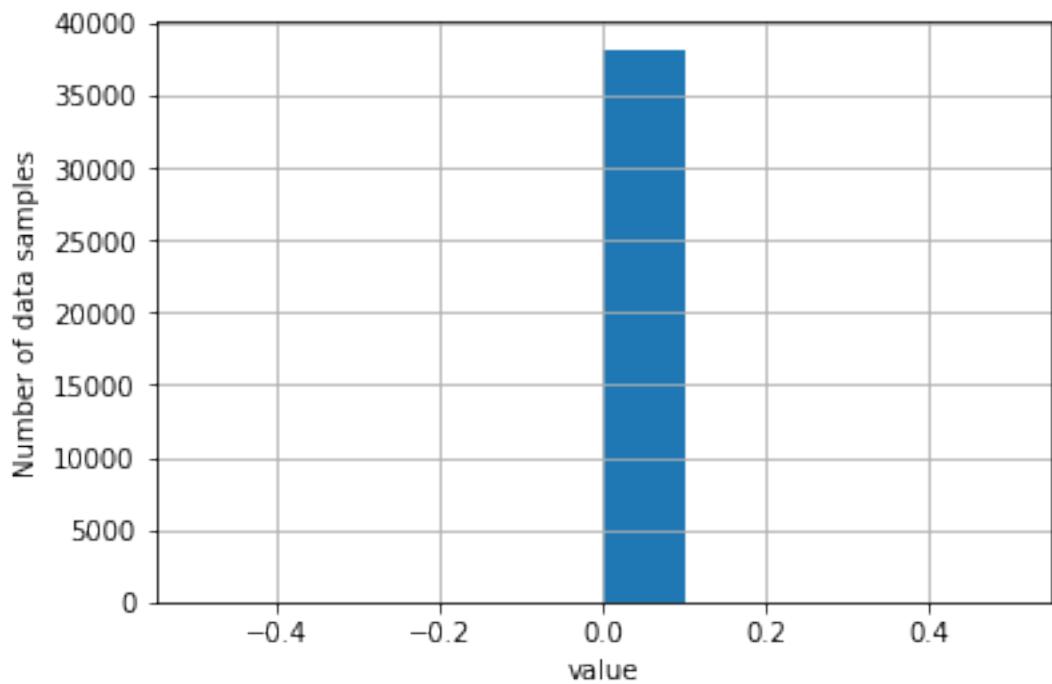
Distribution of data for cpu_value host bb1localdomain type_instance user



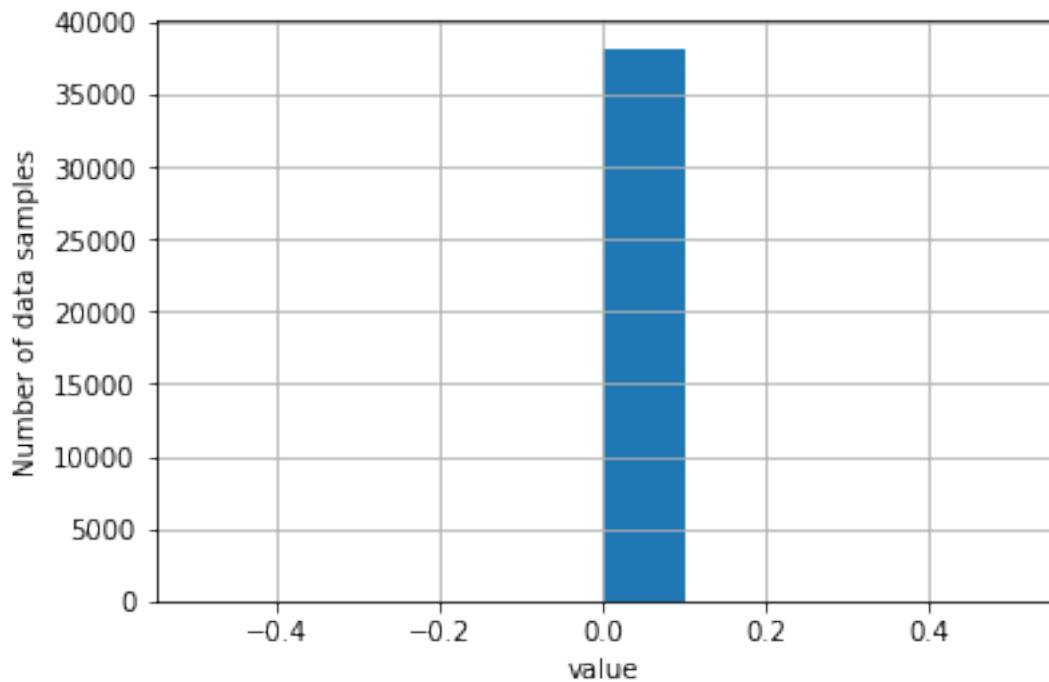
Distribution of data for cpu_value host bb1localdomain type_instance wait



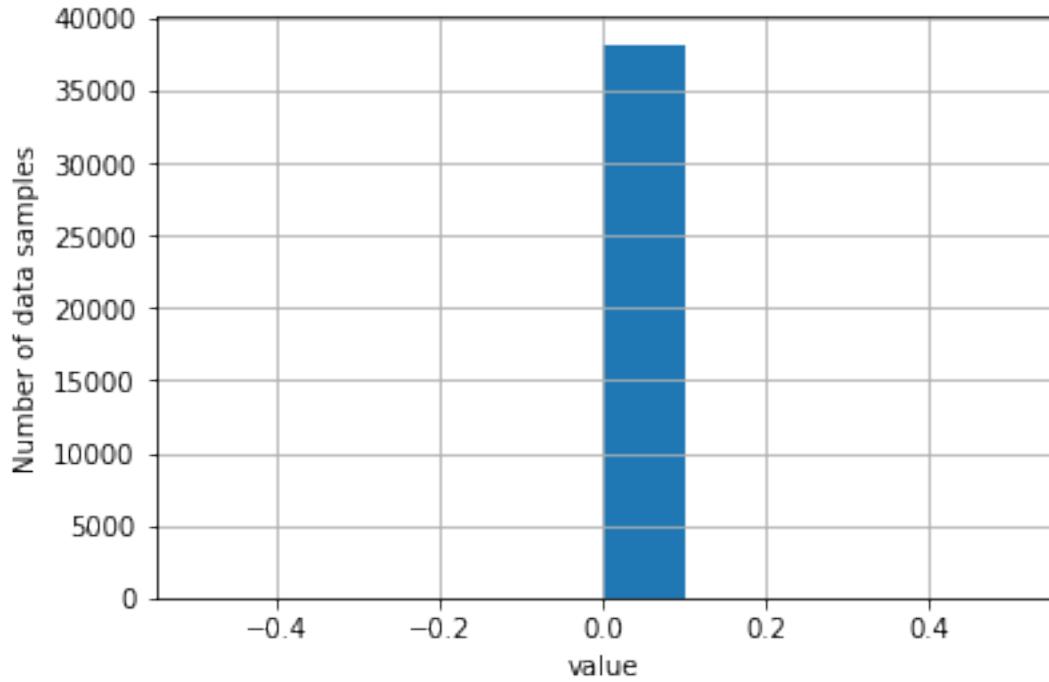
Distribution of data for interface_tx host bb1localdomain instance lo type if_dropped



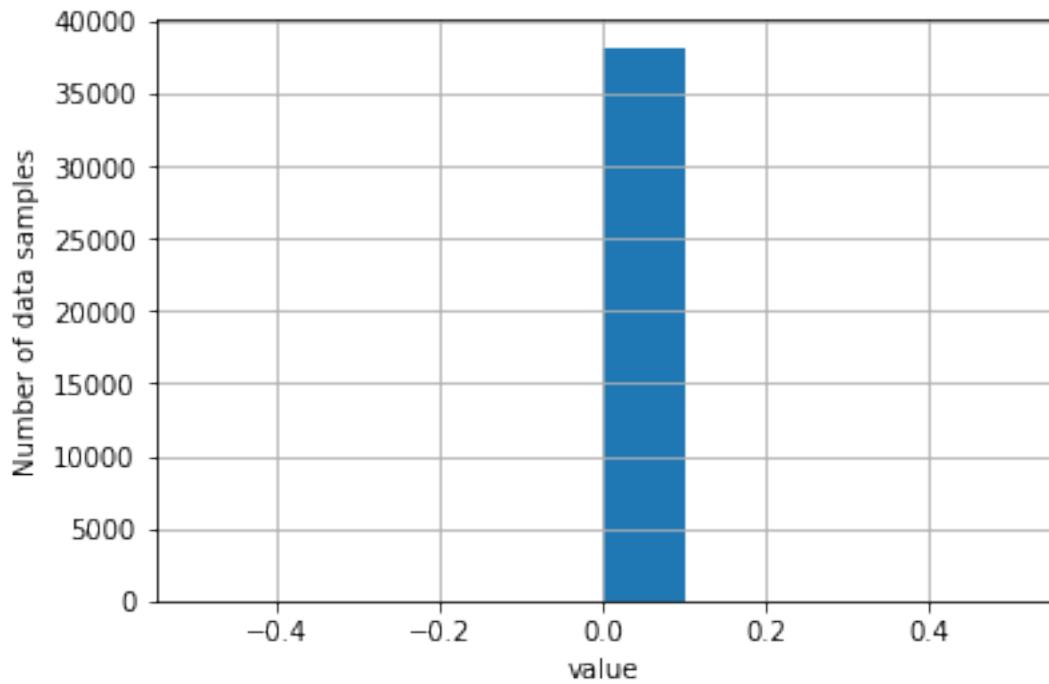
Distribution of data for interface_tx host bb1localdomain instance lo type if_errors



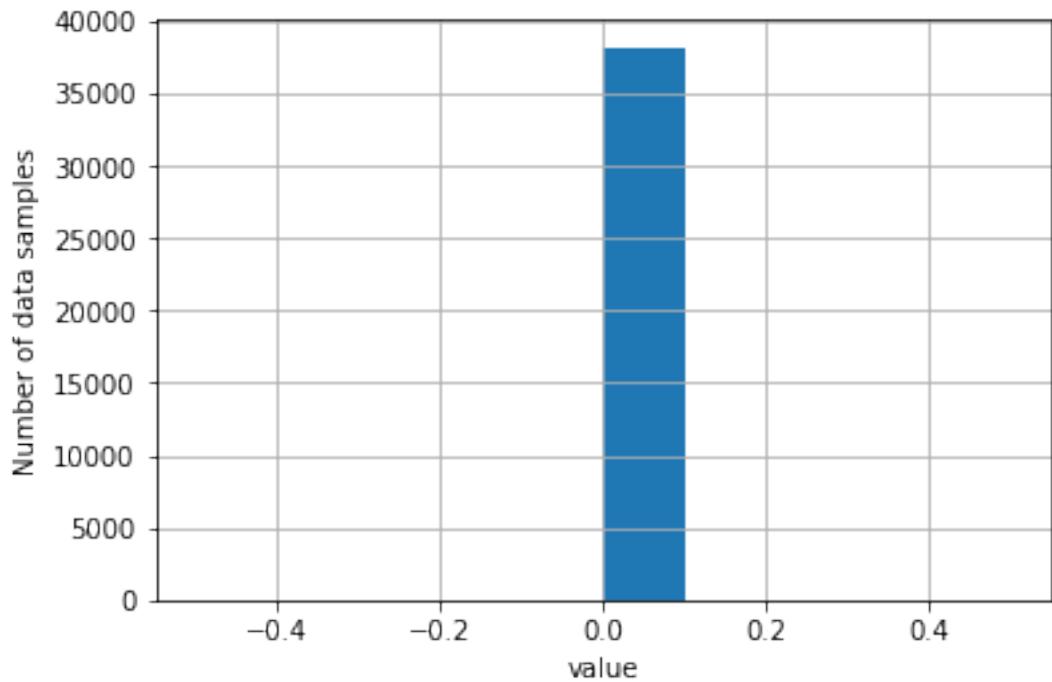
Distribution of data for interface_tx host bb1localdomain instance lo type if_octets



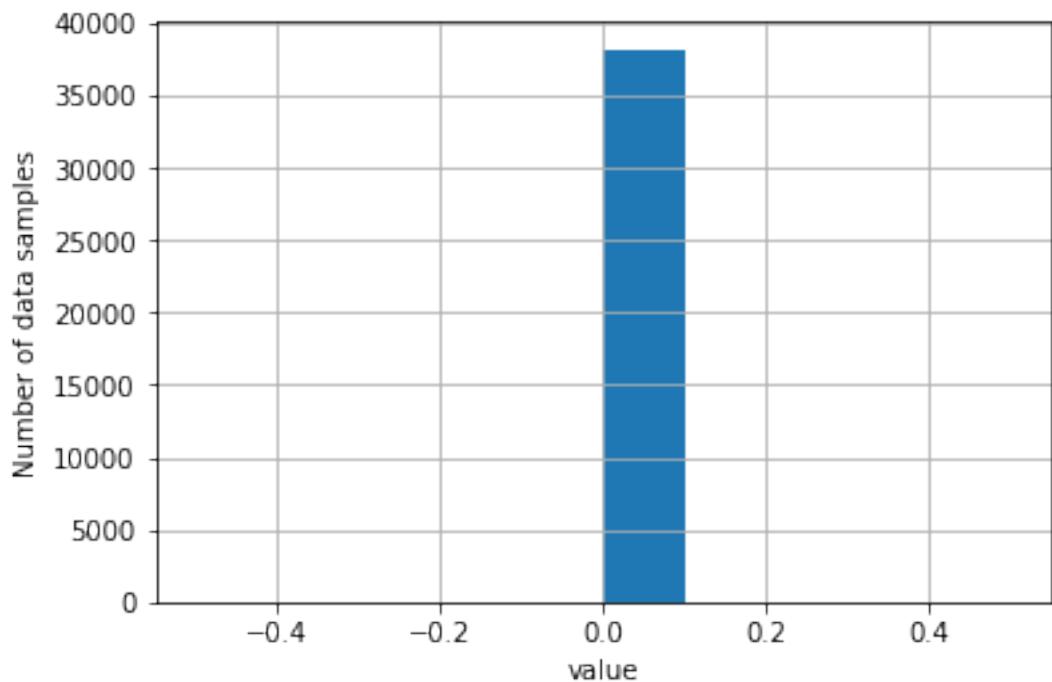
Distribution of data for interface_tx host bb1localdomain instance lo type if_packets



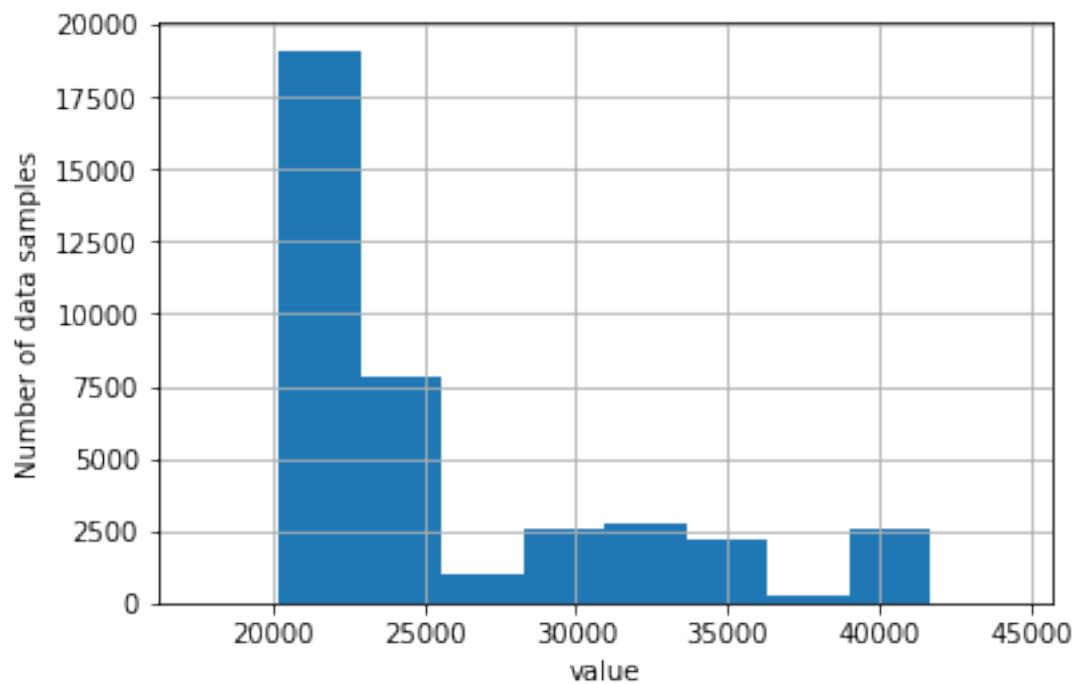
Distribution of data for interface_tx host bbilocaldomain instance wlan0 type if_dropped



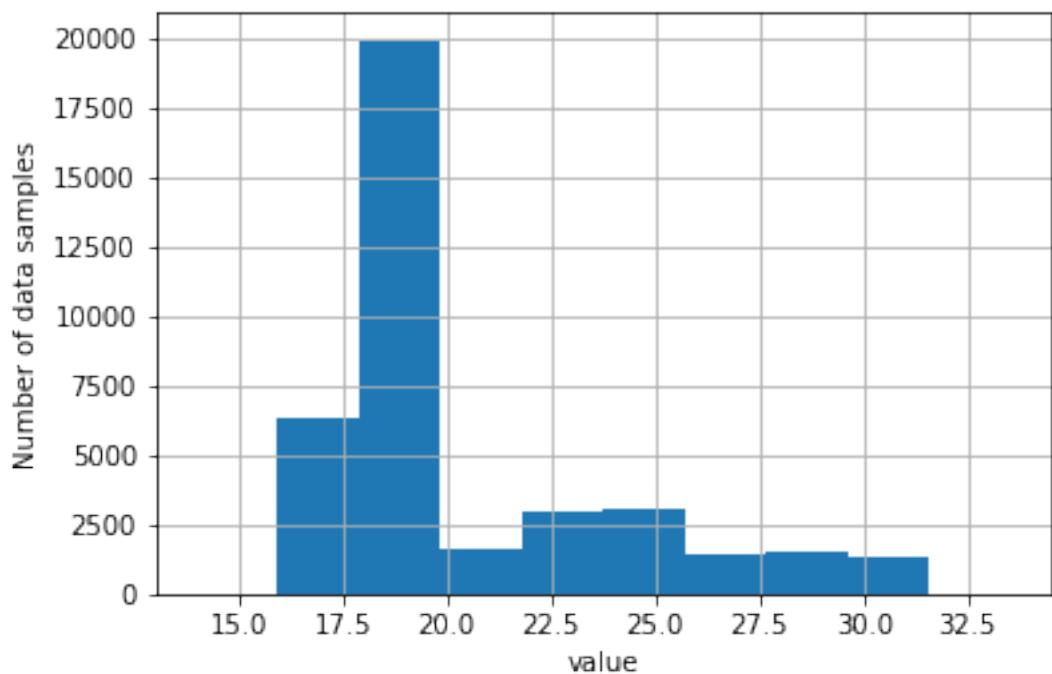
Distribution of data for interface_tx host bbilocaldomain instance wlan0 type if_errors



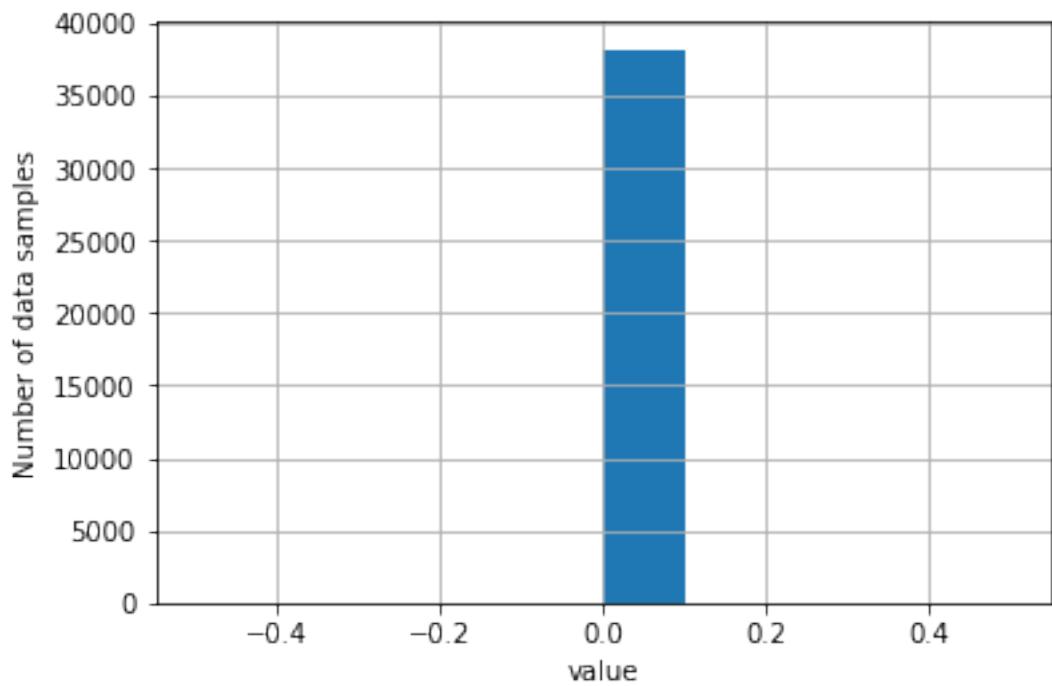
Distribution of data for interface_tx host bb1localdomain instance wlan0 type if_octets



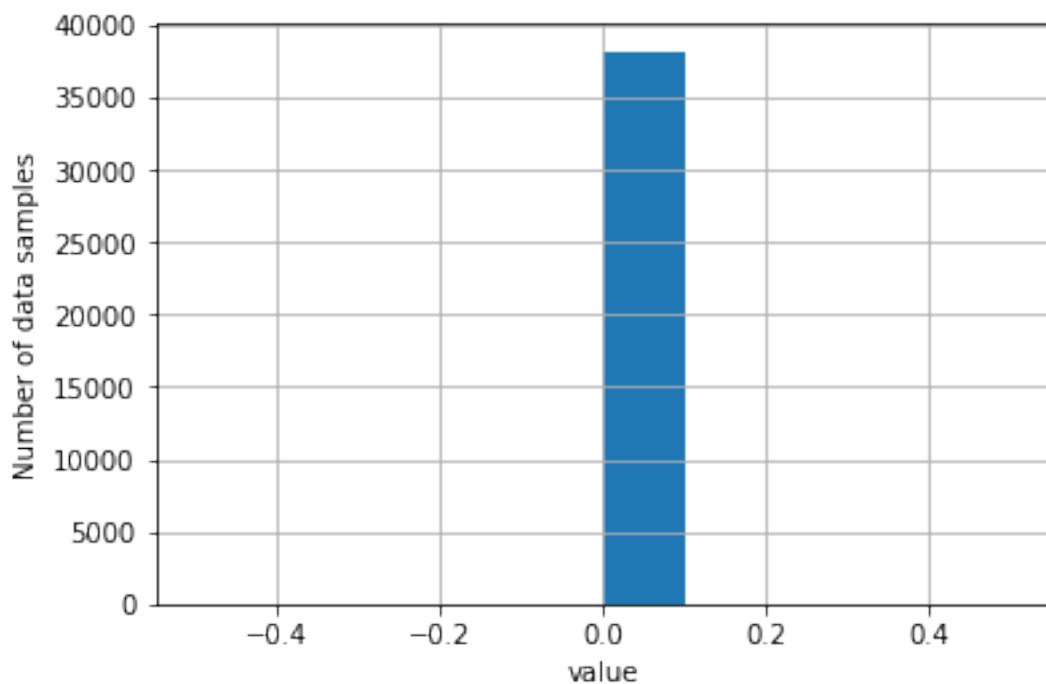
Distribution of data for interface_tx host bb1localdomain instance wlan0 type if_packets



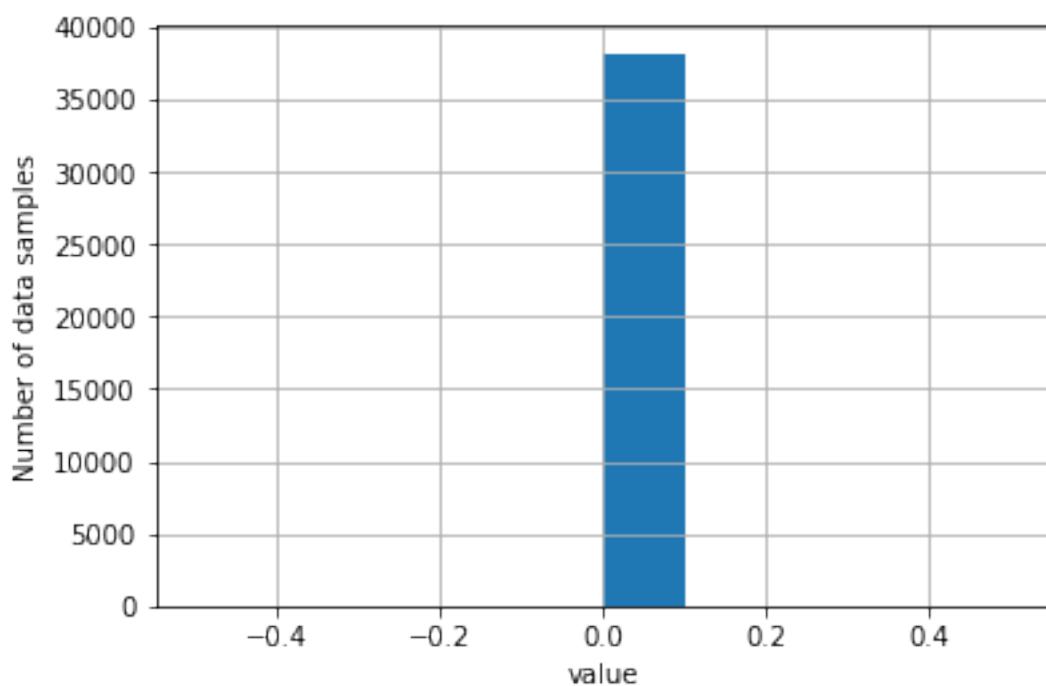
Distribution of data for interface_rx host bb1localdomain instance lo type if_dropped



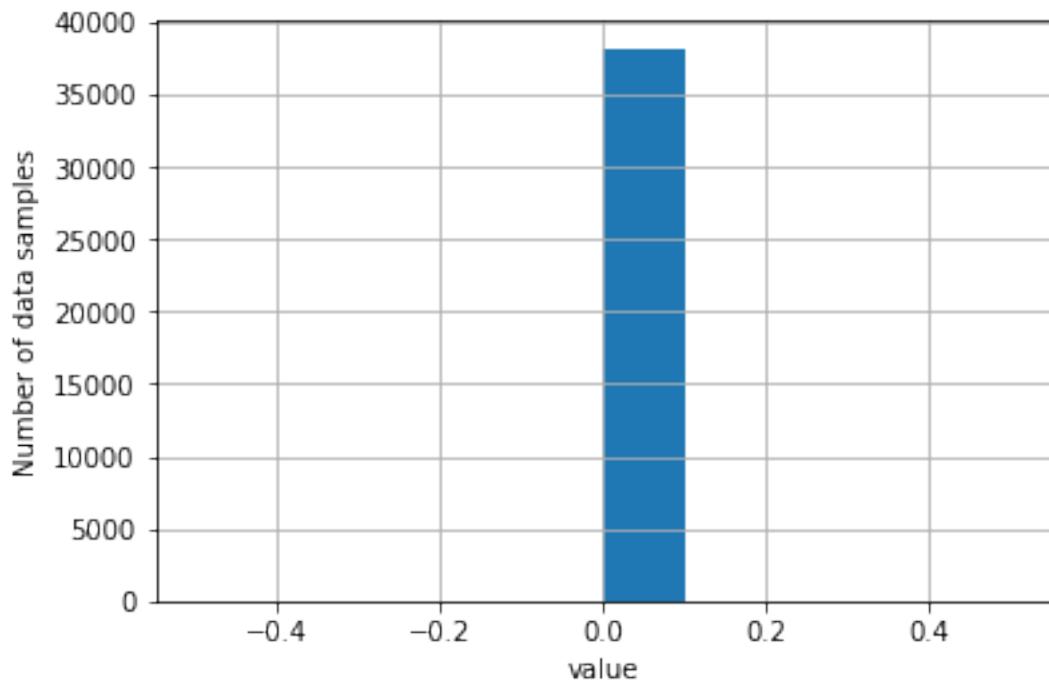
Distribution of data for interface_rx host bbilocaldomain instance lo type if_errors



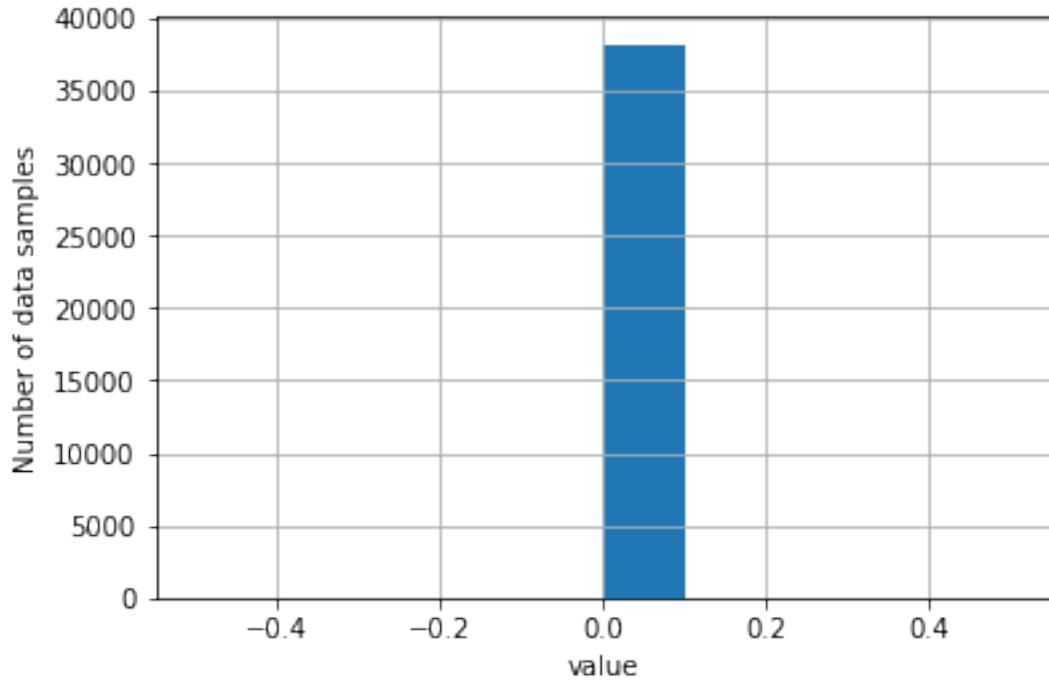
Distribution of data for interface_rx host bbilocaldomain instance lo type if_octets



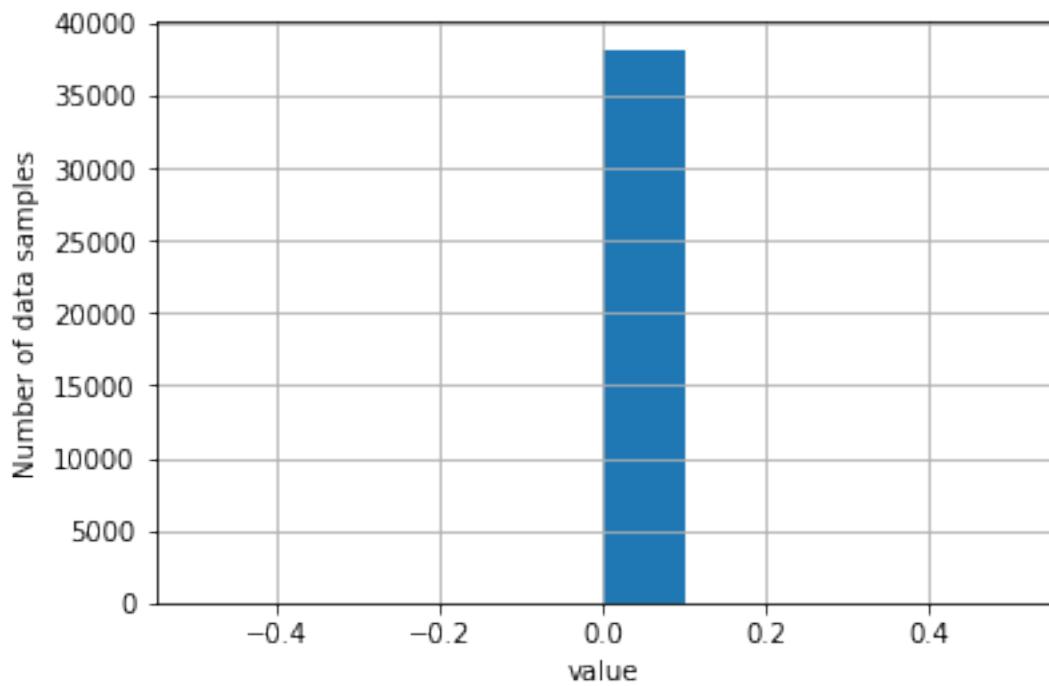
Distribution of data for interface_rx host bb1localdomain instance lo type if_packets



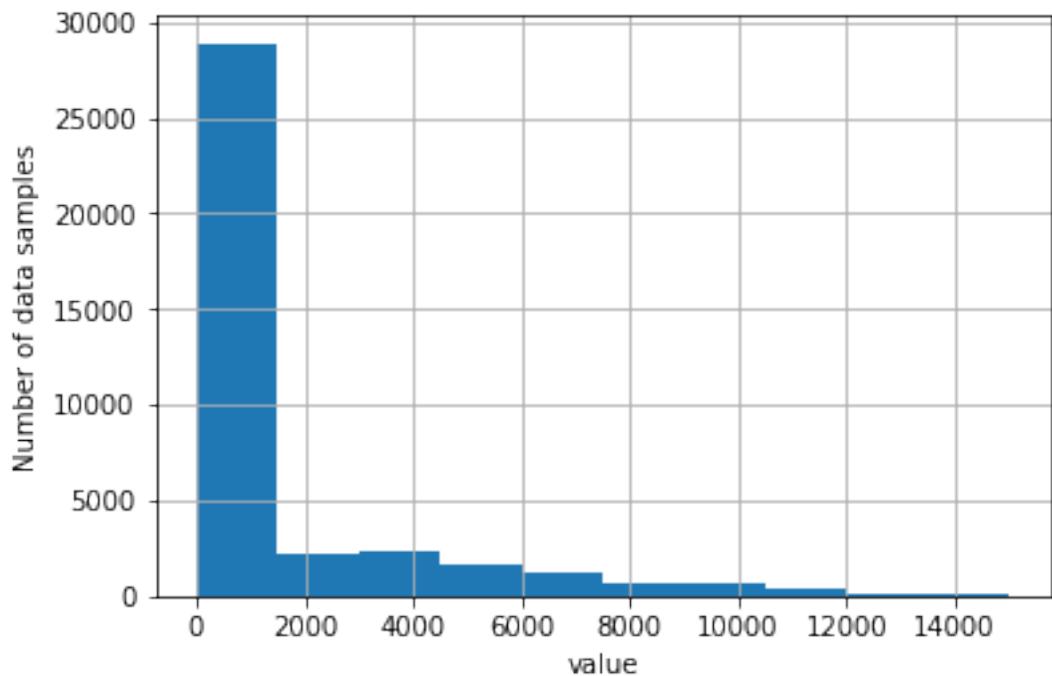
Distribution of data for interface_rx host bb1localdomain instance wlan0 type if_dropped



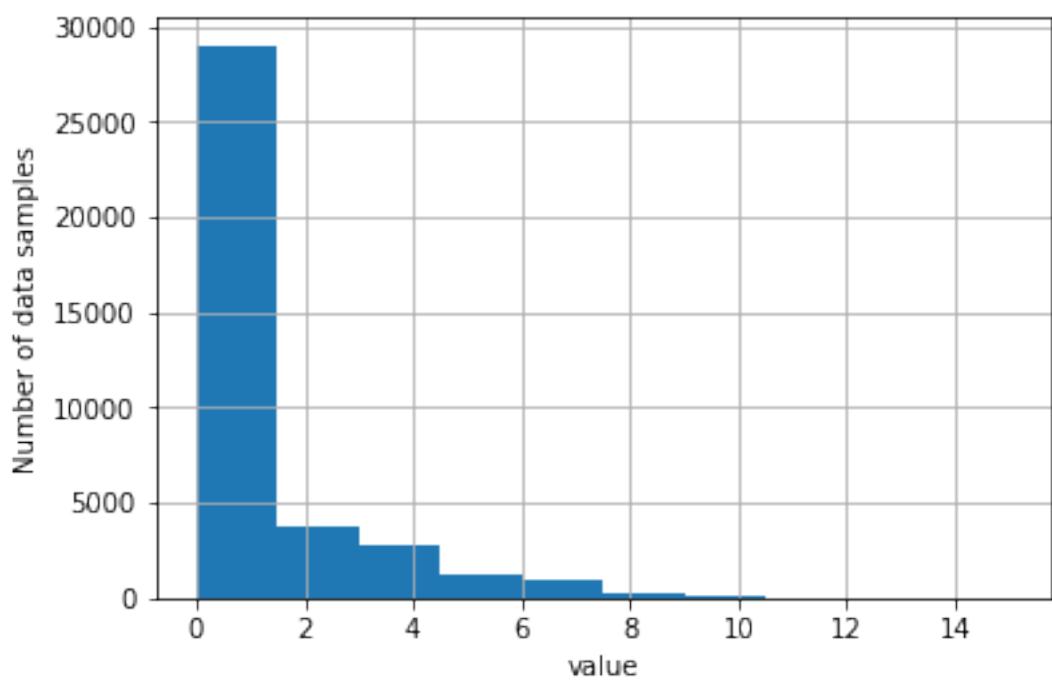
Distribution of data for interface_rx host bb1localdomain instance wlan0 type if_errors



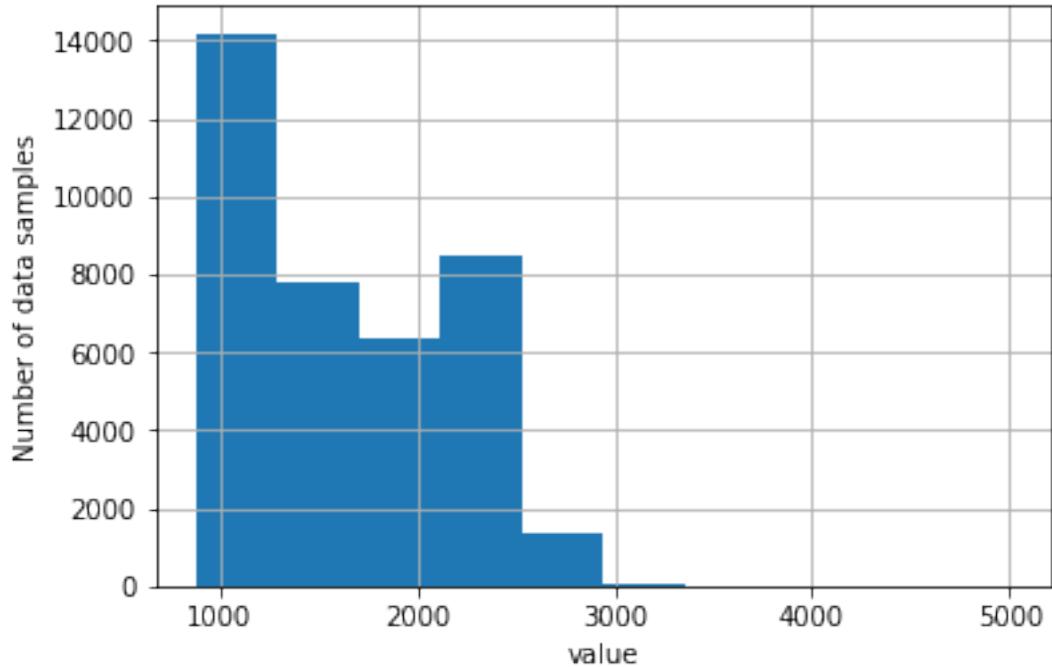
Distribution of data for interface_rx host bb1localdomain instance wlan0 type if_octets



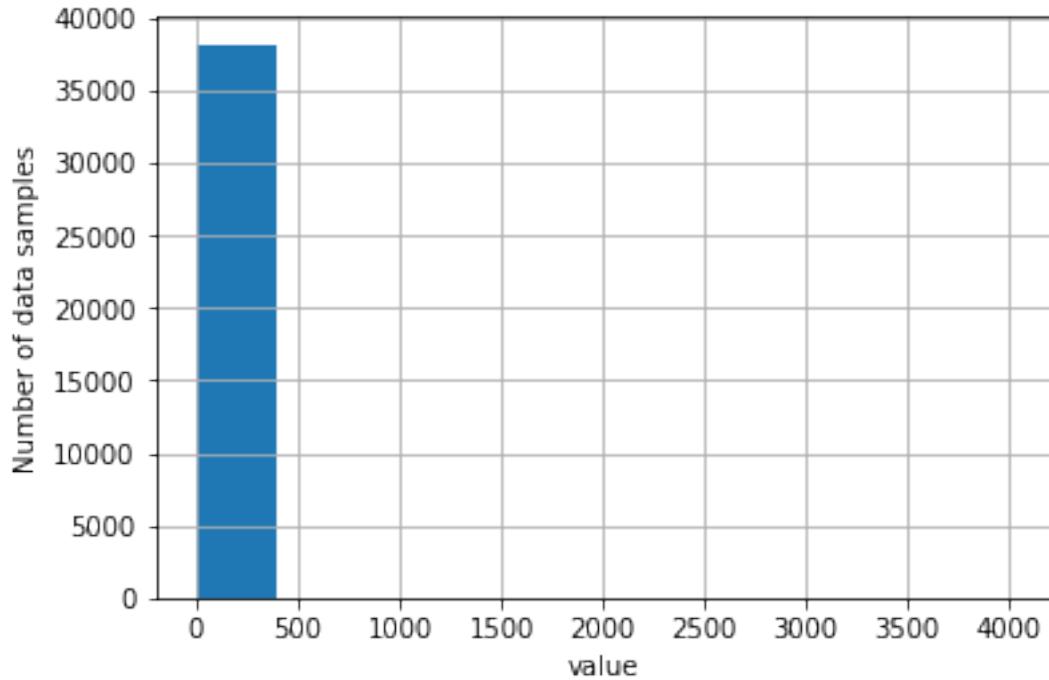
Distribution of data for interface_rx host bb1localdomain instance wlan0 type if_packets



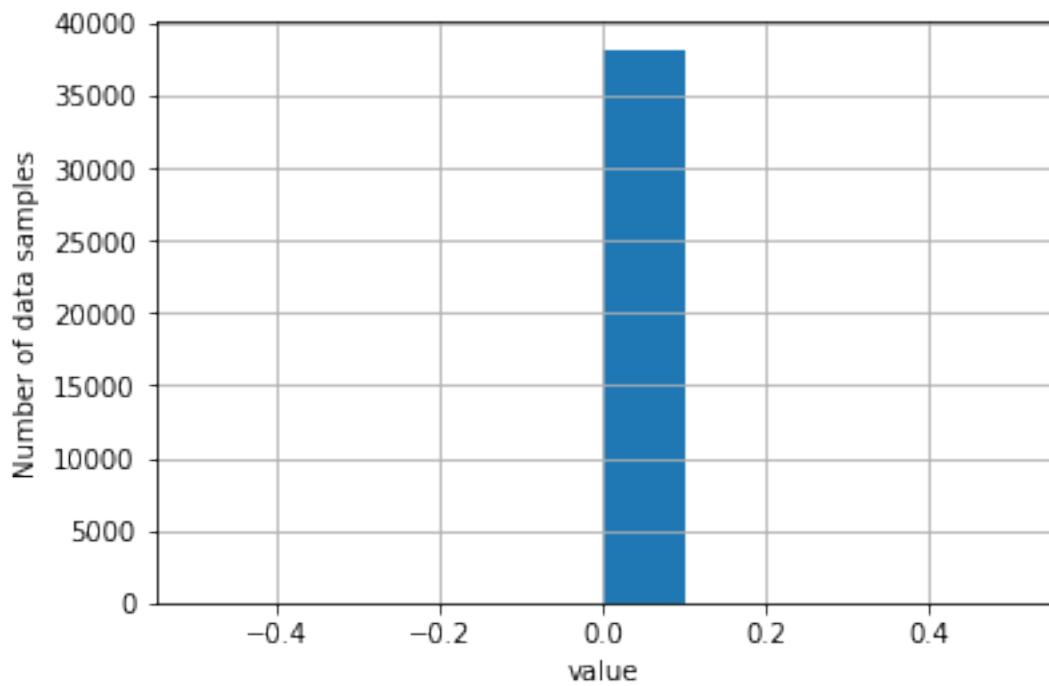
Distribution of data for contextswitch_value host bb1localdomain type contextswitch



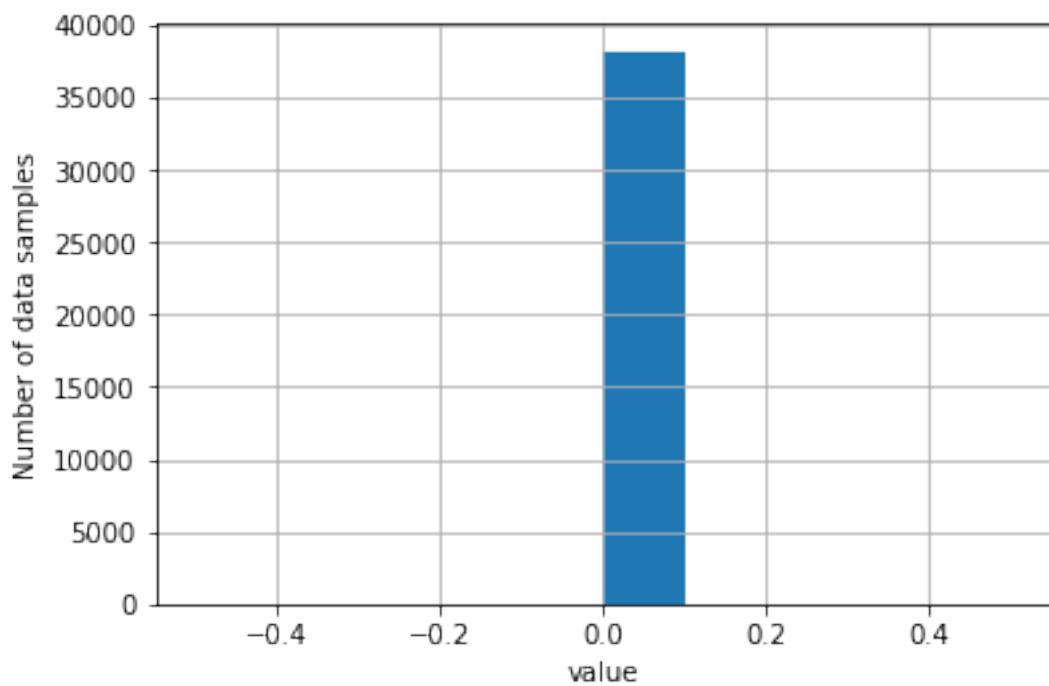
Distribution of data for disk_io_time host bb1localdomain instance mmcblk1 type disk_io_time



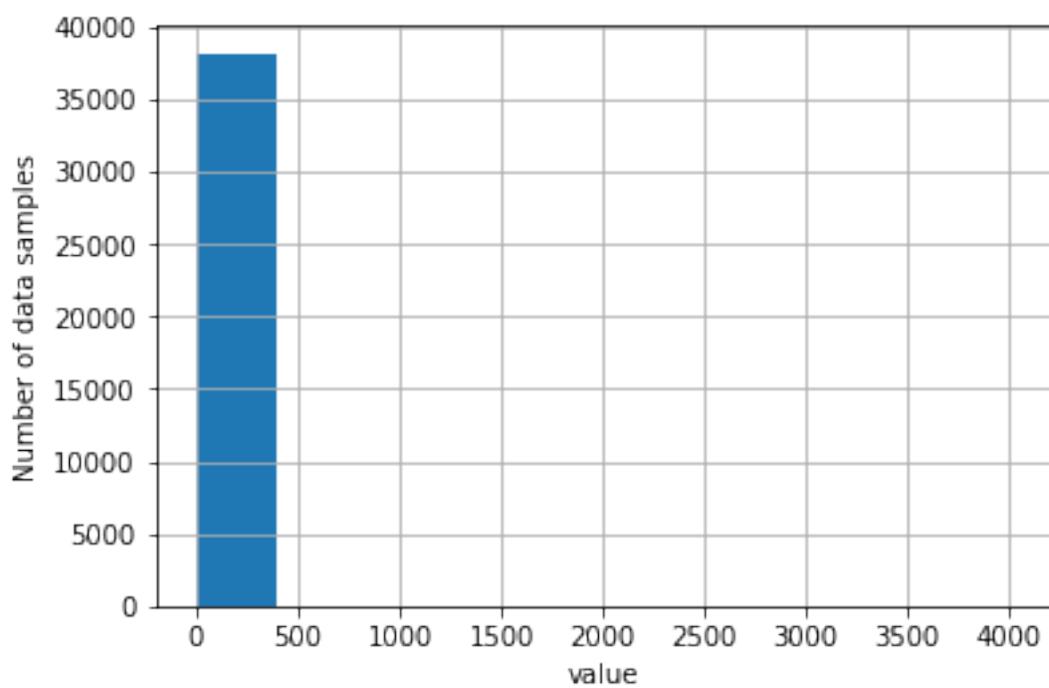
Distribution of data for disk_io_time host bb1localdomain instance mmcblk1boot0 type disk_io_t



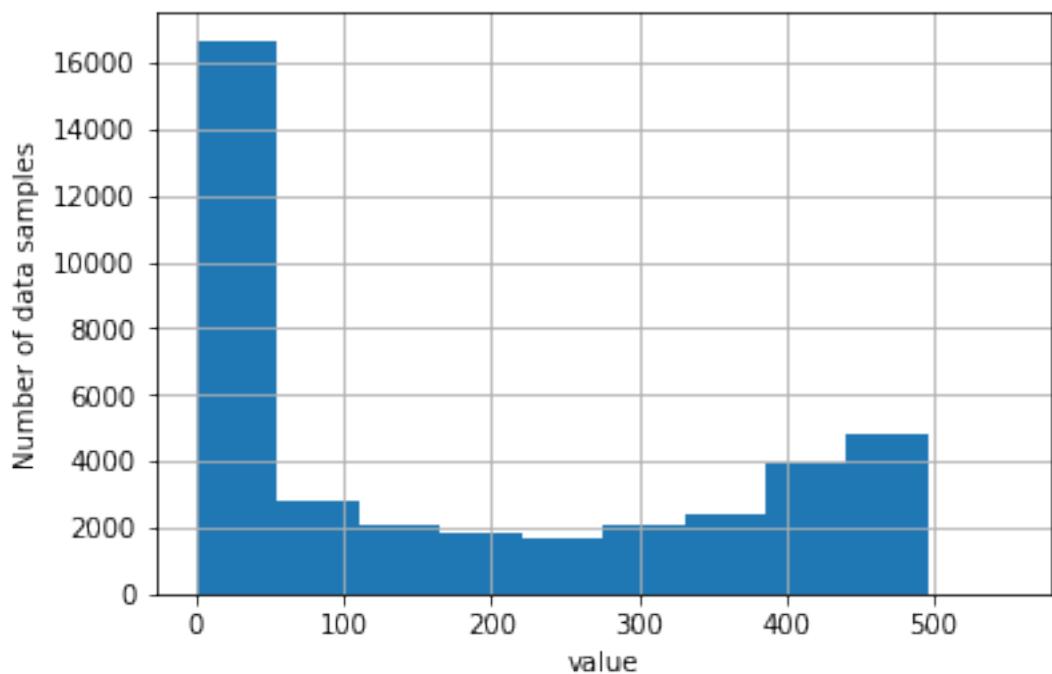
Distribution of data for disk_io_time host bb1localdomain instance mmcblk1boot1 type disk_io_time



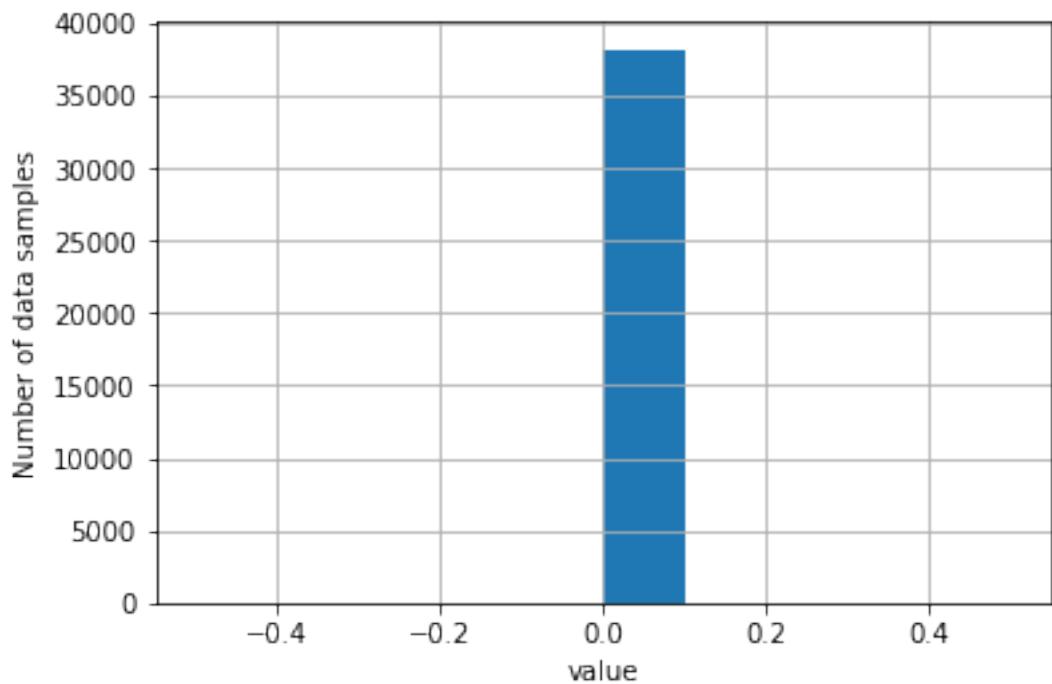
Distribution of data for disk_io_time host bb1localdomain instance mmcblk1p1 type disk_io_time



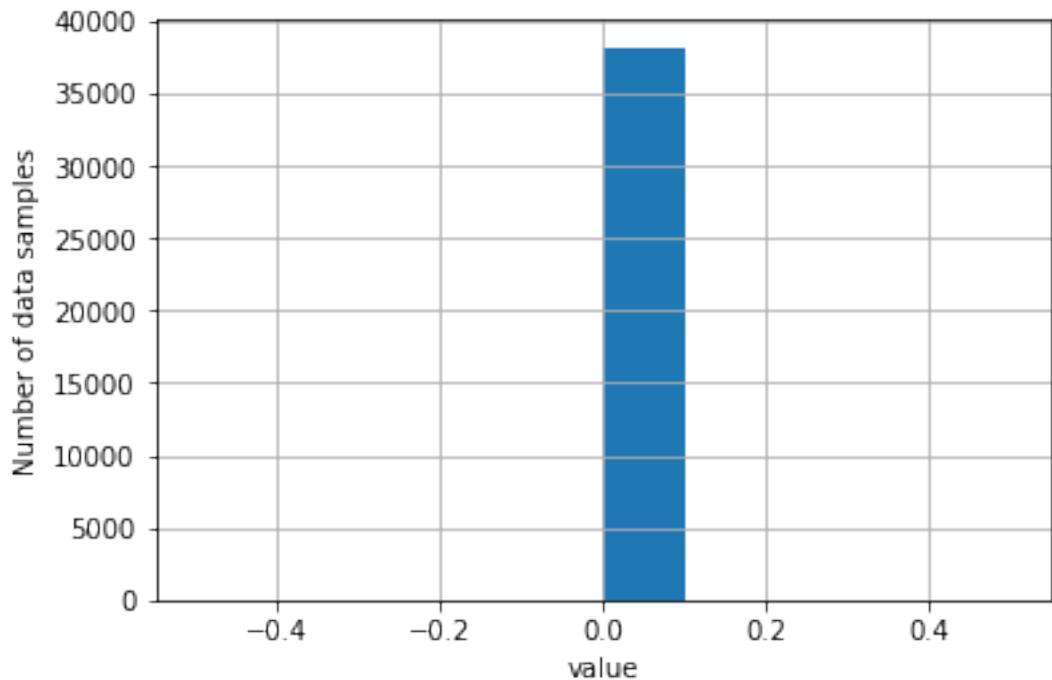
```
(38200, 29)
Index(['cpu_value host bb2localdomain type_instance idle',
       'cpu_value host bb2localdomain type_instance interrupt',
       'cpu_value host bb2localdomain type_instance nice',
       'cpu_value host bb2localdomain type_instance softirq',
       'cpu_value host bb2localdomain type_instance steal',
       'cpu_value host bb2localdomain type_instance system',
       'cpu_value host bb2localdomain type_instance user',
       'cpu_value host bb2localdomain type_instance wait',
       'interface_tx host bb2localdomain instance lo type if_dropped',
       'interface_tx host bb2localdomain instance lo type if_errors',
       'interface_tx host bb2localdomain instance lo type if_octets',
       'interface_tx host bb2localdomain instance lo type if_packets',
       'interface_tx host bb2localdomain instance wlan0 type if_dropped',
       'interface_tx host bb2localdomain instance wlan0 type if_errors',
       'interface_tx host bb2localdomain instance wlan0 type if_octets',
       'interface_tx host bb2localdomain instance wlan0 type if_packets',
       'interface_rx host bb2localdomain instance lo type if_dropped',
       'interface_rx host bb2localdomain instance lo type if_errors',
       'interface_rx host bb2localdomain instance lo type if_octets',
       'interface_rx host bb2localdomain instance lo type if_packets',
       'interface_rx host bb2localdomain instance wlan0 type if_dropped',
       'interface_rx host bb2localdomain instance wlan0 type if_errors',
       'interface_rx host bb2localdomain instance wlan0 type if_octets',
       'interface_rx host bb2localdomain instance wlan0 type if_packets',
       'contextswitch_value host bb2localdomain type contextswitch',
       'disk_io_time host bb2localdomain instance mmcblk1 type disk_io_time',
       'disk_io_time host bb2localdomain instance mmcblk1boot0 type disk_io_time',
       'disk_io_time host bb2localdomain instance mmcblk1boot1 type disk_io_time',
       'disk_io_time host bb2localdomain instance mmcblk1p1 type disk_io_time'],
      dtype='object')
Distribution of data for cpu_value host bb2localdomain type_instance idle
```



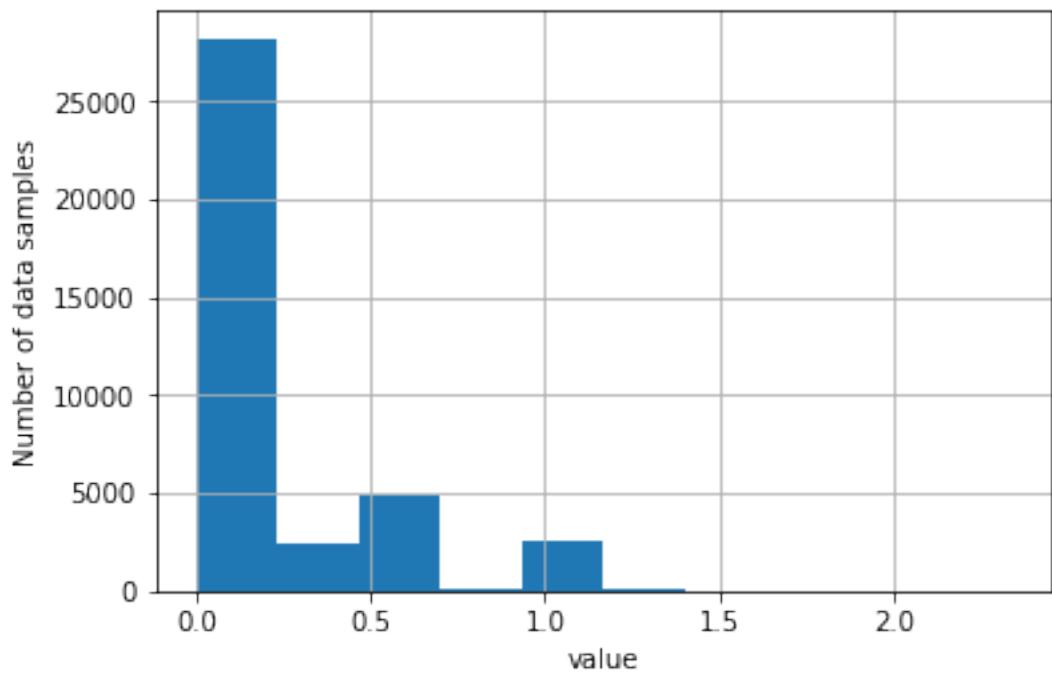
Distribution of data for `cpu_value host bb2localdomain type_instance interrupt`



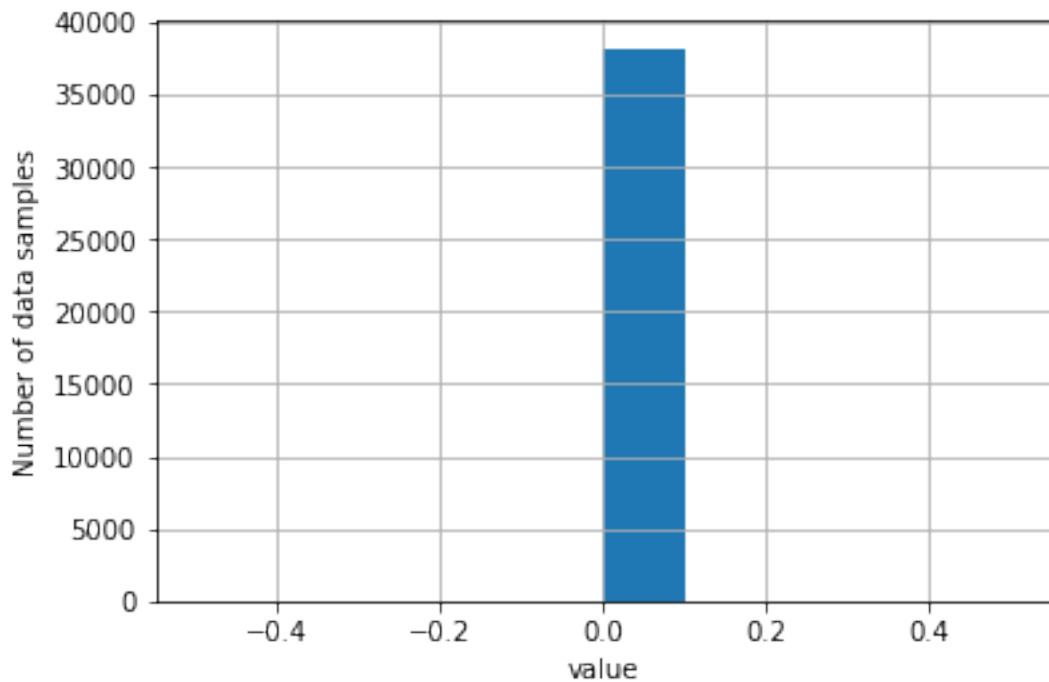
Distribution of data for cpu_value host bb2localdomain type_instance nice



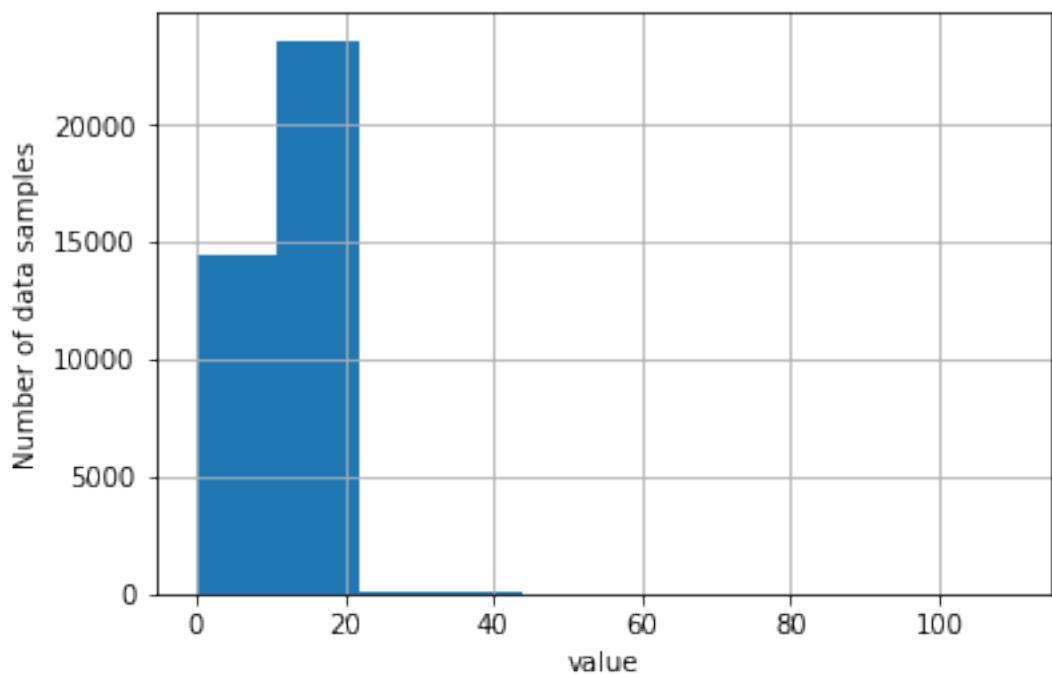
Distribution of data for cpu_value host bb2localdomain type_instance softirq



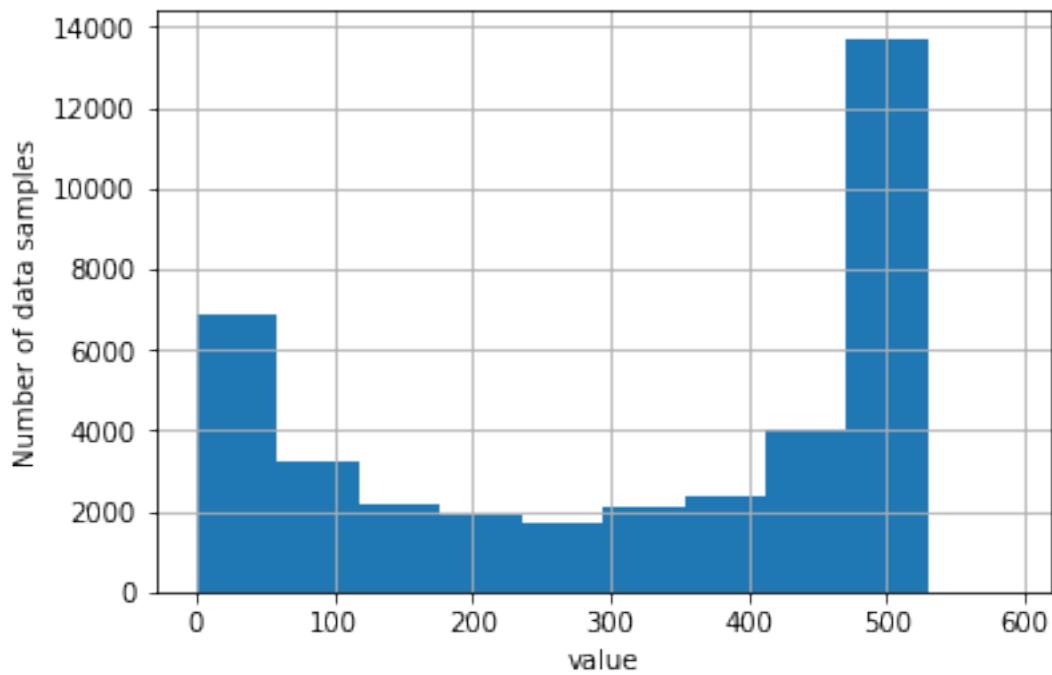
Distribution of data for cpu_value host bb2localdomain type_instance steal



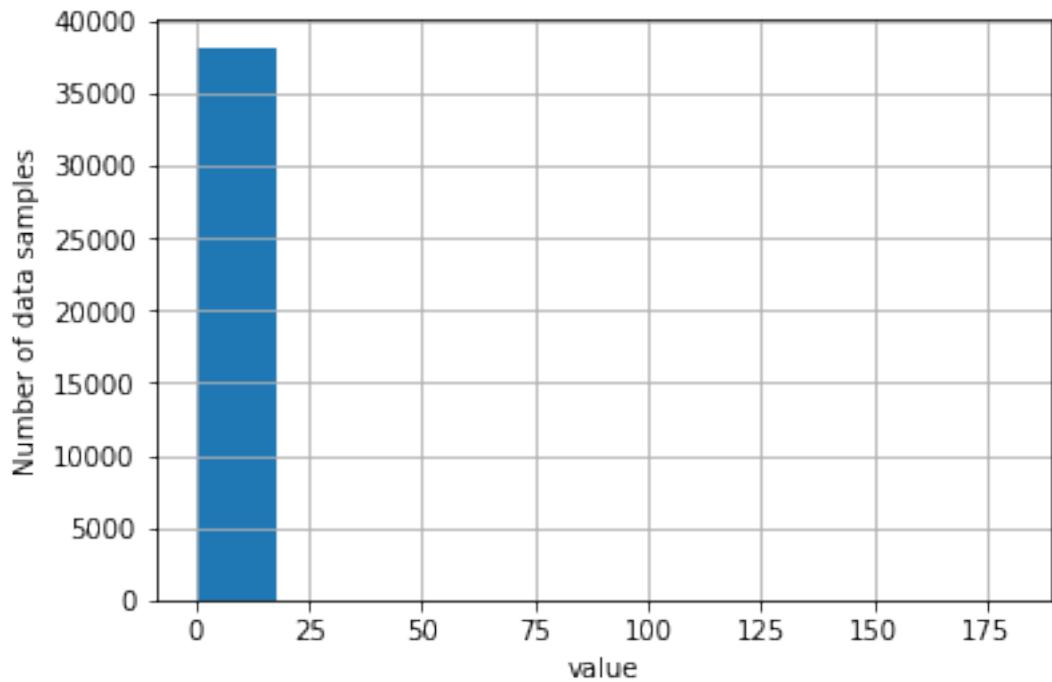
Distribution of data for cpu_value host bb2localdomain type_instance system



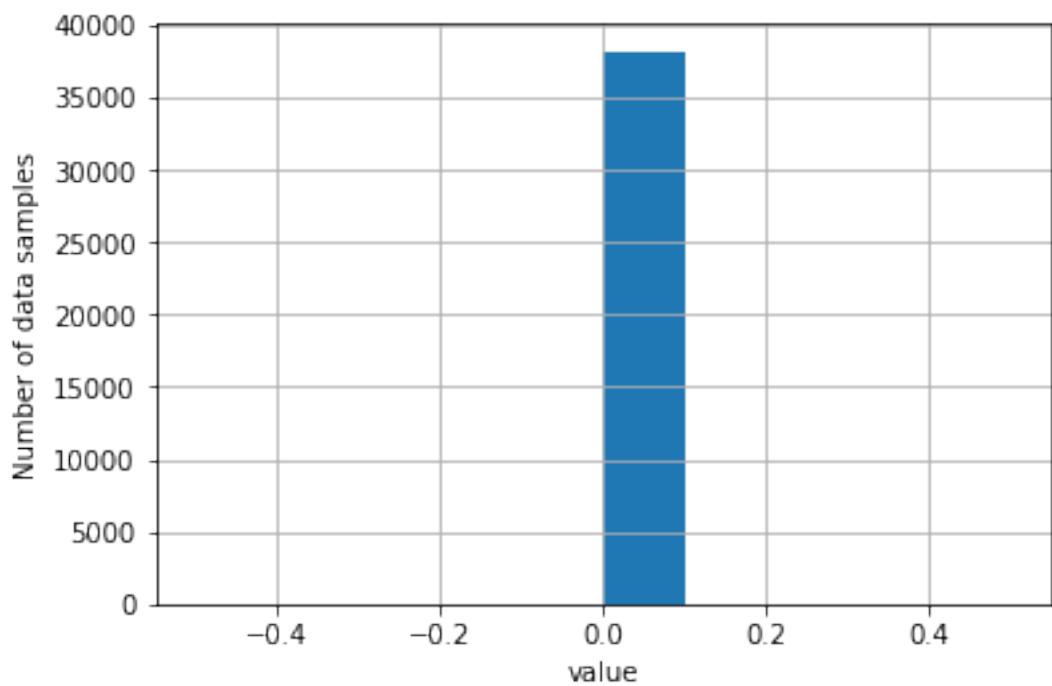
Distribution of data for cpu_value host bb2localdomain type_instance user



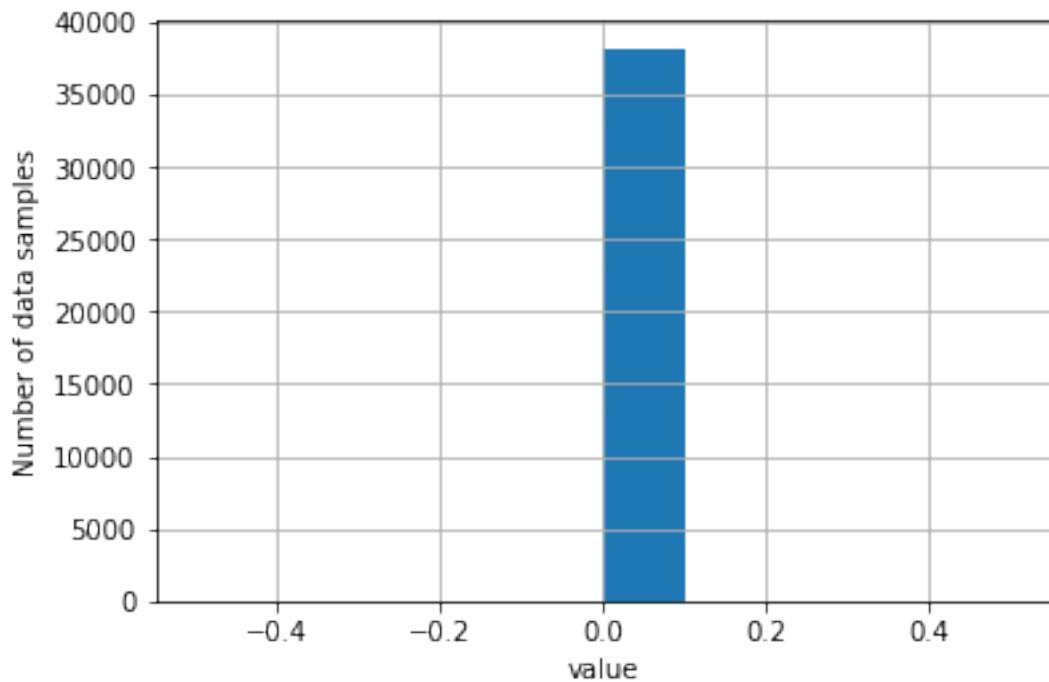
Distribution of data for cpu_value host bb2localdomain type_instance wait



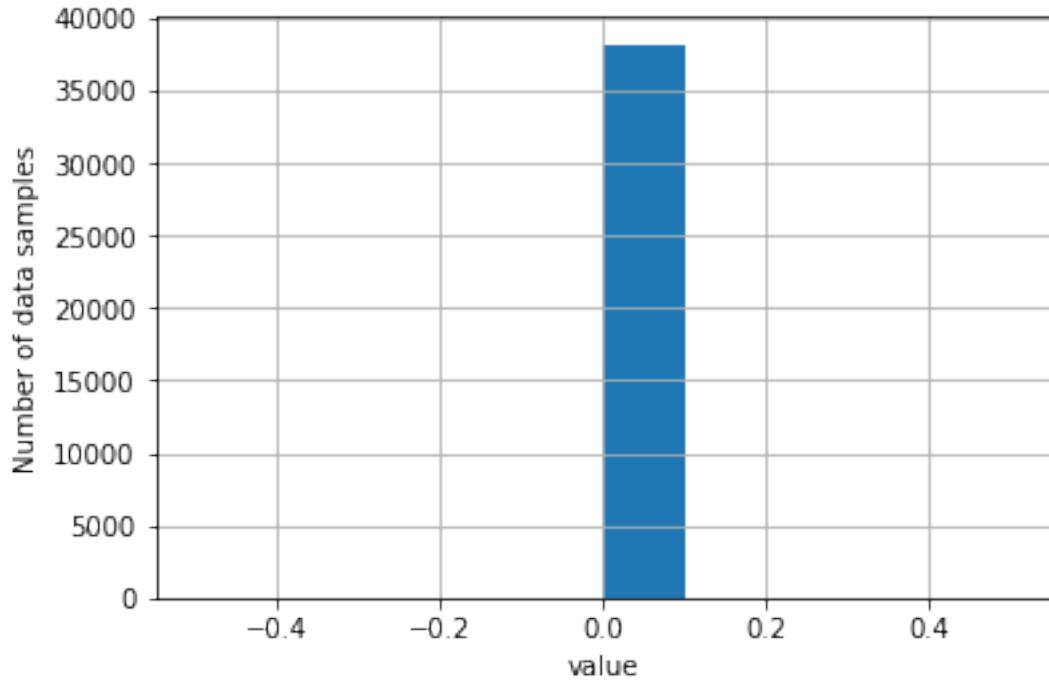
Distribution of data for interface_tx host bb2localdomain instance lo type if_dropped



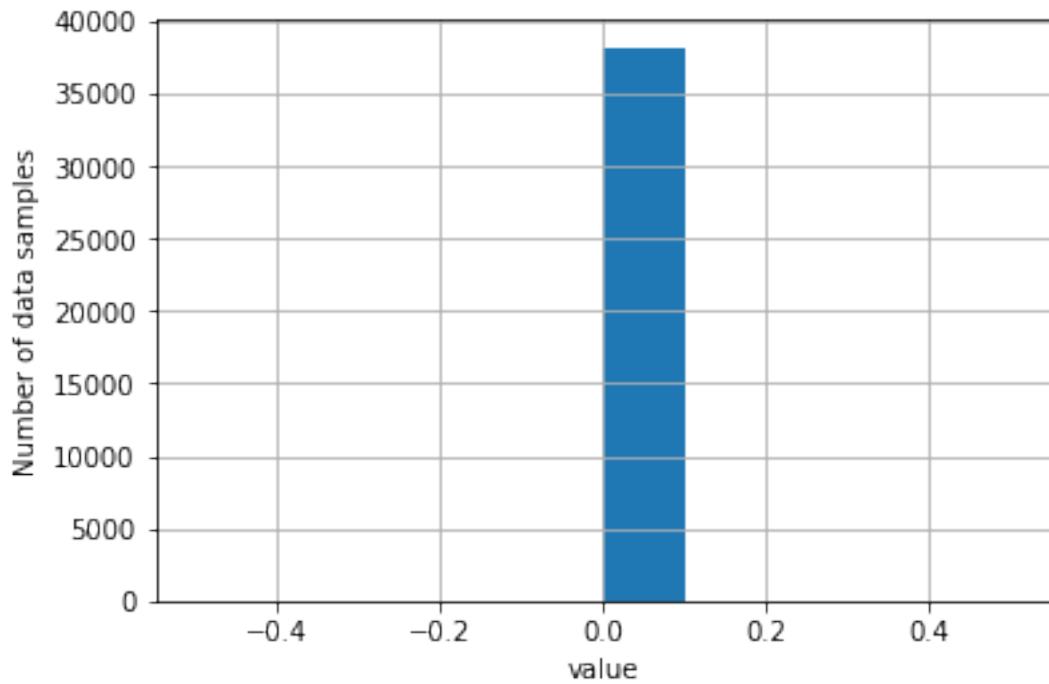
Distribution of data for interface_tx host bb2localdomain instance lo type if_errors



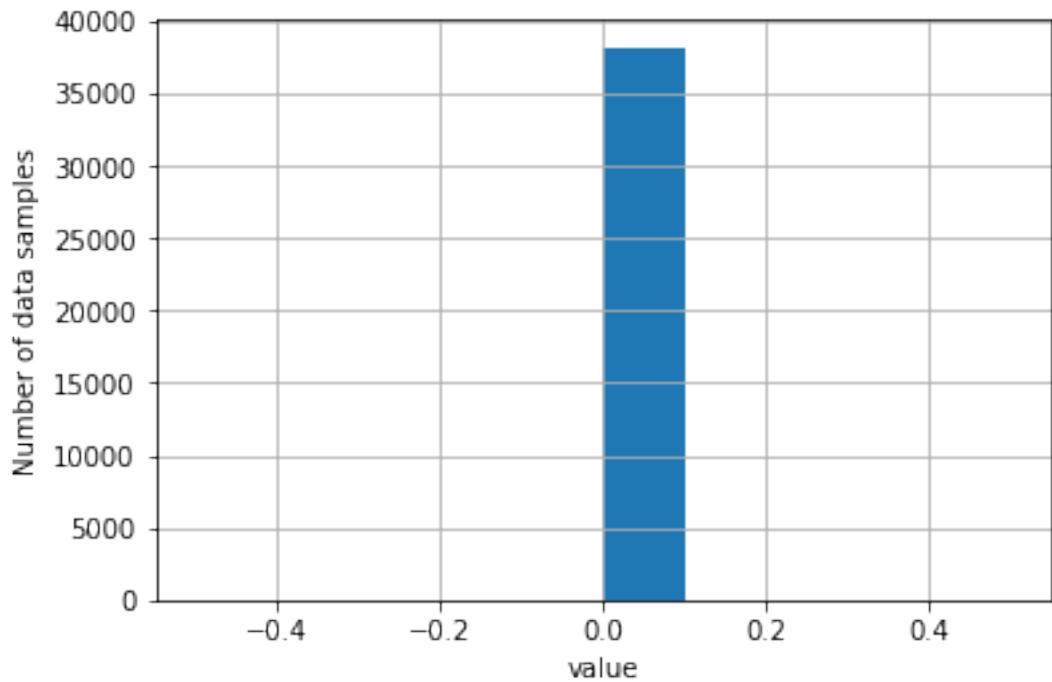
Distribution of data for interface_tx host bb2localdomain instance lo type if_octets



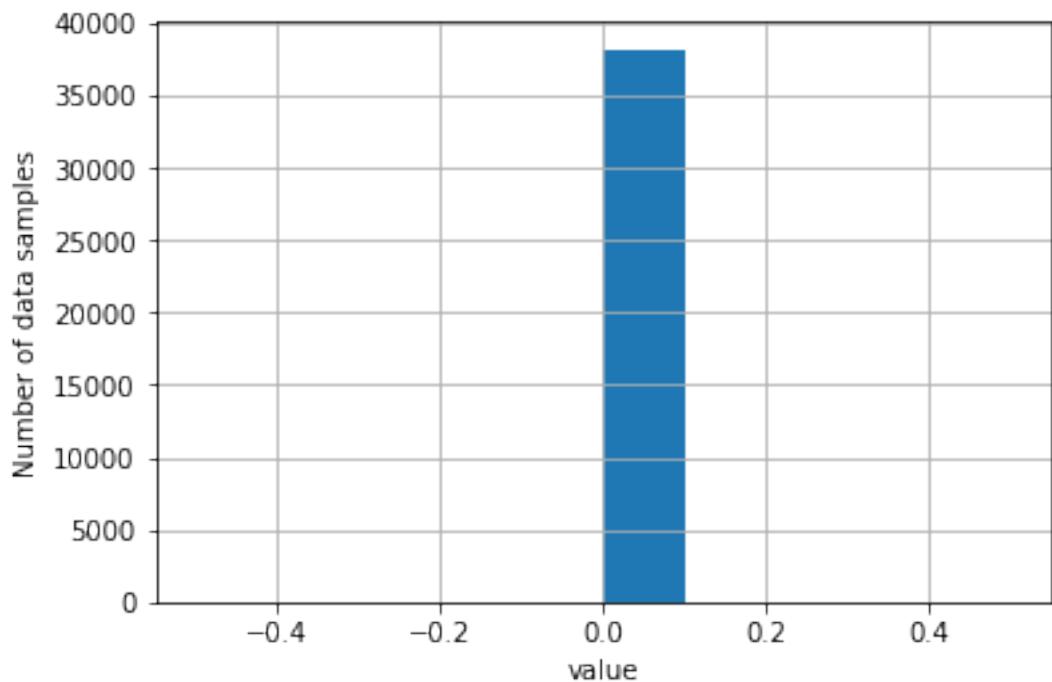
Distribution of data for interface_tx host bb2localdomain instance lo type if_packets



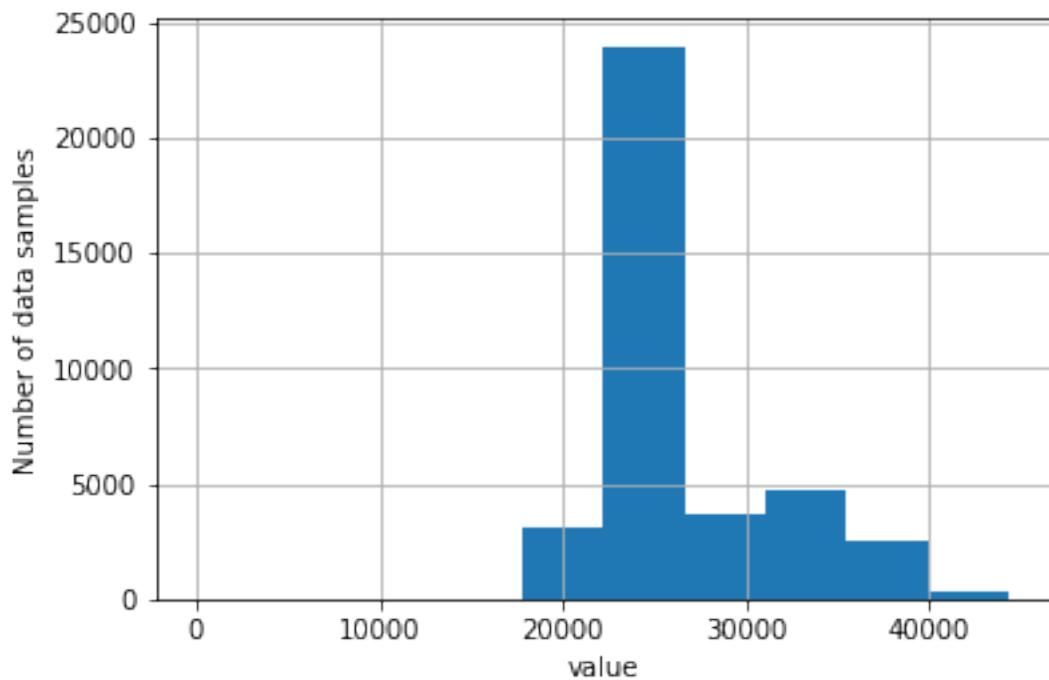
Distribution of data for interface_tx host bb2localdomain instance wlan0 type if_dropped



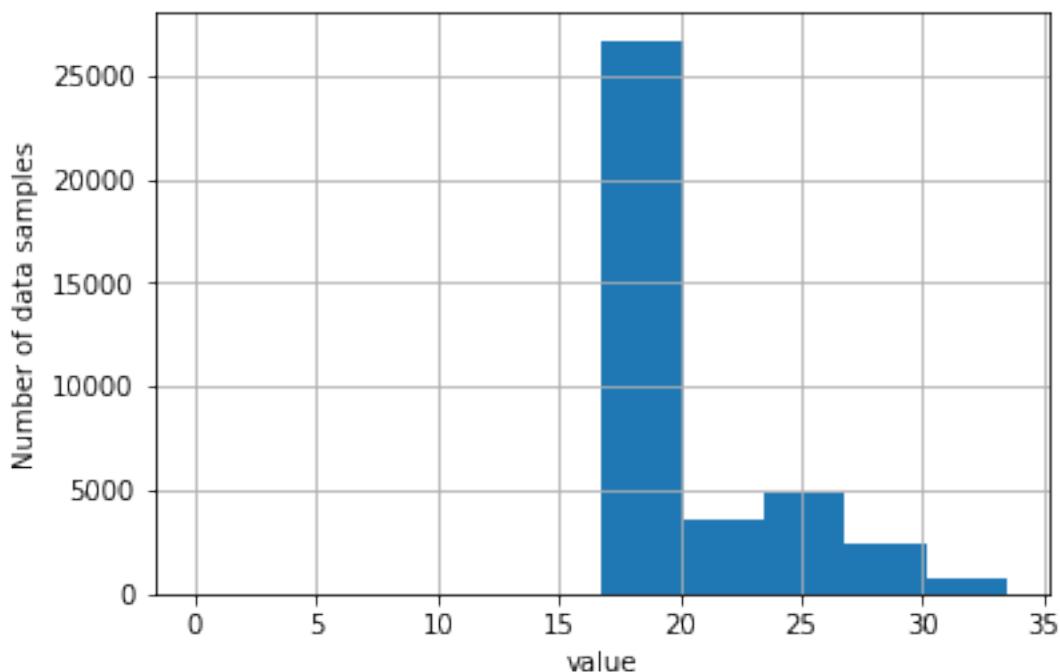
Distribution of data for interface_tx host bb2localdomain instance wlan0 type if_errors



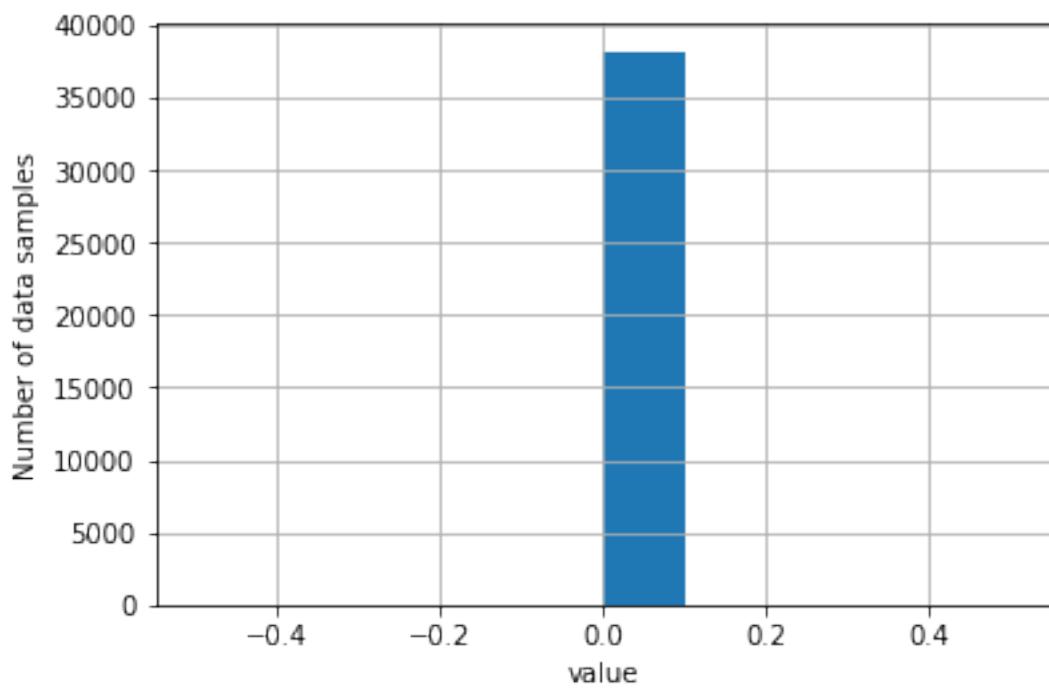
Distribution of data for interface_tx host bb2localdomain instance wlan0 type if_octets



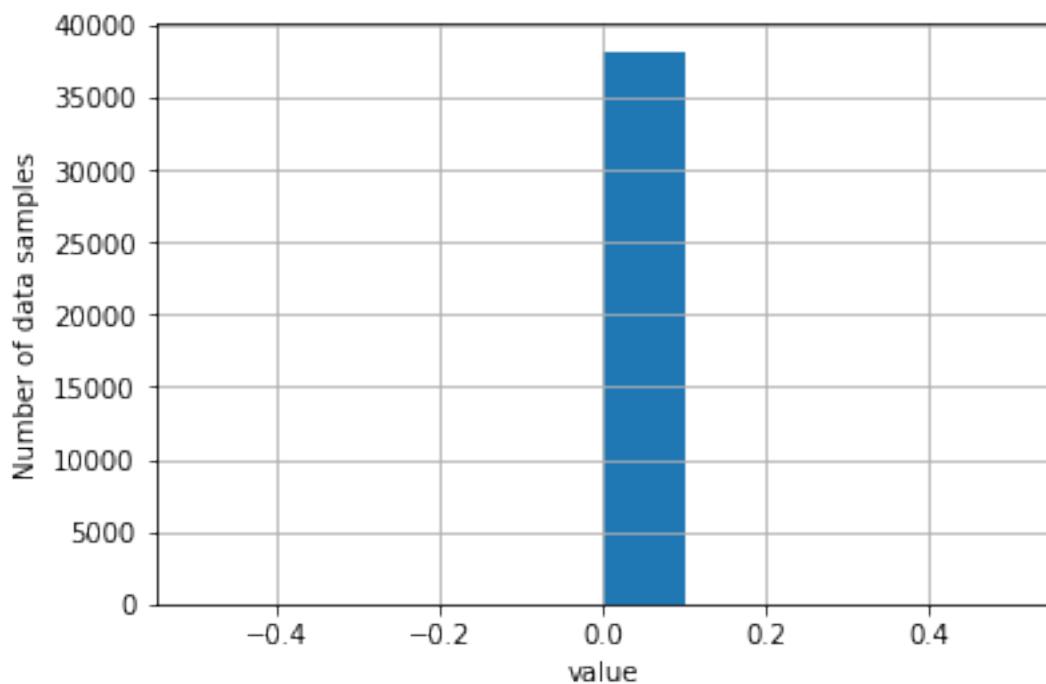
Distribution of data for interface_tx host bb2localdomain instance wlan0 type if_packets



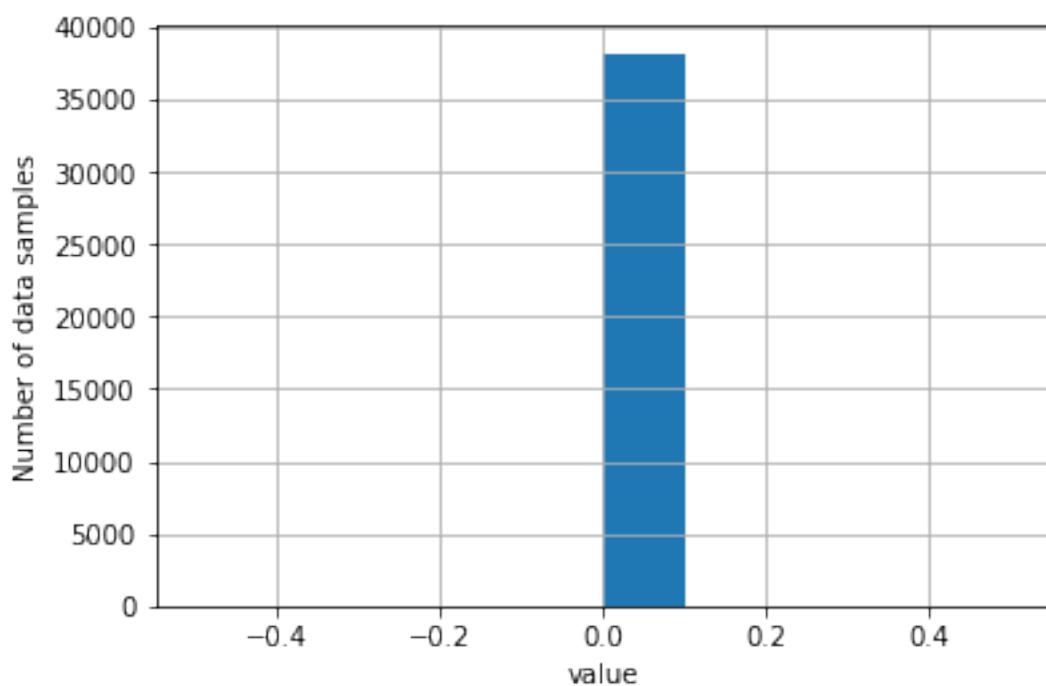
Distribution of data for interface_rx host bb2localdomain instance lo type if_dropped



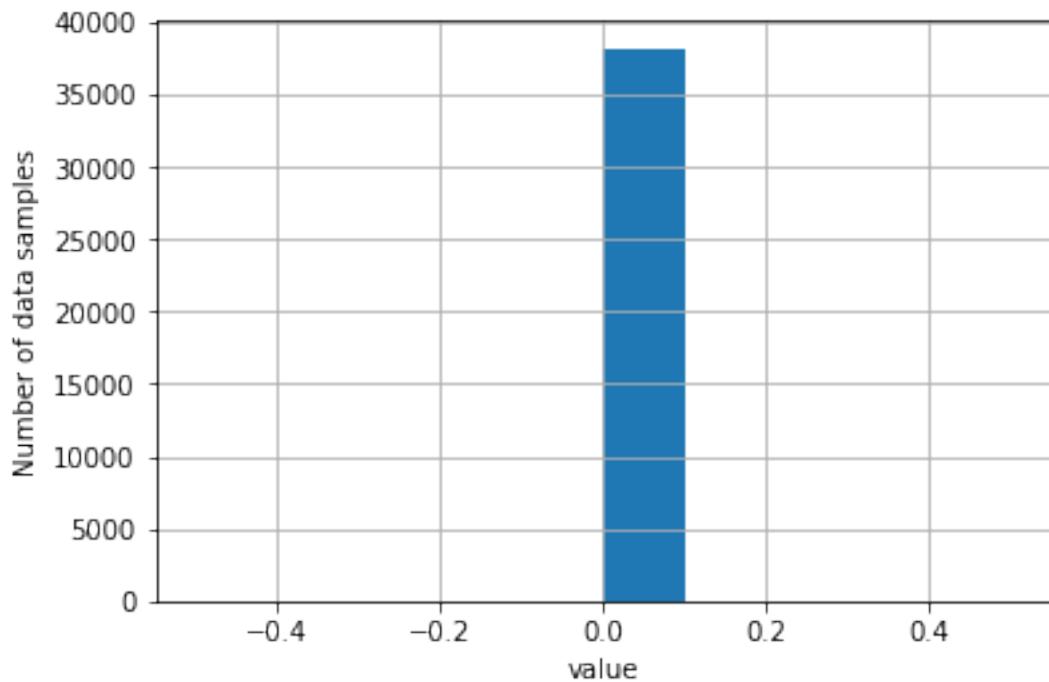
Distribution of data for interface_rx host bb2localdomain instance lo type if_errors



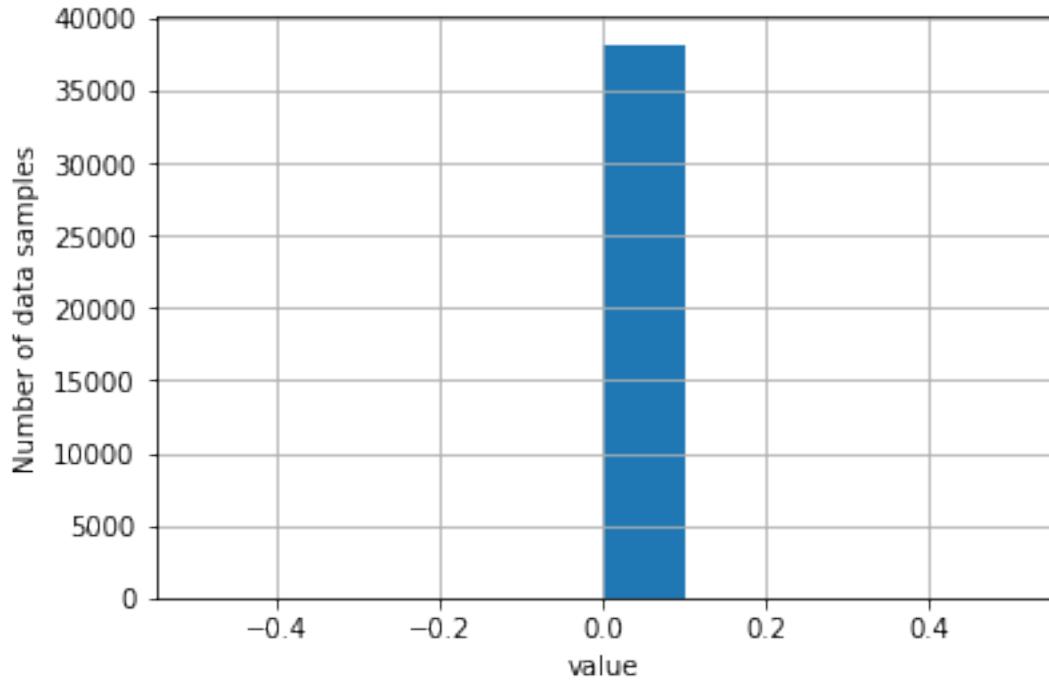
Distribution of data for interface_rx host bb2localdomain instance lo type if_octets



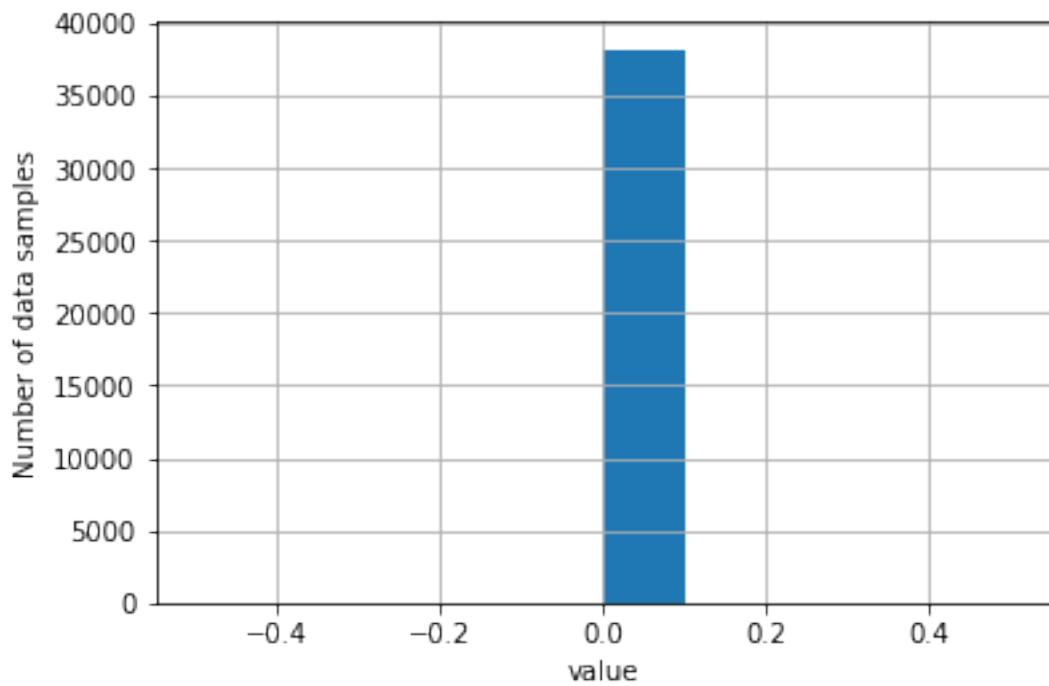
Distribution of data for interface_rx host bb2localdomain instance lo type if_packets



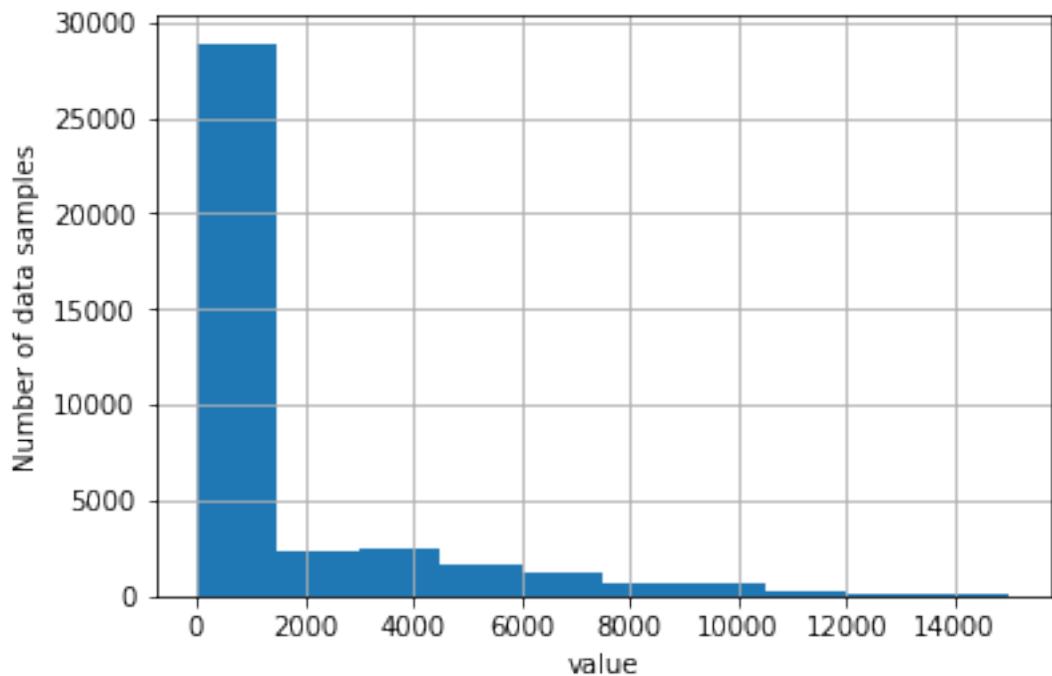
Distribution of data for interface_rx host bb2localdomain instance wlan0 type if_dropped



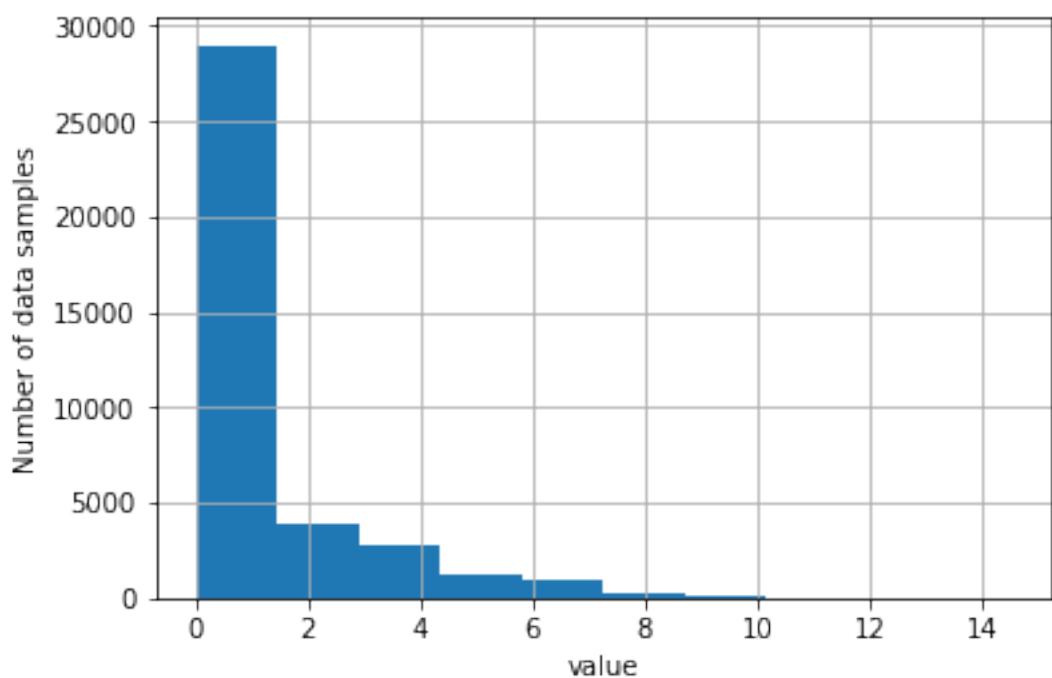
Distribution of data for interface_rx host bb2localdomain instance wlan0 type if_errors



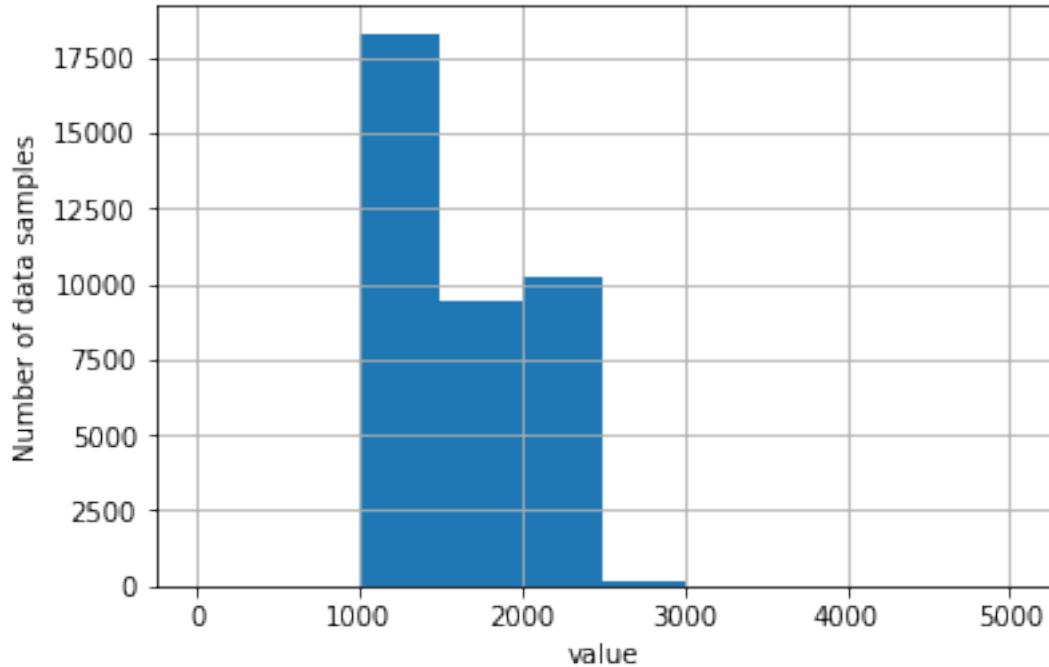
Distribution of data for interface_rx host bb2localdomain instance wlan0 type if_octets



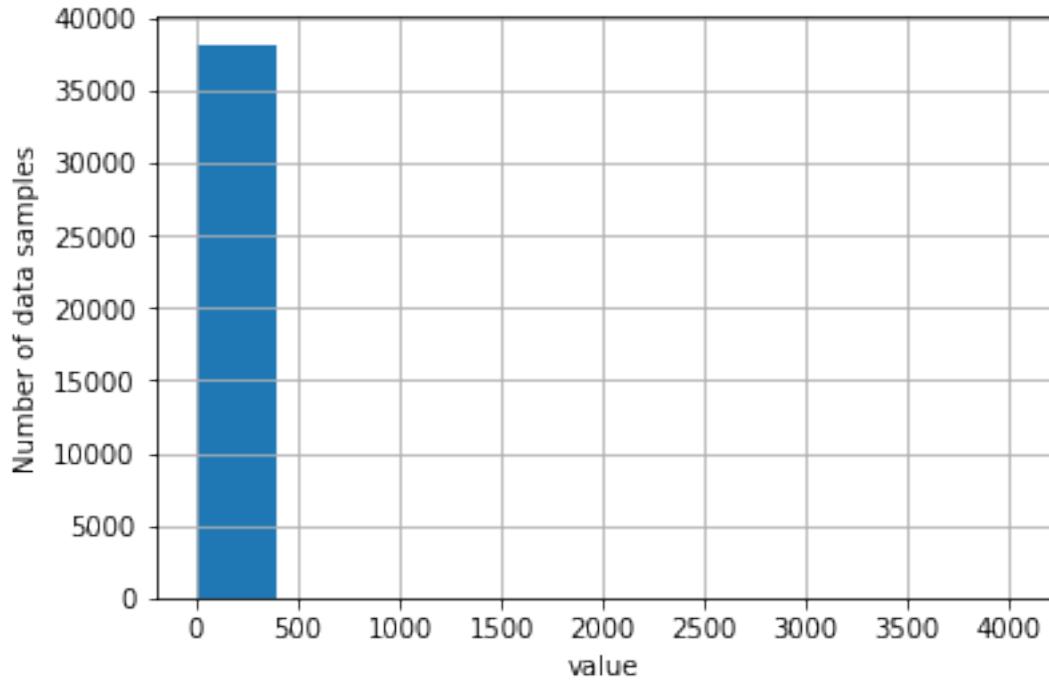
Distribution of data for interface_rx host bb2localdomain instance wlan0 type if_packets



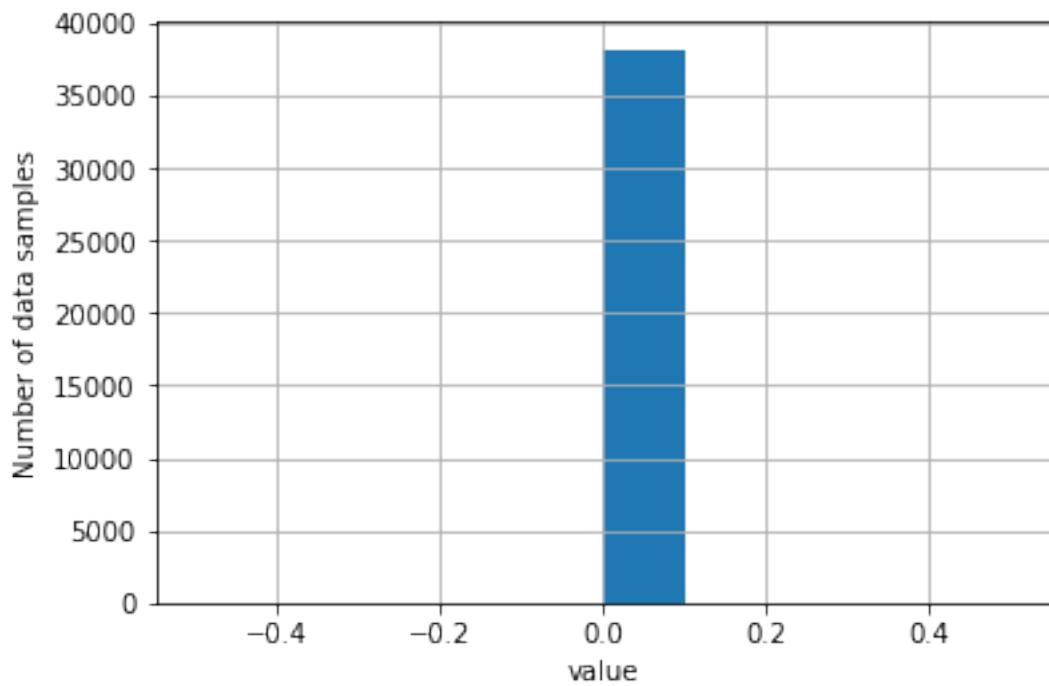
Distribution of data for contextswitch_value host bb2localdomain type contextswitch



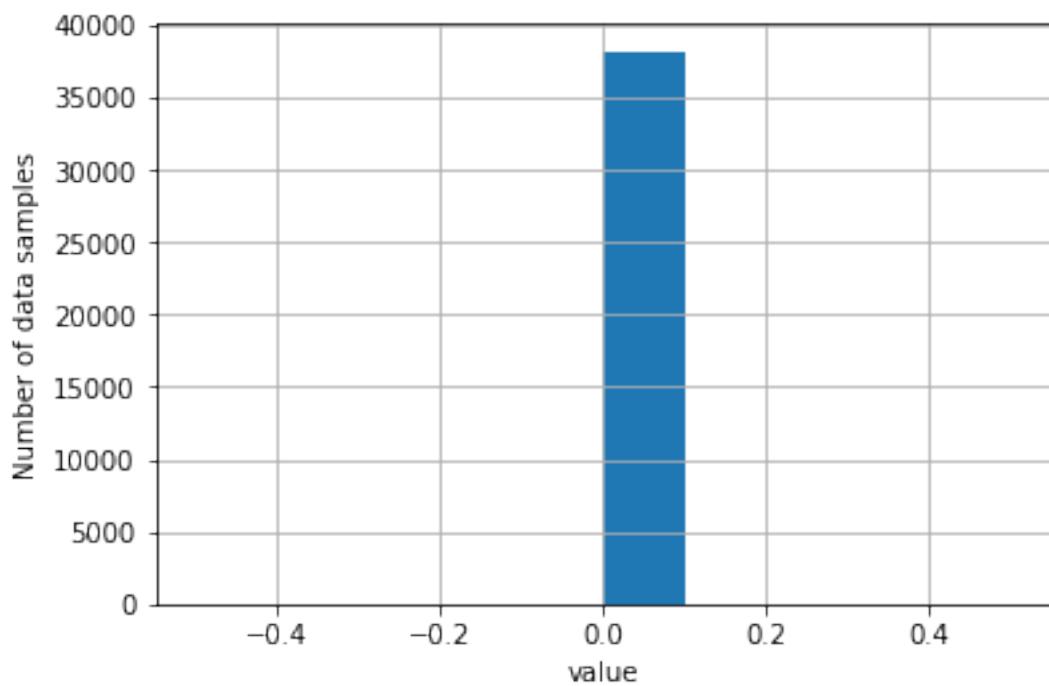
Distribution of data for disk_io_time host bb2localdomain instance mmcblk1 type disk_io_time



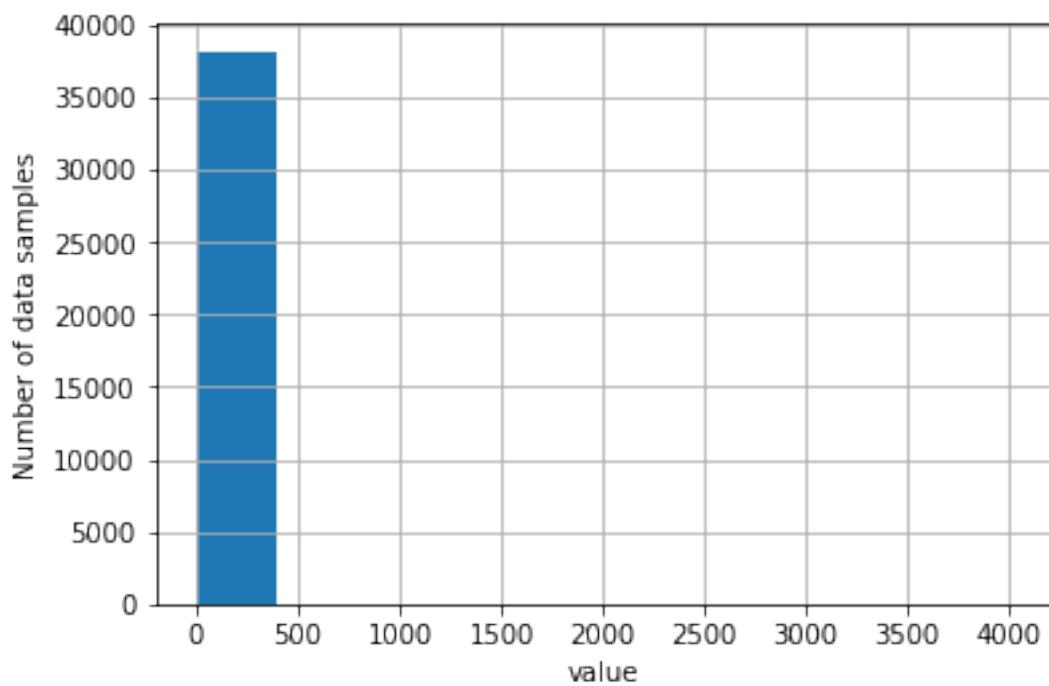
Distribution of data for disk_io_time host bb2localdomain instance mmcblk1boot0 type disk_io_t



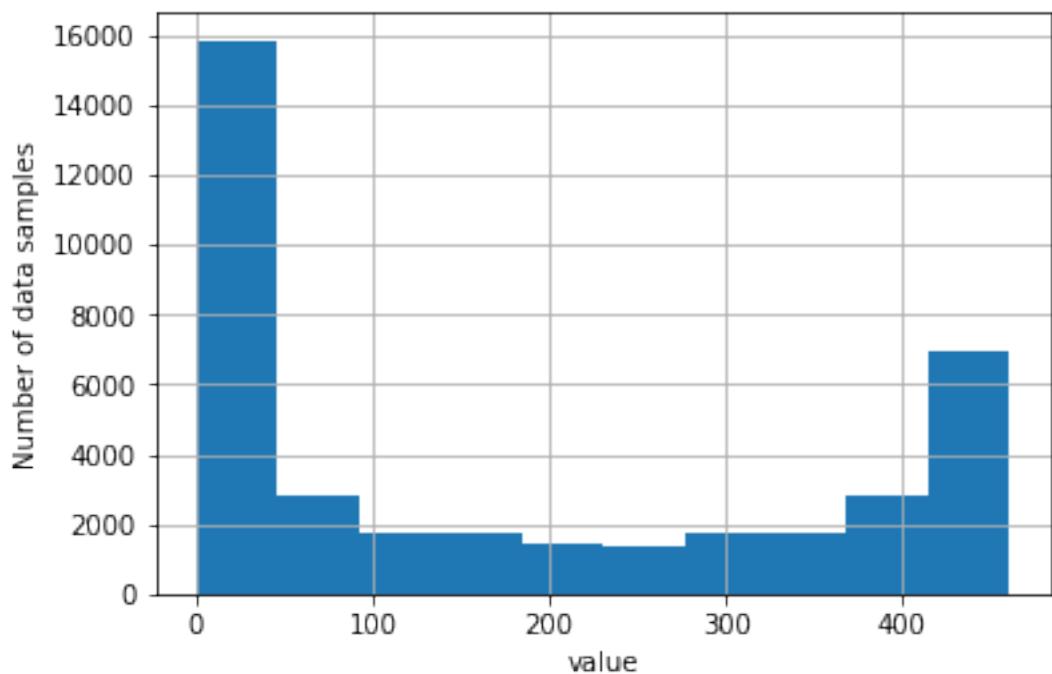
Distribution of data for disk_io_time host bb2localdomain instance mmcblk1boot1 type disk_io_time



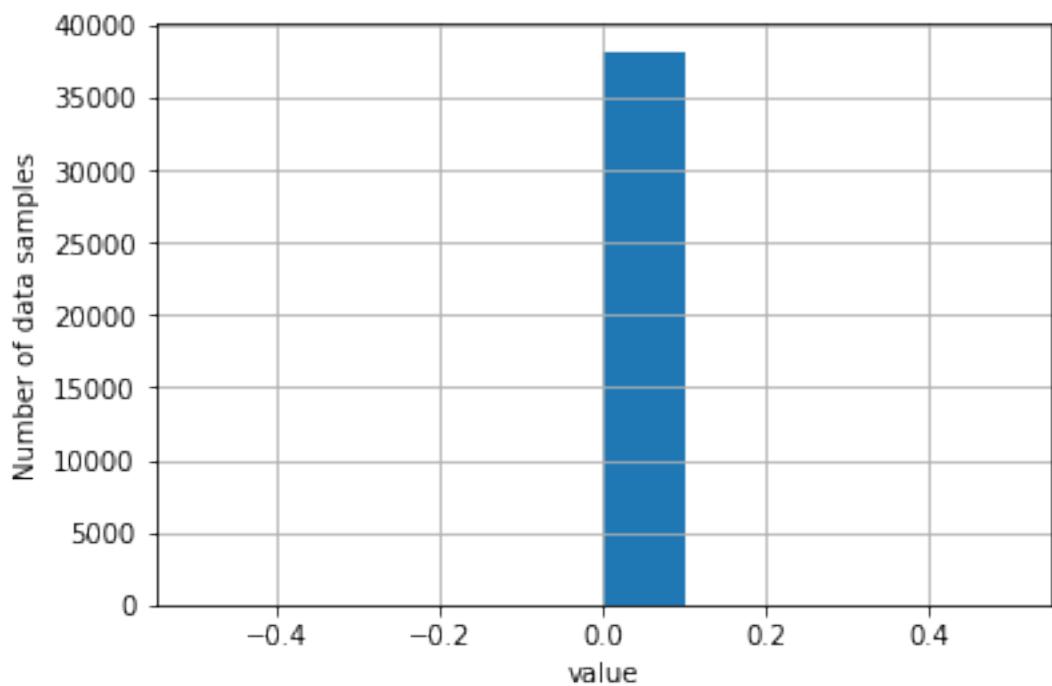
Distribution of data for disk_io_time host bb2localdomain instance mmcblk1p1 type disk_io_time



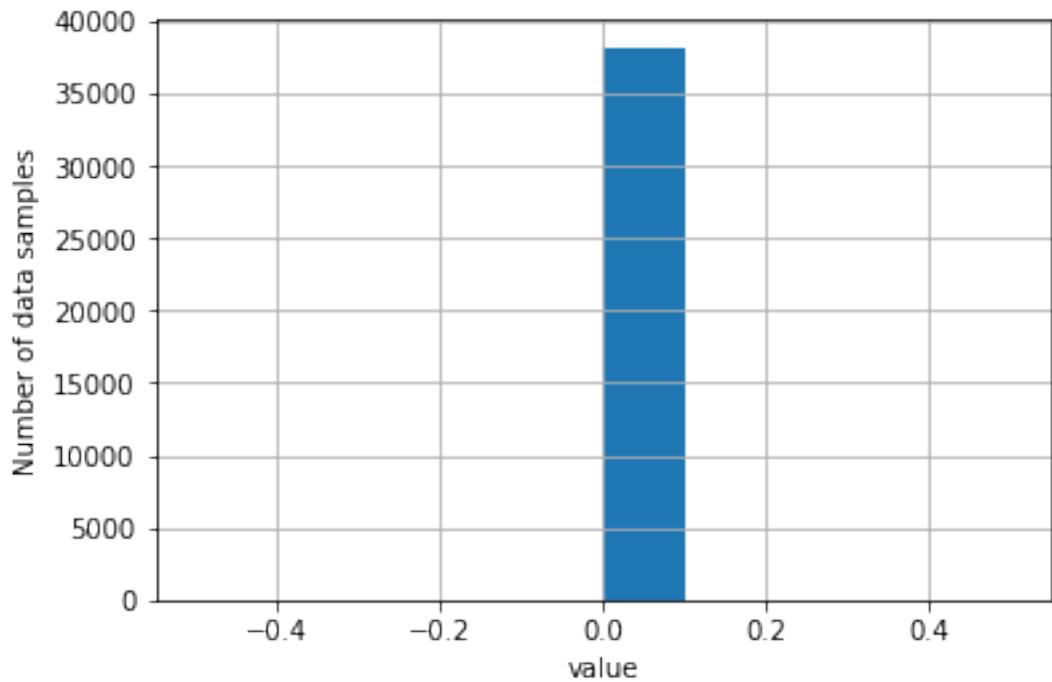
```
(38200, 29)
Index(['cpu_value host bb3localdomain type_instance idle',
       'cpu_value host bb3localdomain type_instance interrupt',
       'cpu_value host bb3localdomain type_instance nice',
       'cpu_value host bb3localdomain type_instance softirq',
       'cpu_value host bb3localdomain type_instance steal',
       'cpu_value host bb3localdomain type_instance system',
       'cpu_value host bb3localdomain type_instance user',
       'cpu_value host bb3localdomain type_instance wait',
       'interface_tx host bb3localdomain instance lo type if_dropped',
       'interface_tx host bb3localdomain instance lo type if_errors',
       'interface_tx host bb3localdomain instance lo type if_octets',
       'interface_tx host bb3localdomain instance lo type if_packets',
       'interface_tx host bb3localdomain instance wlan0 type if_dropped',
       'interface_tx host bb3localdomain instance wlan0 type if_errors',
       'interface_tx host bb3localdomain instance wlan0 type if_octets',
       'interface_tx host bb3localdomain instance wlan0 type if_packets',
       'interface_rx host bb3localdomain instance lo type if_dropped',
       'interface_rx host bb3localdomain instance lo type if_errors',
       'interface_rx host bb3localdomain instance lo type if_octets',
       'interface_rx host bb3localdomain instance lo type if_packets',
       'interface_rx host bb3localdomain instance wlan0 type if_dropped',
       'interface_rx host bb3localdomain instance wlan0 type if_errors',
       'interface_rx host bb3localdomain instance wlan0 type if_octets',
       'interface_rx host bb3localdomain instance wlan0 type if_packets',
       'contextswitch_value host bb3localdomain type contextswitch',
       'disk_io_time host bb3localdomain instance mmcblk1 type disk_io_time',
       'disk_io_time host bb3localdomain instance mmcblk1boot0 type disk_io_time',
       'disk_io_time host bb3localdomain instance mmcblk1boot1 type disk_io_time',
       'disk_io_time host bb3localdomain instance mmcblk1p1 type disk_io_time'],
      dtype='object')
Distribution of data for cpu_value host bb3localdomain type_instance idle
```



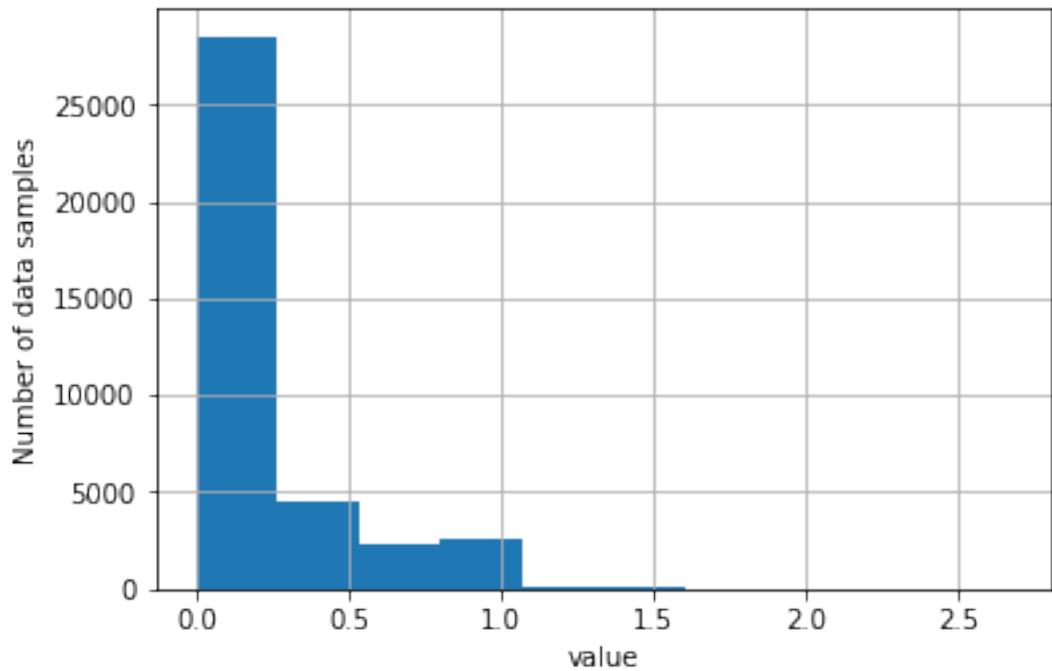
Distribution of data for cpu_value host bb3localdomain type_instance interrupt



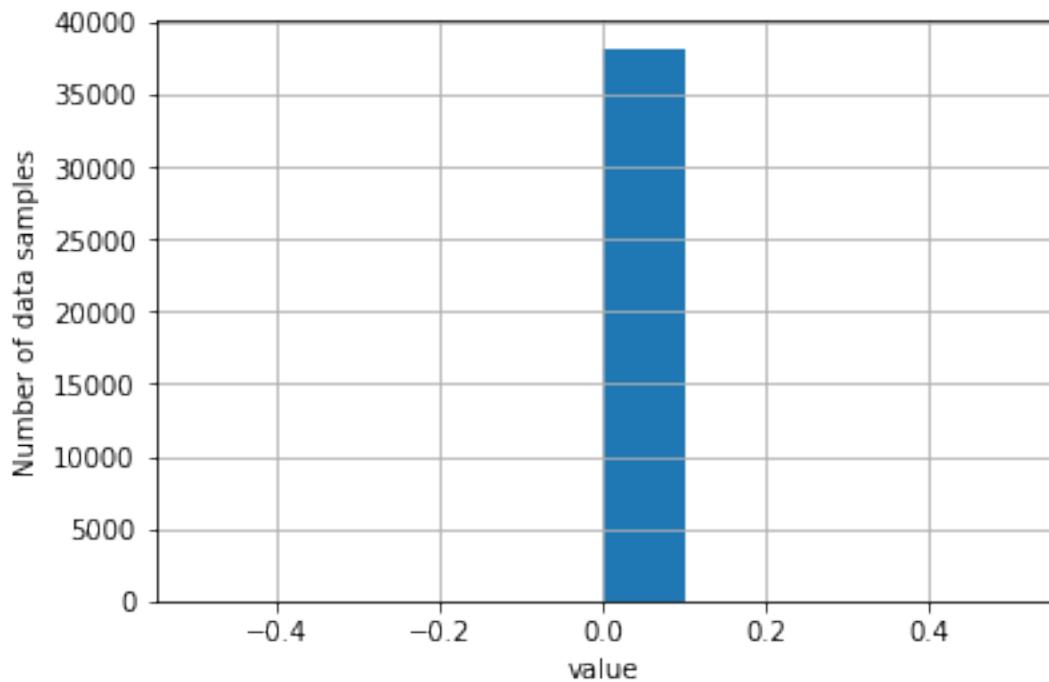
Distribution of data for cpu_value host bb3localdomain type_instance nice



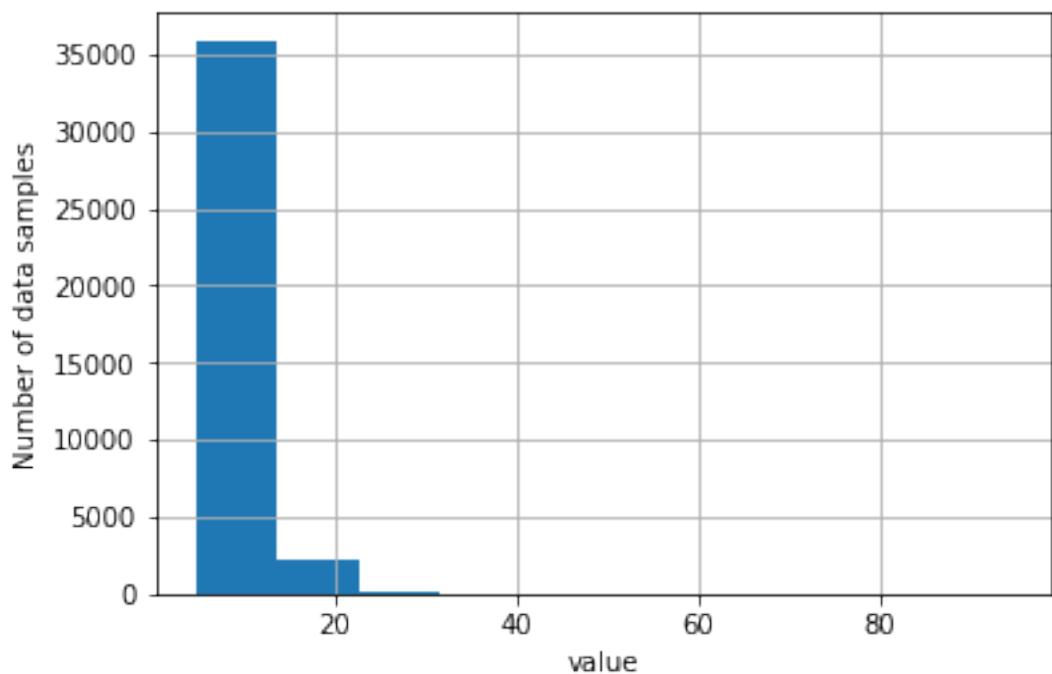
Distribution of data for cpu_value host bb3localdomain type_instance softirq



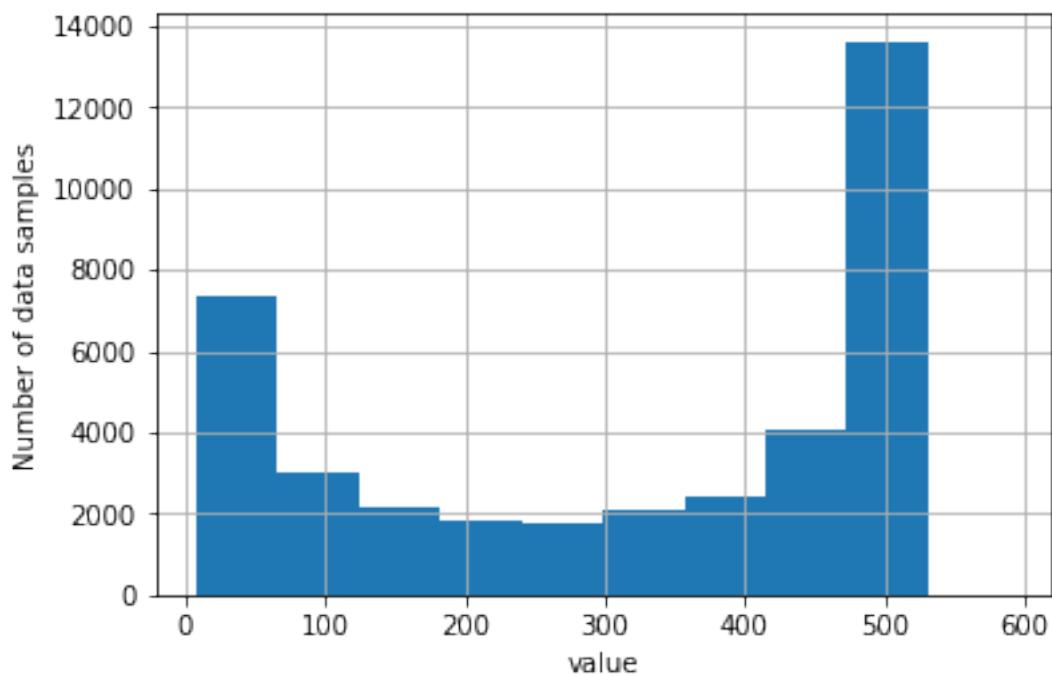
Distribution of data for cpu_value host bb3localdomain type_instance steal



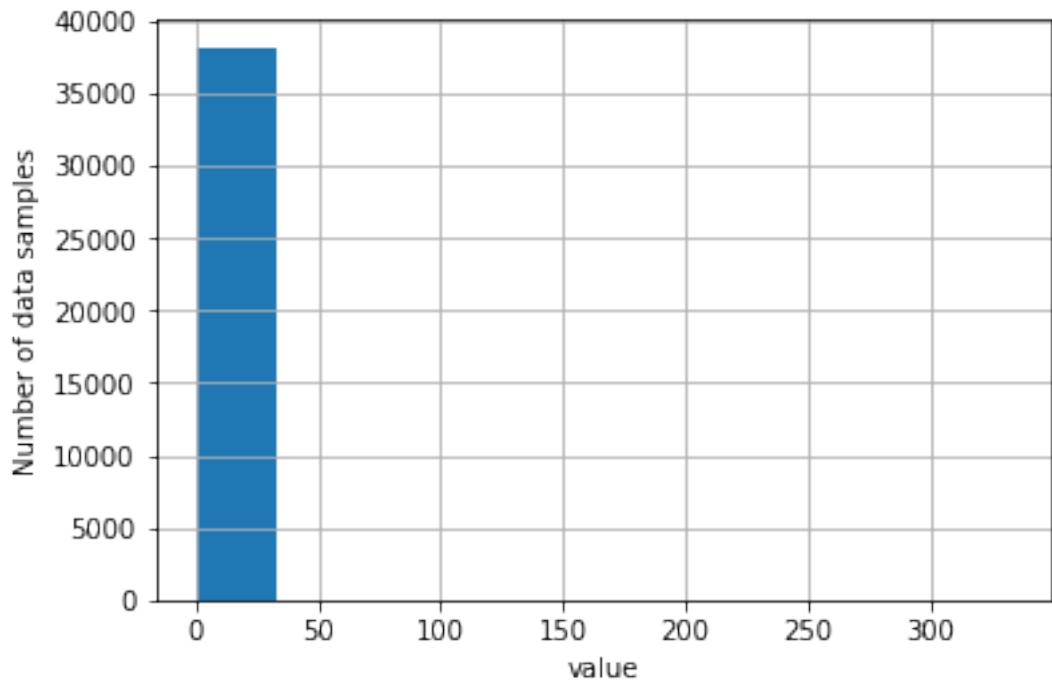
Distribution of data for cpu_value host bb3localdomain type_instance system



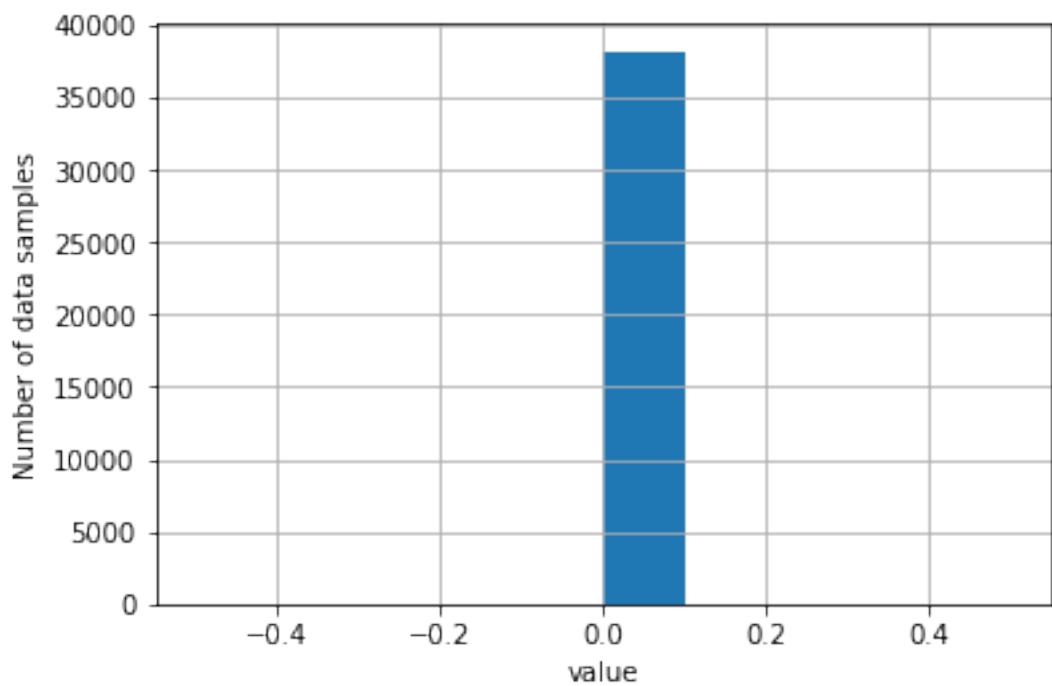
Distribution of data for cpu_value host bb3localdomain type_instance user



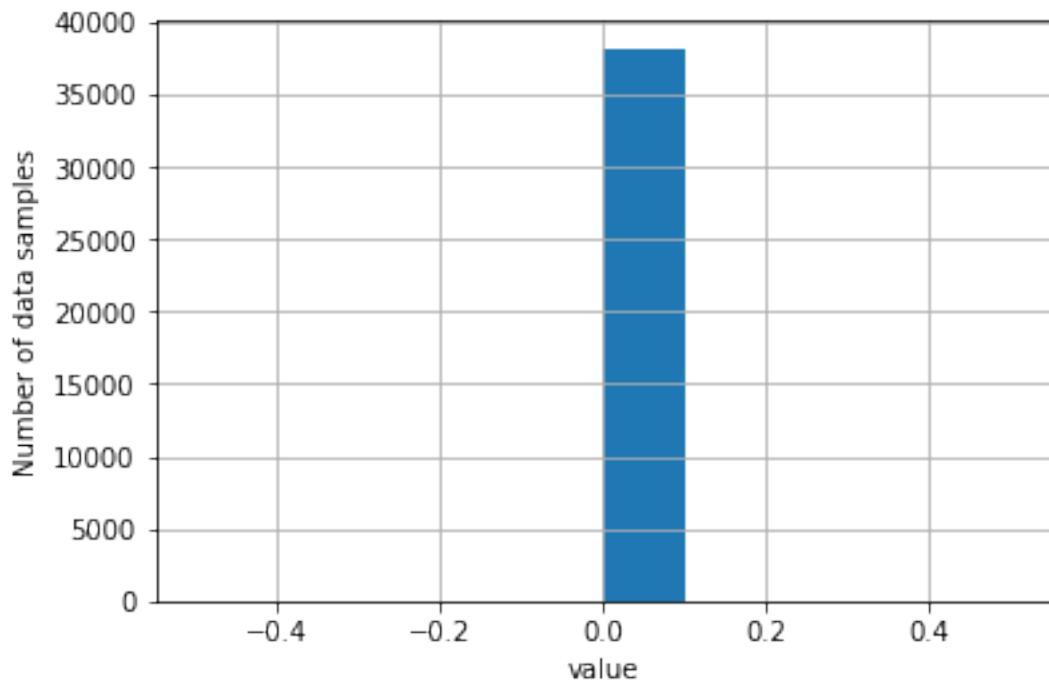
Distribution of data for cpu_value host bb3localdomain type_instance wait



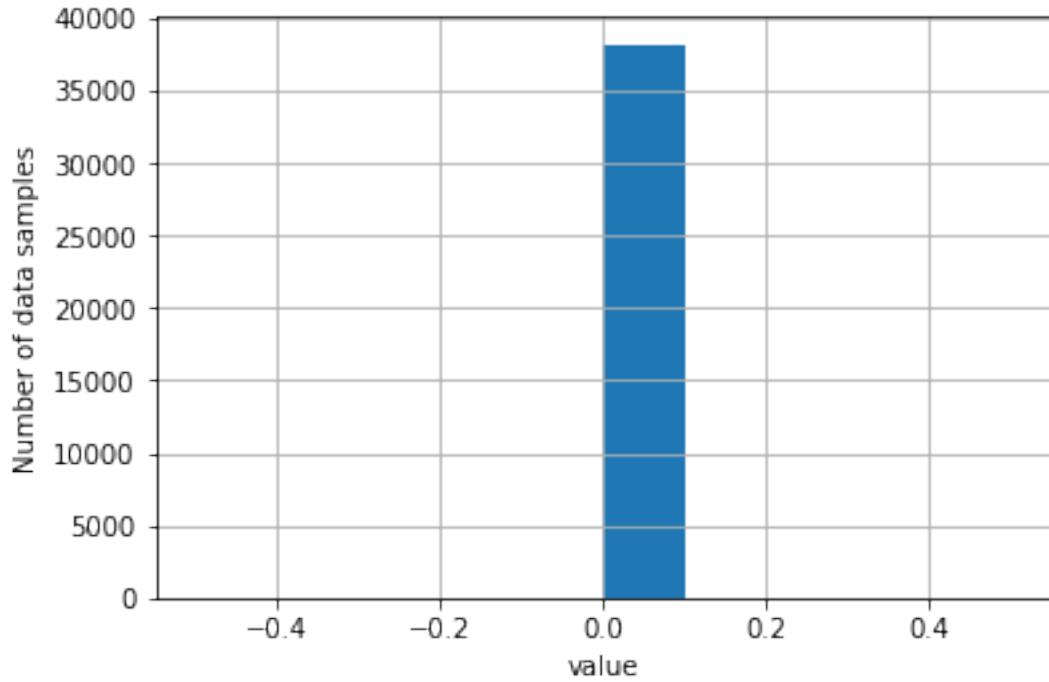
Distribution of data for interface_tx host bb3localdomain instance lo type if_dropped



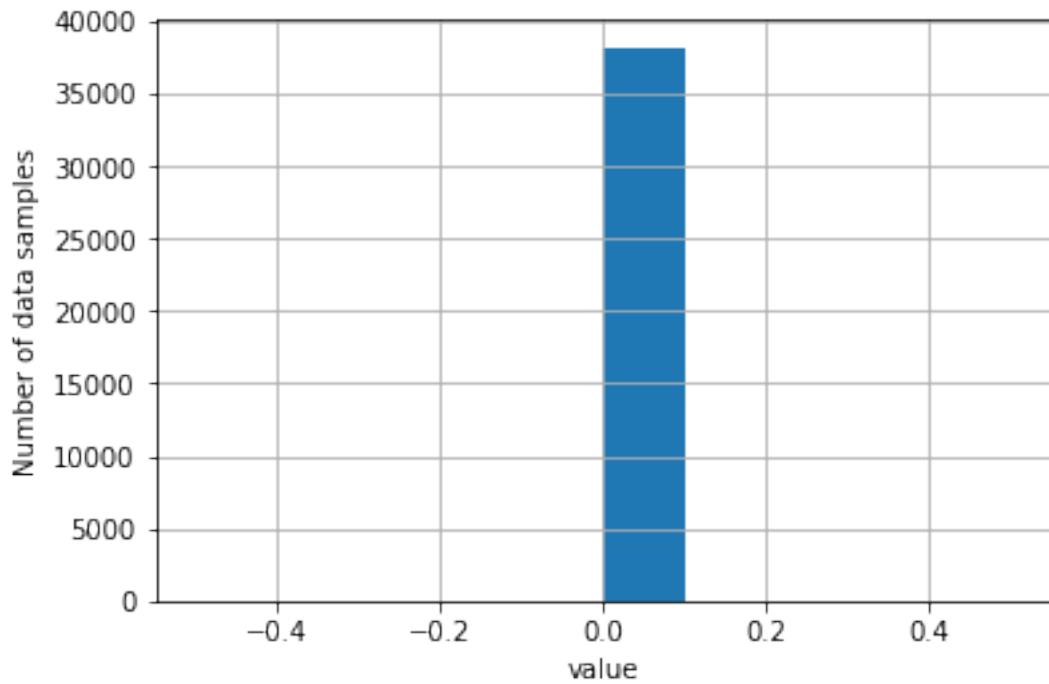
Distribution of data for interface_tx host bb3localdomain instance lo type if_errors



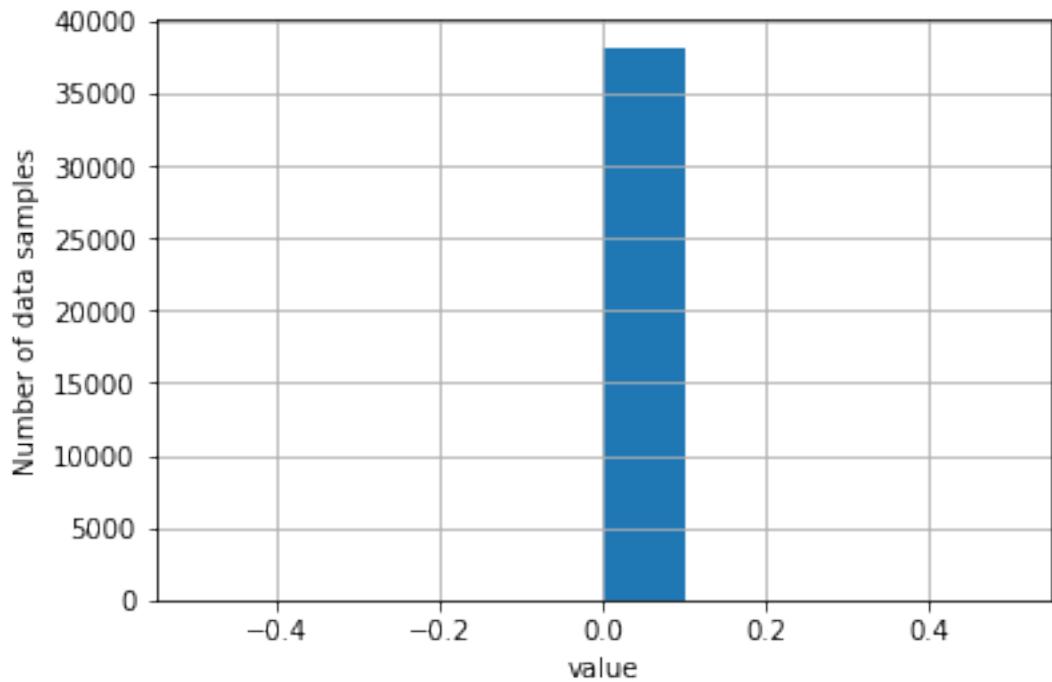
Distribution of data for interface_tx host bb3localdomain instance lo type if_octets



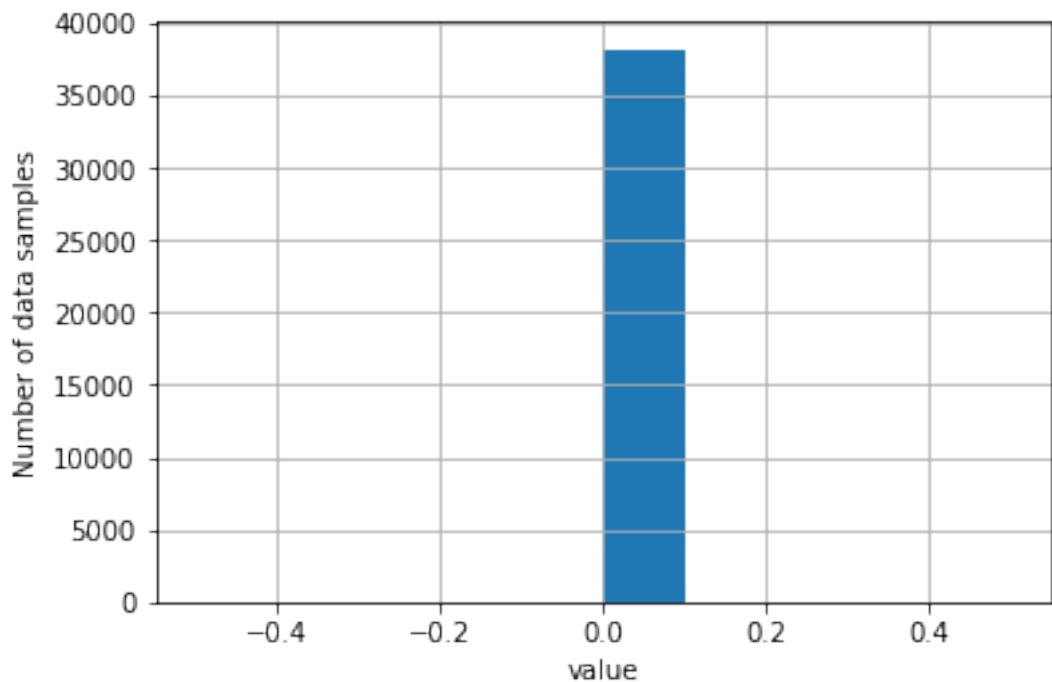
Distribution of data for interface_tx host bb3localdomain instance lo type if_packets



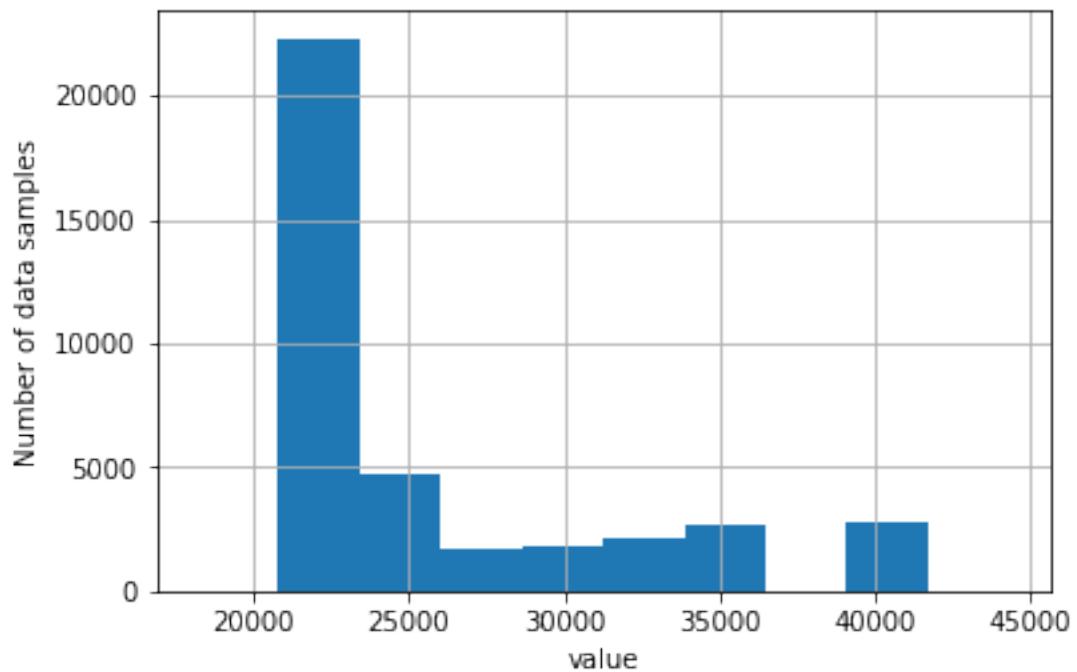
Distribution of data for interface_tx host bb3localdomain instance wlan0 type if_dropped



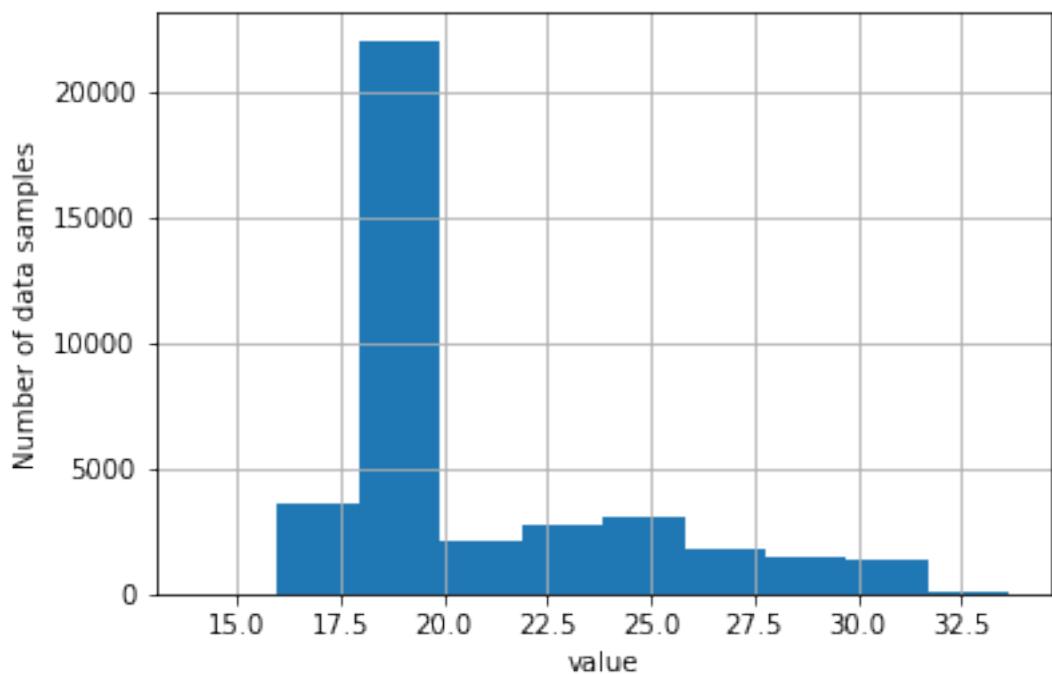
Distribution of data for interface_tx host bb3localdomain instance wlan0 type if_errors



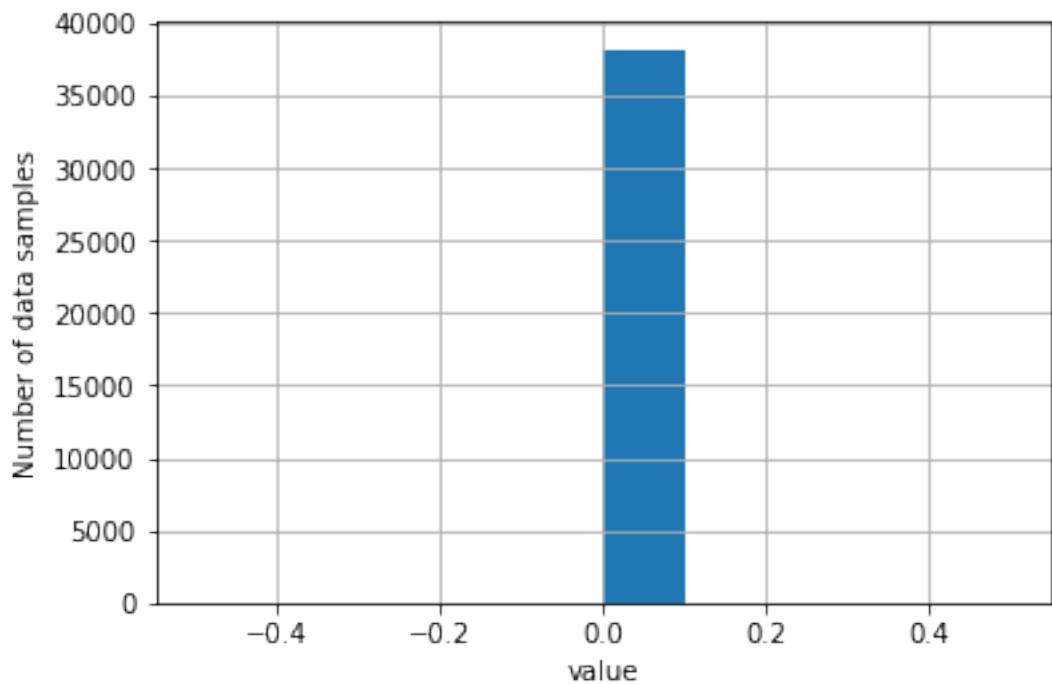
Distribution of data for interface_tx host bb3localdomain instance wlan0 type if_octets



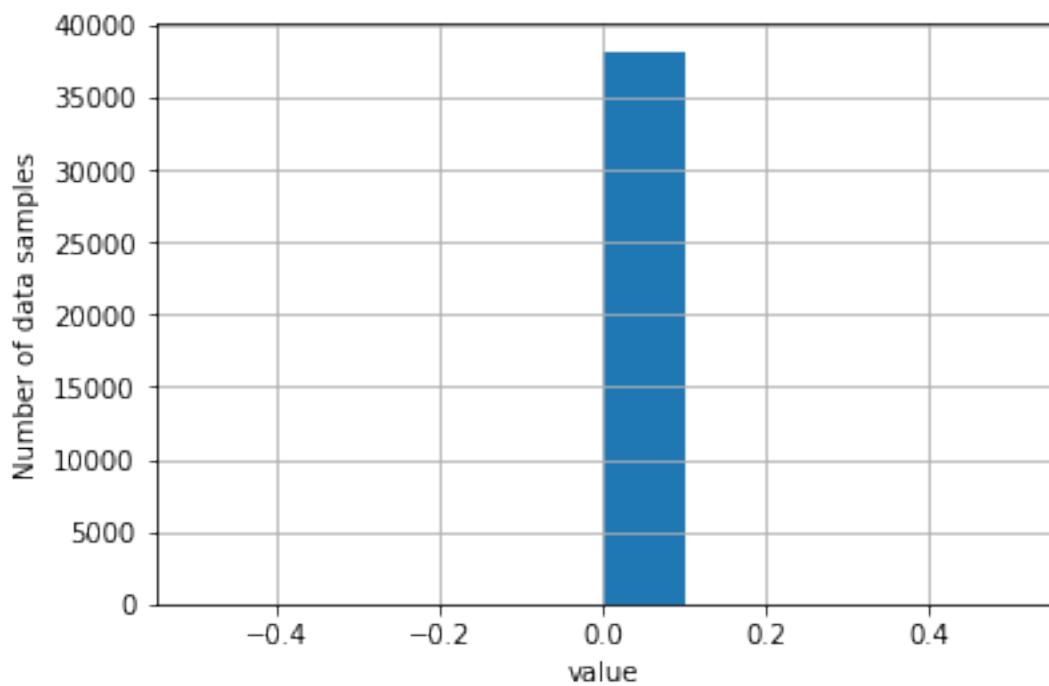
Distribution of data for interface_tx host bb3localdomain instance wlan0 type if_packets



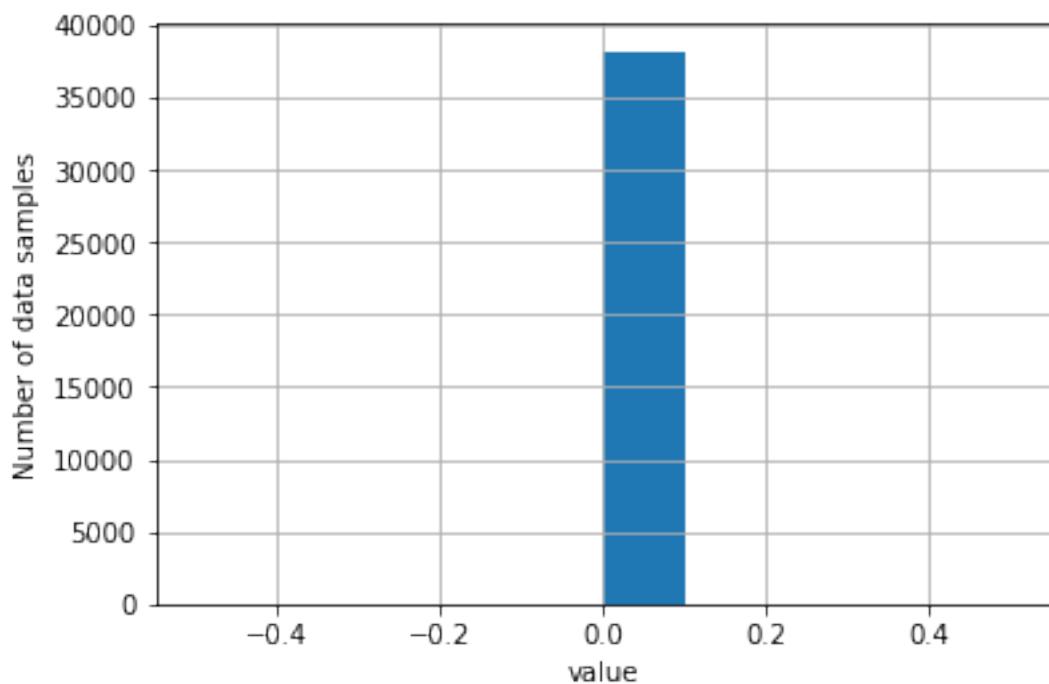
Distribution of data for interface_rx host bb3localdomain instance lo type if_dropped



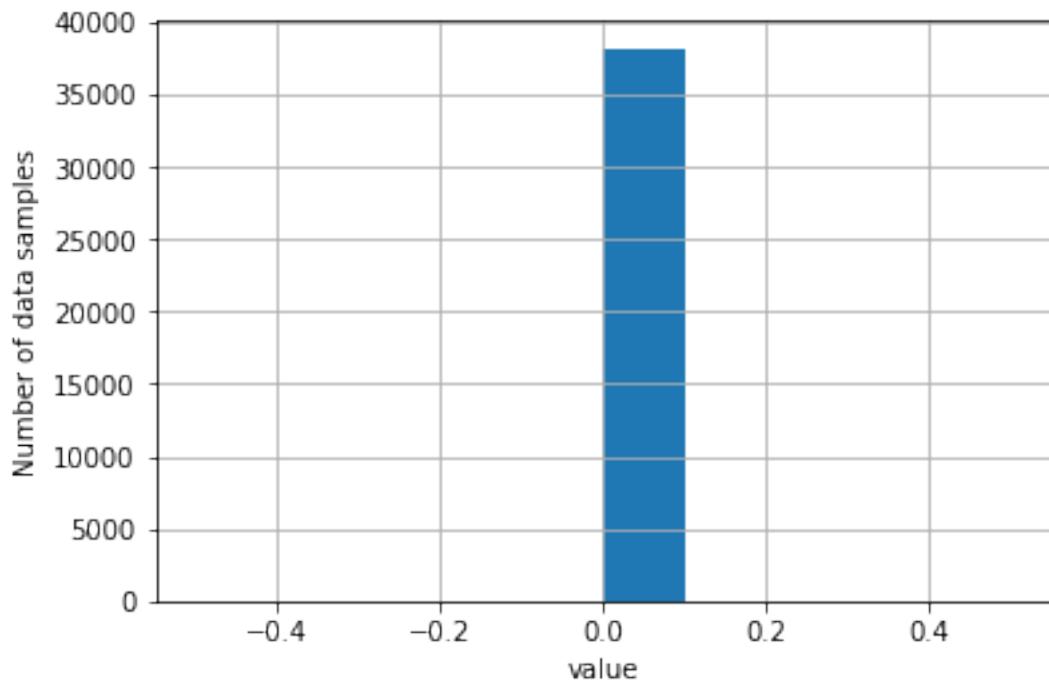
Distribution of data for interface_rx host bb3localdomain instance lo type if_errors



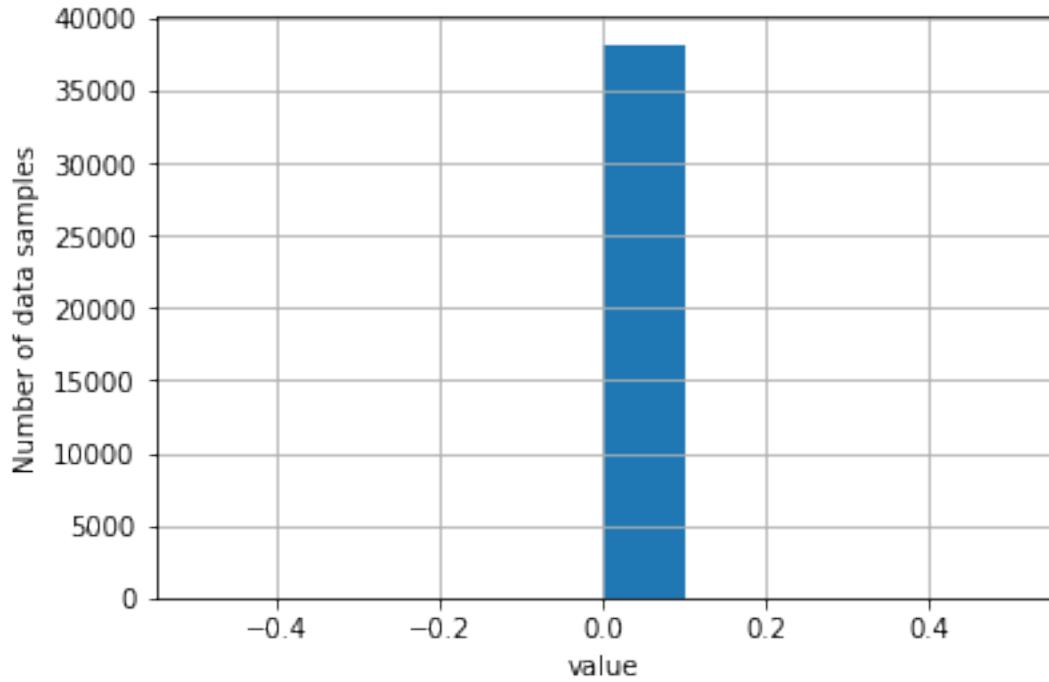
Distribution of data for interface_rx host bb3localdomain instance lo type if_octets



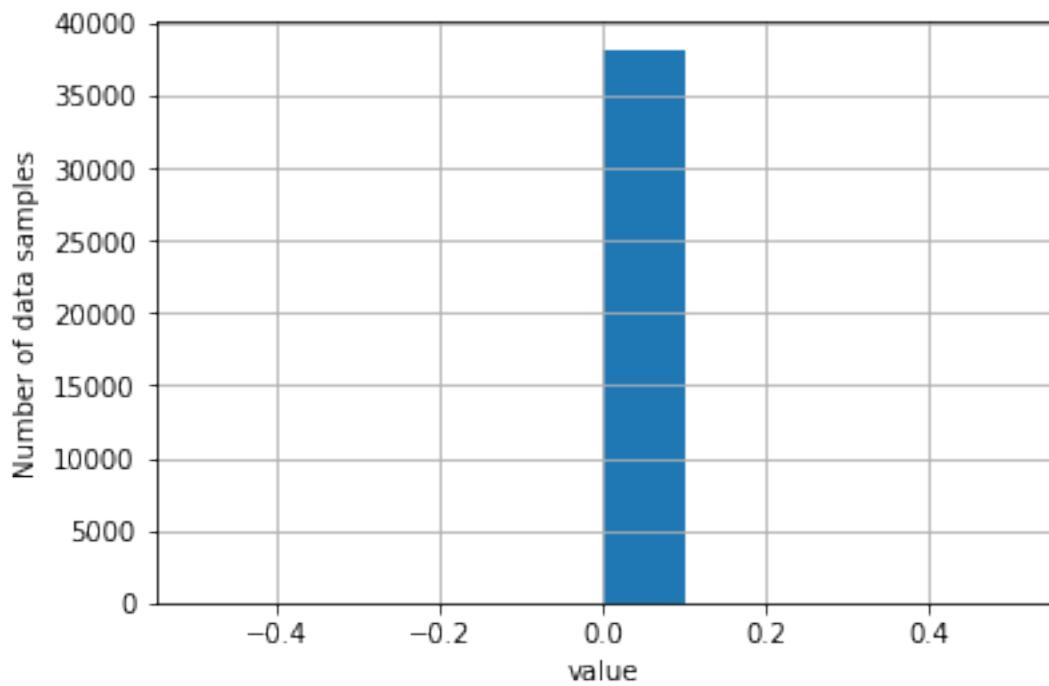
Distribution of data for interface_rx host bb3localdomain instance lo type if_packets



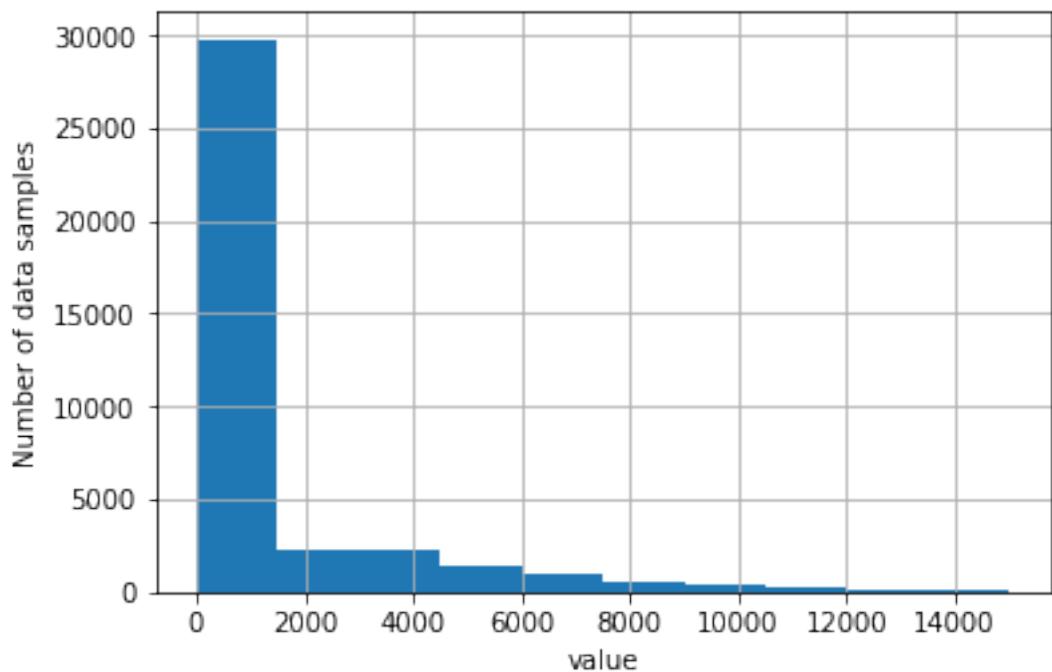
Distribution of data for interface_rx host bb3localdomain instance wlan0 type if_dropped



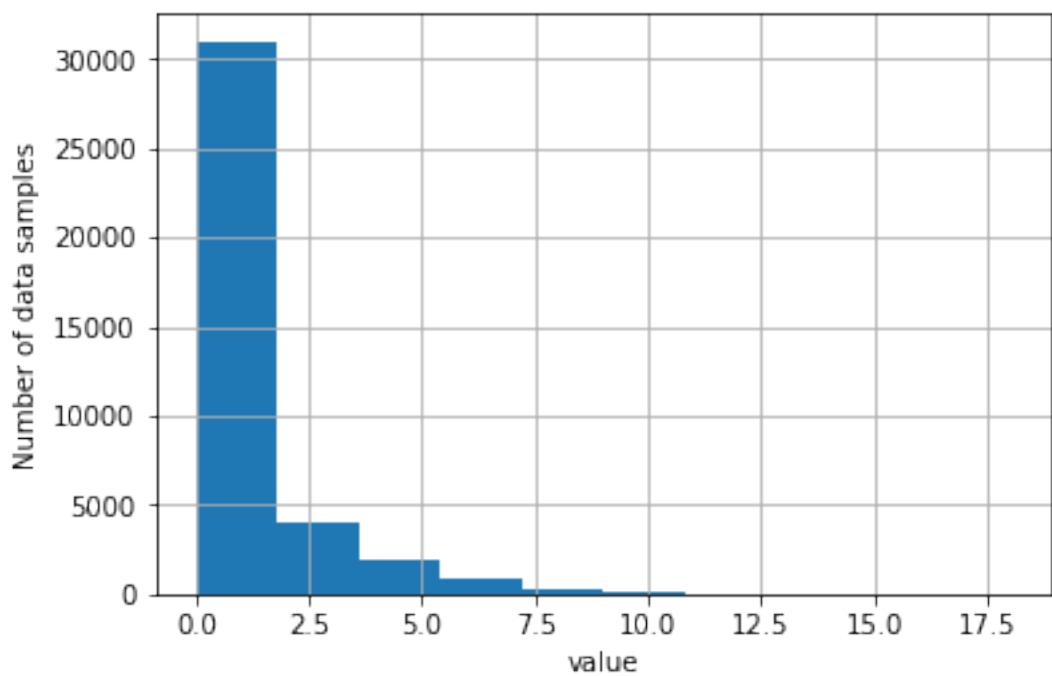
Distribution of data for interface_rx host bb3localdomain instance wlan0 type if_errors



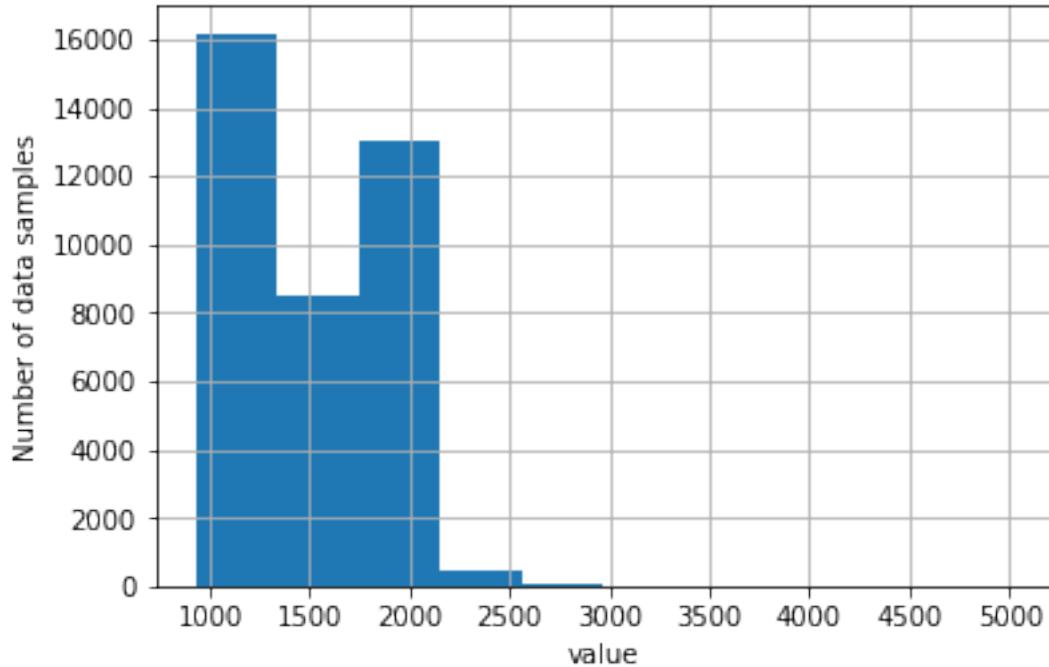
Distribution of data for interface_rx host bb3localdomain instance wlan0 type if_octets



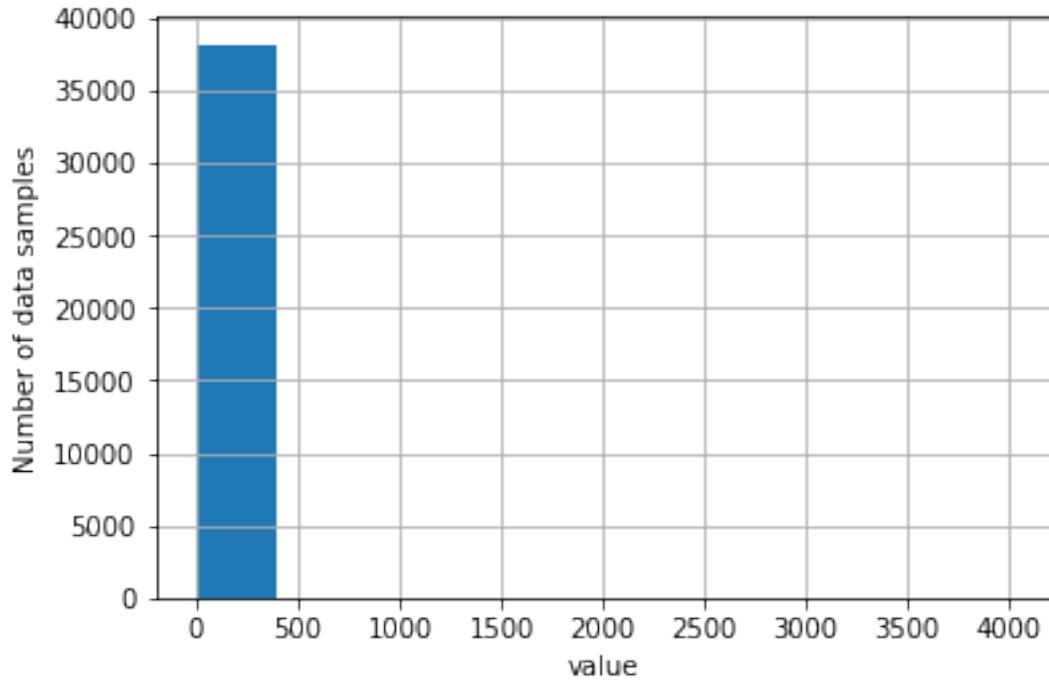
Distribution of data for interface_rx host bb3localdomain instance wlan0 type if_packets



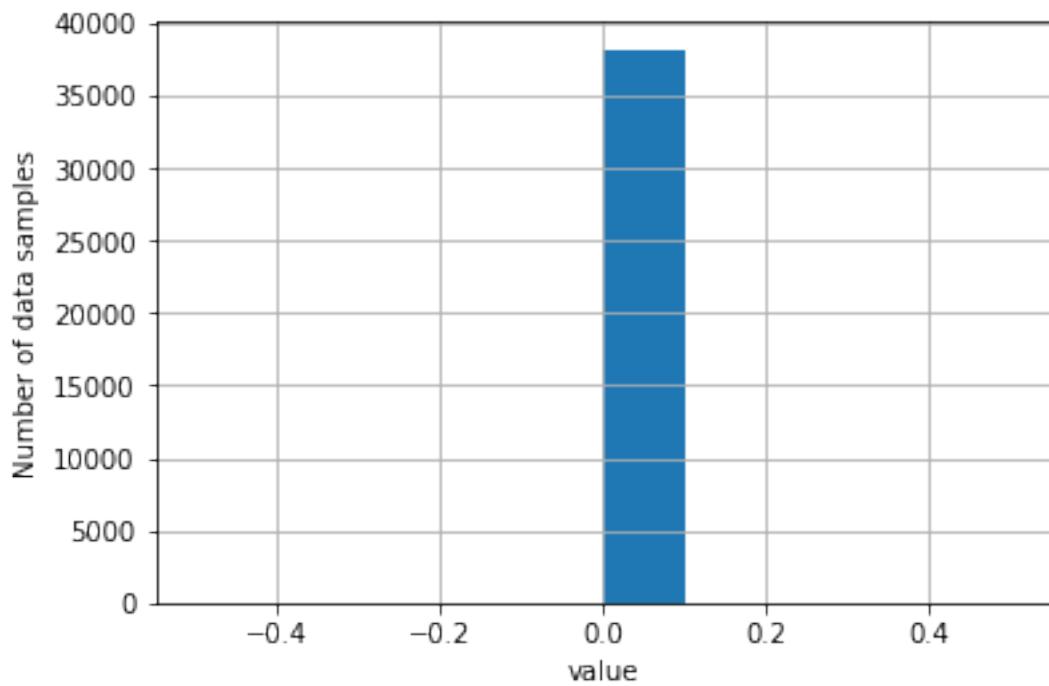
Distribution of data for contextswitch_value host bb3localdomain type contextswitch



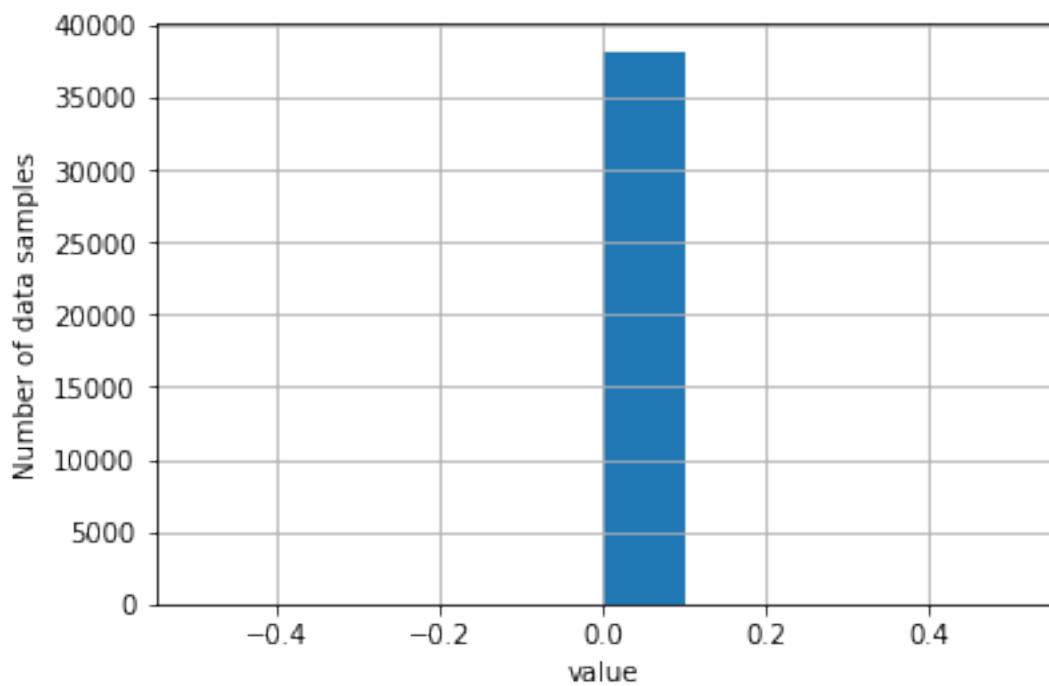
Distribution of data for disk_io_time host bb3localdomain instance mmcblk1 type disk_io_time



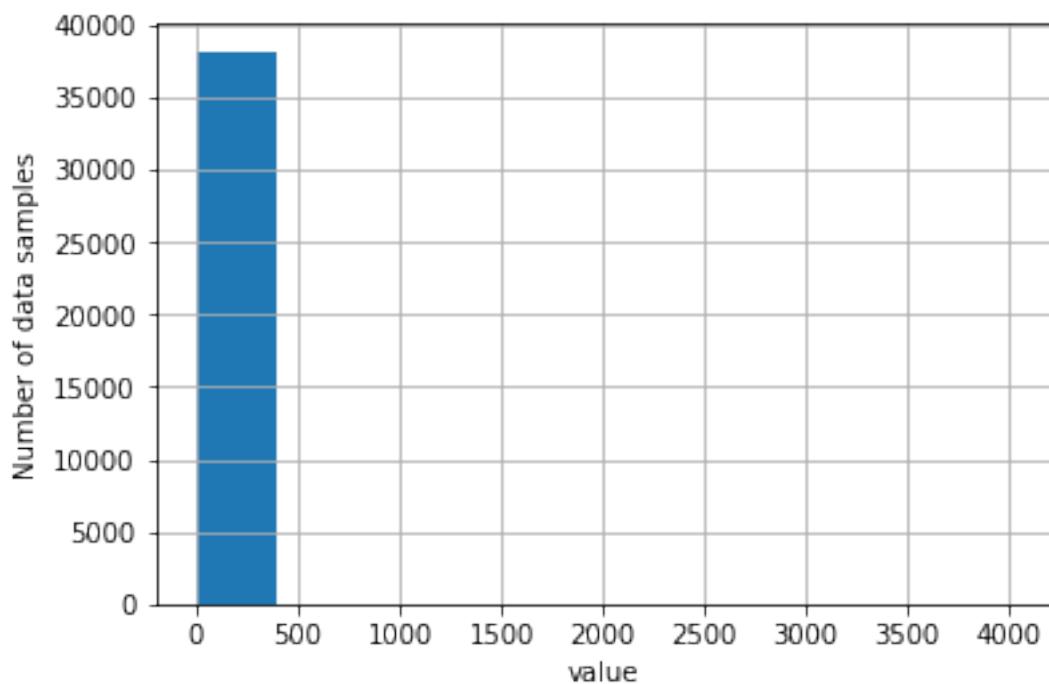
Distribution of data for disk_io_time host bb3localdomain instance mmcblk1boot0 type disk_io_t



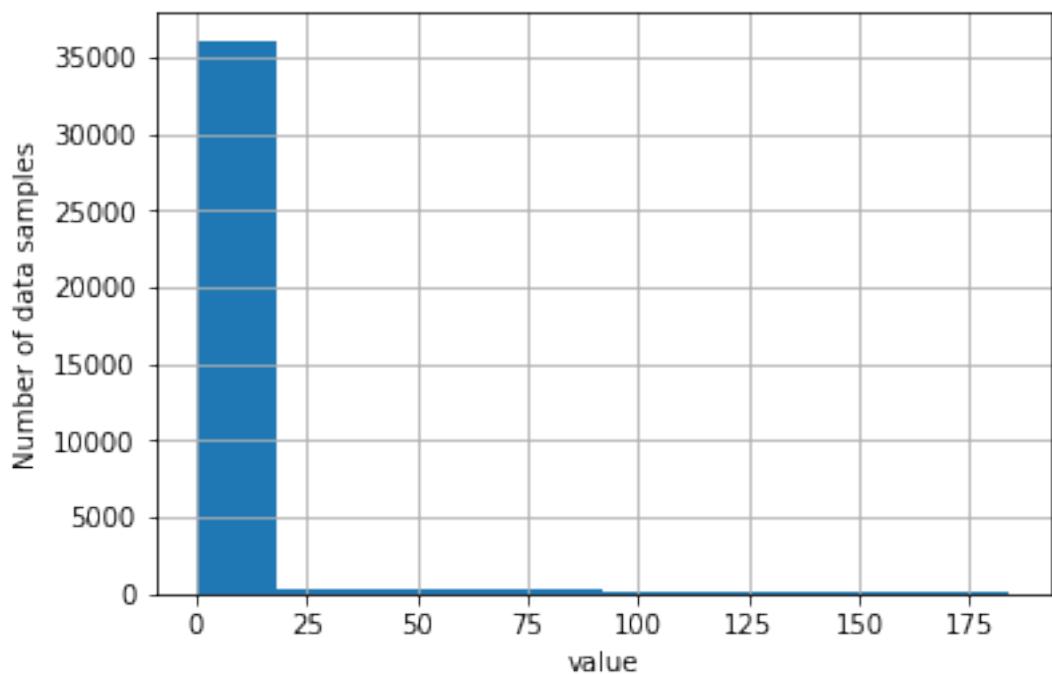
Distribution of data for disk_io_time host bb3localdomain instance mmcblk1boot1 type disk_io_time



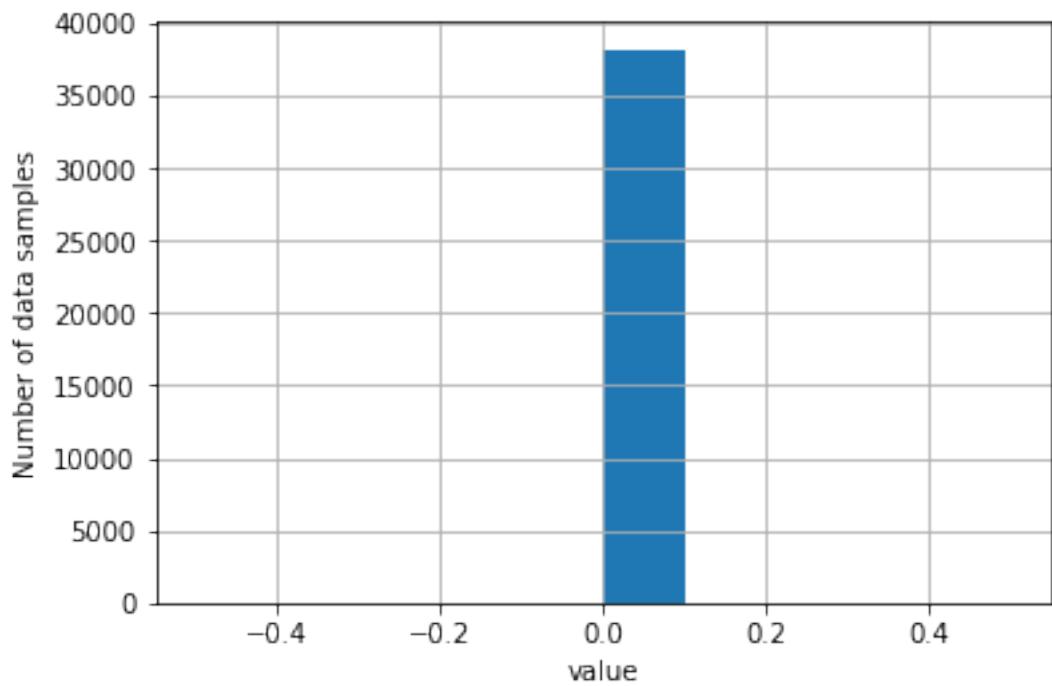
Distribution of data for disk_io_time host bb3localdomain instance mmcblk1p1 type disk_io_time



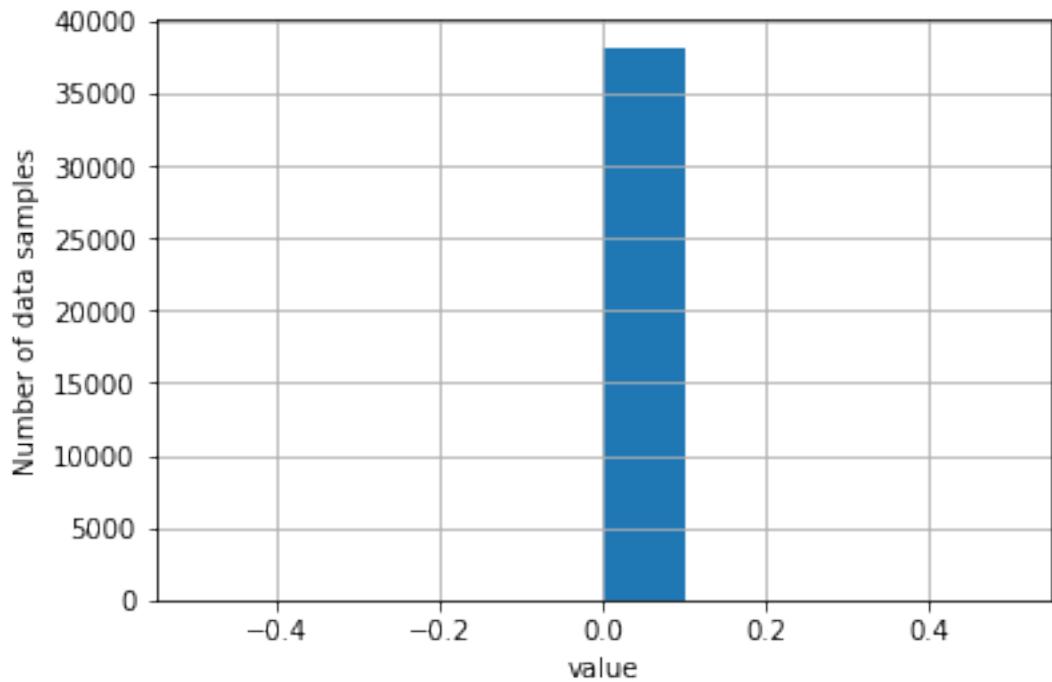
```
(38200, 29)
Index(['cpu_value host bb4localdomain type_instance idle',
       'cpu_value host bb4localdomain type_instance interrupt',
       'cpu_value host bb4localdomain type_instance nice',
       'cpu_value host bb4localdomain type_instance softirq',
       'cpu_value host bb4localdomain type_instance steal',
       'cpu_value host bb4localdomain type_instance system',
       'cpu_value host bb4localdomain type_instance user',
       'cpu_value host bb4localdomain type_instance wait',
       'interface_tx host bb4localdomain instance lo type if_dropped',
       'interface_tx host bb4localdomain instance lo type if_errors',
       'interface_tx host bb4localdomain instance lo type if_octets',
       'interface_tx host bb4localdomain instance lo type if_packets',
       'interface_tx host bb4localdomain instance wlan0 type if_dropped',
       'interface_tx host bb4localdomain instance wlan0 type if_errors',
       'interface_tx host bb4localdomain instance wlan0 type if_octets',
       'interface_tx host bb4localdomain instance wlan0 type if_packets',
       'interface_rx host bb4localdomain instance lo type if_dropped',
       'interface_rx host bb4localdomain instance lo type if_errors',
       'interface_rx host bb4localdomain instance lo type if_octets',
       'interface_rx host bb4localdomain instance lo type if_packets',
       'interface_rx host bb4localdomain instance wlan0 type if_dropped',
       'interface_rx host bb4localdomain instance wlan0 type if_errors',
       'interface_rx host bb4localdomain instance wlan0 type if_octets',
       'interface_rx host bb4localdomain instance wlan0 type if_packets',
       'contextswitch_value host bb4localdomain type contextswitch',
       'disk_io_time host bb4localdomain instance mmcblk1 type disk_io_time',
       'disk_io_time host bb4localdomain instance mmcblk1boot0 type disk_io_time',
       'disk_io_time host bb4localdomain instance mmcblk1boot1 type disk_io_time',
       'disk_io_time host bb4localdomain instance mmcblk1p1 type disk_io_time'],
      dtype='object')
Distribution of data for cpu_value host bb4localdomain type_instance idle
```



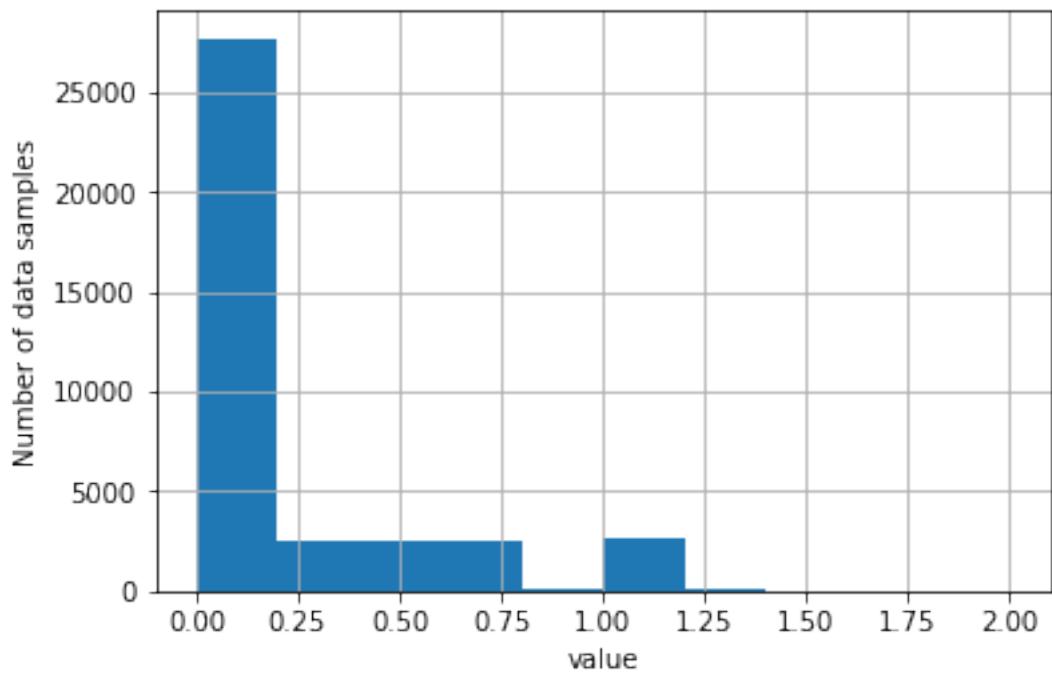
Distribution of data for cpu_value host bb4localdomain type_instance interrupt



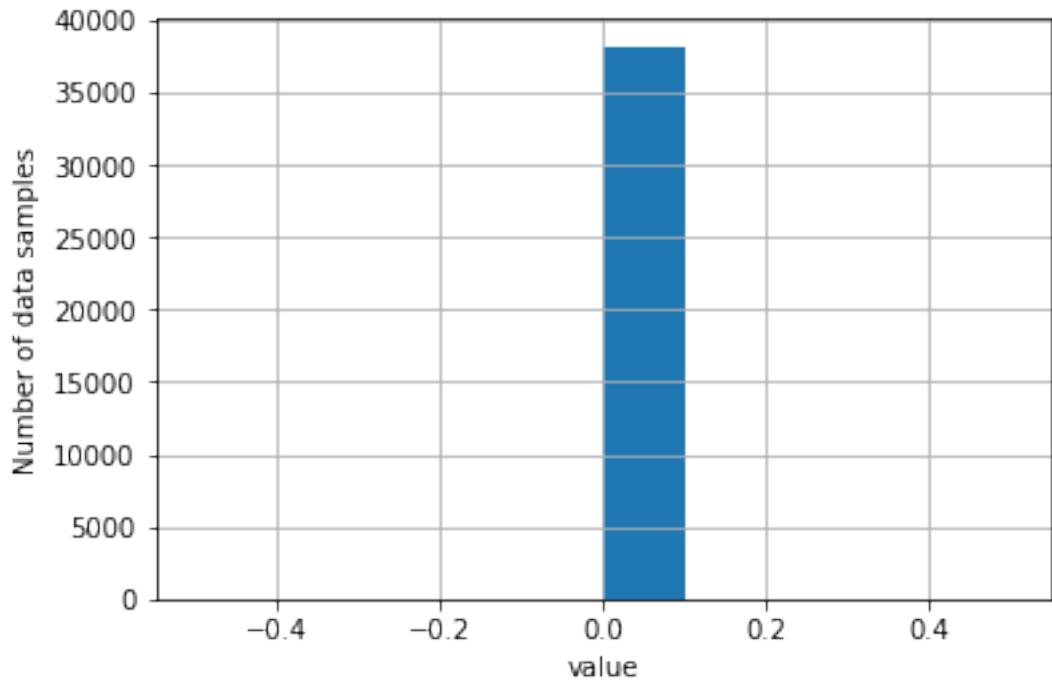
Distribution of data for cpu_value host bb4localdomain type_instance nice



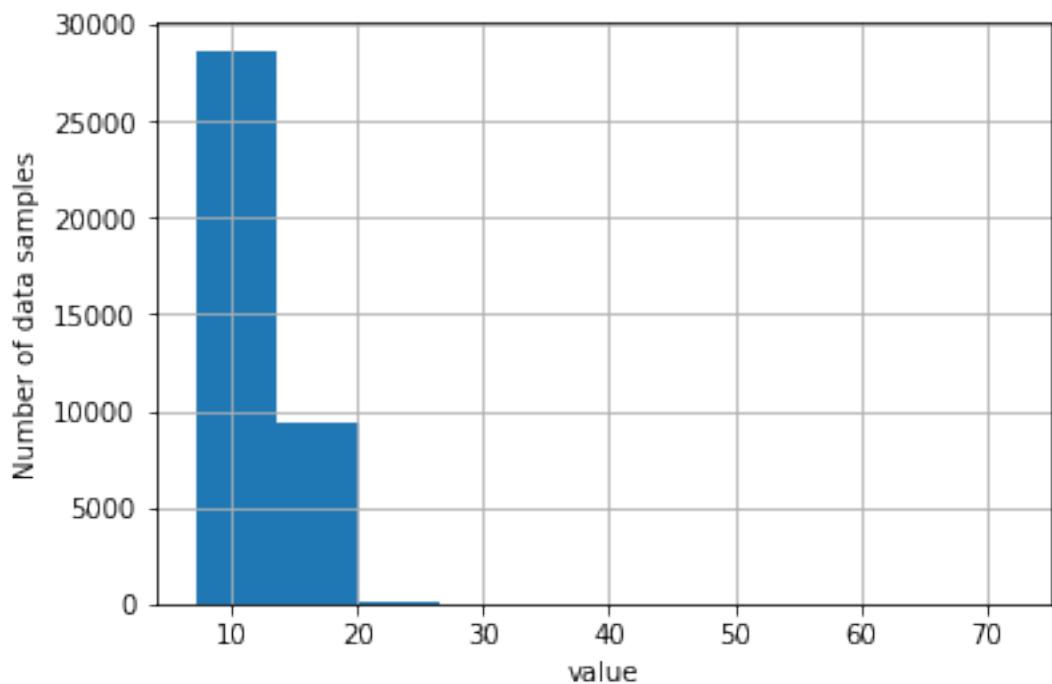
Distribution of data for cpu_value host bb4localdomain type_instance softirq



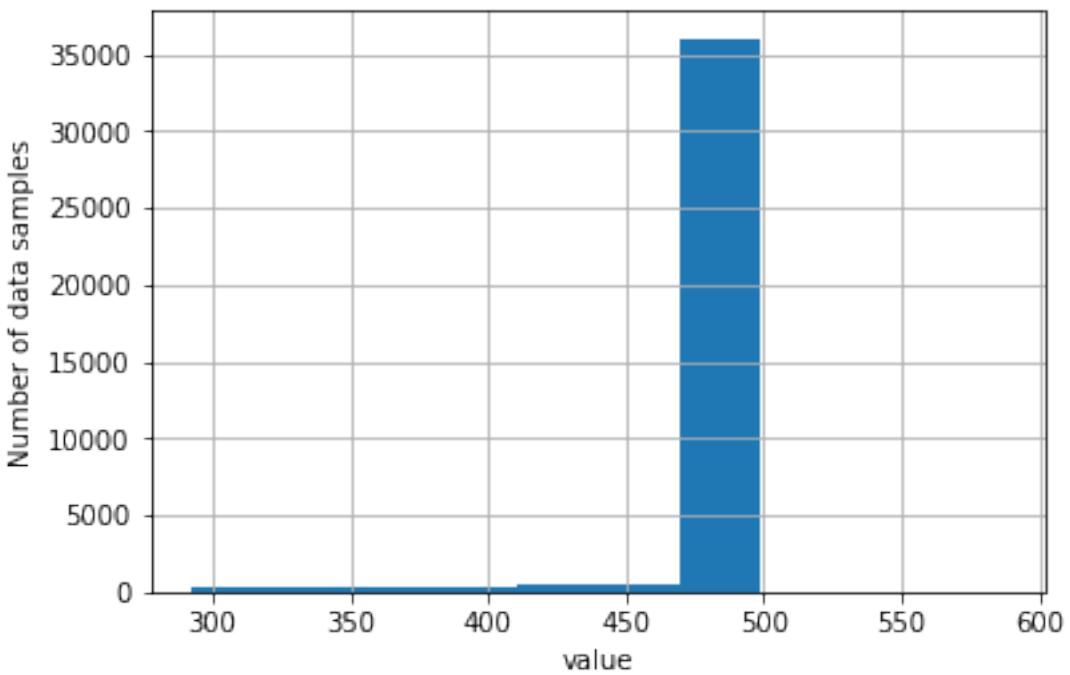
Distribution of data for cpu_value host bb4localdomain type_instance steal



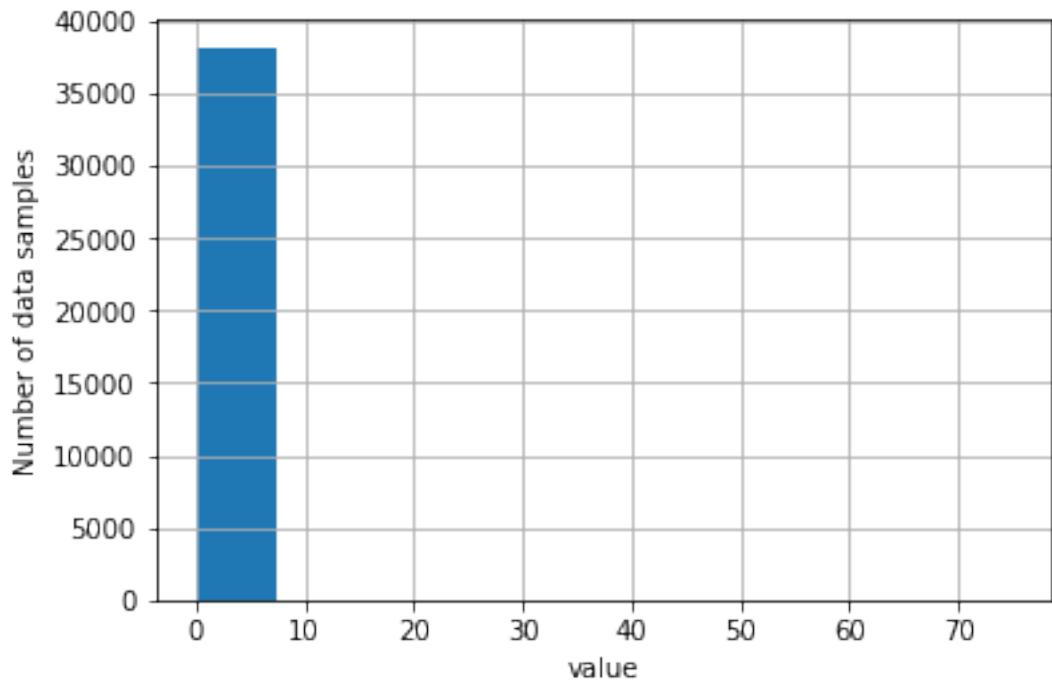
Distribution of data for cpu_value host bb4localdomain type_instance system



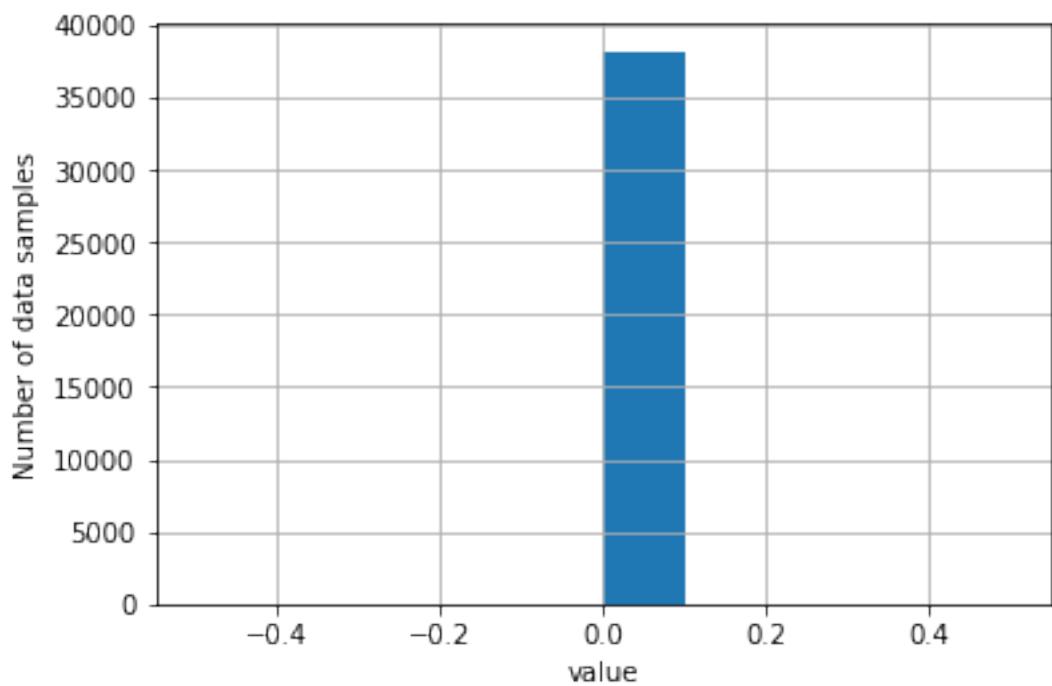
Distribution of data for cpu_value host bb4localdomain type_instance user



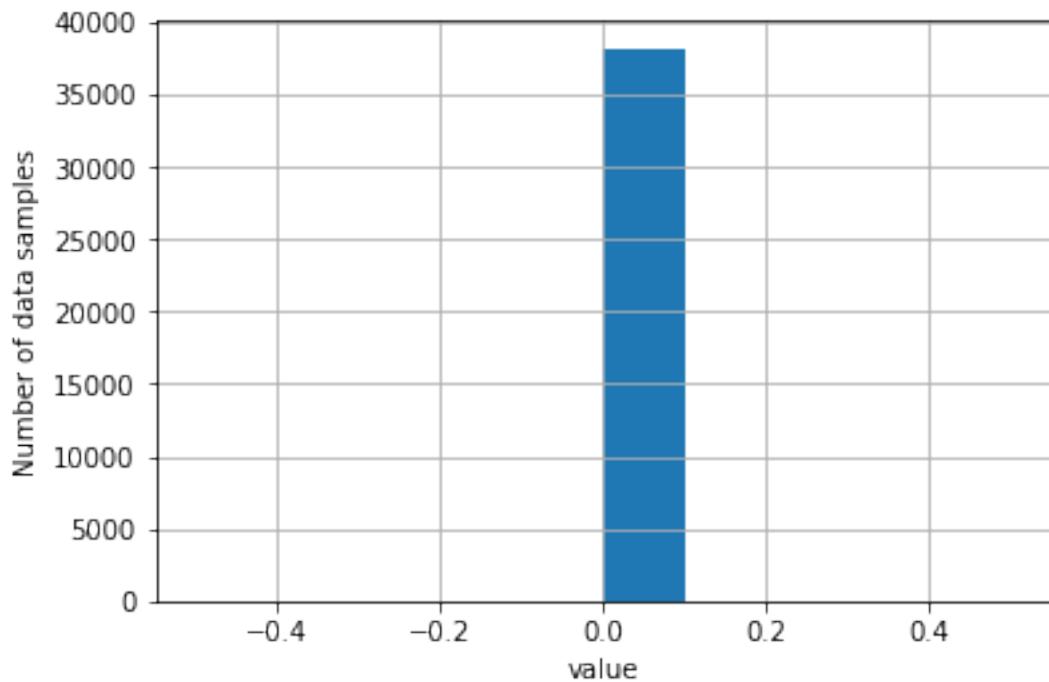
Distribution of data for cpu_value host bb4localdomain type_instance wait



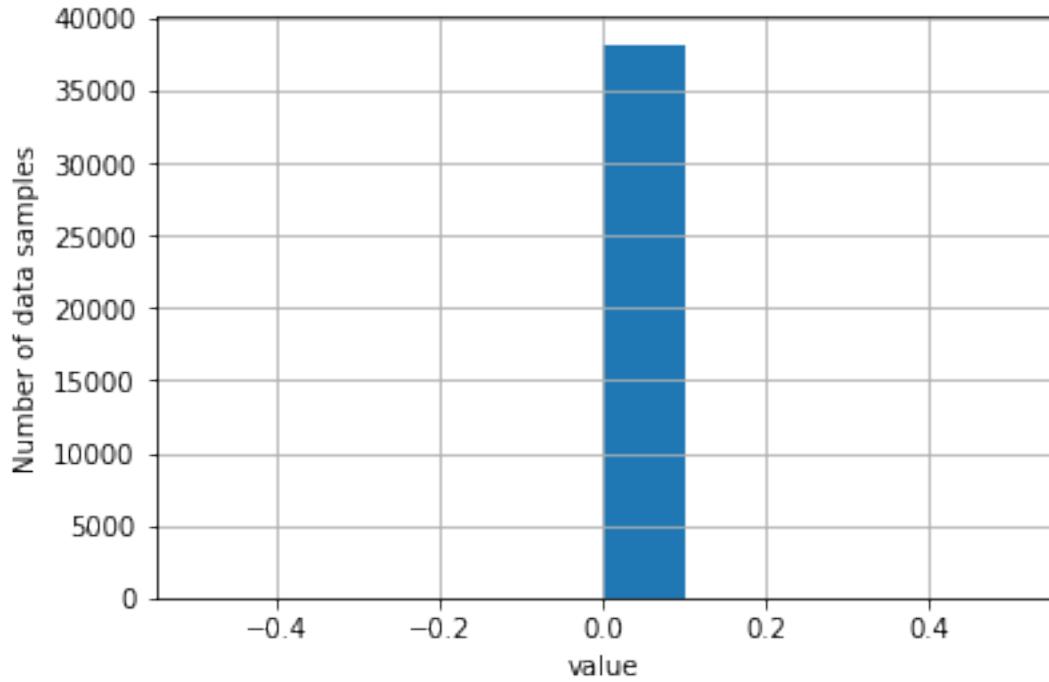
Distribution of data for interface_tx host bb4localdomain instance lo type if_dropped



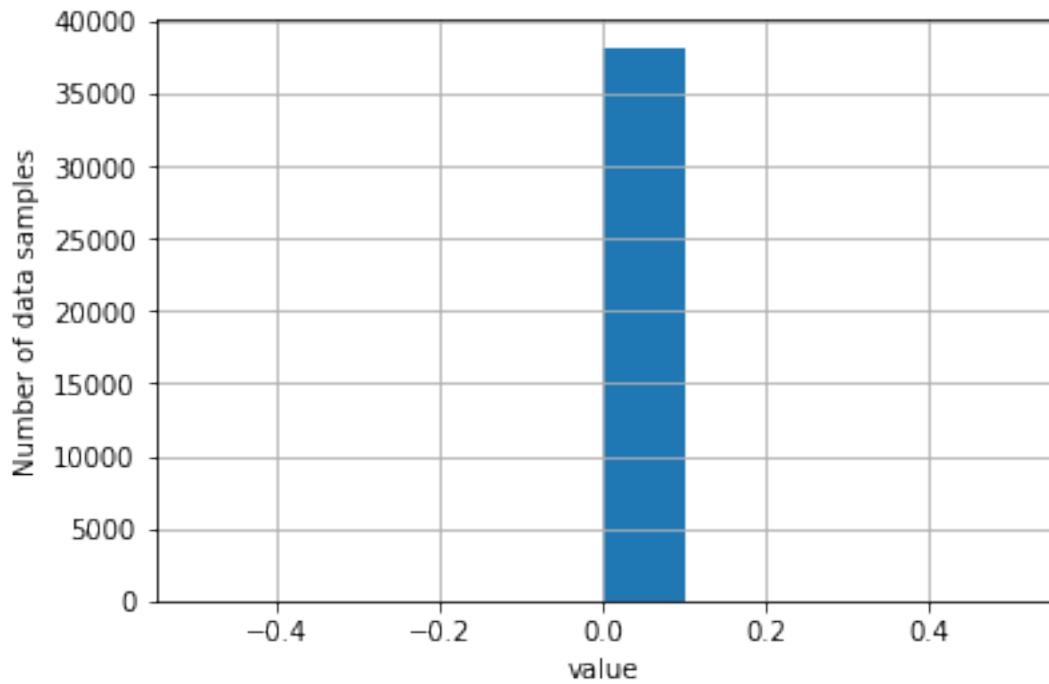
Distribution of data for interface_tx host bb4localdomain instance lo type if_errors



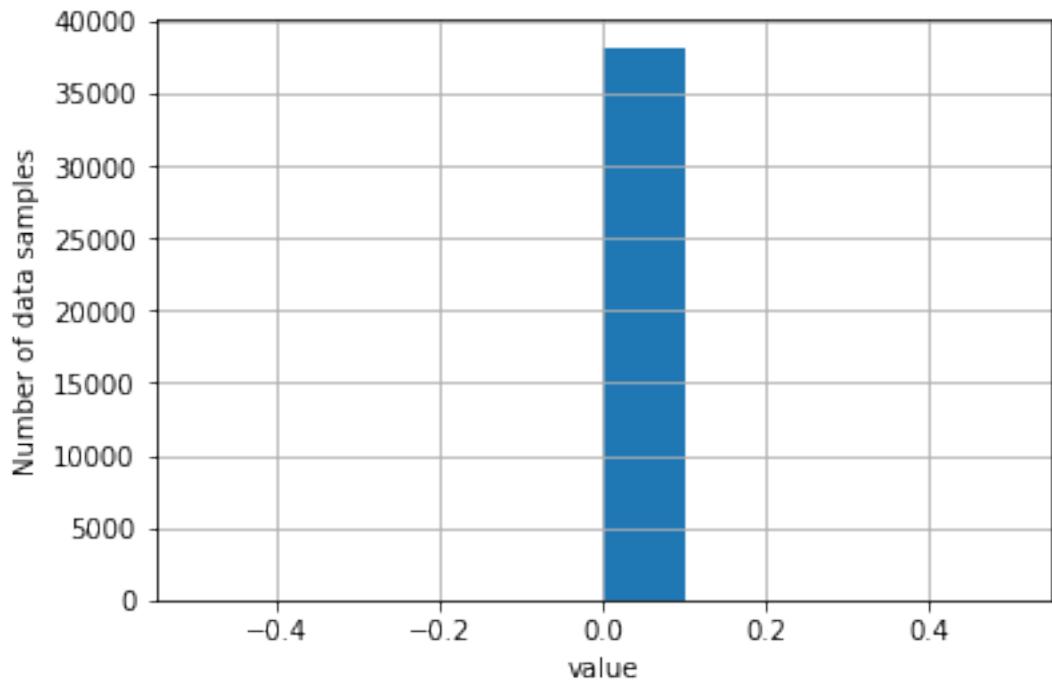
Distribution of data for interface_tx host bb4localdomain instance lo type if_octets



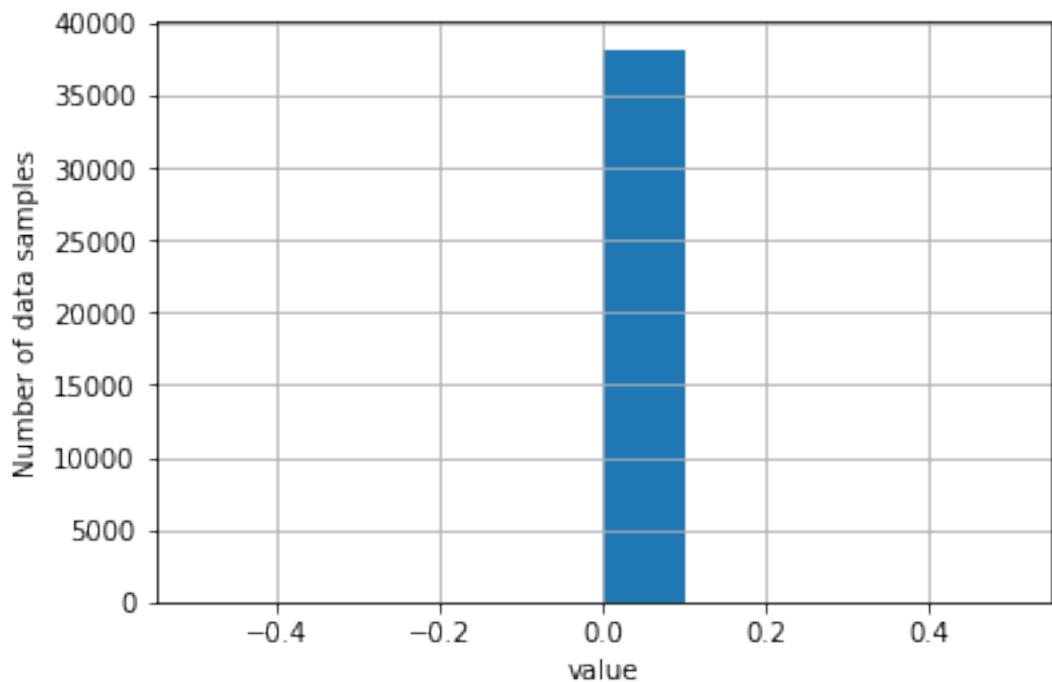
Distribution of data for interface_tx host bb4localdomain instance lo type if_packets



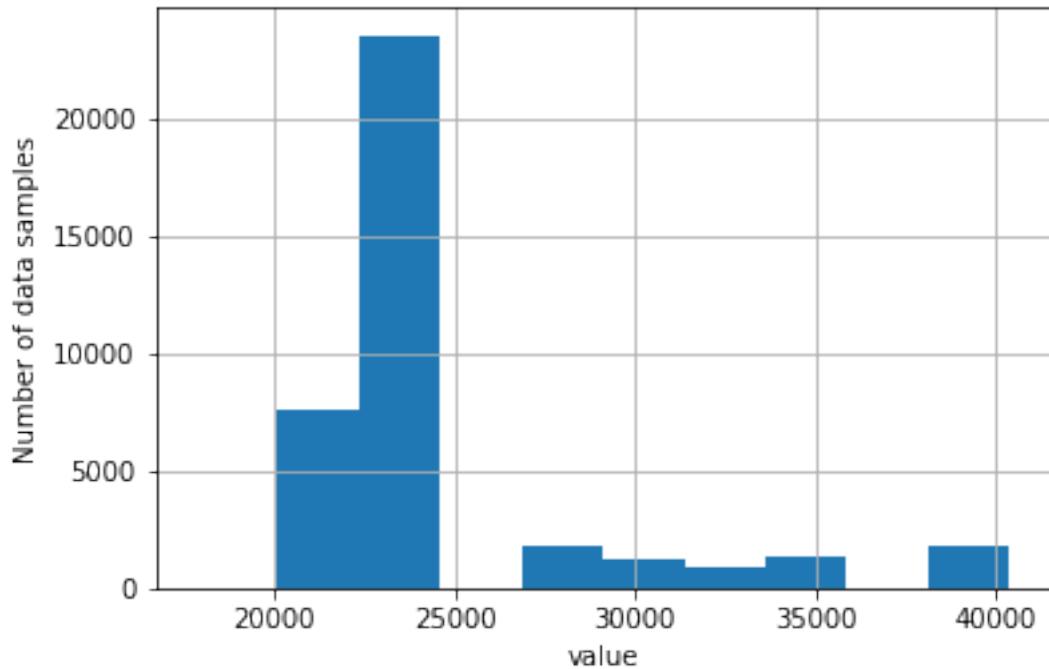
Distribution of data for interface_tx host bb4localdomain instance wlan0 type if_dropped



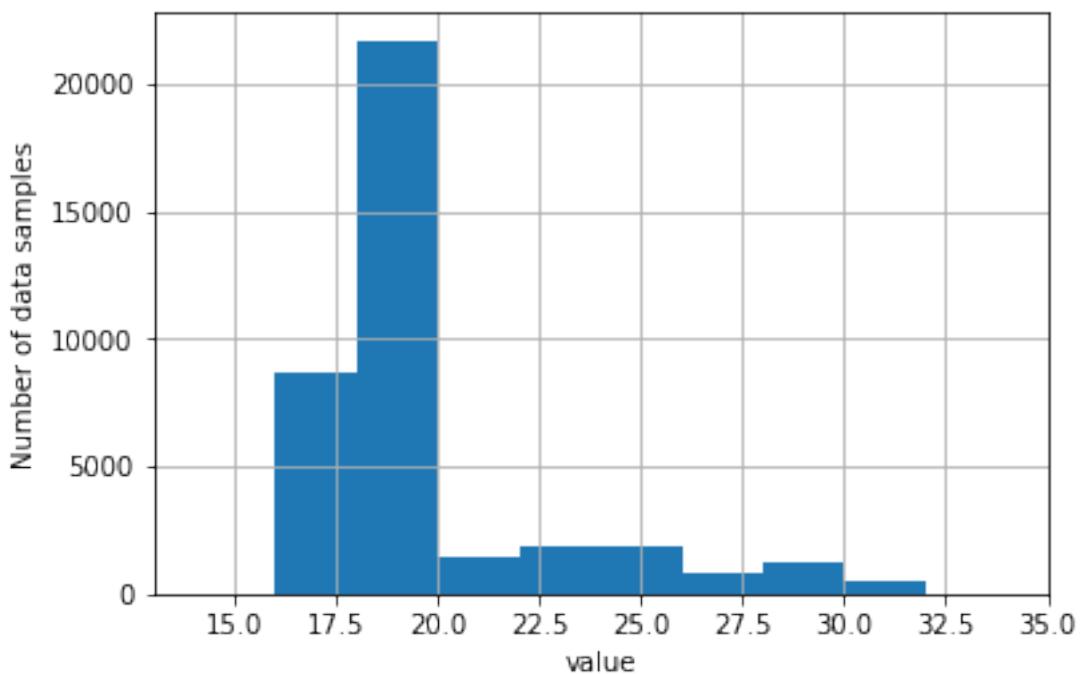
Distribution of data for interface_tx host bb4localdomain instance wlan0 type if_errors



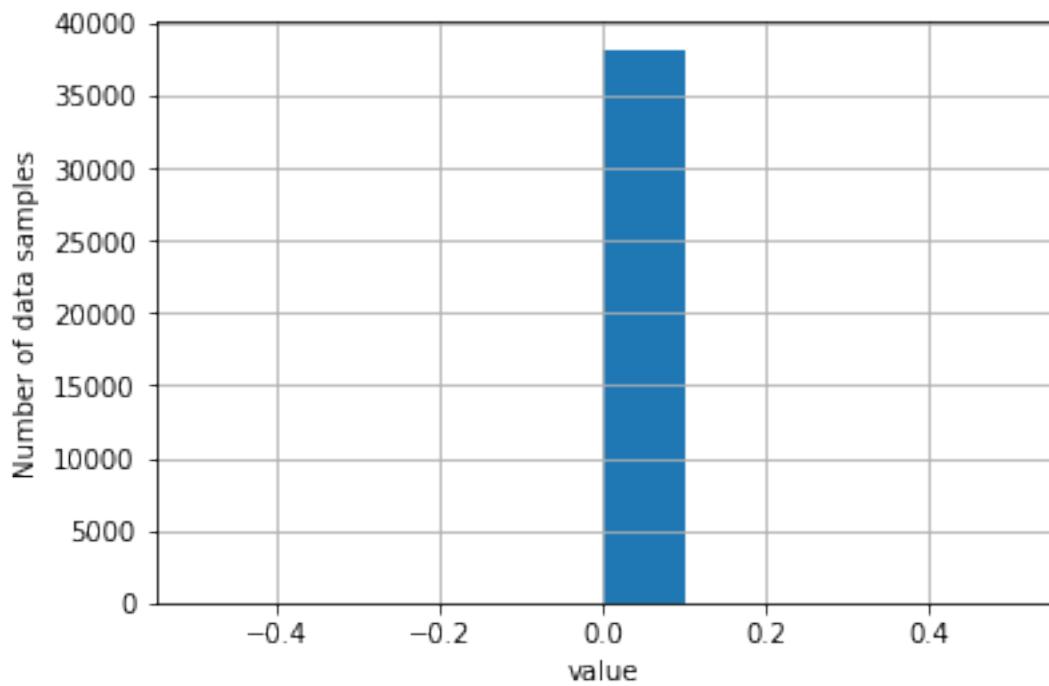
Distribution of data for interface_tx host bb4localdomain instance wlan0 type if_octets



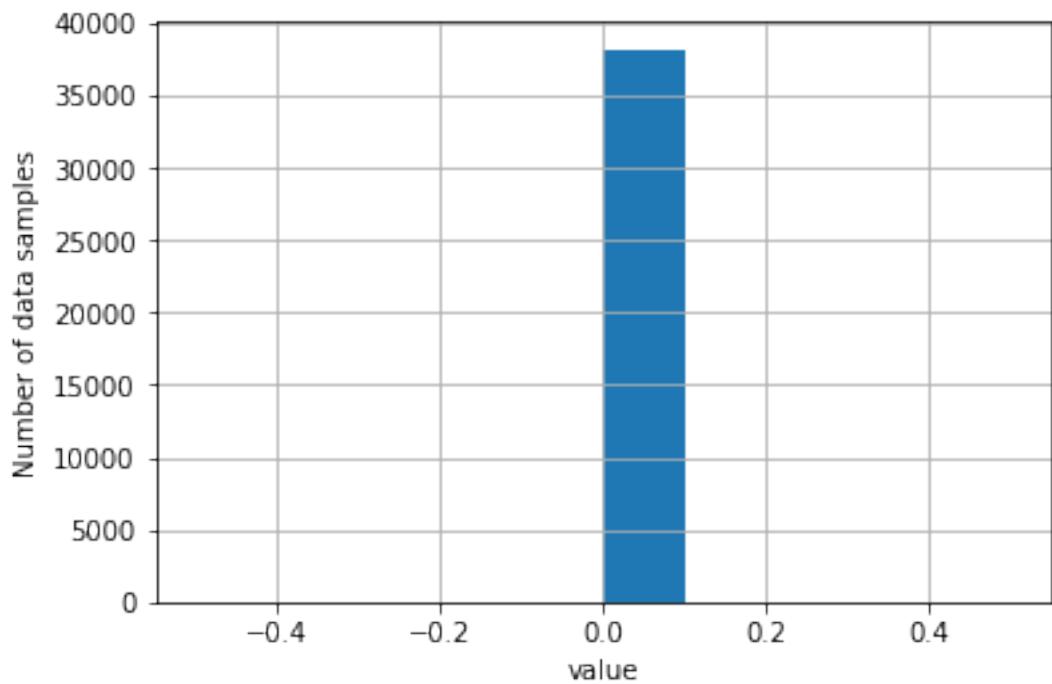
Distribution of data for interface_tx host bb4localdomain instance wlan0 type if_packets



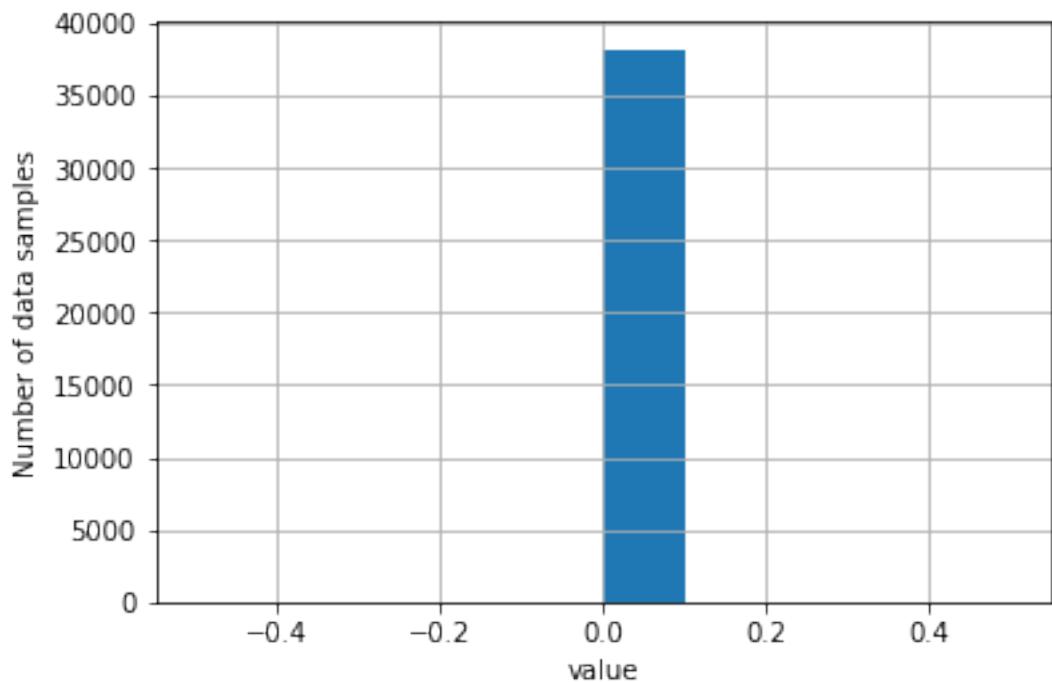
Distribution of data for interface_rx host bb4localdomain instance lo type if_dropped



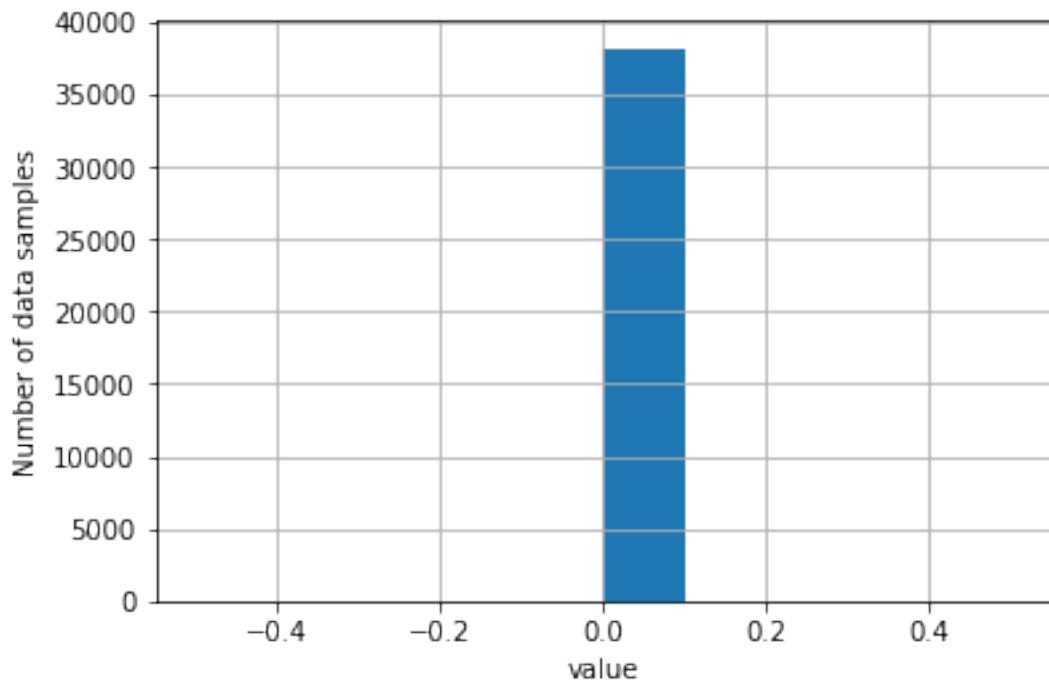
Distribution of data for interface_rx host bb4localdomain instance lo type if_errors



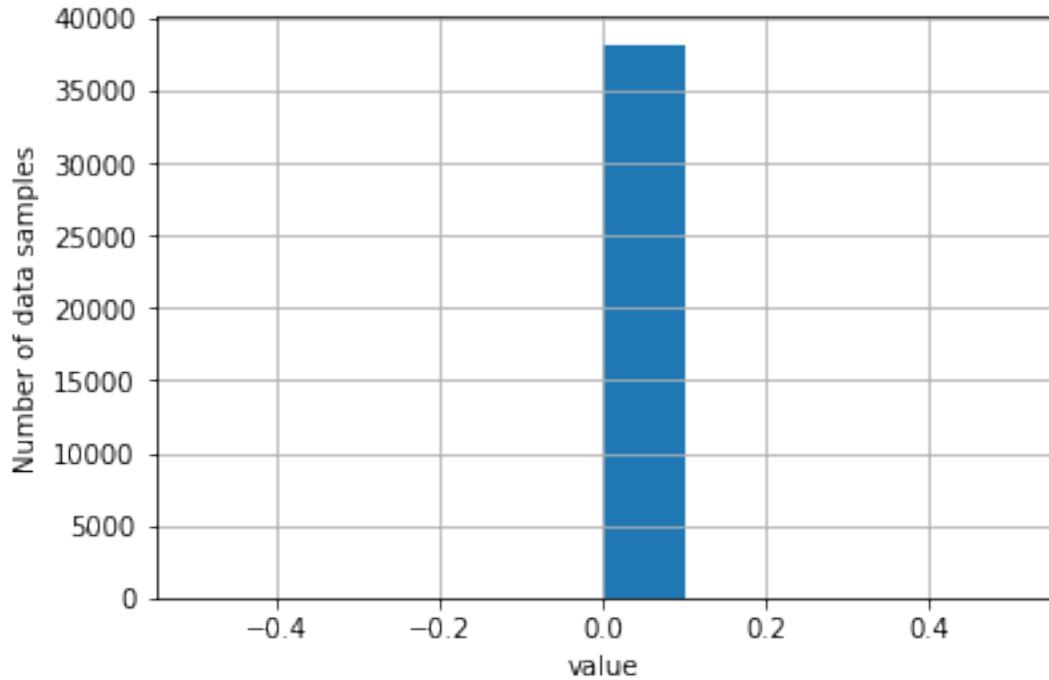
Distribution of data for interface_rx host bb4localdomain instance lo type if_octets



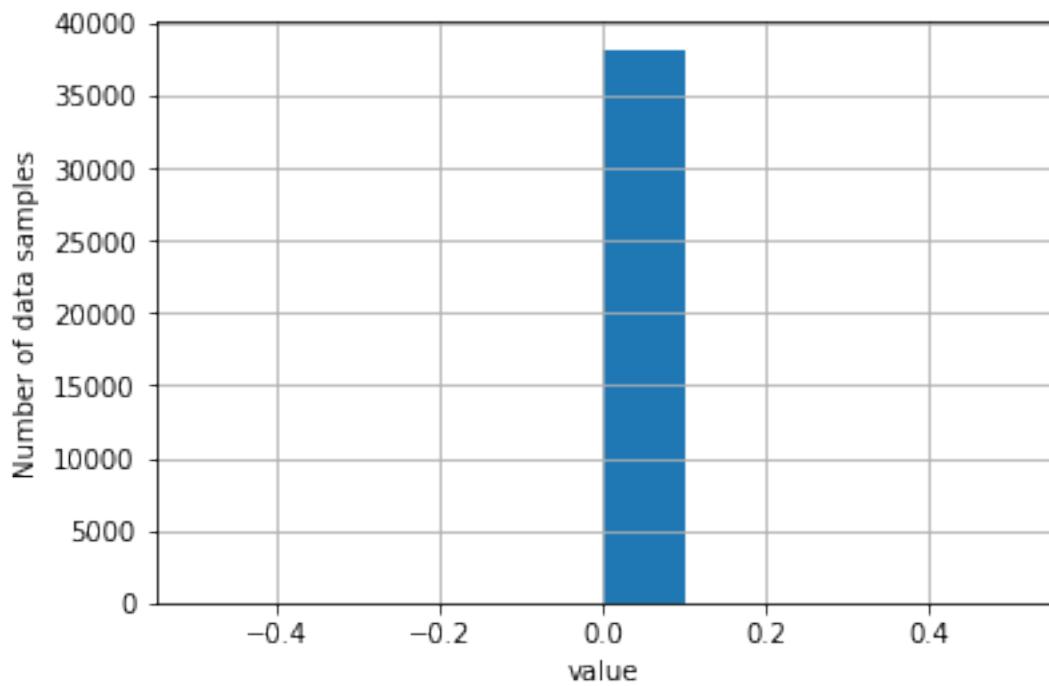
Distribution of data for interface_rx host bb4localdomain instance lo type if_packets



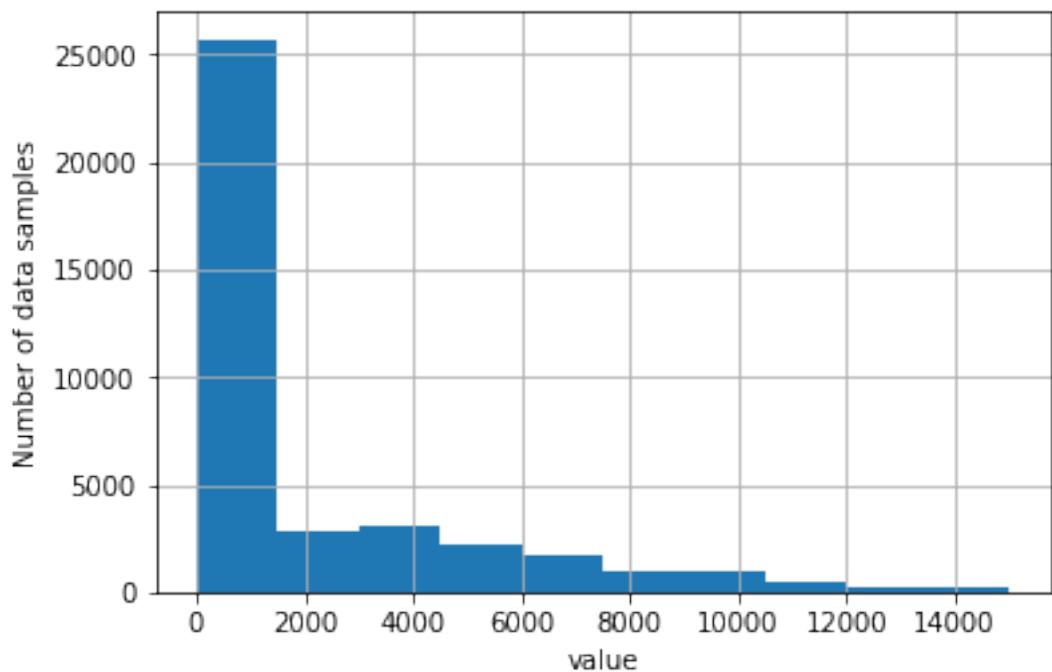
Distribution of data for interface_rx host bb4localdomain instance wlan0 type if_dropped



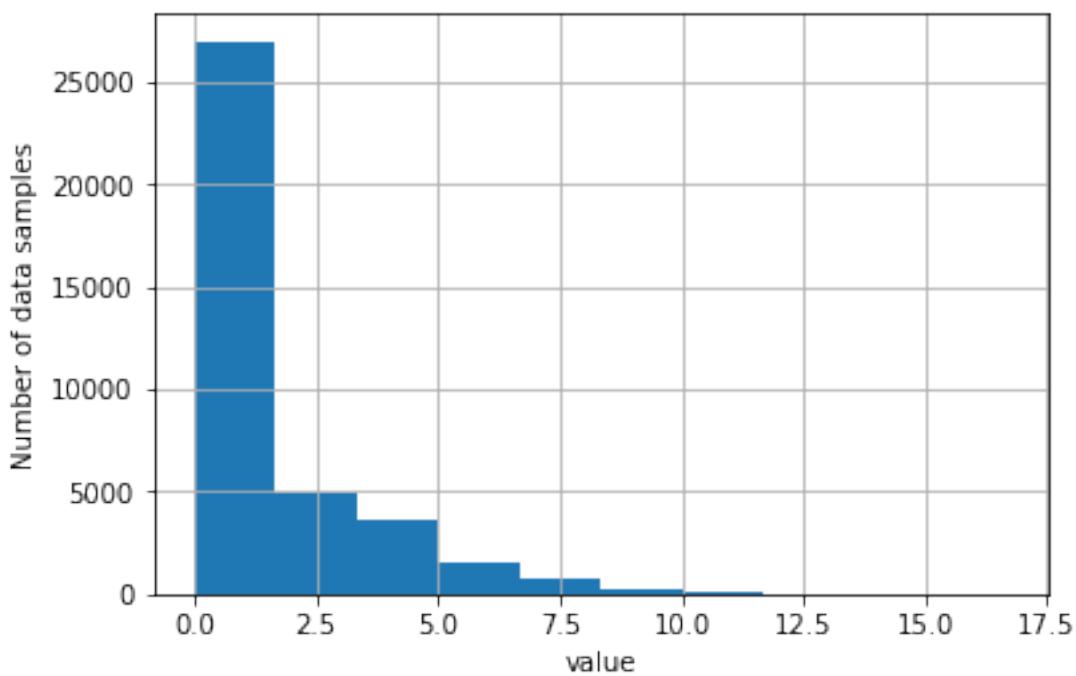
Distribution of data for interface_rx host bb4localdomain instance wlan0 type if_errors



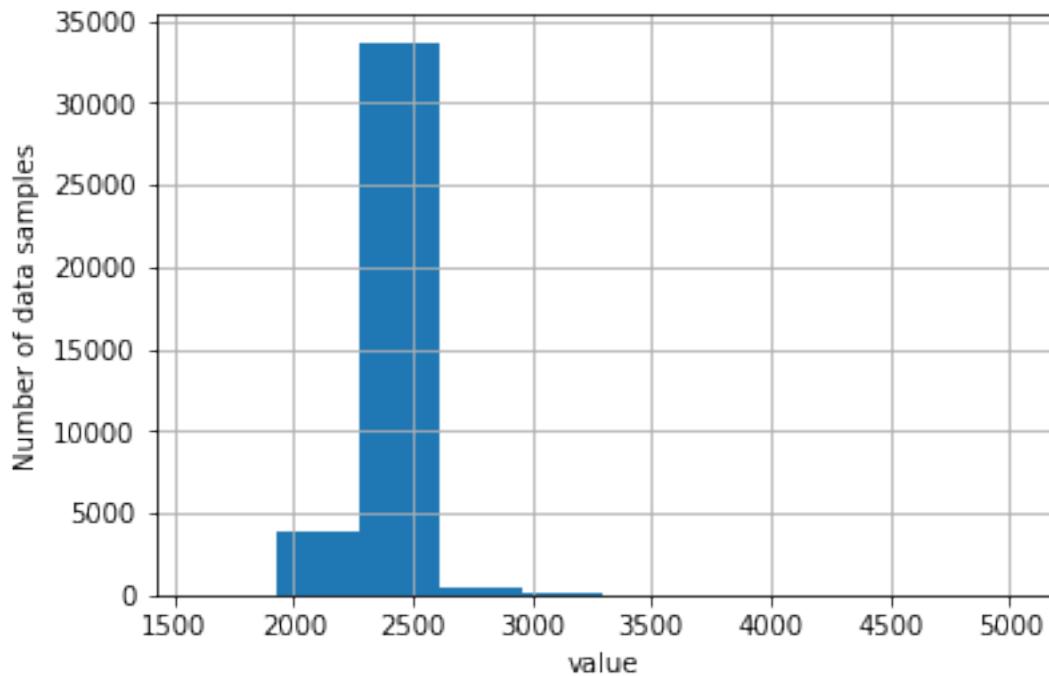
Distribution of data for interface_rx host bb4localdomain instance wlan0 type if_octets



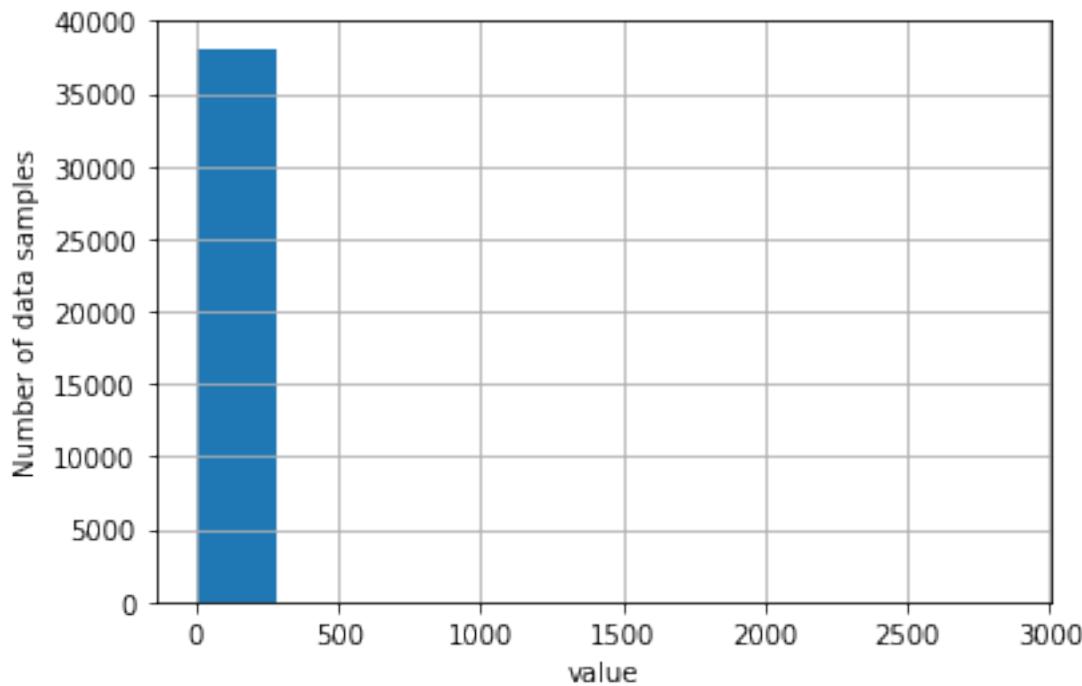
Distribution of data for interface_rx host bb4localdomain instance wlan0 type if_packets



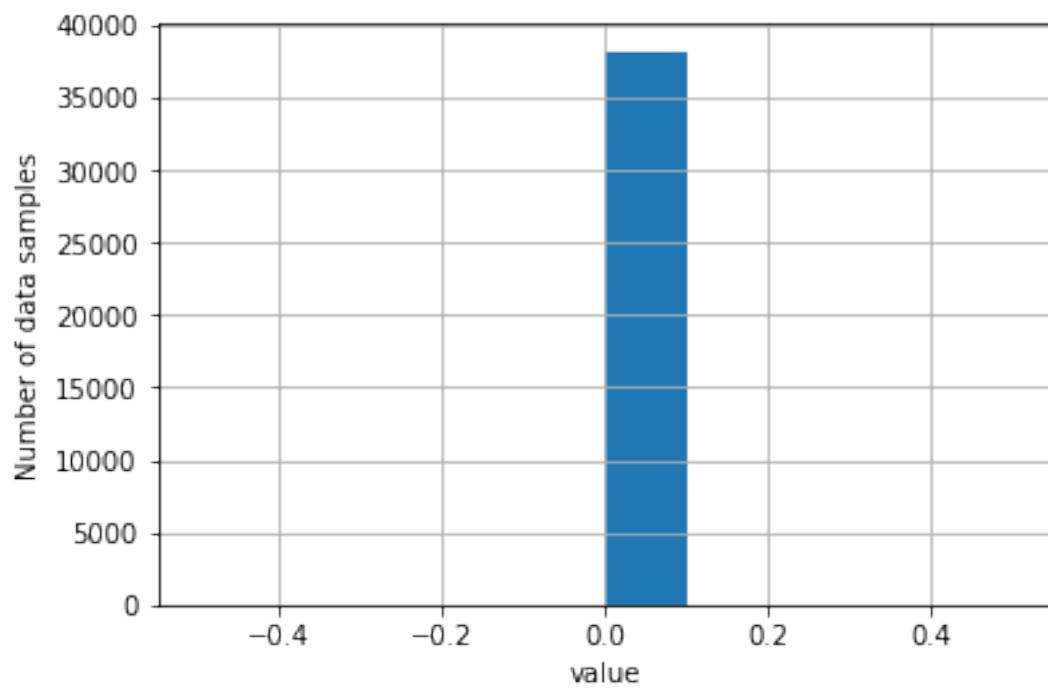
Distribution of data for contextswitch_value host bb4localdomain type contextswitch



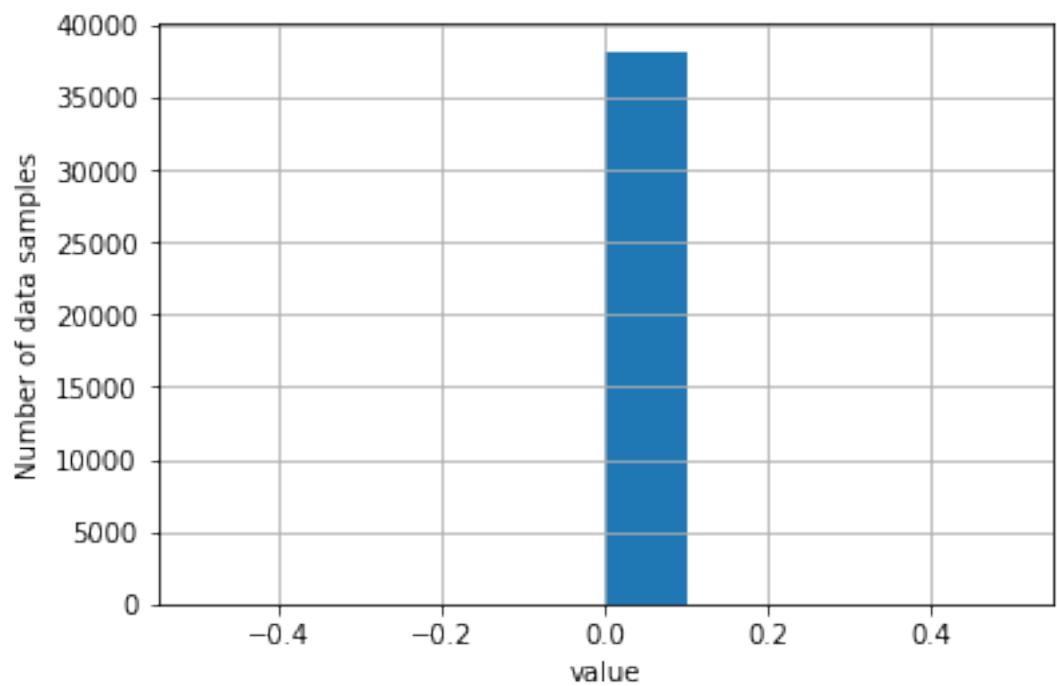
Distribution of data for disk_io_time host bb4localdomain instance mmcblk1 type disk_io_time



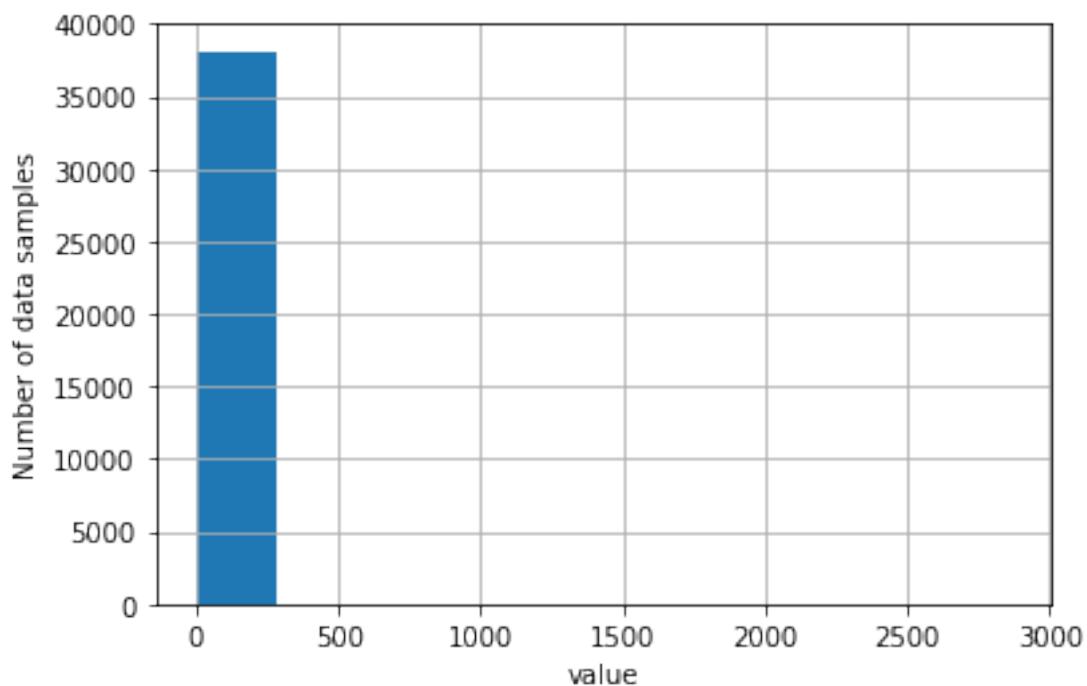
Distribution of data for disk_io_time host bb4localdomain instance mmcblk1boot0 type disk_io_t



Distribution of data for disk_io_time host bb4localdomain instance mmcblk1boot1 type disk_io_time



Distribution of data for disk_io_time host bb4localdomain instance mmcblk1p1 type disk_io_time



1.7 Get data

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

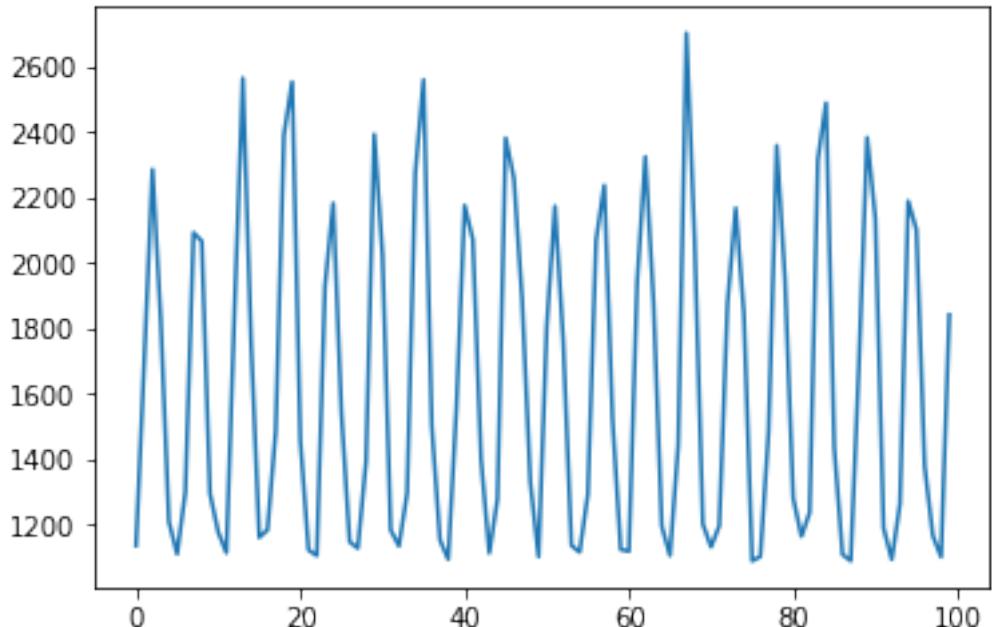
```
for i in range(1,5):
    data_matrices.append(node[i].as_matrix())

data = numpy.vstack(data_matrices)
```

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

```
Out[21]: (152800, 29)
```

```
In [22]: tdata = data[:,24]
plotter.plot(tdata.T[:100])
plotter.show()
print(data.shape)
```



```
(152800, 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 Correlation measurement

2 Prediction

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

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

    X = numpy.stack(data_matrices[:-1], axis=1)
    print(data_matrices[3].shape)

(38200, 29)
```

```
In [27]: print(X.shape)
LEN = X.shape[0]
SPLIT = int(0.85*LEN)

train_X = X[:SPLIT,:,:]
val_X = X[SPLIT:SPLIT+1000,:,:]
test_X = X[SPLIT+1000,:,:]
diff_app_X = data_matrices[3][SPLIT+1000,:,:]

(38200, 3, 29)
```

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

```

ano_test = test_X[0]
norm_test = test_X[2]
diff_app_test = diff_app_X
app_change_test = numpy.vstack([norm_test[:3500,:], diff_app_test[3500:,:]])
net_flood_test = numpy.copy(norm_test)
net_flood_test[3000:,22] = 1.0
net_flood_test[3000:,23] = 1.0

print(f"Validation shape is {val_X.shape}")
print(f"Anomaly test shape is {ano_test.shape}")
print(f"Normal test shape is {norm_test.shape}")
print(f"Different app test shape is {diff_app_test.shape}")
print(f"App change test shape is {app_change_test.shape}")
print(f"Net flood test shape is {net_flood_test.shape}")

Validation shape is (3, 1000, 29)
Anomaly test shape is (4730, 29)
Normal test shape is (4730, 29)
Different app test shape is (4730, 29)
App change test shape is (4730, 29)
Net flood test shape is (4730, 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

```

2.1 Flat models

```

In [30]: from keras.models import Model
        from keras.layers import Dense, Input, Dropout, GRU
        from keras.callbacks import EarlyStopping

```

Using TensorFlow backend.

```
In [31]: 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=[estopper])
    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=1000)
    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]}\n")

In [32]: 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="beta", sparms=(a, b), plot=plotter)
        plotter.legend()
        plotter.savefig(f"{name}_qq.png",dpi=1000)
        plotter.show()
```

```

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

    fig = plotter.figure()
    plotter.plot(error, 'g-', alpha=0.4, label="Error", linewidth=0.5)
    plotter.plot(global_mean, 'r-', alpha=0.9, label="Mean", linewidth=0.5)
    #plotter.plot(upper,'b-', alpha=0.2, label="Upper Bound", linewidth=0.5)
    plotter.plot(bound,'b-', alpha=0.2, label="Bound", linewidth=0.5)
    plotter.plot(anomaly,'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=1000)
    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[0]}")

    return (global_mean, bound)

```

```

In [33]: def data_test(model, dataset=test_X[0], ravel=1, write=0, name="none", window=5, bounds=[0, 1]):
    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=False)

```

```

#print(error)

In [34]: 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 [35]: def test(model, ravel=1, name="none", window=20):
    print(f"----- Beginning tests for {name} -----")
    print(f"Testing on normal data.")
    bounds = data_test(model, dataset=norm_test, ravel=ravel, name=(name+"_normal_"),
    print(f"Testing on anomaly data.")
    data_test(model, dataset=ano_test, ravel=ravel, name=(name+"_anomaly_"), window=window)
    print(f"Testing on different app data.")
    data_test(model, dataset=diff_app_test, ravel=ravel, name=(name+"_diff_app_"),
    print(f"Testing on App change synthetic data.")
    data_test(model, dataset=app_change_test, ravel=ravel, name=(name+"_app_change_"))
    print(f"Testing on Net flood synthetic data.")
    data_test(model, dataset=net_flood_test, ravel=ravel, name=(name+"_net_flood_"),
    print("=="*20)
    print("\r\n\r\n")
```

2.1.1 Linear Regression

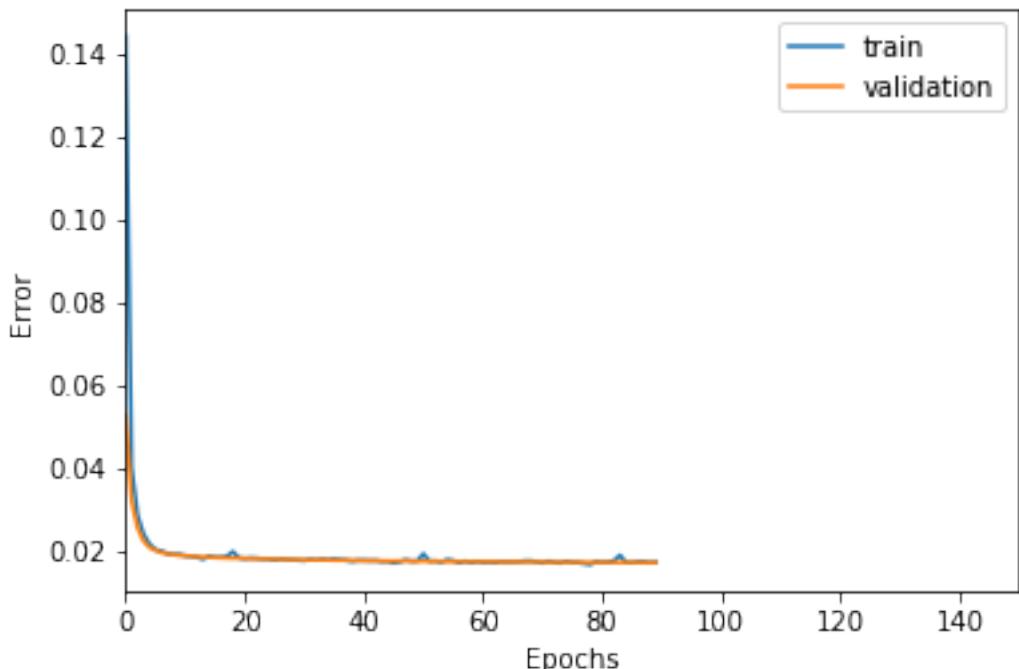
2 steps

```
In [36]: TIMESTEPS = 2
        DIM = 29
        tgen = flat_generator(X, TIMESTEPS)
        vgen = flat_generator(val_X, TIMESTEPS)
        name = "lin2"

In [37]: input_layer = Input(shape=(TIMESTEPS*DIM,))
        output = Dense(DIM, activation='sigmoid')(input_layer)

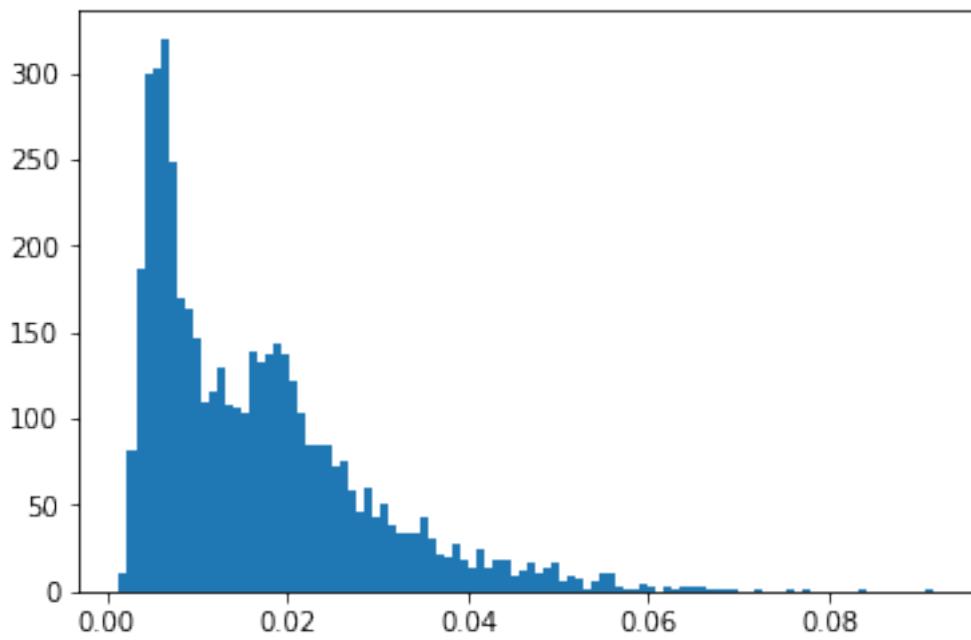
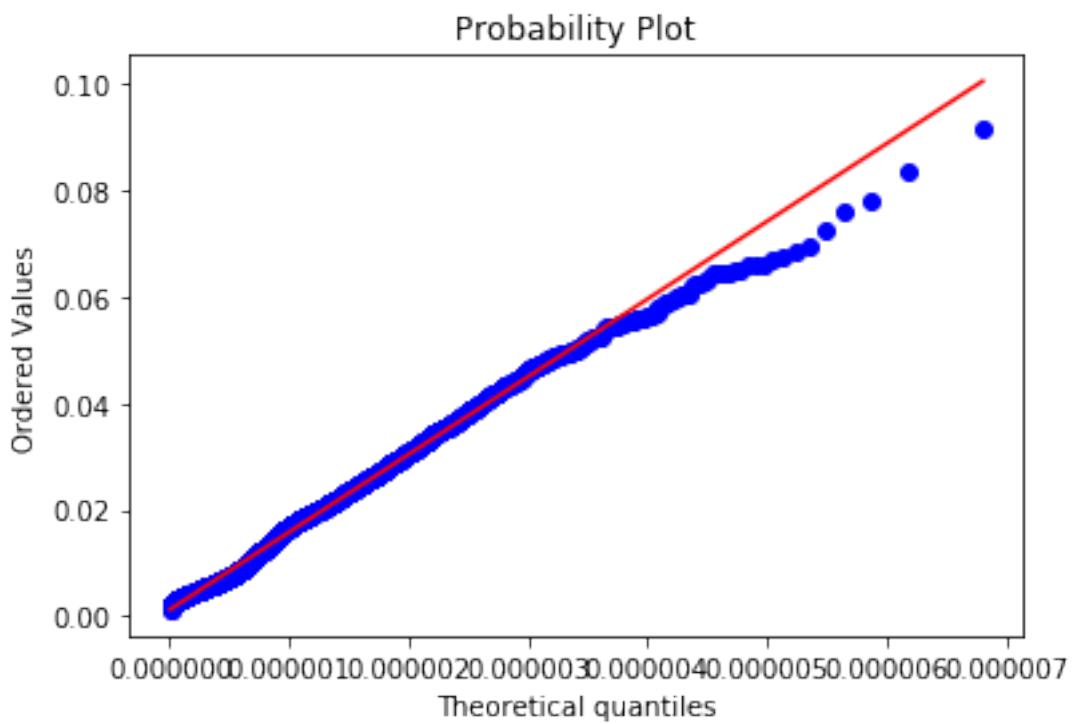
In [38]: model = Model(input_layer, output)
        model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

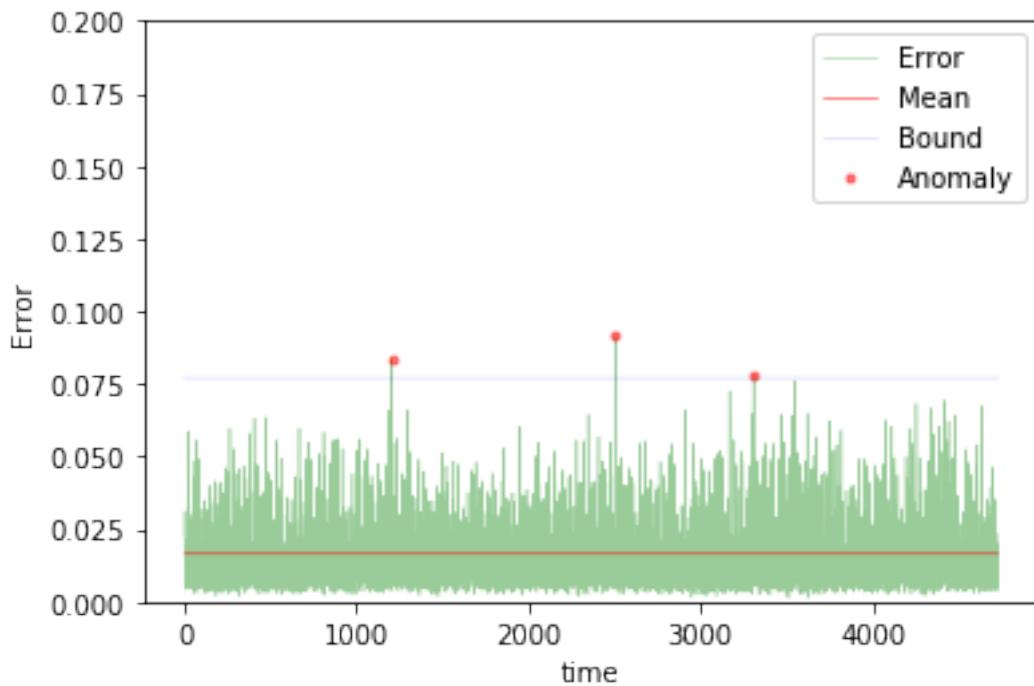
In [39]: train(model, tgen, vgen, name=name)
        test(model, name=name, window=TIMESTEPS)
```



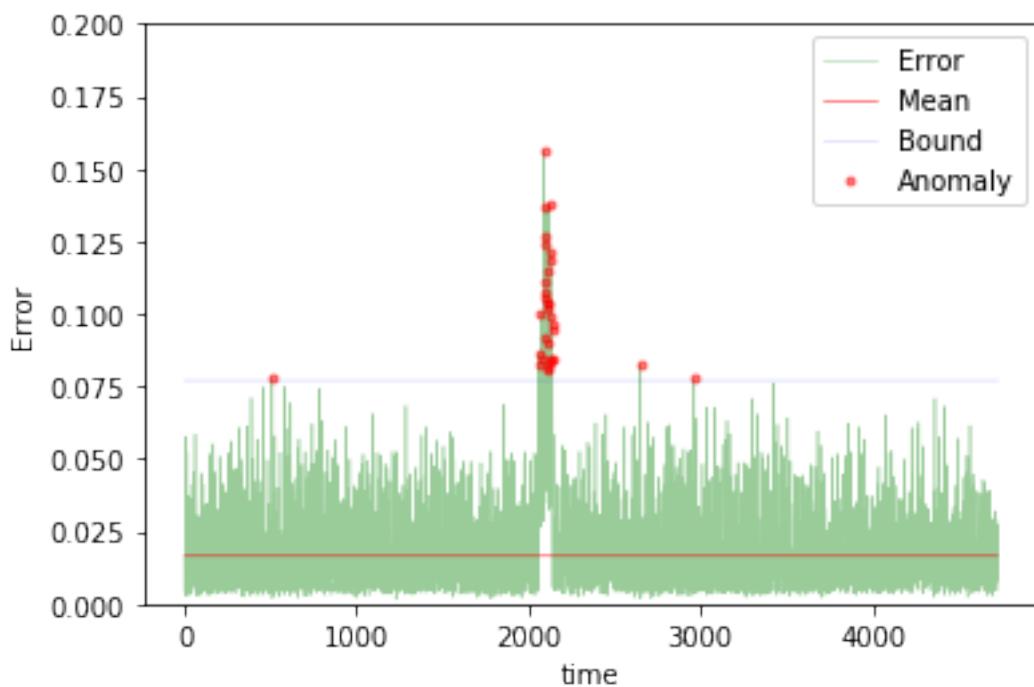
```
Training loss for final epoch is 0.017389860161347314
Validation loss for final epoch is 0.01731316165509634
----- Beginning tests for lin2 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
```

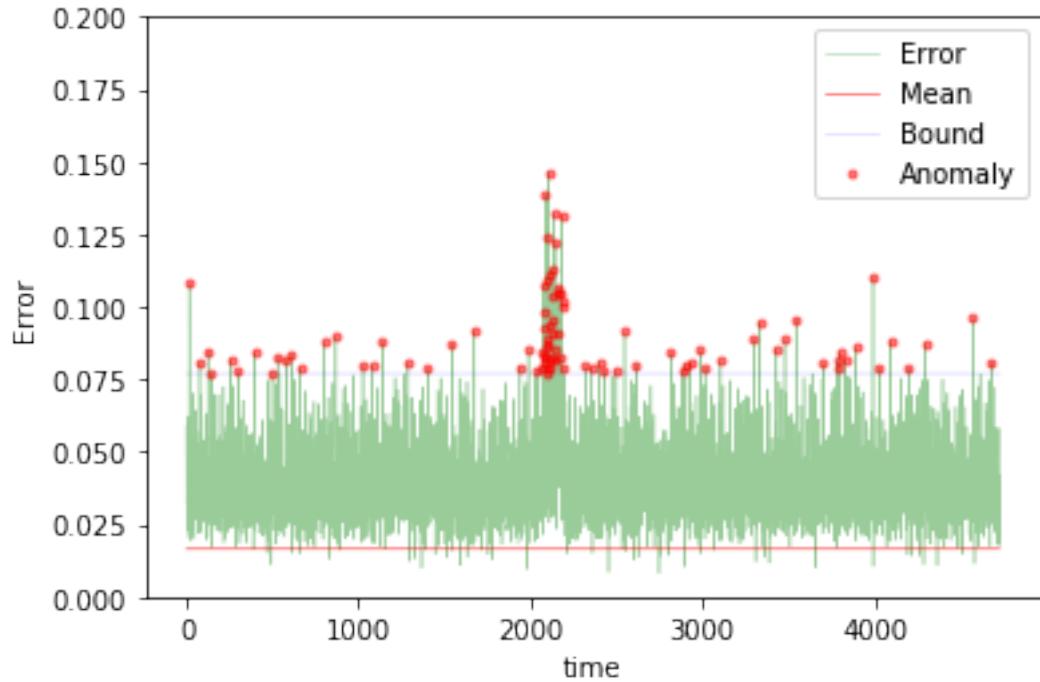




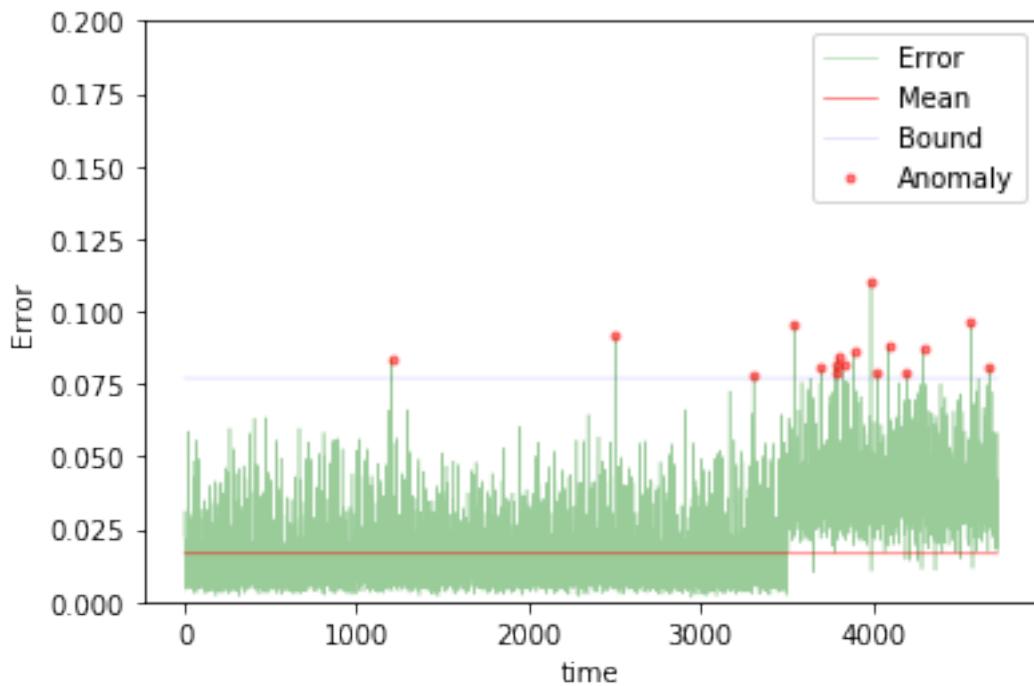
The mean error for lin2_normal_ is 0.01655509486734453 for length 4727
Testing on anomaly data.



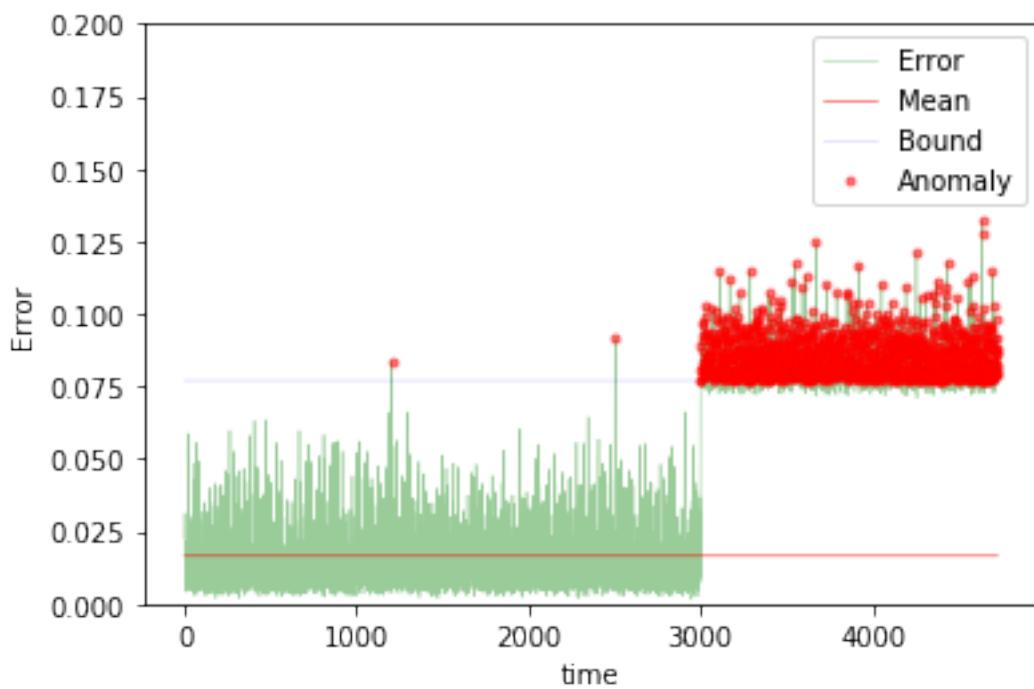
The mean error for lin2_anomaly_ is 0.018563440967838674 for length 4727
Testing on different app data.



The mean error for lin2_diff_app_ is 0.03878612360331067 for length 4727
Testing on App change synthetic data.



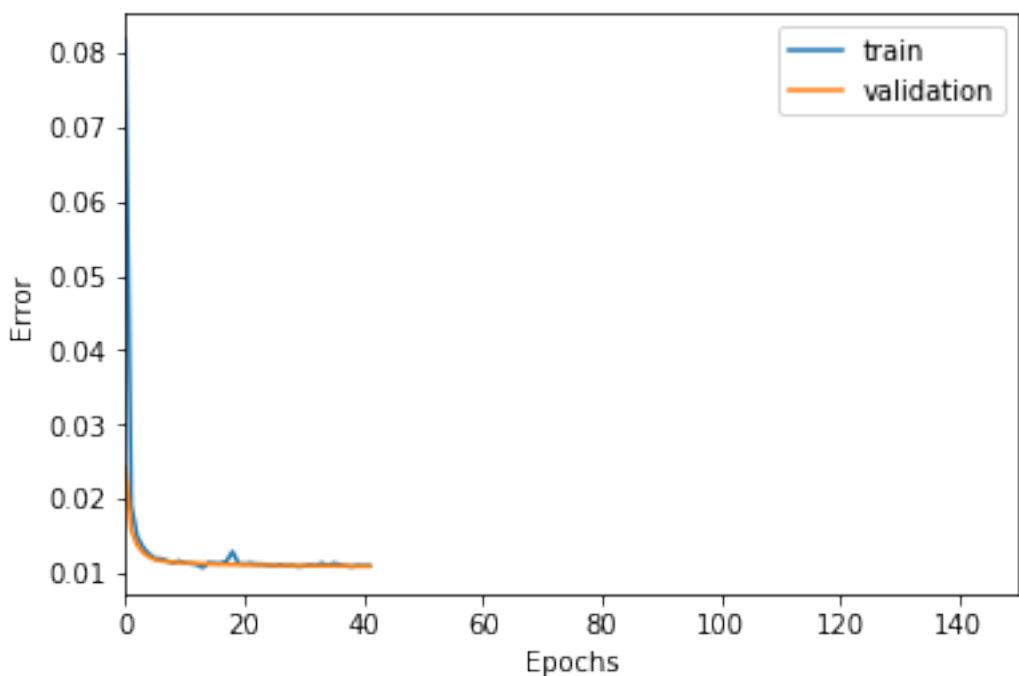
The mean error for lin2_app_change_ is 0.022111713874579796 for length 4727
Testing on Net flood synthetic data.



```
The mean error for lin2_net_flood_ is 0.04080915023322091 for length 4727  
=====
```

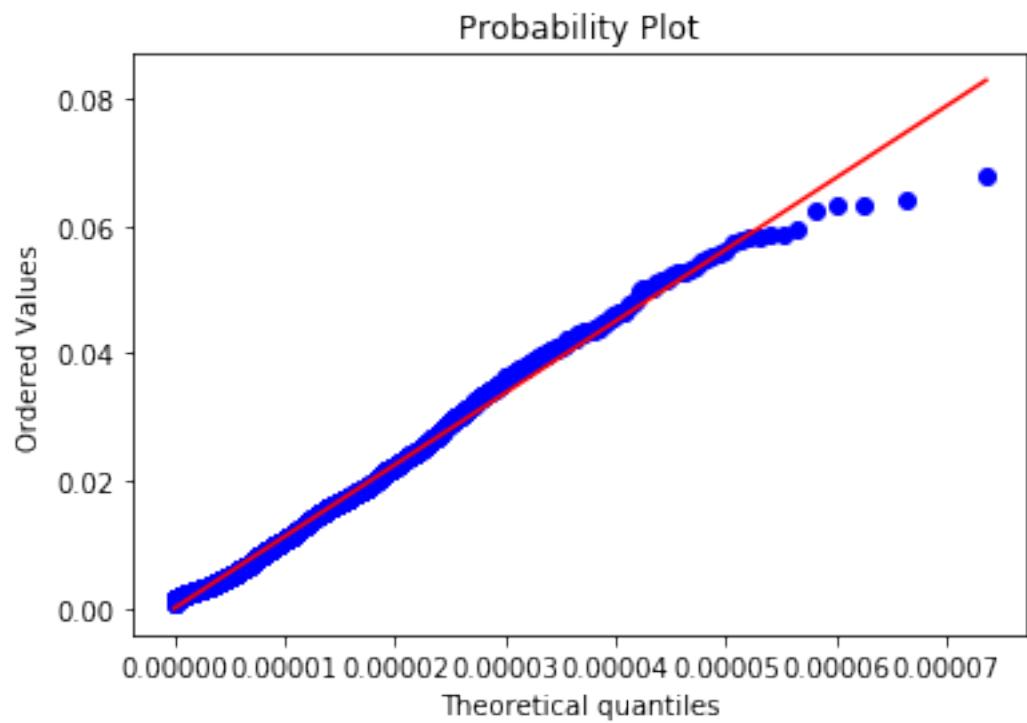
5 steps

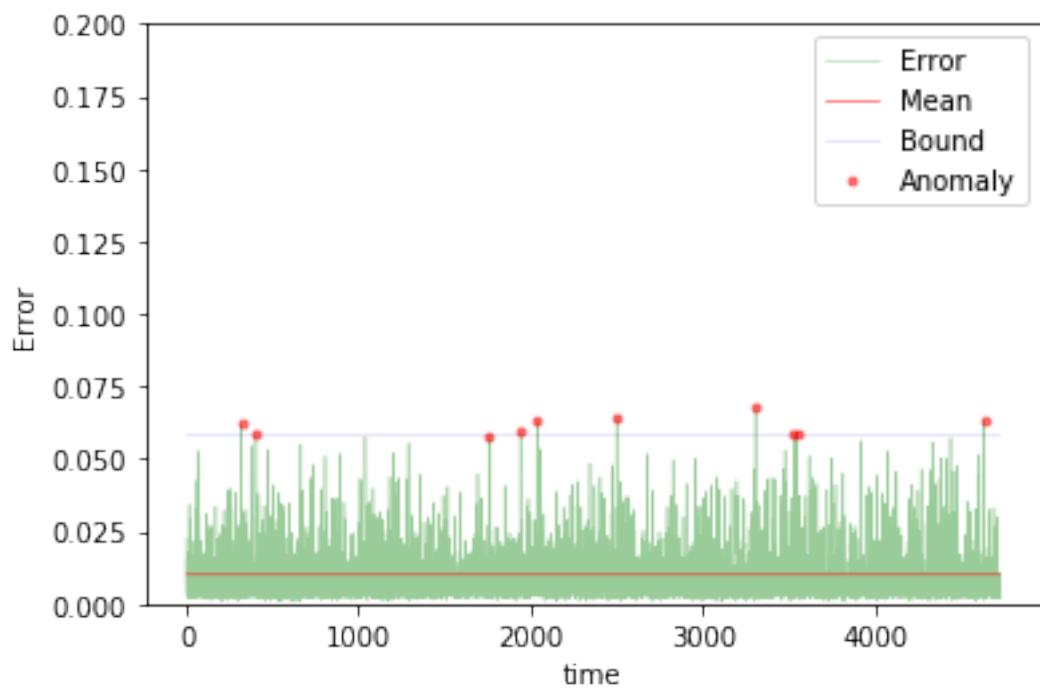
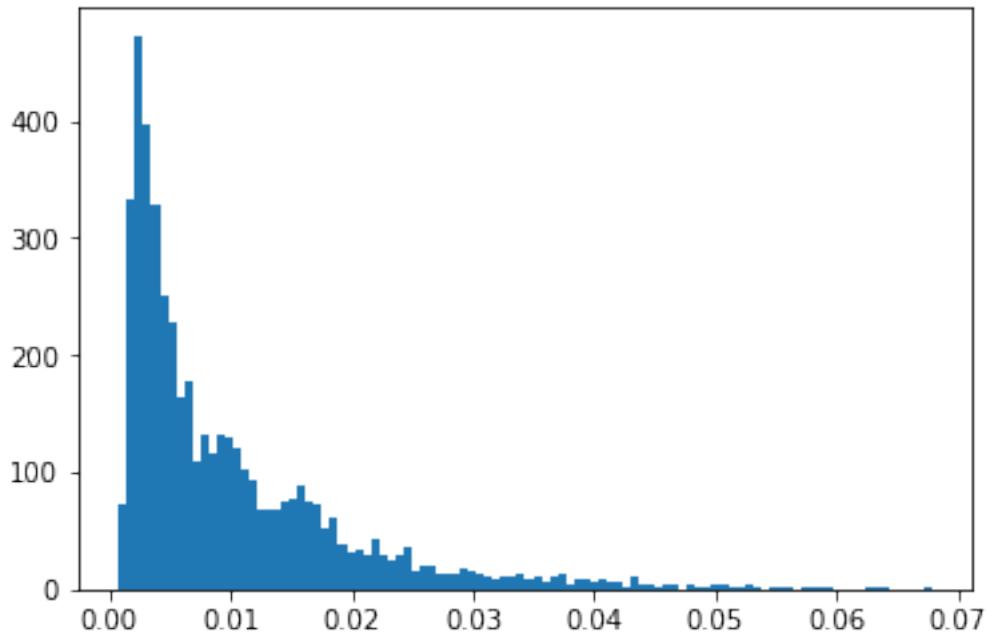
```
In [40]: TIMESTEPS = 5  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS)  
vgen = flat_generator(val_X, TIMESTEPS)  
name = "lin5"  
  
In [41]: input_layer = Input(shape=(TIMESTEPS*DIM,))  
output = Dense(DIM, activation='sigmoid')(input_layer)  
  
In [42]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [43]: train(model, tgen, vgen, name=name)  
test(model, name=name, window=TIMESTEPS)
```



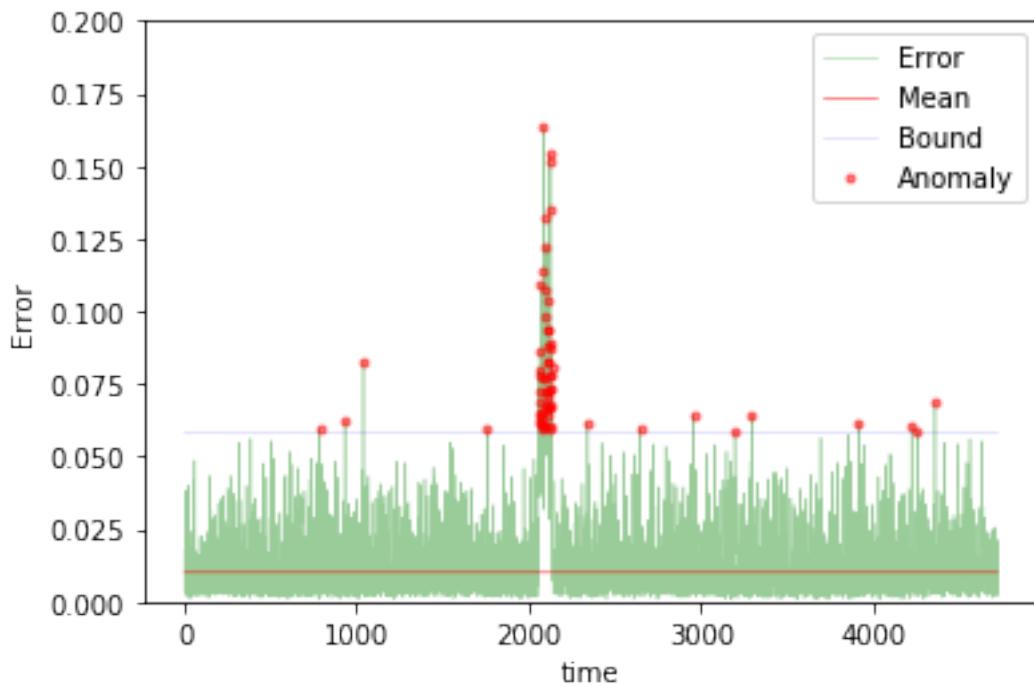
```
Training loss for final epoch is 0.010957506600185298
Validation loss for final epoch is 0.010861165395355784
----- Beginning tests for lin5 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
    sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
improvement from the last ten iterations.
warnings.warn(msg, RuntimeWarning)
```

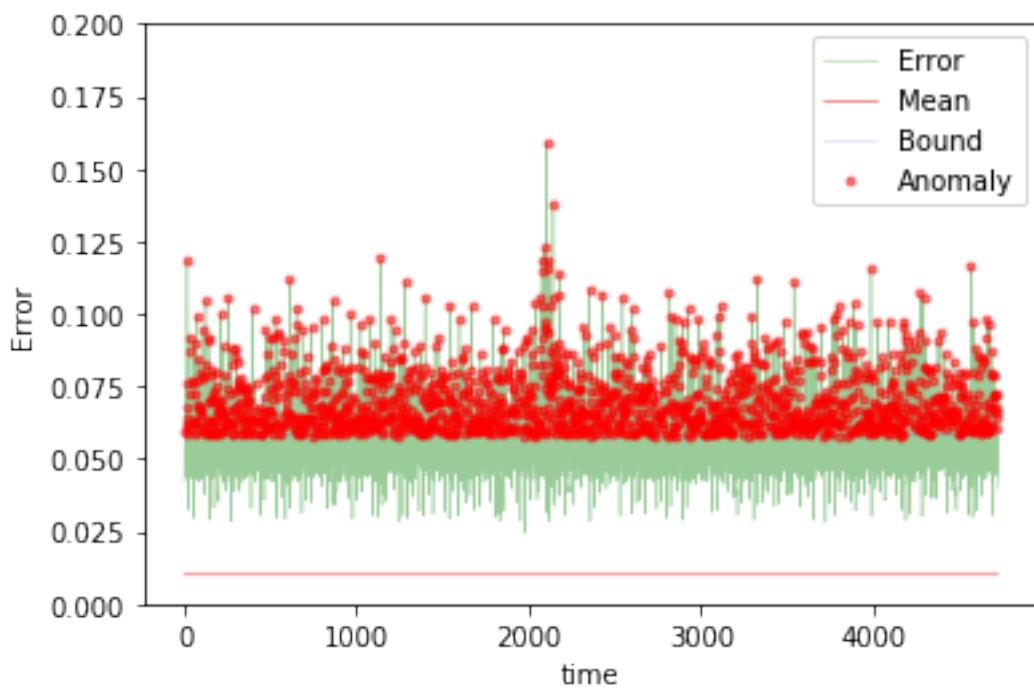




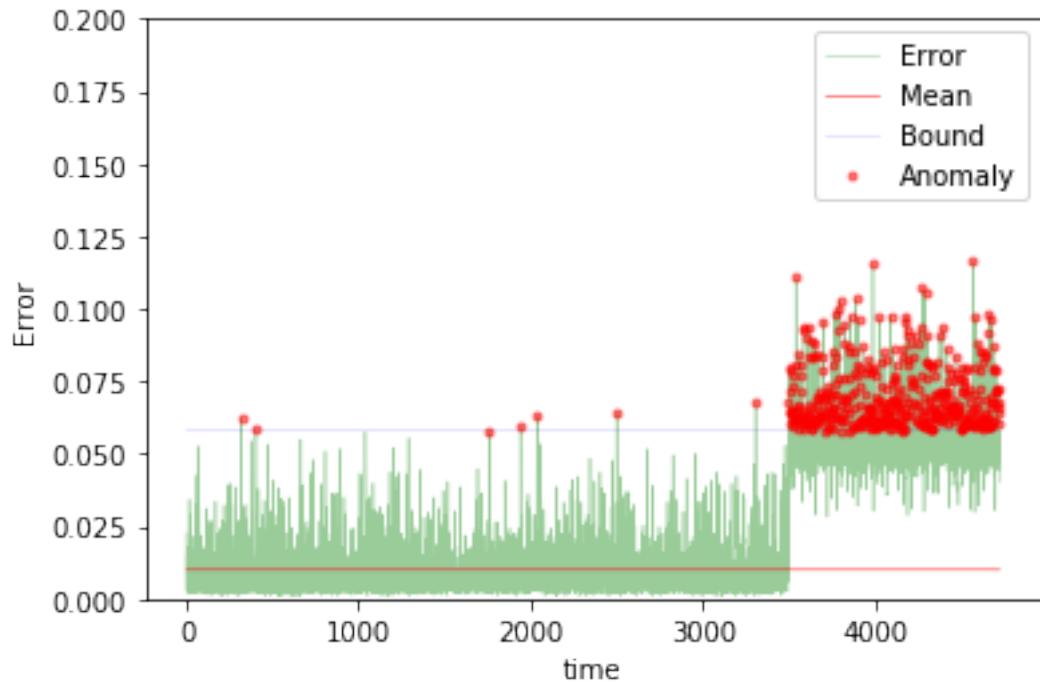
The mean error for lin5_normal_ is 0.01009074691672264 for length 4724
Testing on anomaly data.



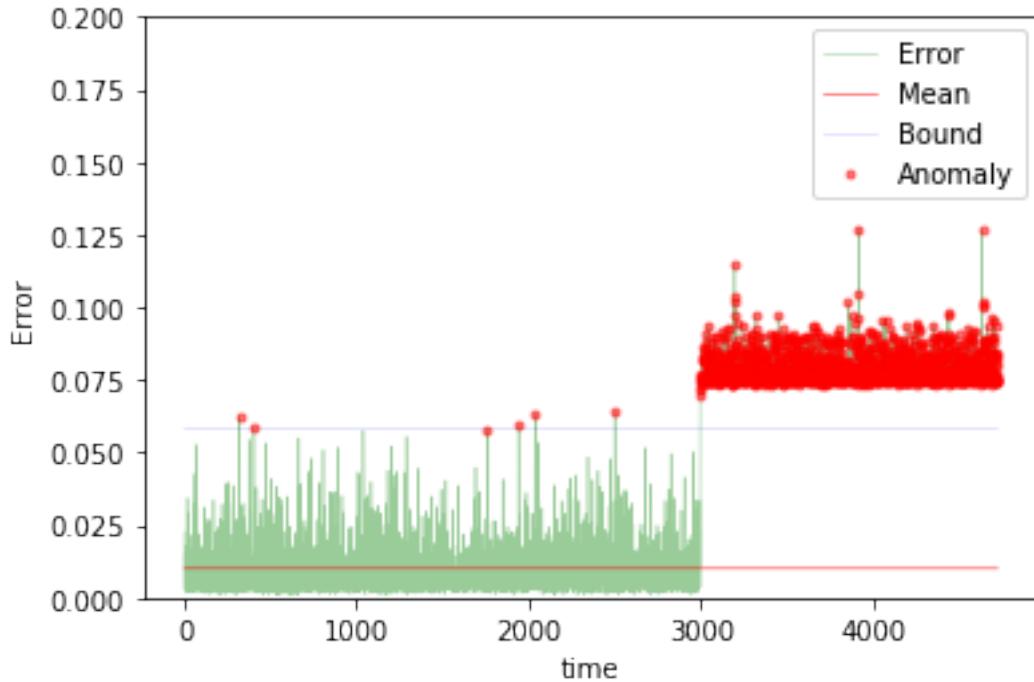
The mean error for lin5_anomaly_ is 0.012369398605104028 for length 4724
Testing on different app data.



The mean error for lin5_diff_app_ is 0.056360324528173224 for length 4724
Testing on App change synthetic data.



The mean error for lin5_app_change_ is 0.02199005424789996 for length 4724
Testing on Net flood synthetic data.



```
The mean error for lin5_net_flood_ is 0.03526307421125053 for length 4724
=====
```

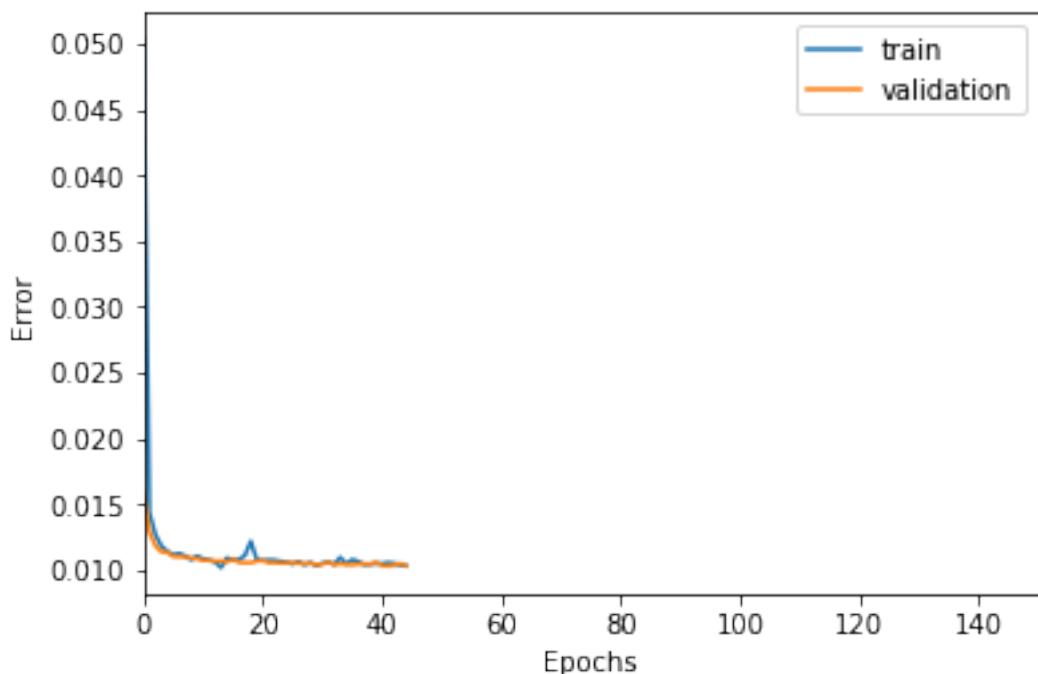
10 steps

```
In [44]: TIMESTEPS = 10
        DIM = 29
        tgen = flat_generator(X, TIMESTEPS)
        vgen = flat_generator(val_X, TIMESTEPS)
        name = "lin10"

In [45]: input_layer = Input(shape=(TIMESTEPS*DIM,))
        output = Dense(DIM, activation='sigmoid')(input_layer)

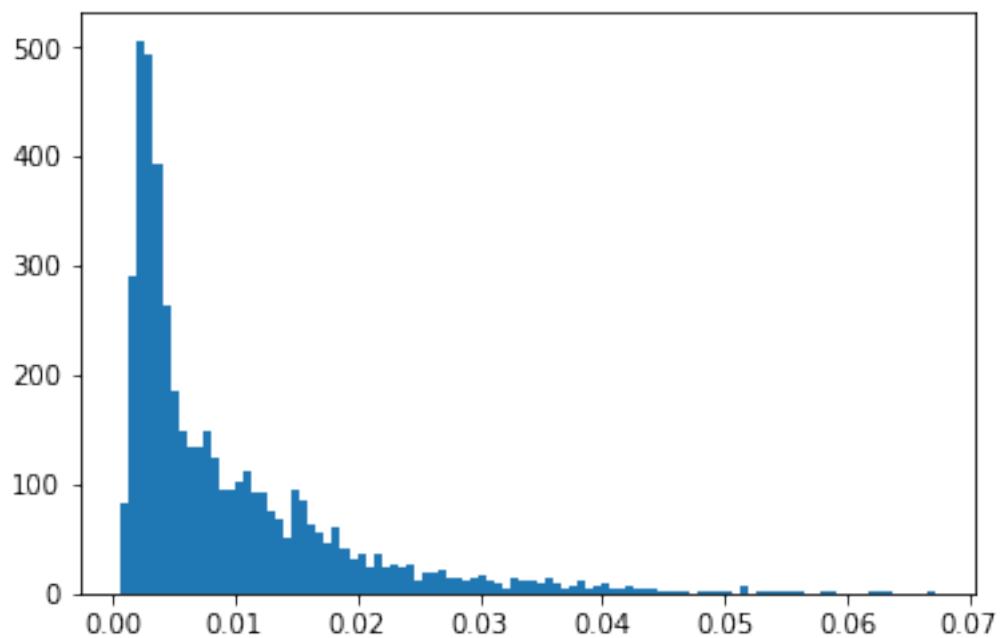
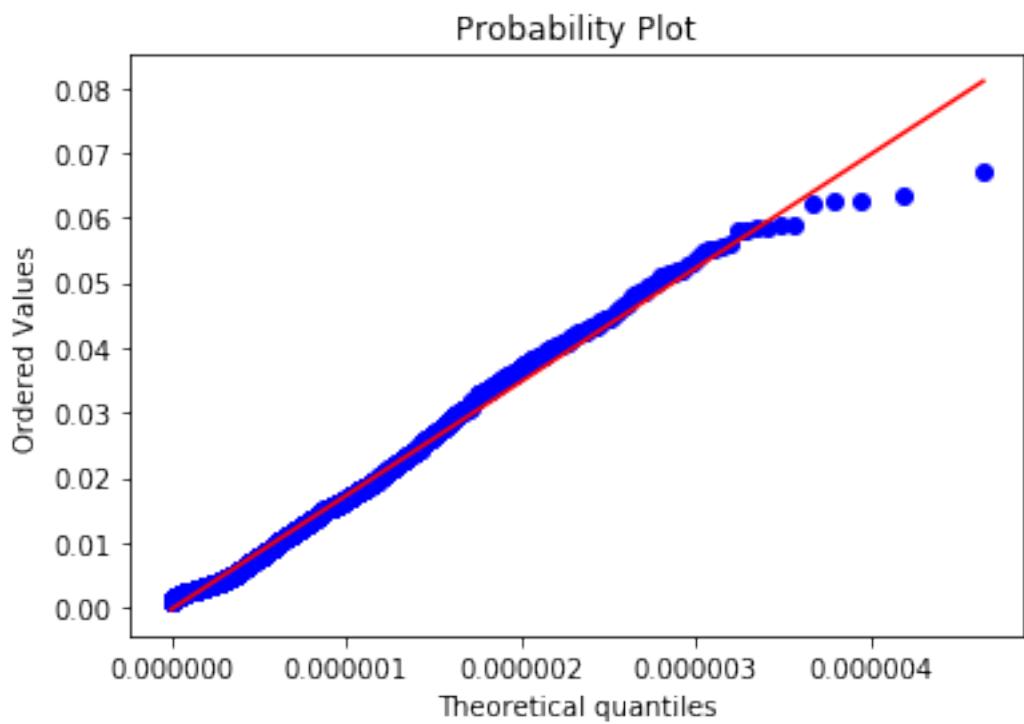
In [46]: model = Model(input_layer, output)
        model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

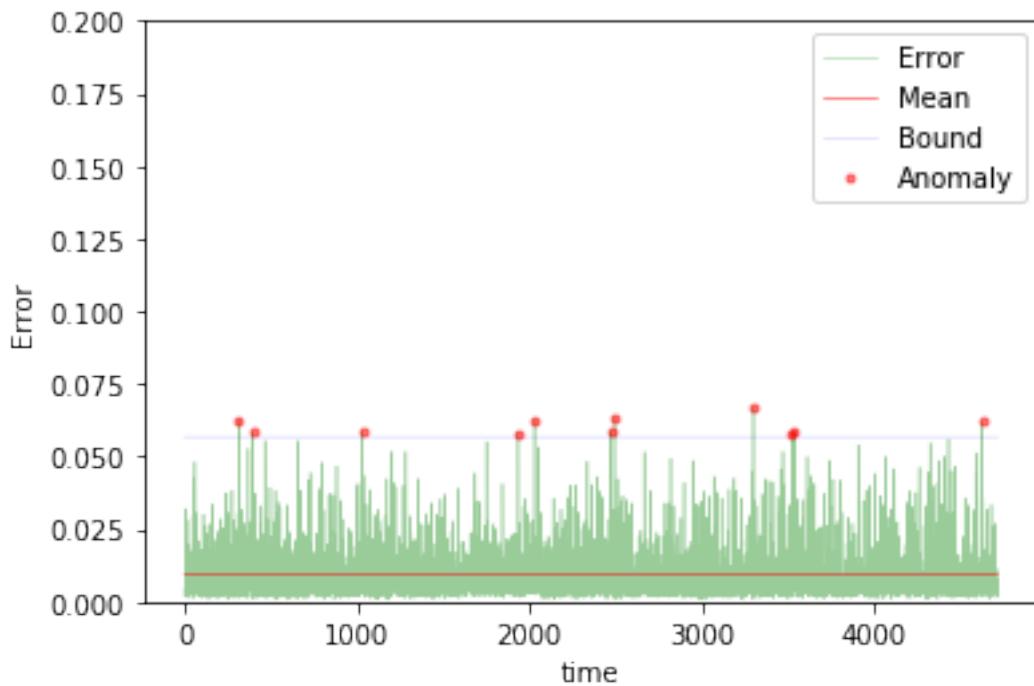
In [47]: train(model, tgen, vgen, name=name)
        test(model, name=name, window=TIMESTEPS)
```



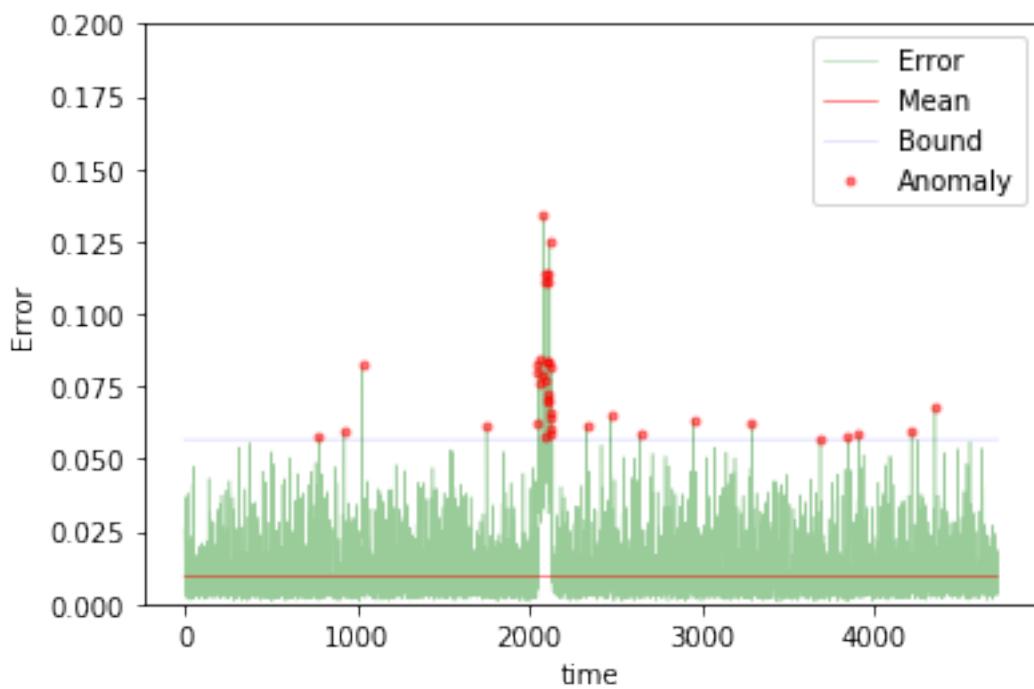
```
Training loss for final epoch is 0.010312253204756417
Validation loss for final epoch is 0.010331870075664483
----- Beginning tests for lin10 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
    warnings.warn(msg, RuntimeWarning)
```

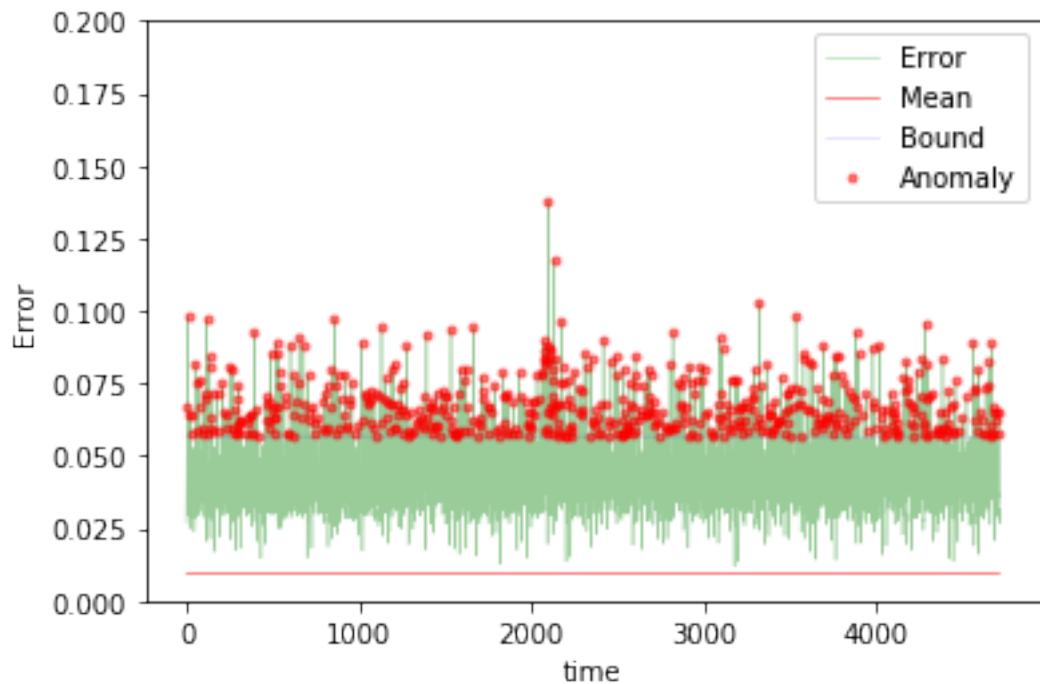




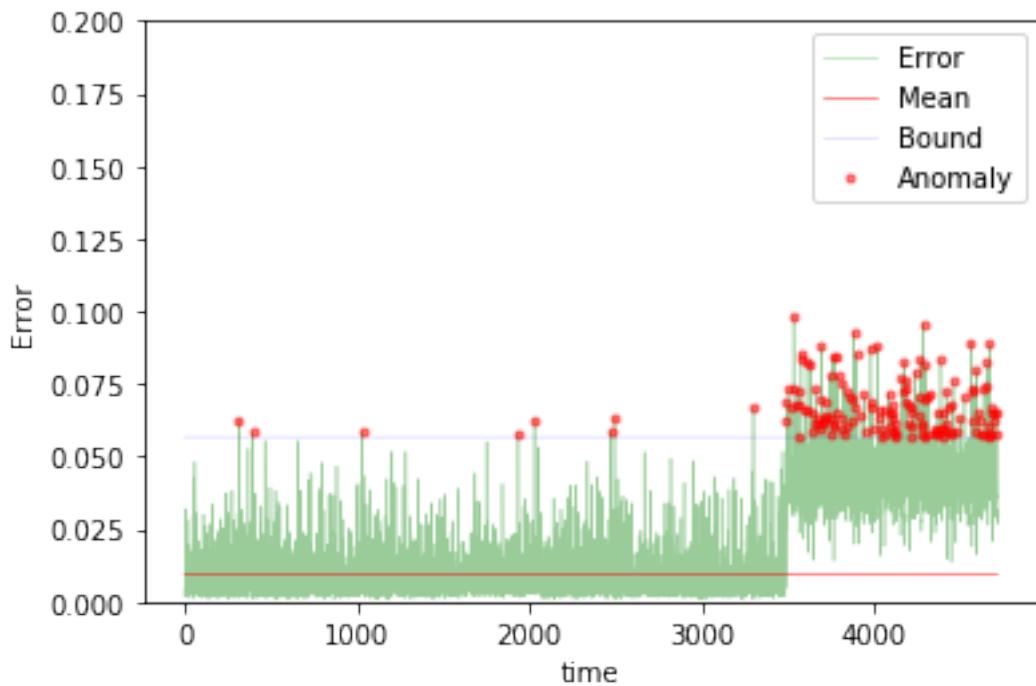
The mean error for lin10_normal_ is 0.009630723271066271 for length 4719
Testing on anomaly data.



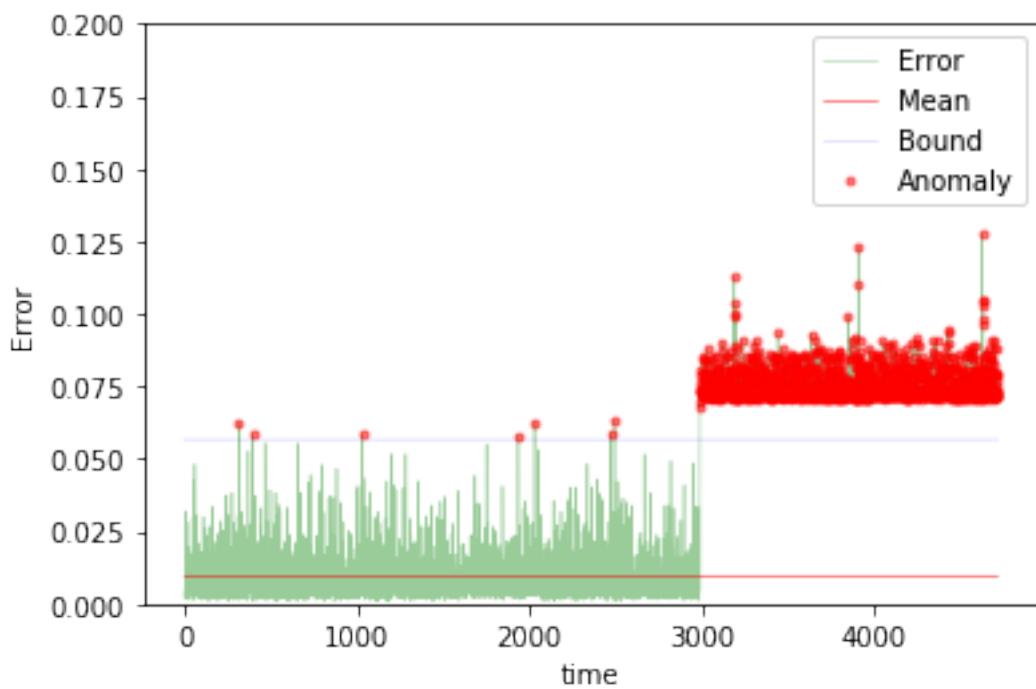
The mean error for lin10_anomaly_ is 0.011594947036201828 for length 4719
Testing on different app data.



The mean error for lin10_diff_app_ is 0.04283260678245369 for length 4719
Testing on App change synthetic data.



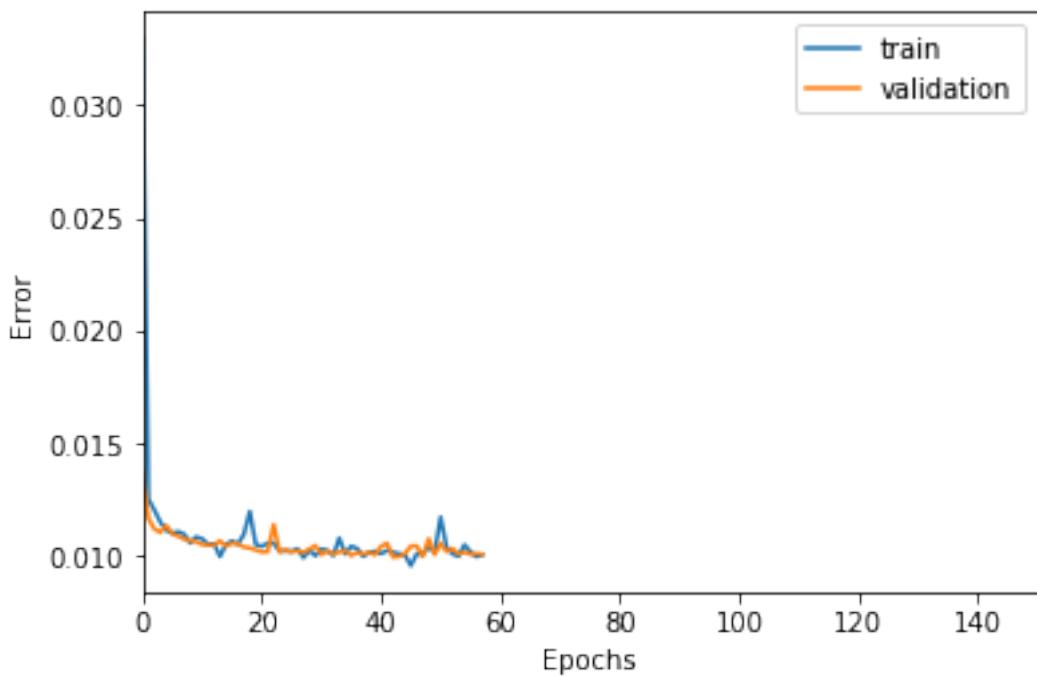
The mean error for lin10_app_change_ is 0.018160527243946203 for length 4719
Testing on Net flood synthetic data.



```
The mean error for lin10_net_flood_ is 0.03377499225988397 for length 4719  
=====
```

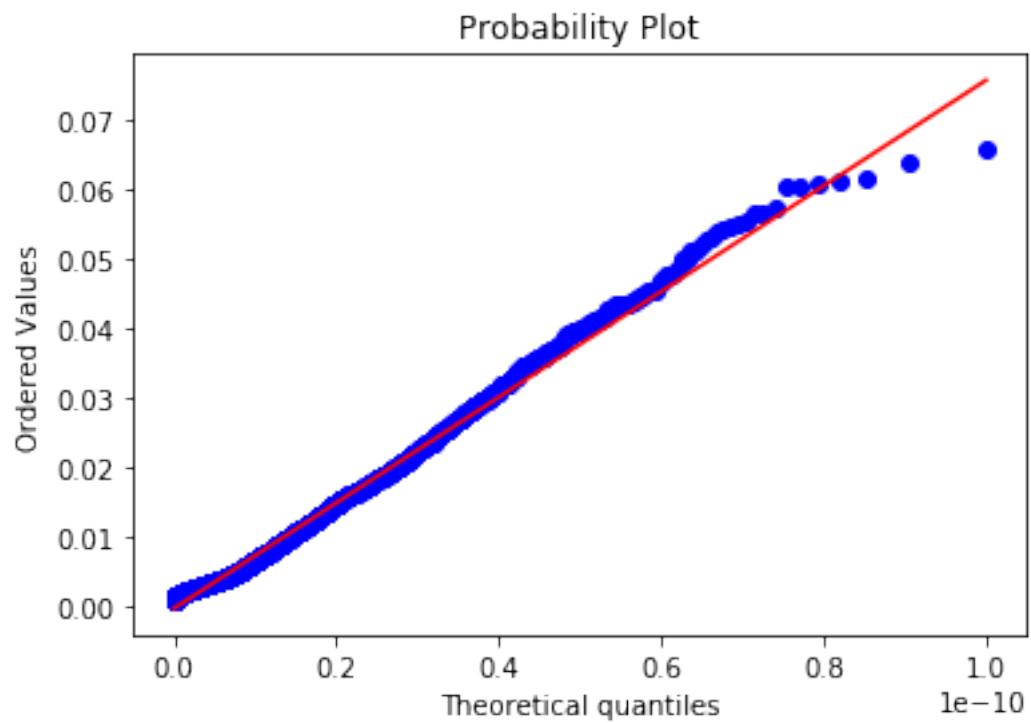
20 steps

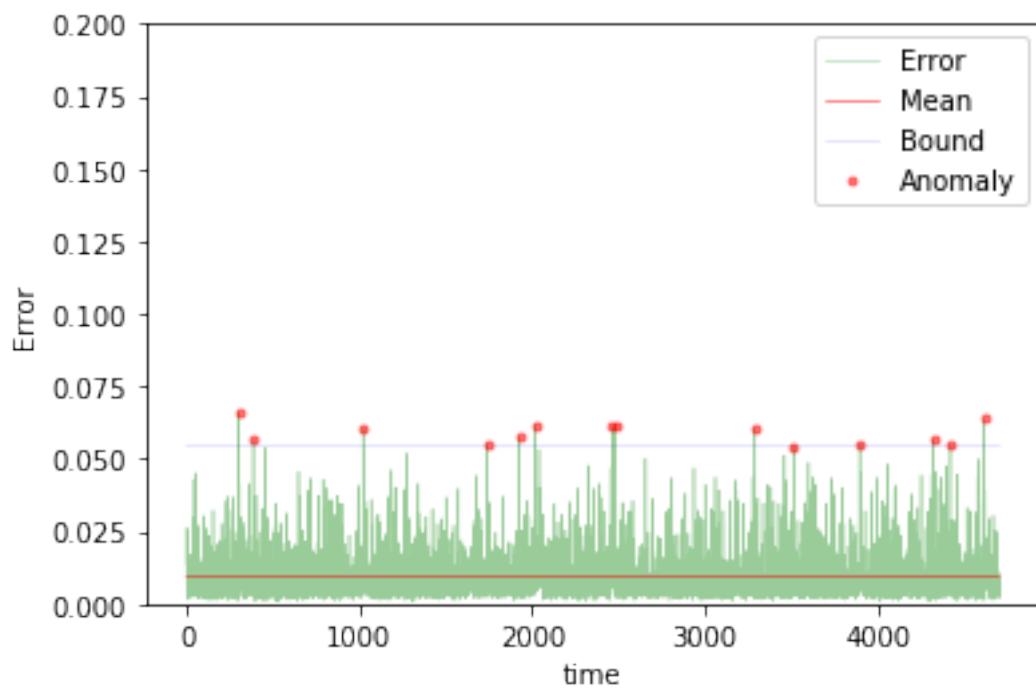
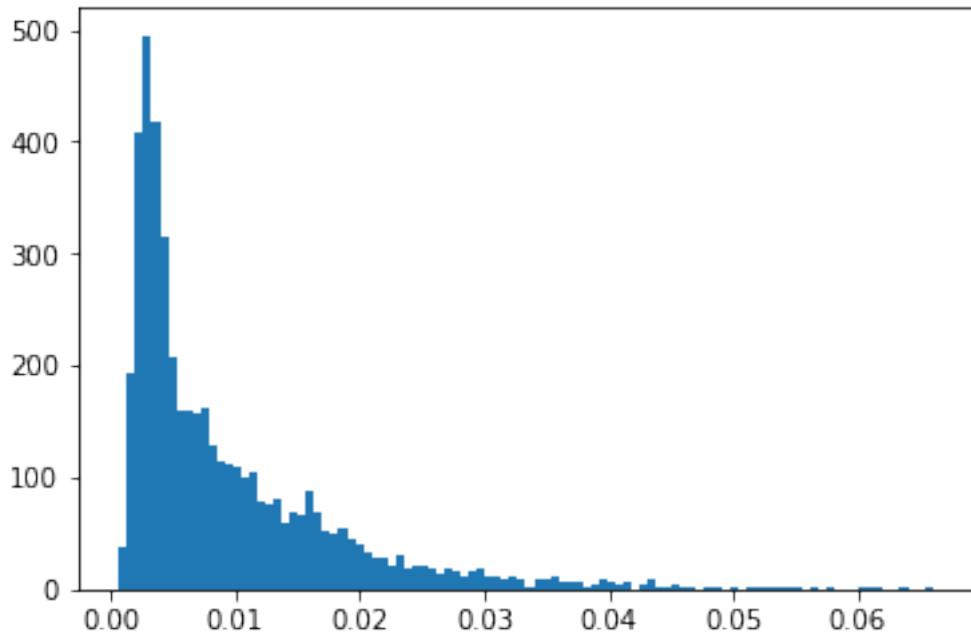
```
In [48]: TIMESTEPS = 20  
        DIM = 29  
        tgen = flat_generator(X, TIMESTEPS)  
        vgen = flat_generator(val_X, TIMESTEPS)  
        name = "lin20"  
  
In [49]: input_layer = Input(shape=(TIMESTEPS*DIM,))  
        output = Dense(DIM, activation='sigmoid')(input_layer)  
  
In [50]: model = Model(input_layer, output)  
        model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [51]: train(model, tgen, vgen, name=name)  
        test(model, name=name, window=TIMESTEPS)
```



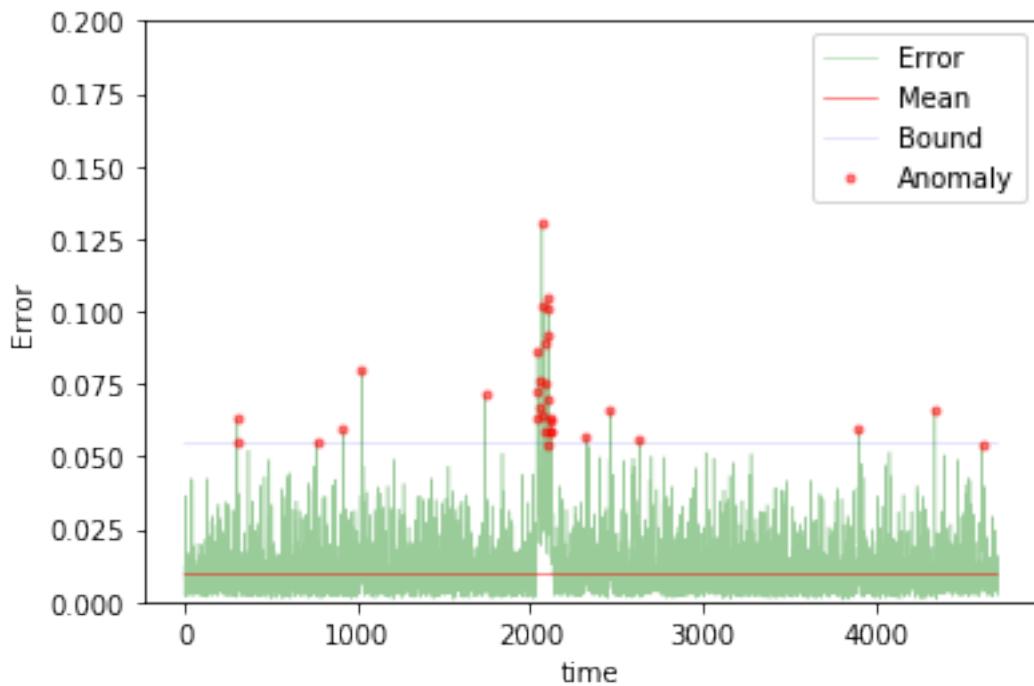
```
Training loss for final epoch is 0.010084782492951489
Validation loss for final epoch is 0.010088384414906613
----- Beginning tests for lin20 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
    sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
improvement from the last ten iterations.
warnings.warn(msg, RuntimeWarning)
```

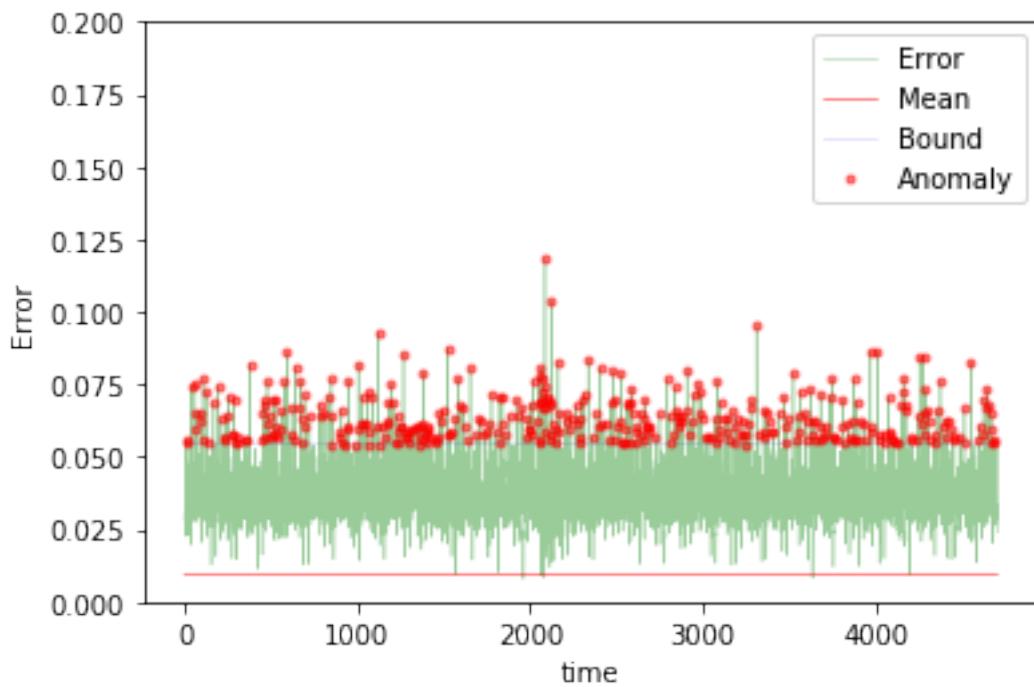




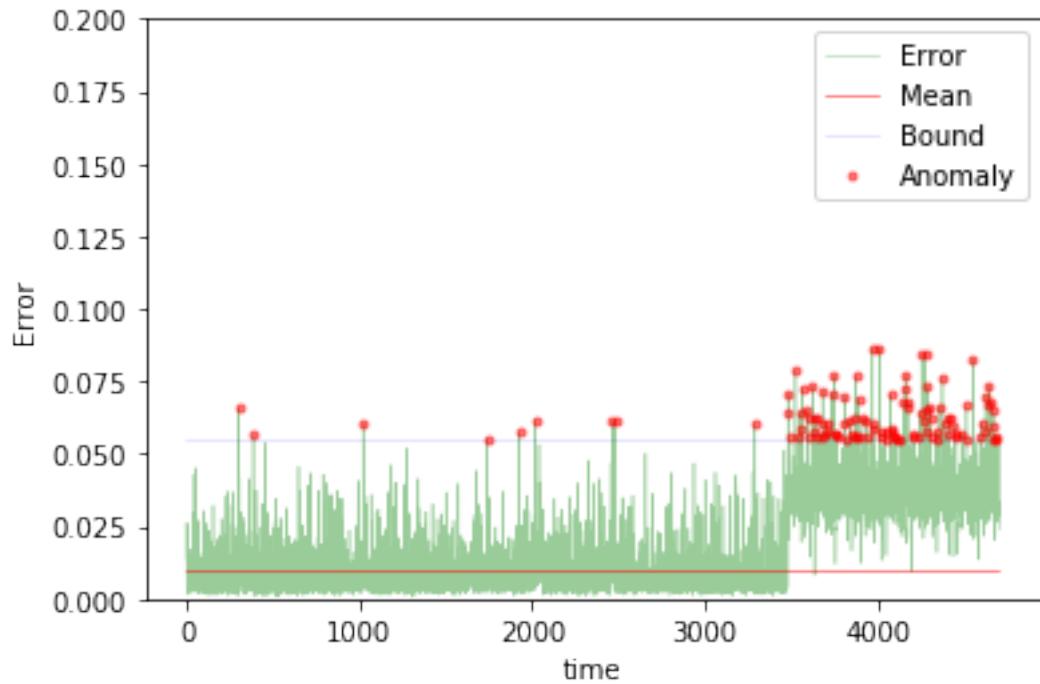
The mean error for lin20_normal_ is 0.009627444817040526 for length 4709
Testing on anomaly data.



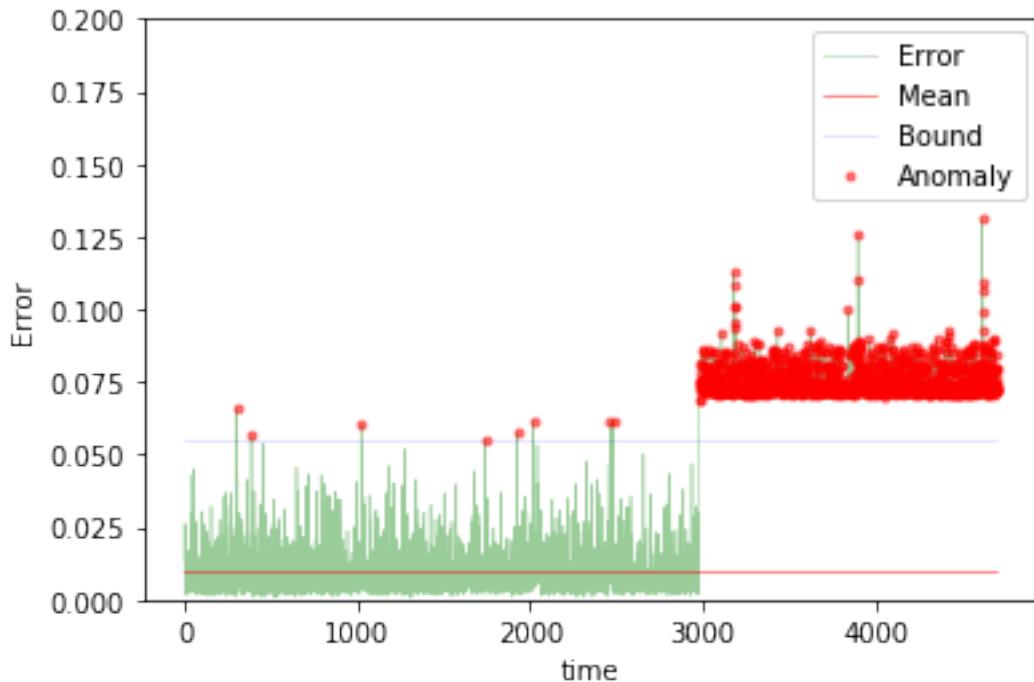
The mean error for lin20_anomaly_ is 0.010888487342045918 for length 4709
Testing on different app data.



The mean error for lin20_diff_app_ is 0.037454910684489244 for length 4709
Testing on App change synthetic data.



The mean error for lin20_app_change_ is 0.016817751211270243 for length 4709
Testing on Net flood synthetic data.



```
The mean error for lin20_net_flood_ is 0.033679282840746394 for length 4709
=====
```

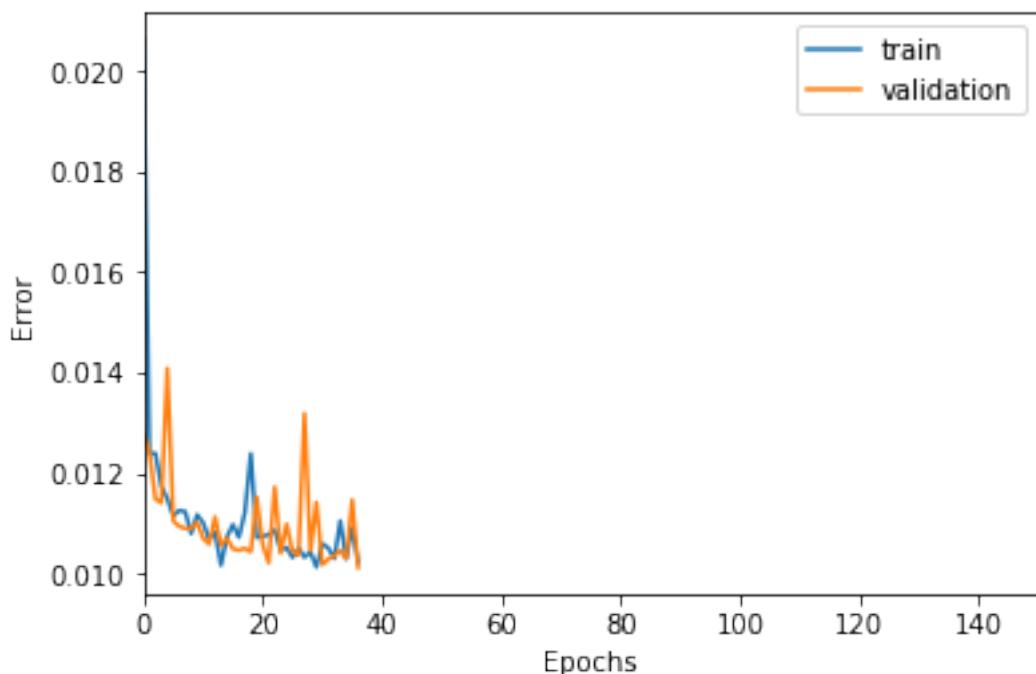
50 steps

```
In [52]: TIMESTEPS = 50
        DIM = 29
        tgen = flat_generator(X, TIMESTEPS)
        vgen = flat_generator(val_X, TIMESTEPS)
        name = "lin50"

In [53]: input_layer = Input(shape=(TIMESTEPS*DIM,))
        output = Dense(DIM, activation='sigmoid')(input_layer)

In [54]: model = Model(input_layer, output)
        model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

In [55]: train(model, tgen, vgen, name=name)
        test(model, name=name, window=TIMESTEPS)
```



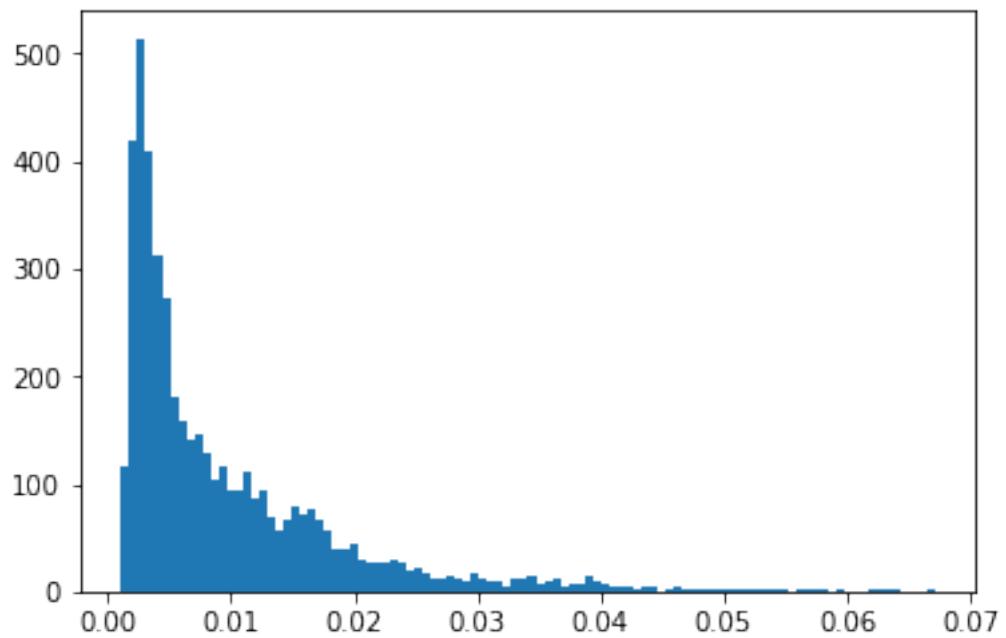
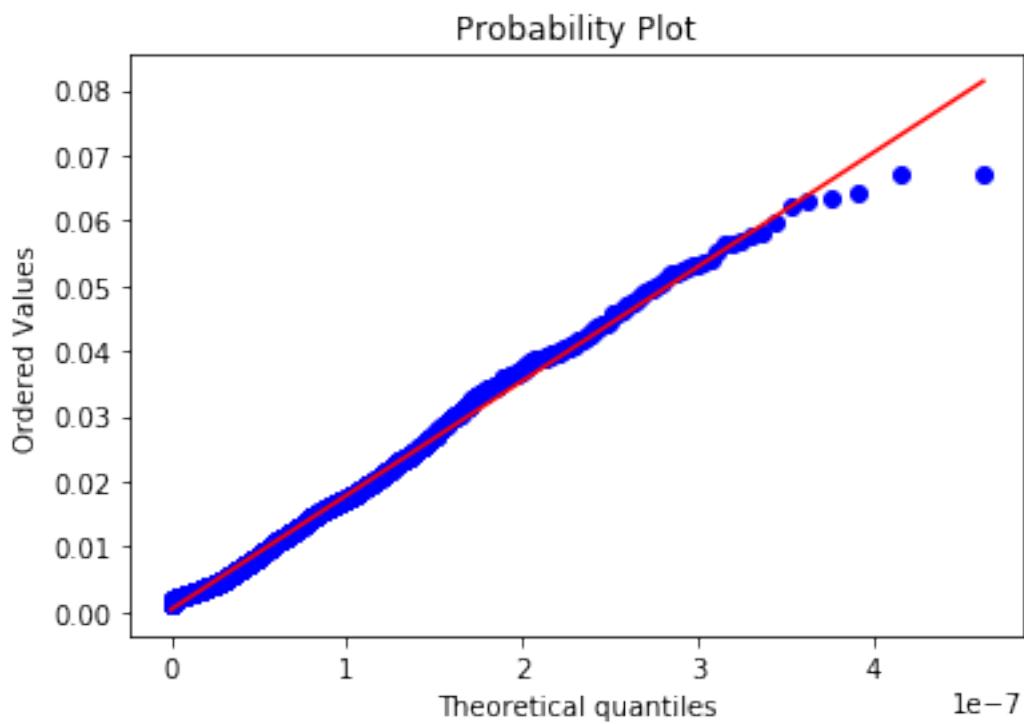
Training loss for final epoch is 0.010230929027311503

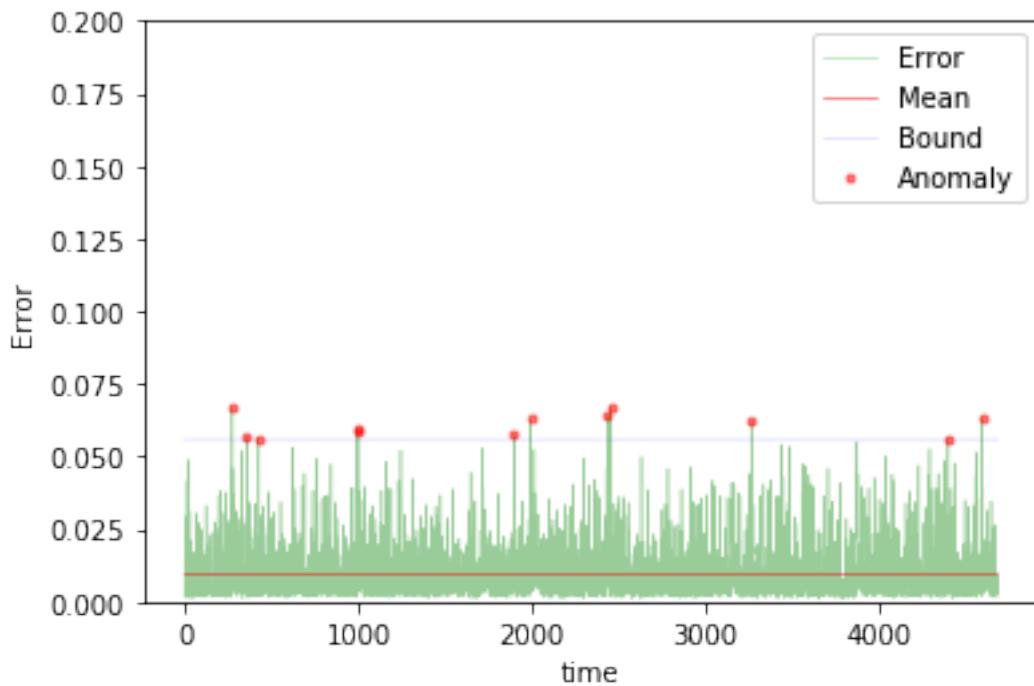
Validation loss for final epoch is 0.010121707968530244

----- Beginning tests for lin50 -----

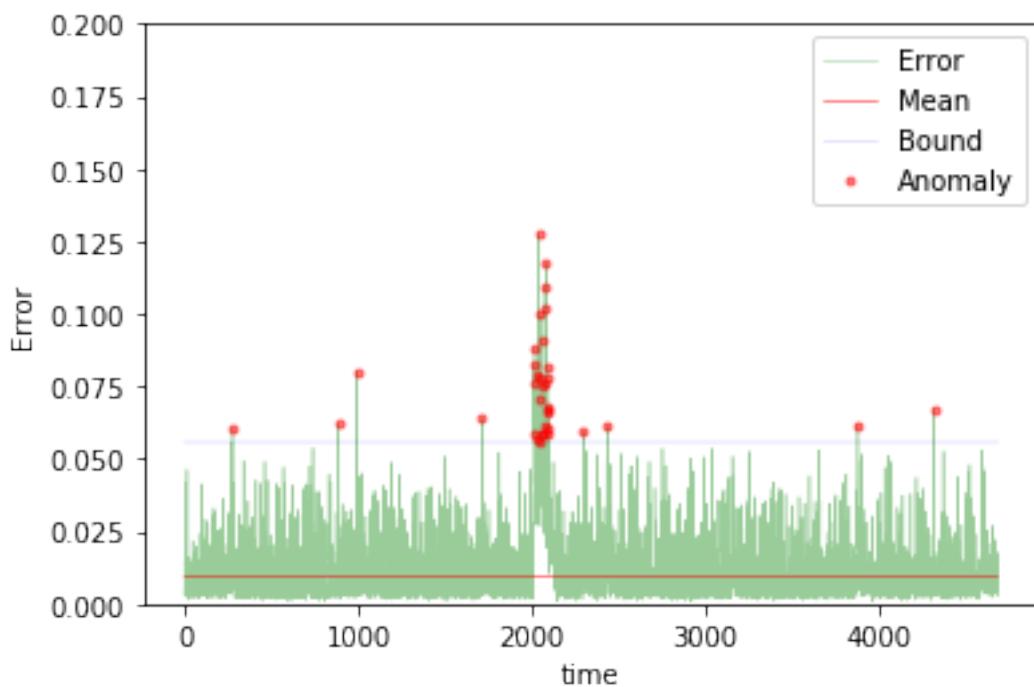
Testing on normal data.

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
    warnings.warn(msg, RuntimeWarning)
```

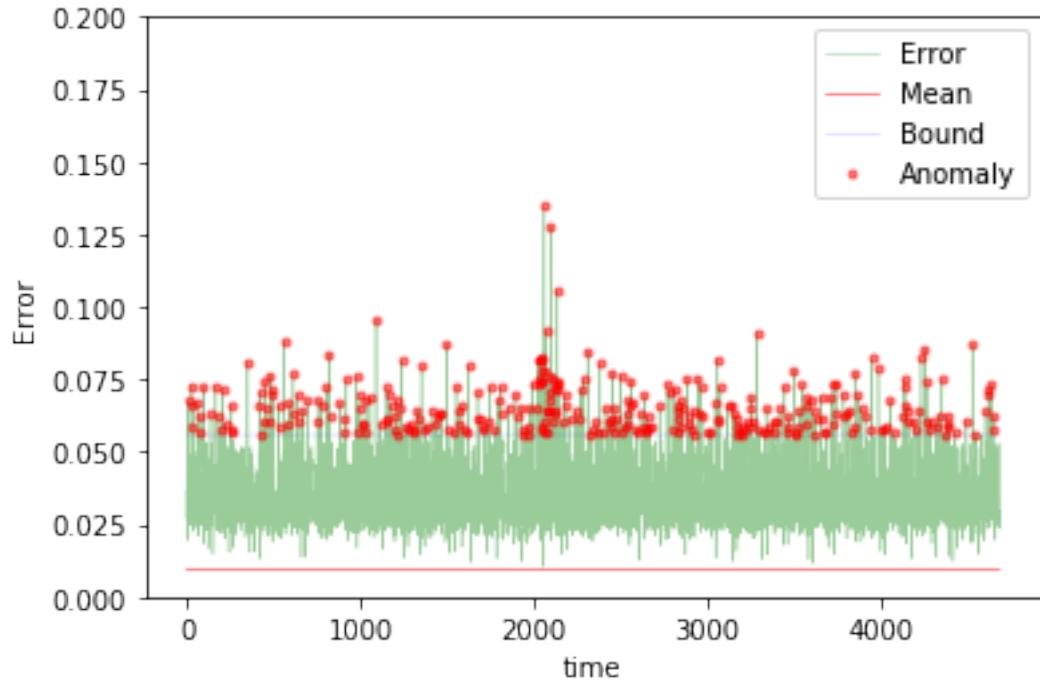




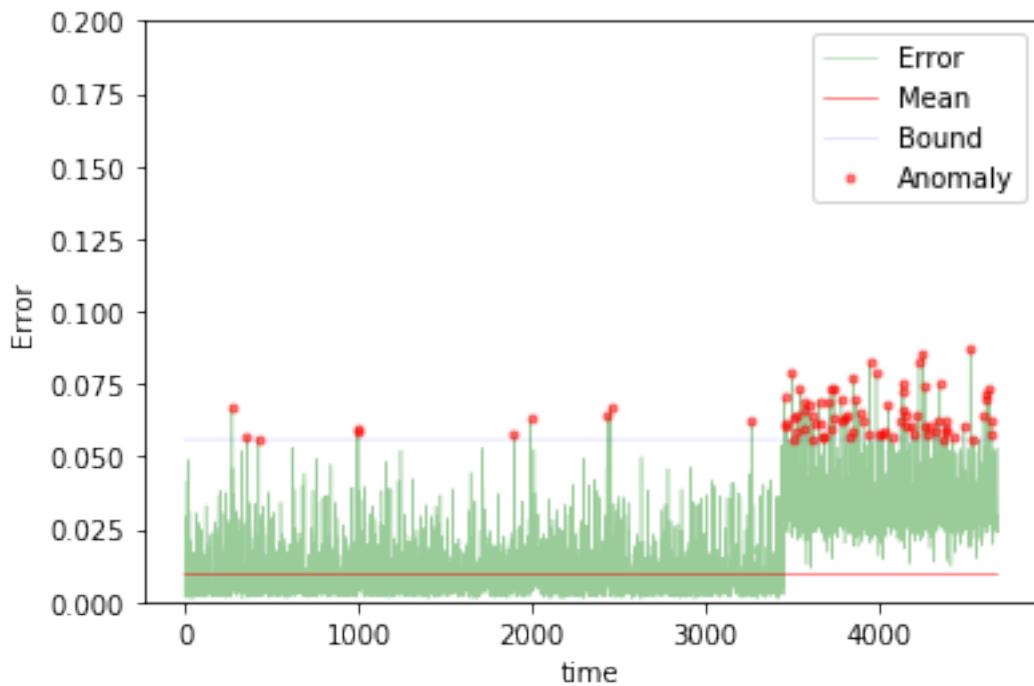
The mean error for lin50_normal_ is 0.009684156224200475 for length 4679
Testing on anomaly data.



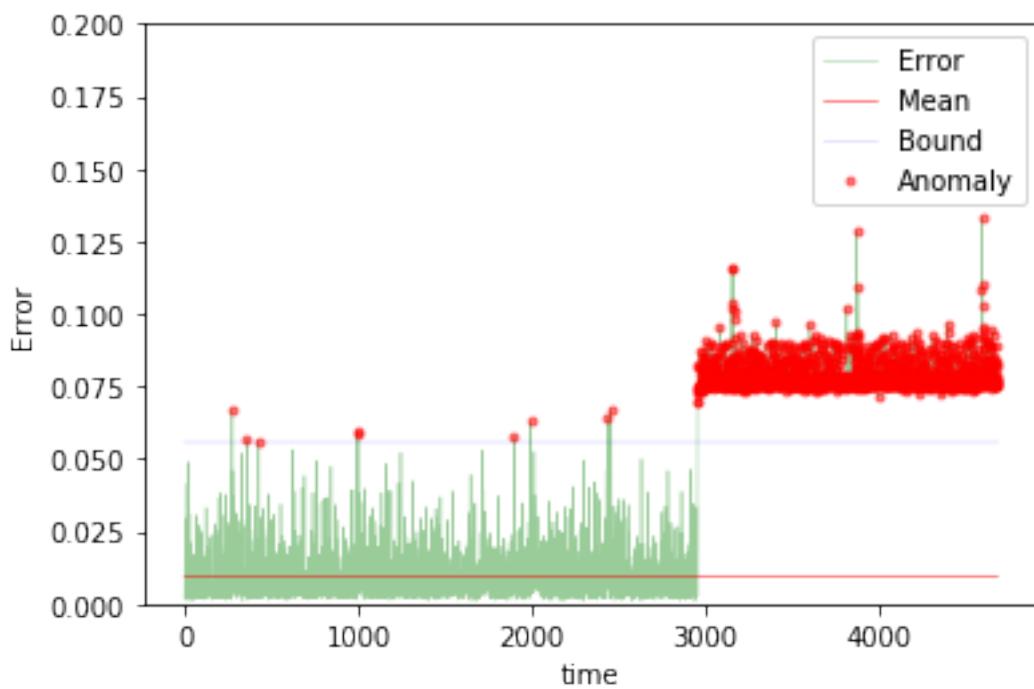
The mean error for lin50_anomaly_ is 0.01125377227224112 for length 4679
Testing on different app data.



The mean error for lin50_diff_app_ is 0.03499984172434301 for length 4679
Testing on App change synthetic data.



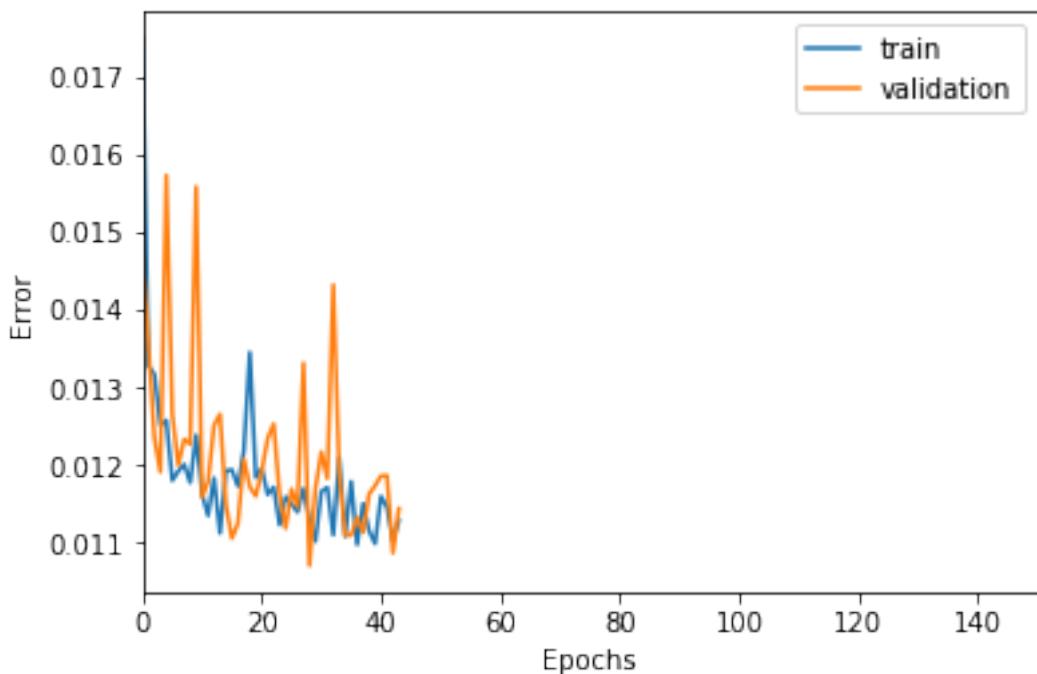
The mean error for lin50_app_change_ is 0.016271085869223337 for length 4679
Testing on Net flood synthetic data.



```
The mean error for lin50_net_flood_ is 0.03525888783488409 for length 4679  
=====
```

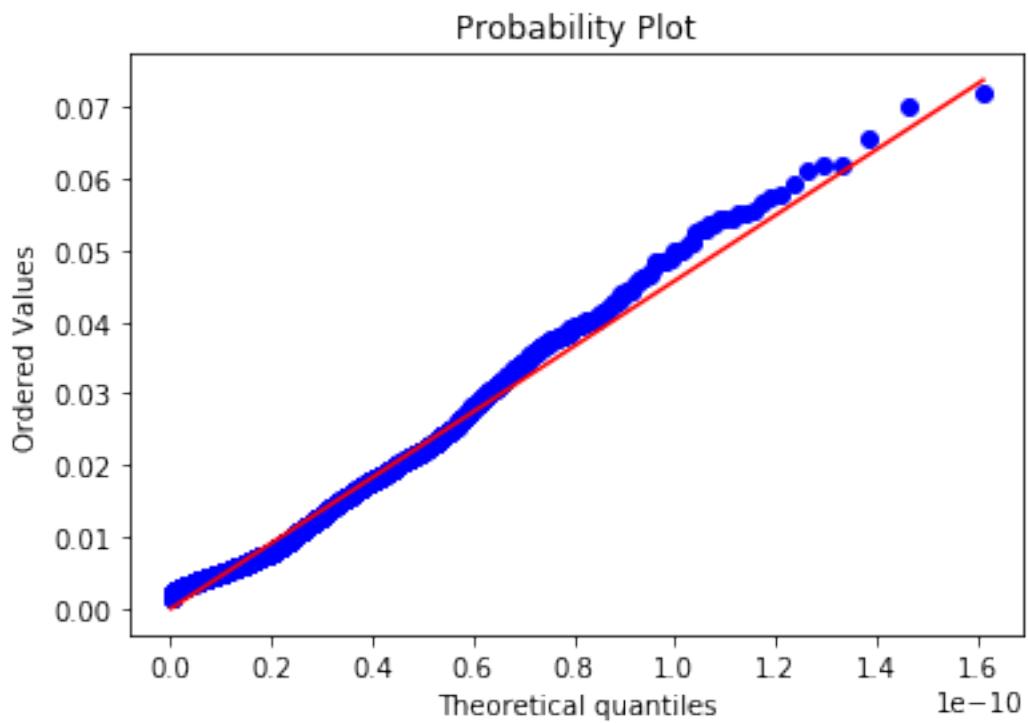
100 steps

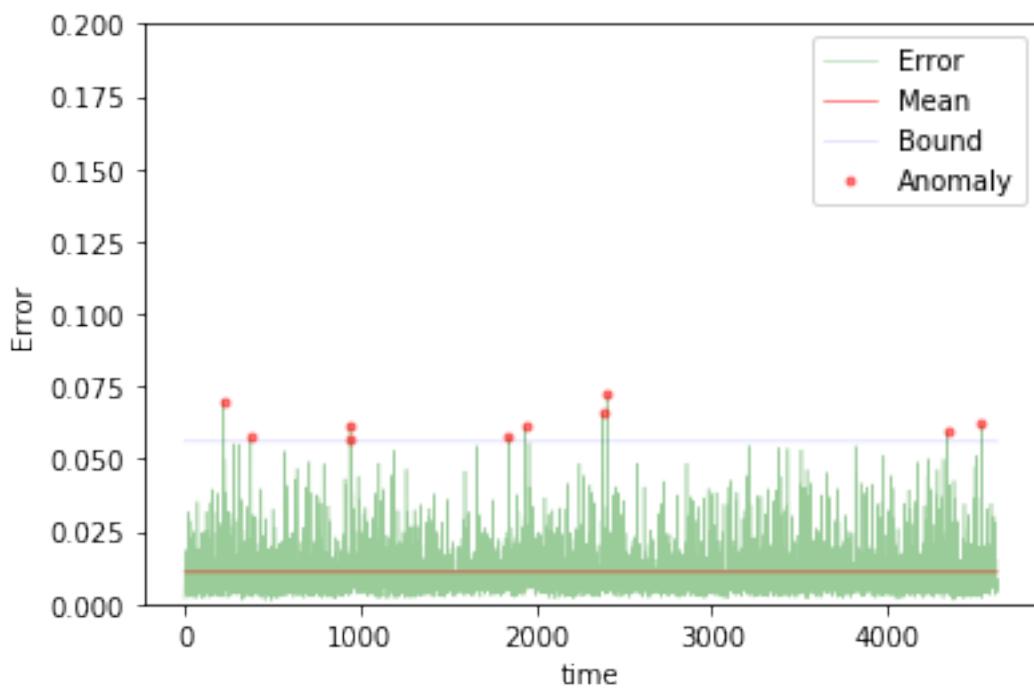
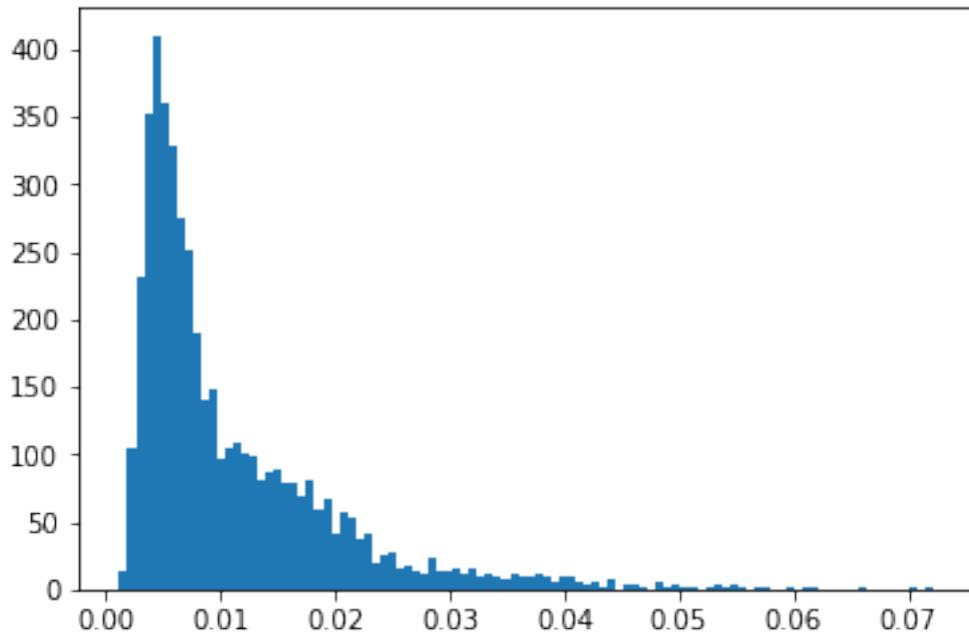
```
In [56]: TIMESTEPS = 100  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS)  
vgen = flat_generator(val_X, TIMESTEPS)  
name = "lin100"  
  
In [57]: input_layer = Input(shape=(TIMESTEPS*DIM,))  
output = Dense(DIM, activation='sigmoid')(input_layer)  
  
In [58]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [59]: train(model, tgen, vgen, name=name)  
test(model, name=name, window=TIMESTEPS)
```



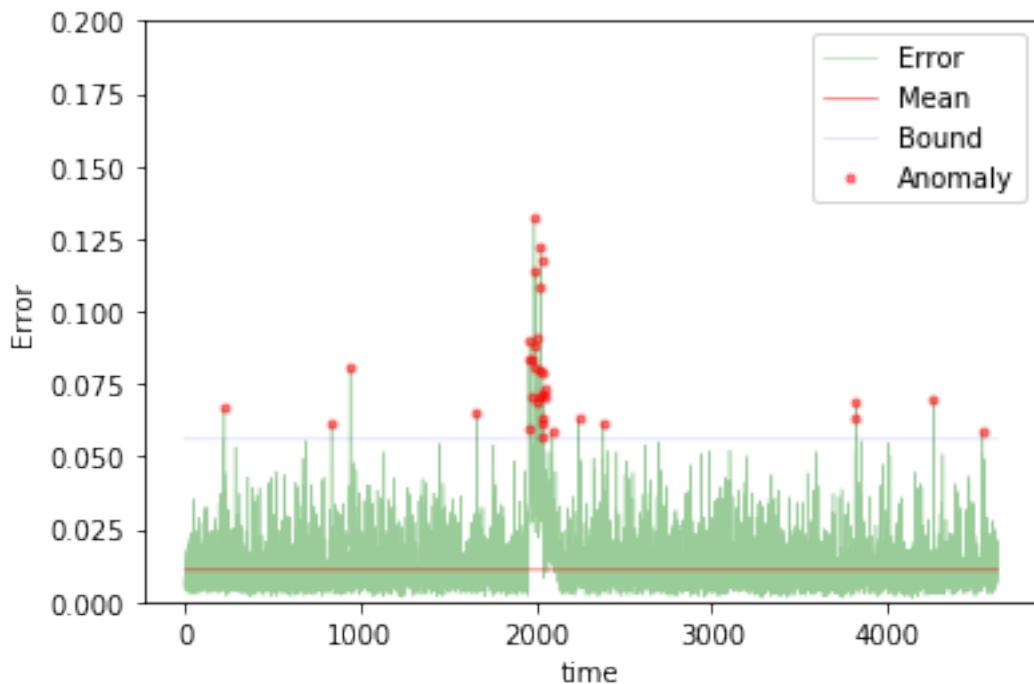
```
Training loss for final epoch is 0.011290103742619977
Validation loss for final epoch is 0.011443300546845421
----- Beginning tests for lin100 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
    sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
improvement from the last ten iterations.
warnings.warn(msg, RuntimeWarning)
```

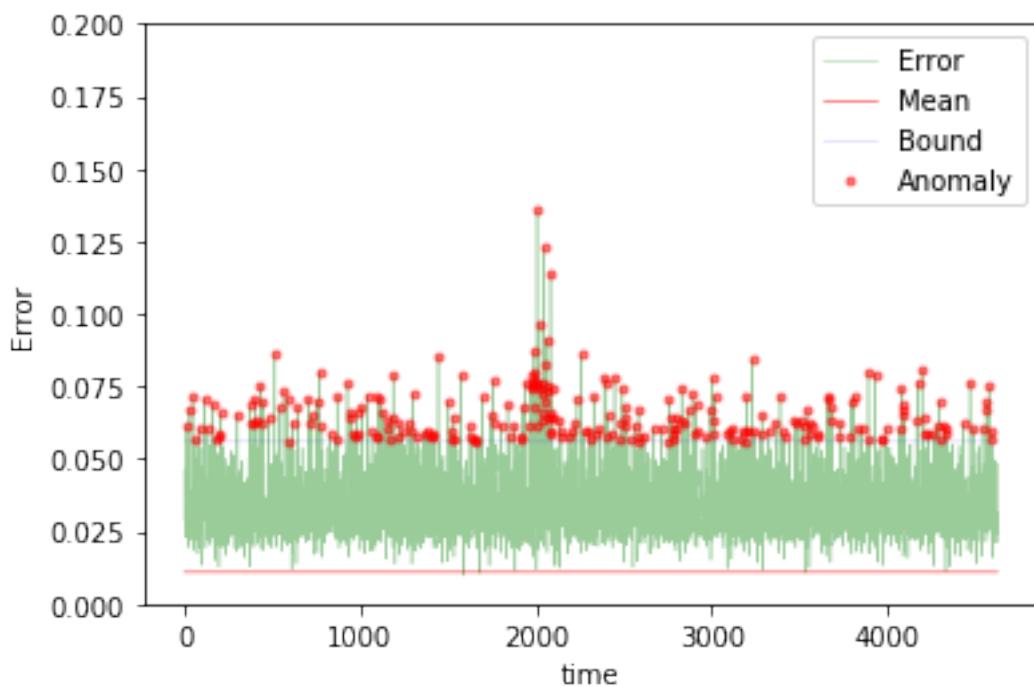




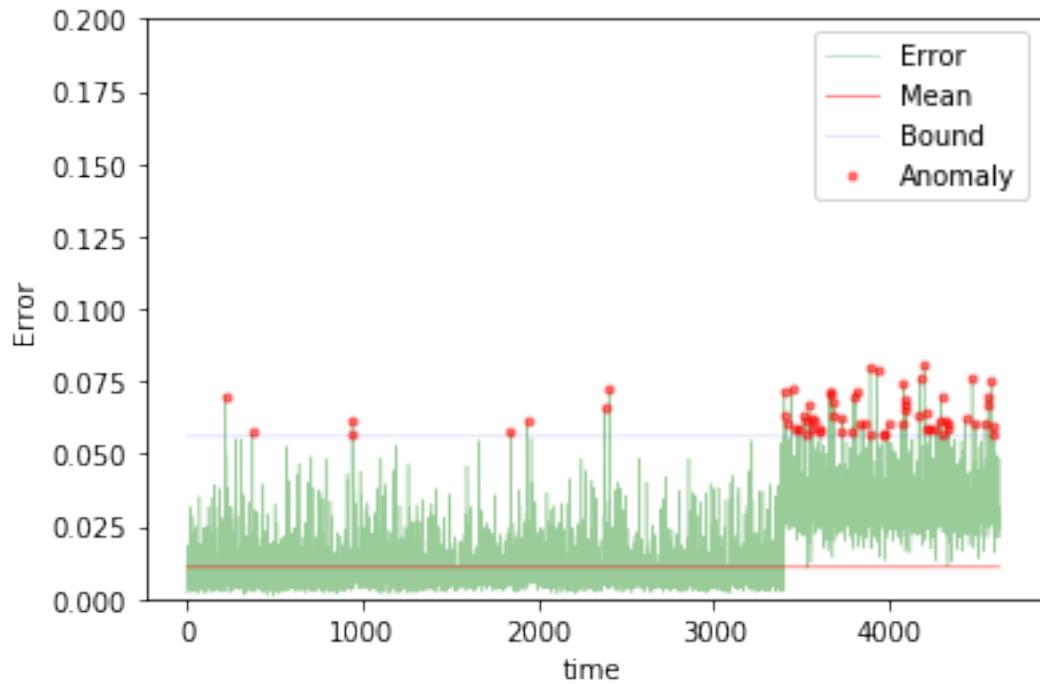
The mean error for lin100_normal_ is 0.010974929266375485 for length 4629
Testing on anomaly data.



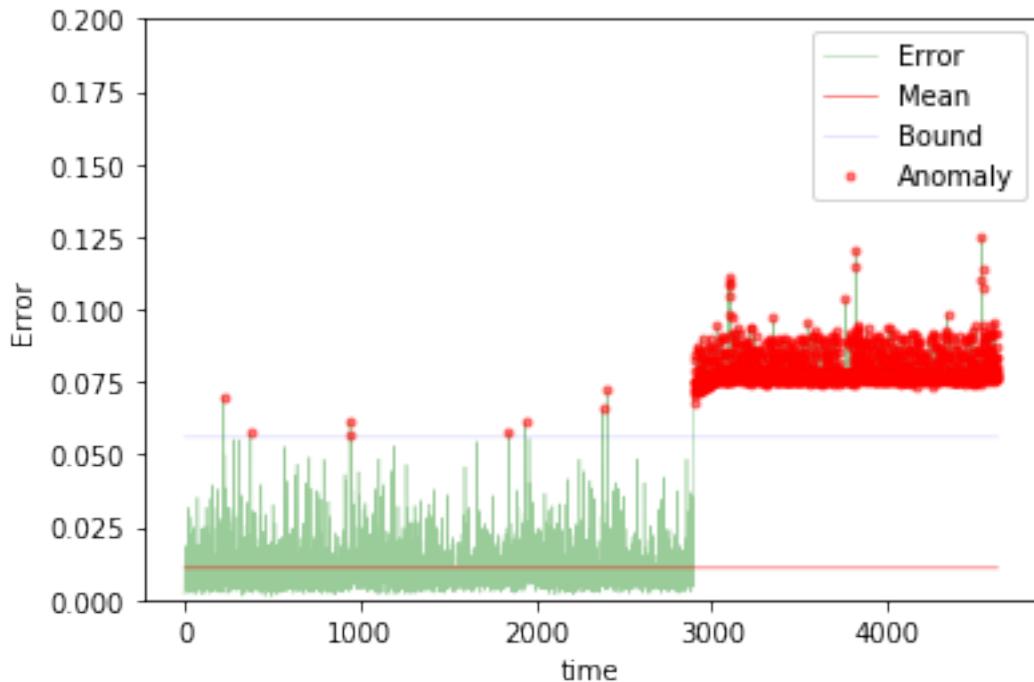
The mean error for lin100_anomaly_ is 0.012634772486407587 for length 4629
Testing on different app data.



The mean error for lin100_diff_app_ is 0.03402547325185798 for length 4629
Testing on App change synthetic data.



The mean error for lin100_app_change_ is 0.017008660773470838 for length 4629
Testing on Net flood synthetic data.



```
The mean error for lin100_net_flood_ is 0.03649632513377071 for length 4629
=====
```

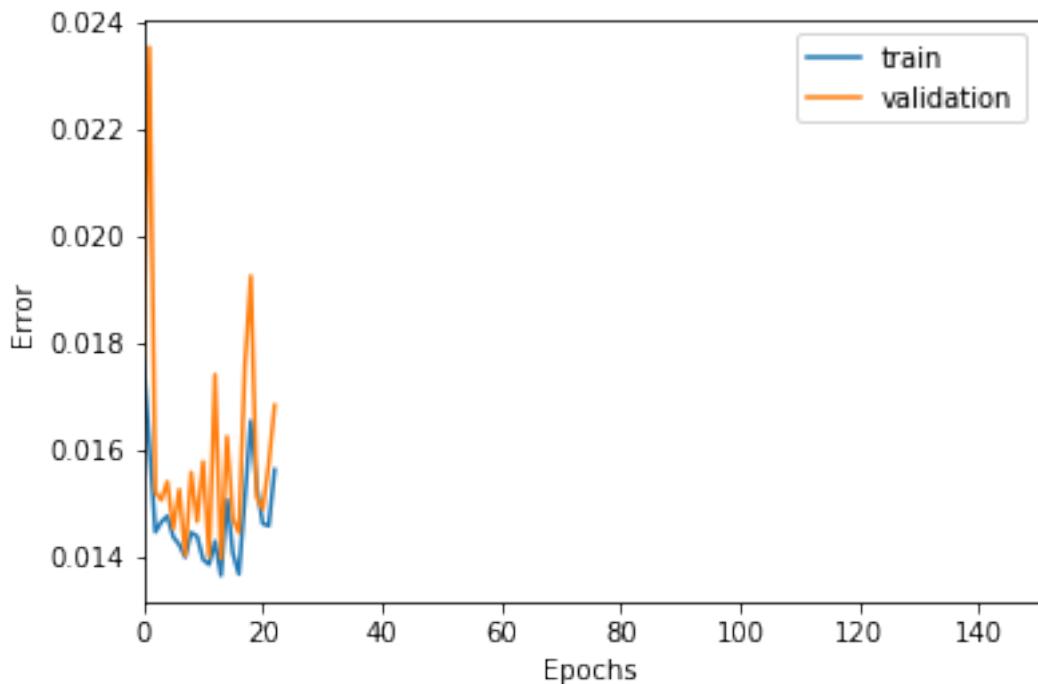
200 steps

```
In [60]: TIMESTEPS = 200
        DIM = 29
        tgen = flat_generator(X, TIMESTEPS)
        vgen = flat_generator(val_X, TIMESTEPS)
        name = "lin200"

In [61]: input_layer = Input(shape=(TIMESTEPS*DIM,))
        output = Dense(DIM, activation='sigmoid')(input_layer)

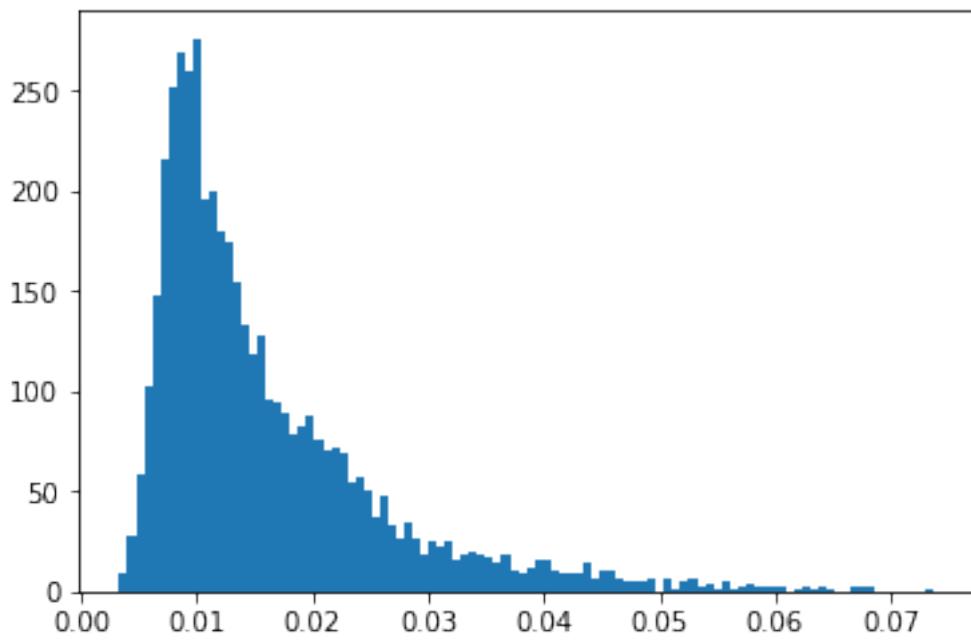
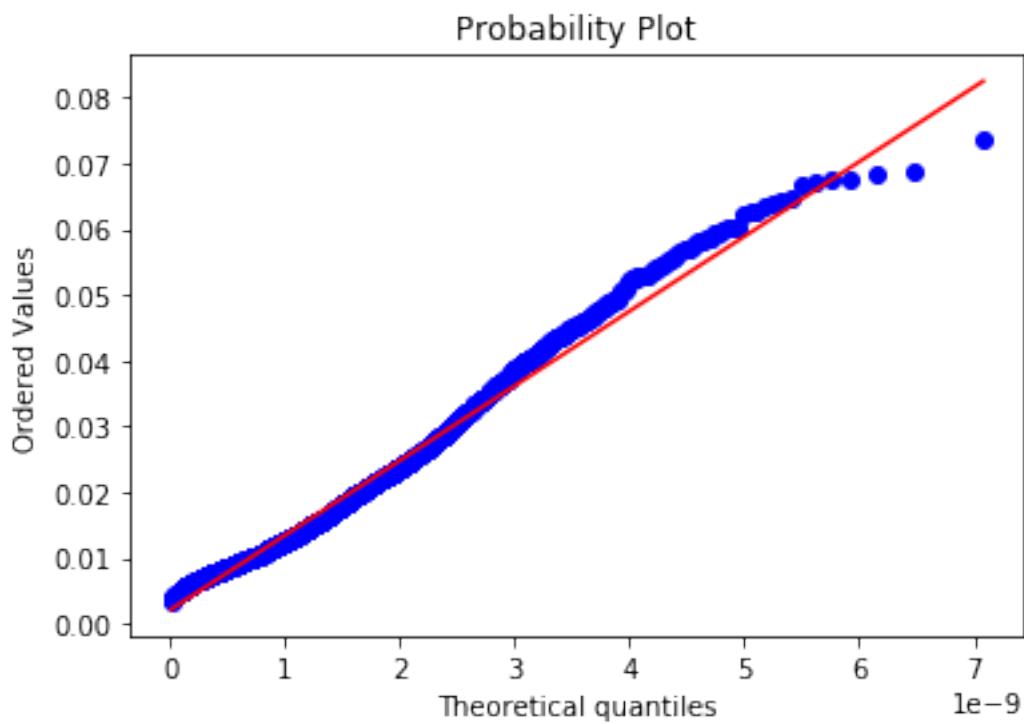
In [62]: model = Model(input_layer, output)
        model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

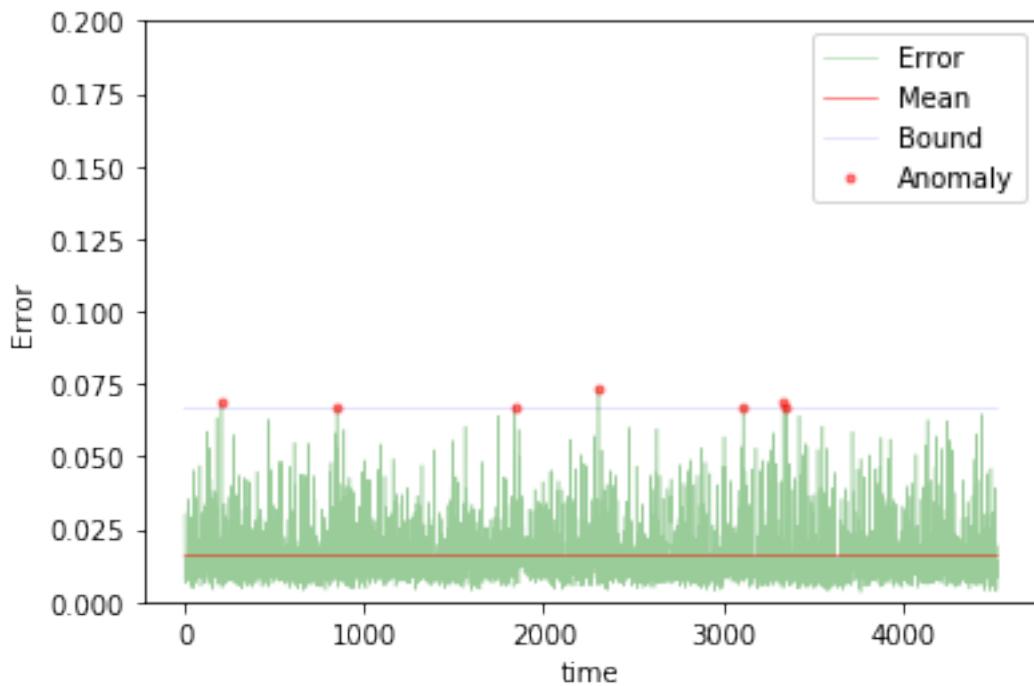
In [63]: train(model, tgen, vgen, name=name)
        test(model, name=name, window=TIMESTEPS)
```



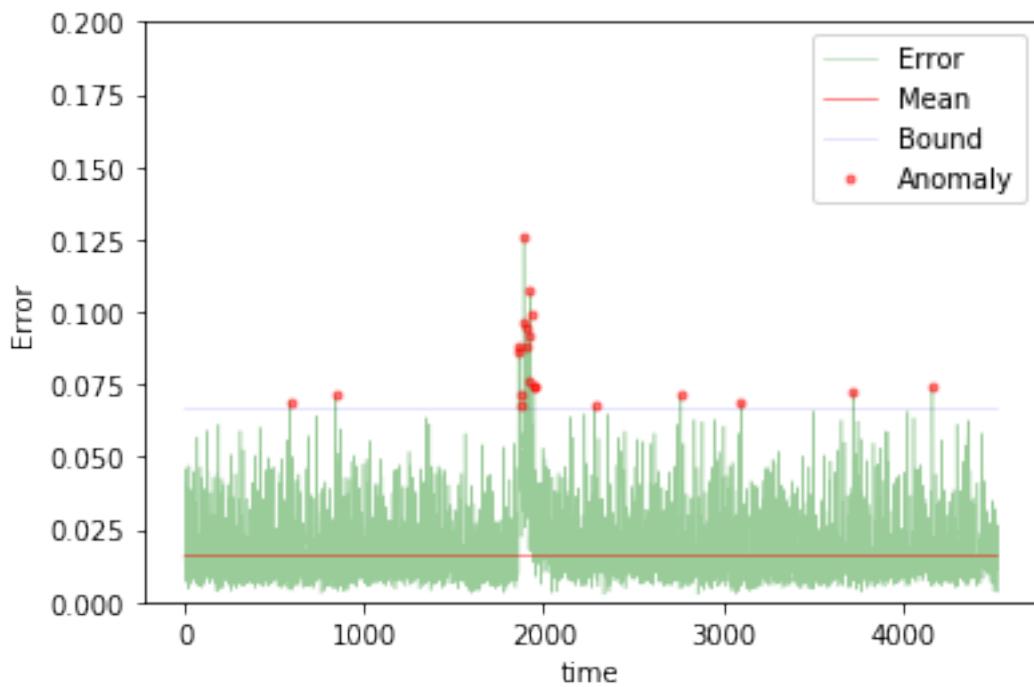
```
Training loss for final epoch is 0.015617035196628421
Validation loss for final epoch is 0.01682050040224567
----- Beginning tests for lin200 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
    warnings.warn(msg, RuntimeWarning)
```

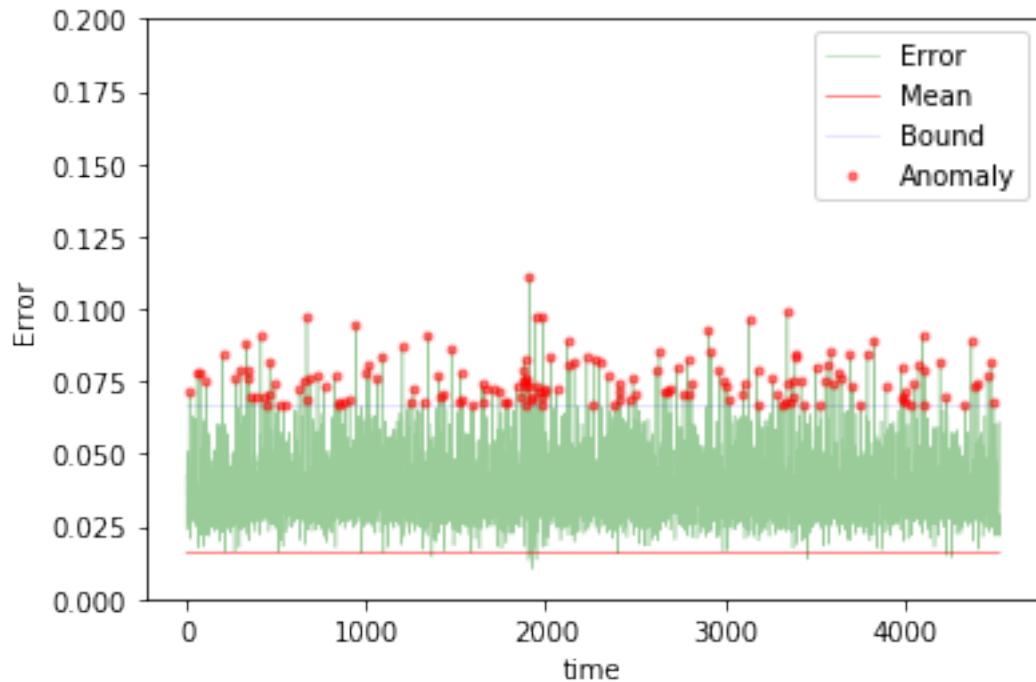




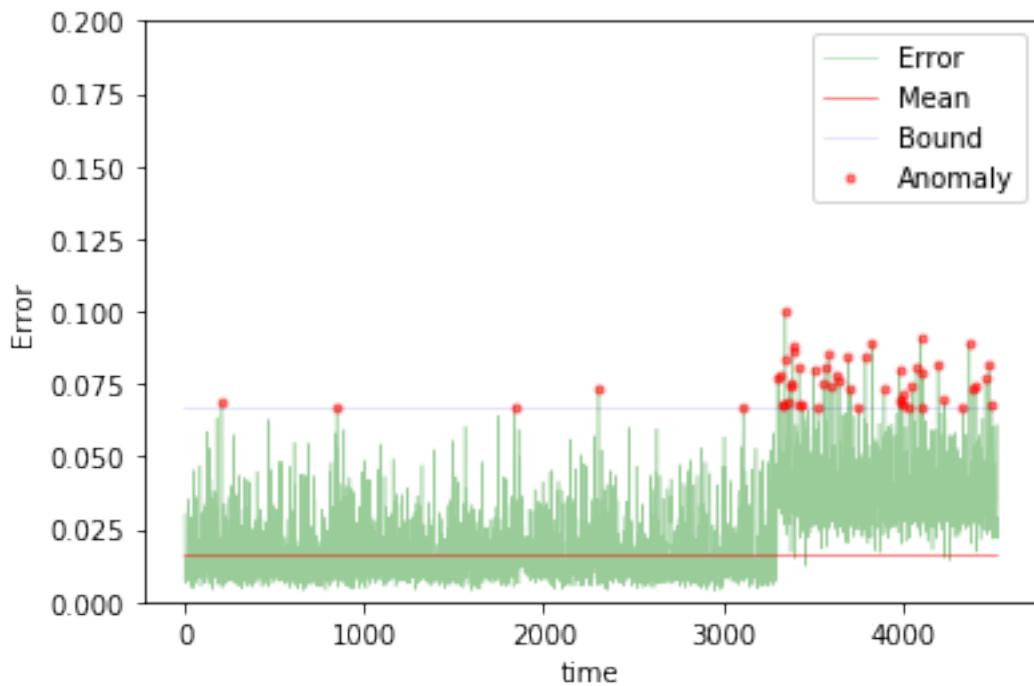
The mean error for lin200_normal_ is 0.01599304859277505 for length 4529
Testing on anomaly data.



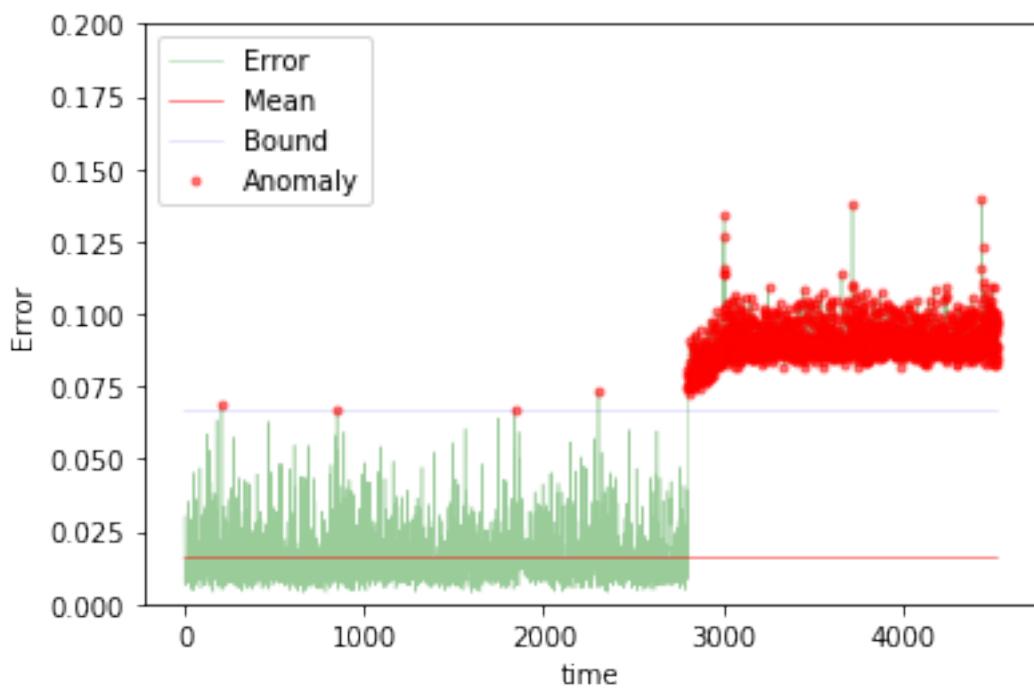
The mean error for lin200_anomaly_ is 0.0181470885692155 for length 4529
Testing on different app data.



The mean error for lin200_diff_app_ is 0.037685847455793786 for length 4529
Testing on App change synthetic data.



The mean error for lin200_app_change_ is 0.02205382806193309 for length 4529
Testing on Net flood synthetic data.

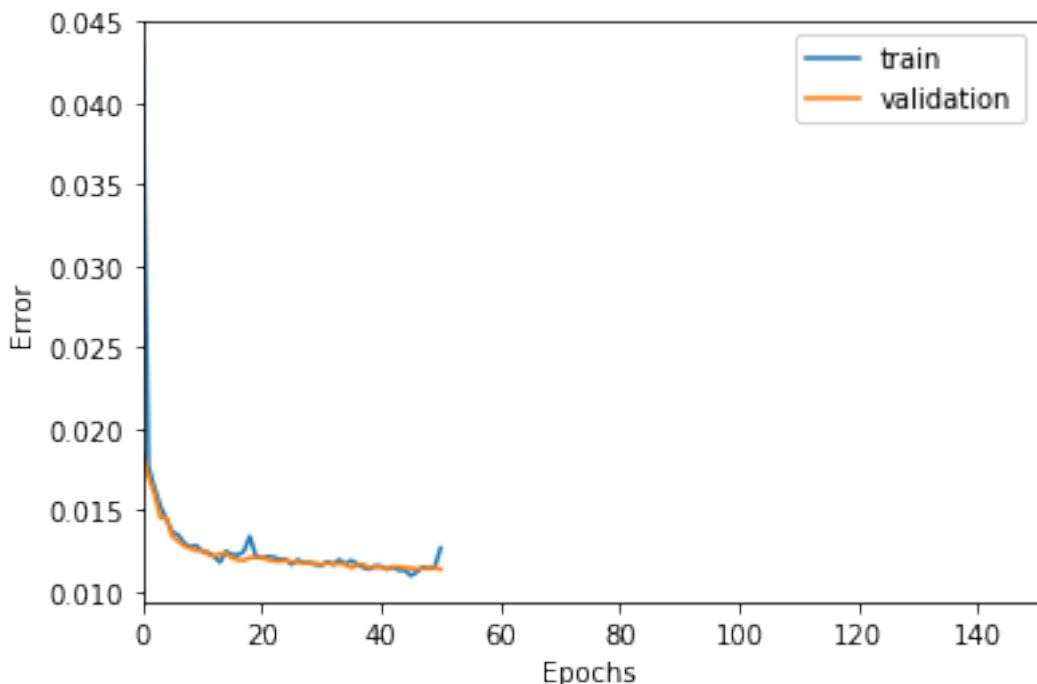


```
The mean error for lin200_net_flood_ is 0.04436544598448197 for length 4529  
=====
```

2.1.2 NN with 1 hidden layer

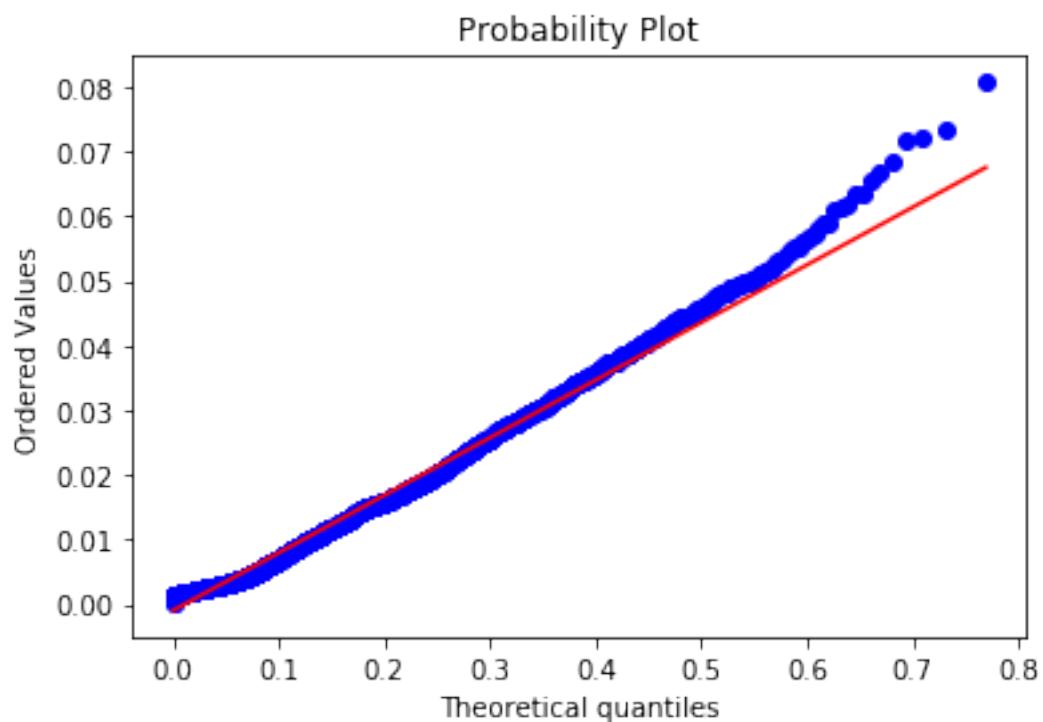
2 steps

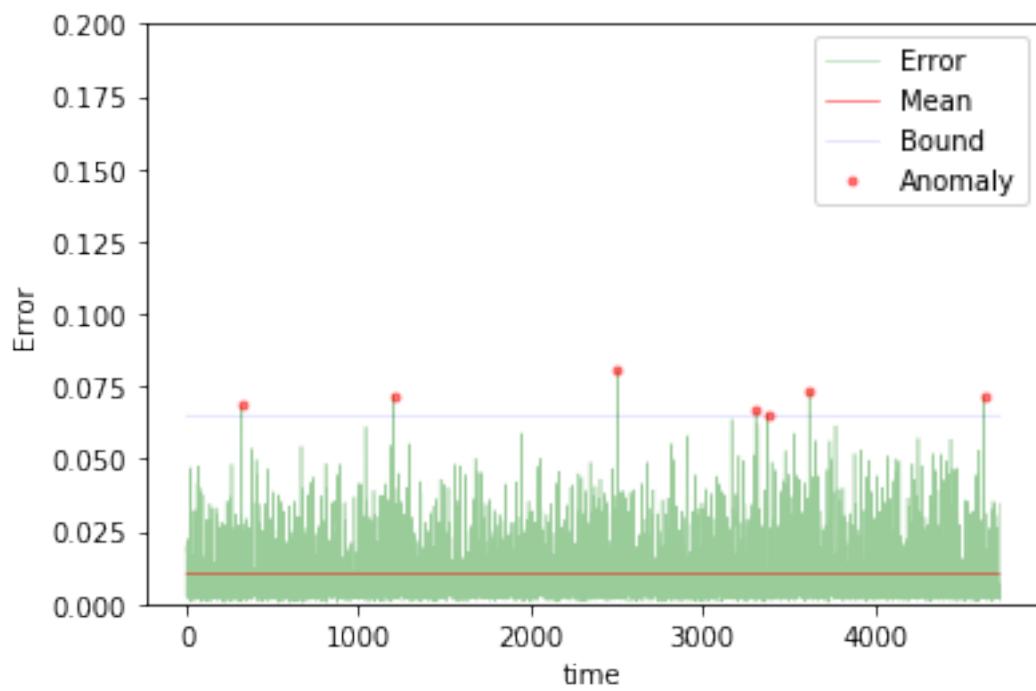
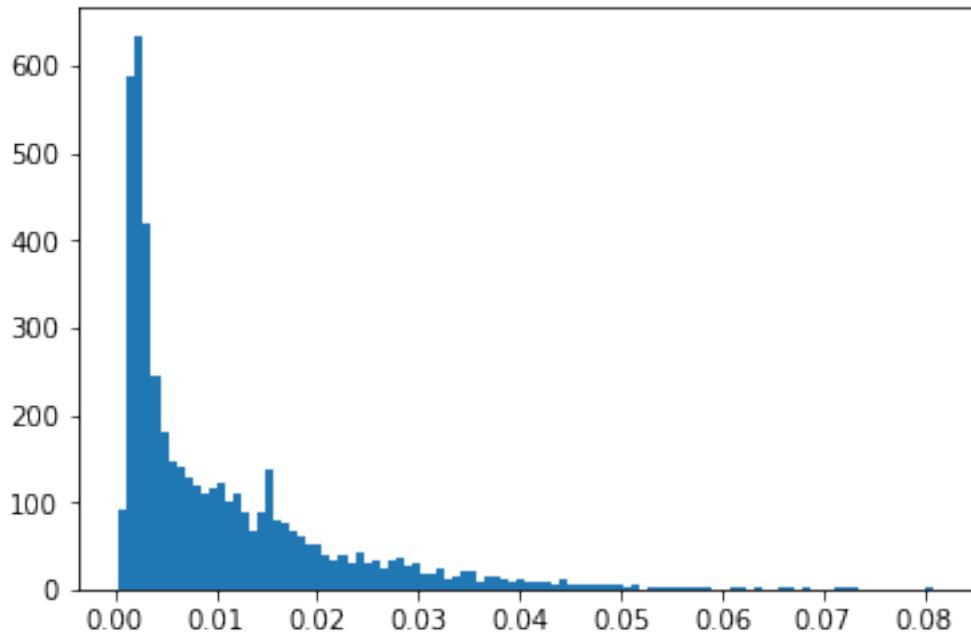
```
In [64]: TIMESTEPS = 2  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS)  
vgen = flat_generator(val_X, TIMESTEPS)  
name = "nn1_2"  
  
In [65]: input_layer = Input(shape=(TIMESTEPS*DIM,))  
hidden = Dense(100, activation='relu')(input_layer)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [66]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [67]: train(model, tgen, vgen, name=name)  
test(model, name=name, window=TIMESTEPS)
```



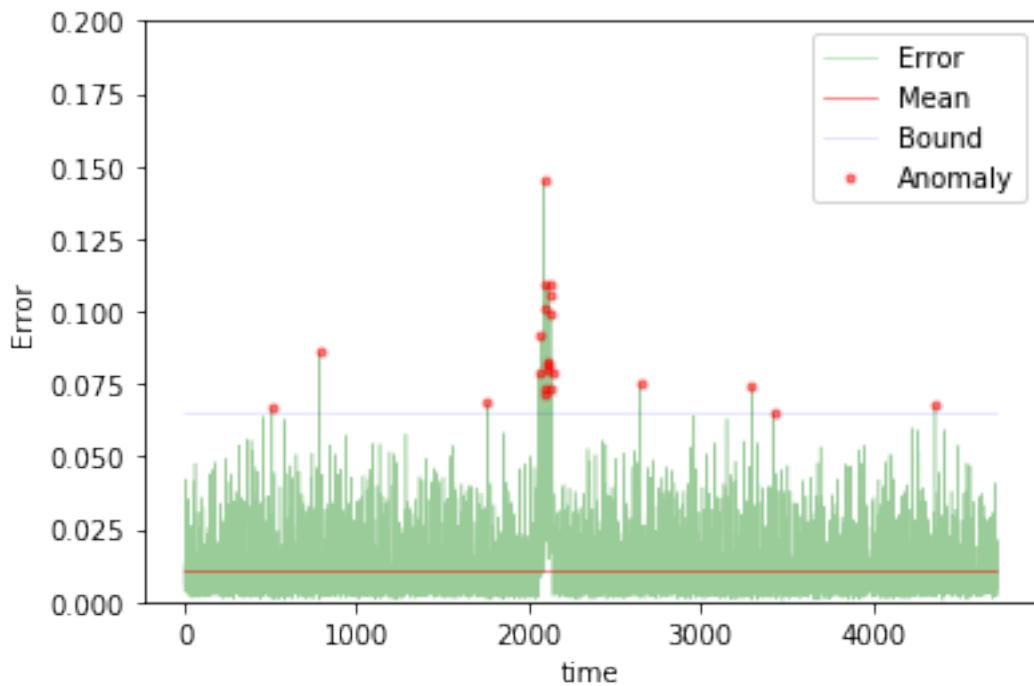
```
Training loss for final epoch is 0.012657105904538184
Validation loss for final epoch is 0.011347391777322628
----- Beginning tests for nn1_2 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
```

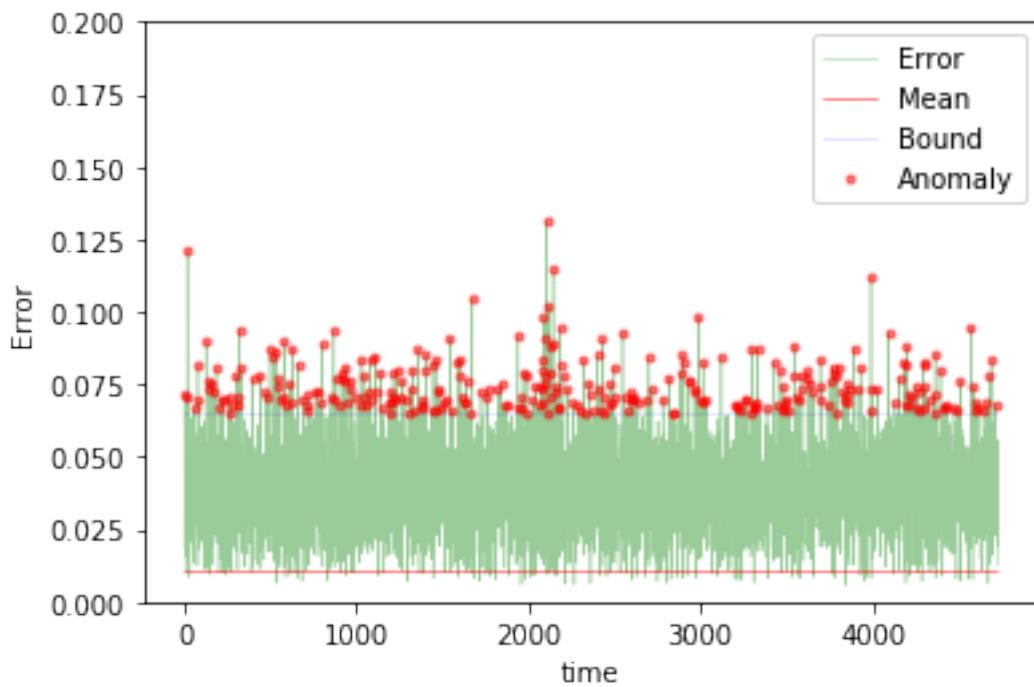




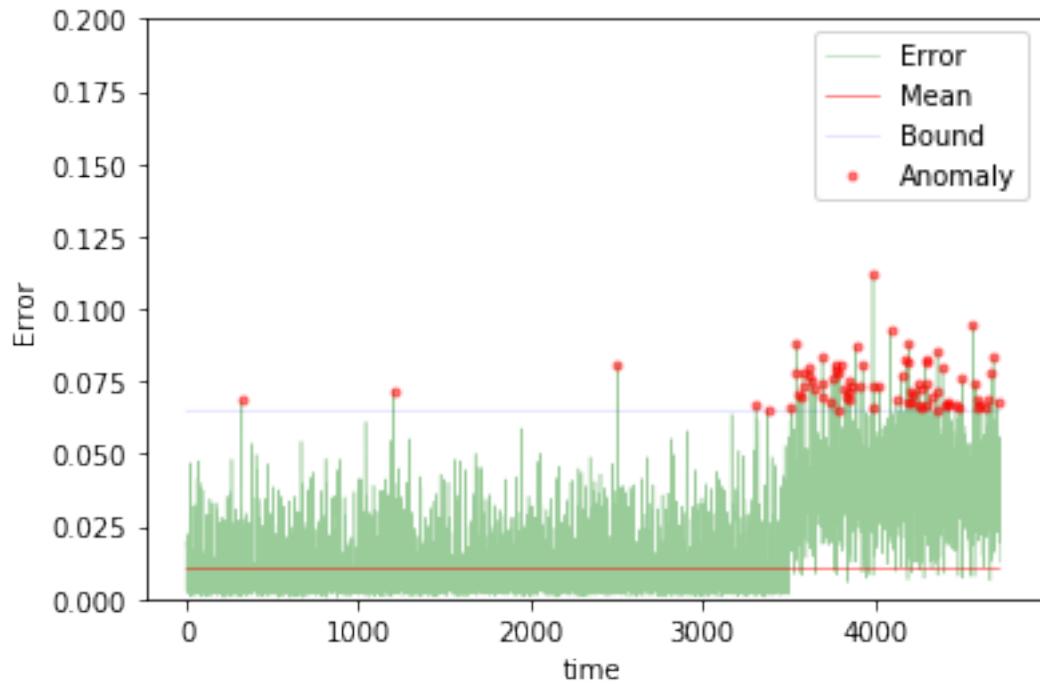
The mean error for nn1_2_normal_ is 0.010610702901873688 for length 4727
Testing on anomaly data.



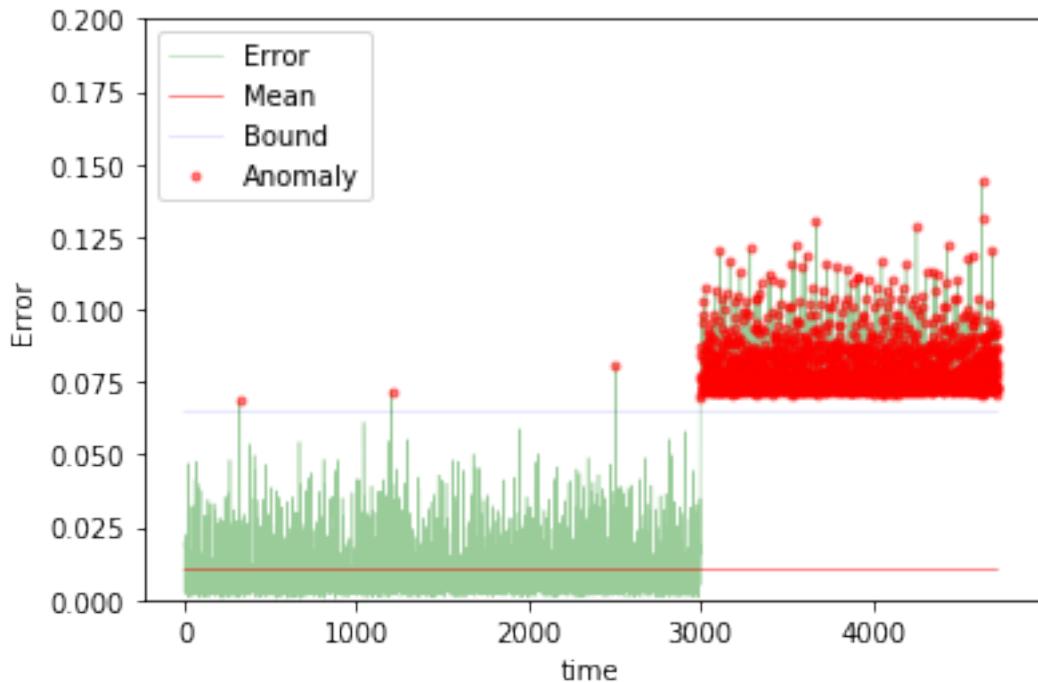
The mean error for nn1_2_anomaly_ is 0.01219715025456409 for length 4727
Testing on different app data.



The mean error for nn1_2_diff_app_ is 0.03870061737845644 for length 4727
Testing on App change synthetic data.



The mean error for nn1_2_app_change_ is 0.017819362561009685 for length 4727
Testing on Net flood synthetic data.



```
The mean error for nn1_2_net_flood_ is 0.03601428622422198 for length 4727
=====
```

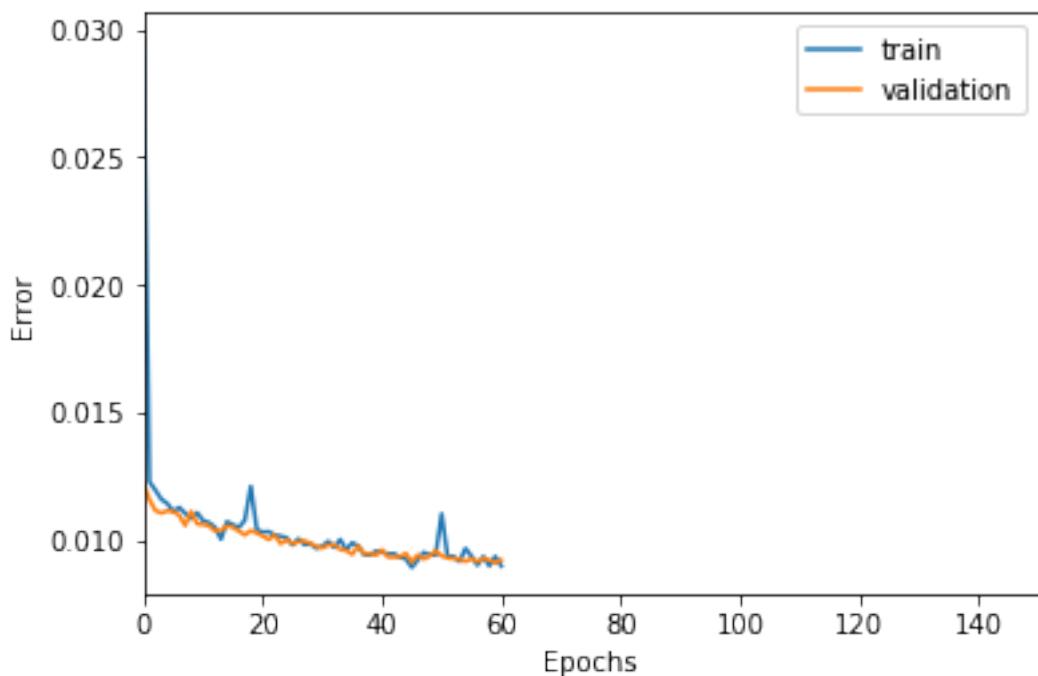
5 steps

```
In [68]: TIMESTEPS = 5
        DIM = 29
        tgen = flat_generator(X, TIMESTEPS)
        vgen = flat_generator(val_X, TIMESTEPS)
        name = "nn1_5"

In [69]: input_layer = Input(shape=(TIMESTEPS*DIM,))
        hidden = Dense(100, activation='relu')(input_layer)
        output = Dense(DIM, activation='sigmoid')(hidden)

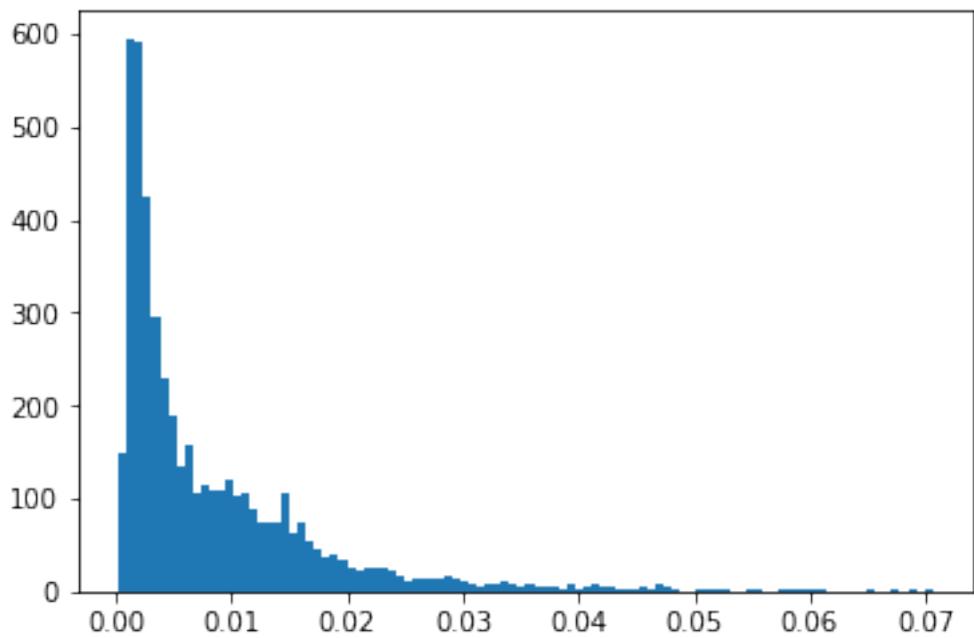
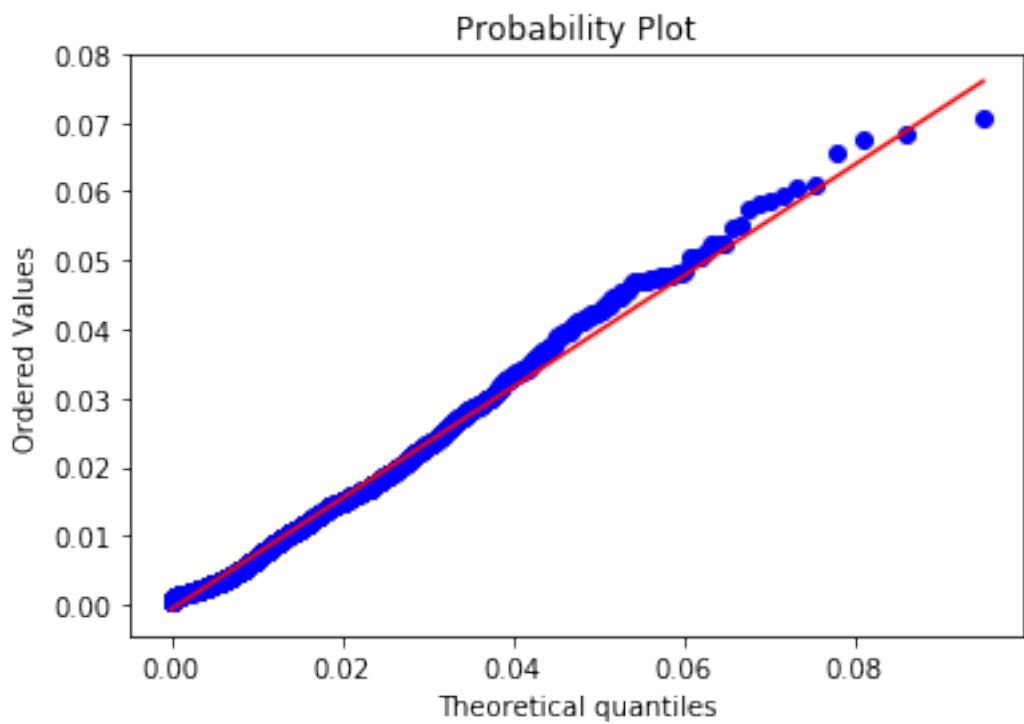
In [70]: model = Model(input_layer, output)
        model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

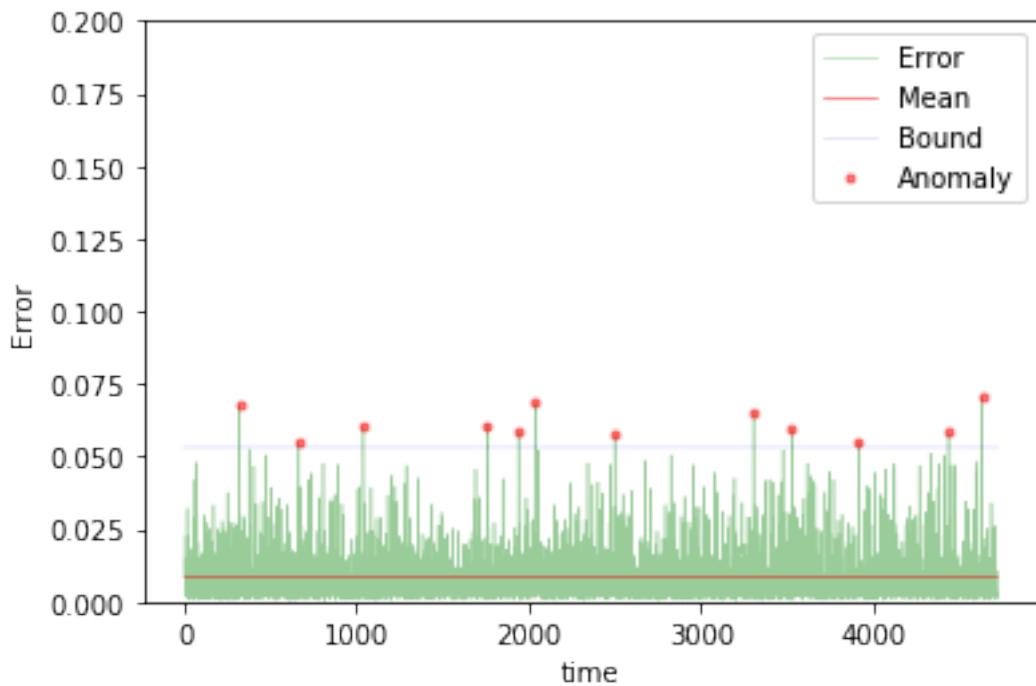
In [71]: train(model, tgen, vgen, name=name)
        test(model, name=name, window=TIMESTEPS)
```



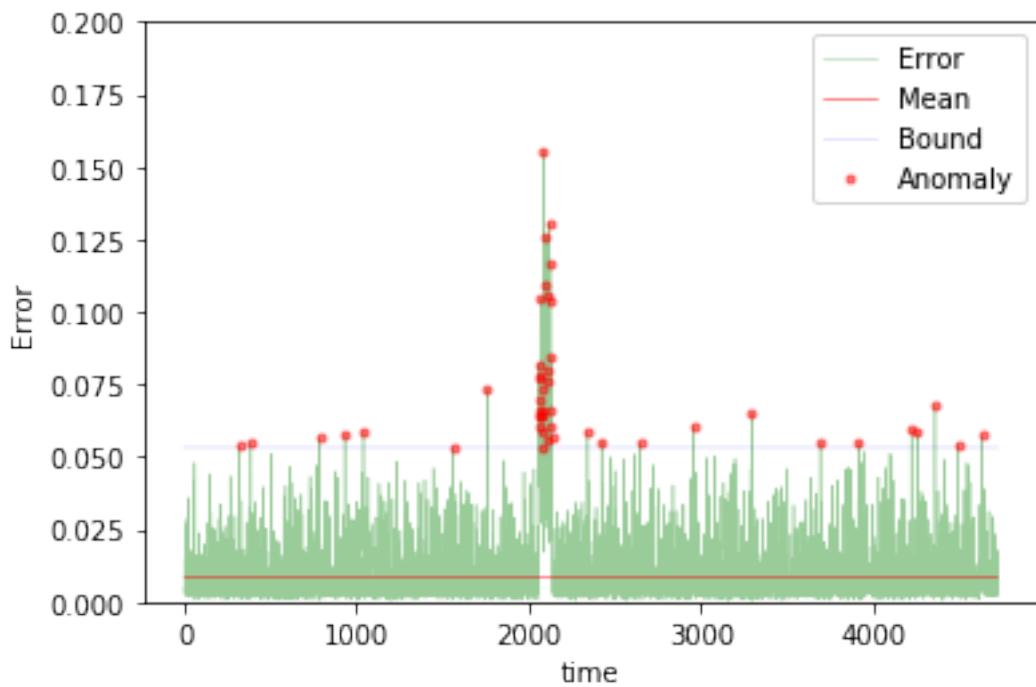
```
Training loss for final epoch is 0.00898358252464095
Validation loss for final epoch is 0.009218202371965162
----- Beginning tests for nn1_5 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
    warnings.warn(msg, RuntimeWarning)
```

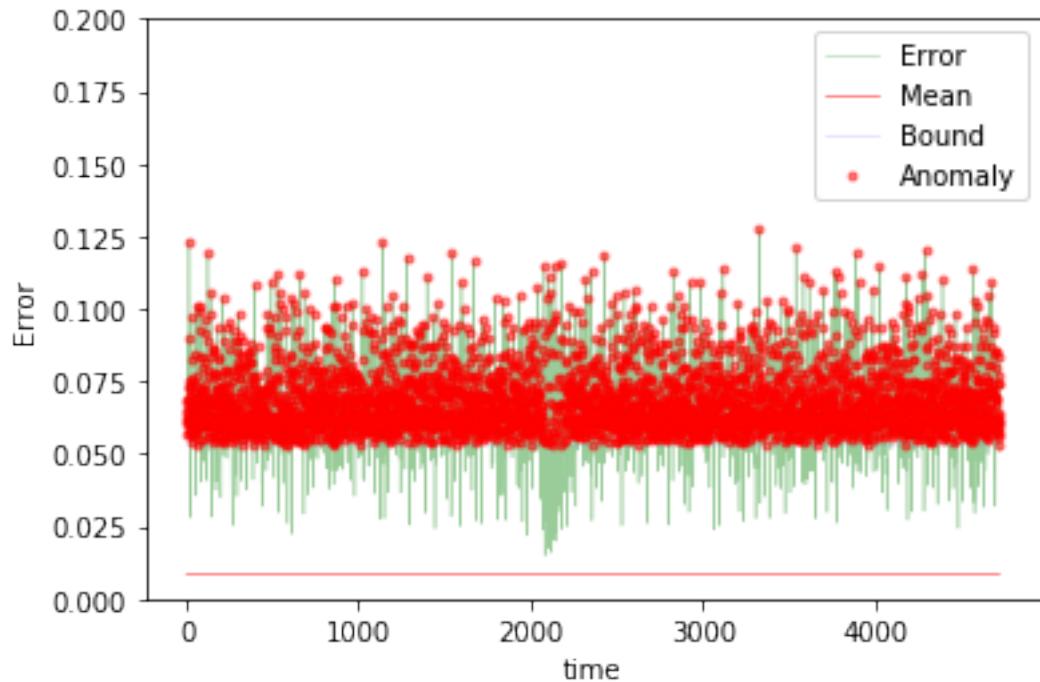




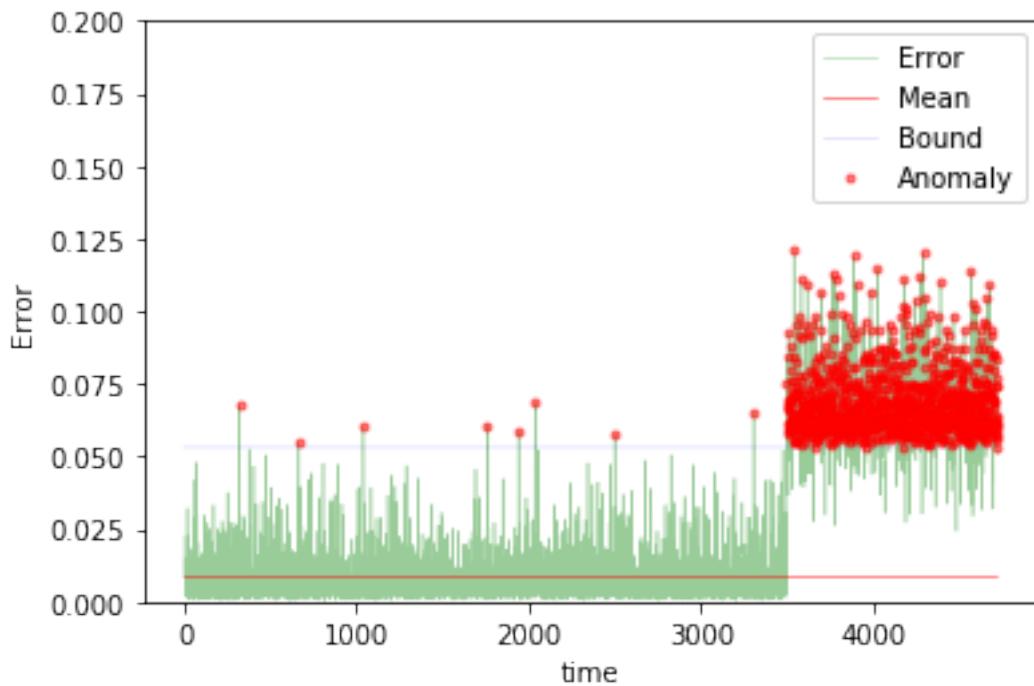
The mean error for nn1_5_normal_ is 0.008384038977260243 for length 4724
Testing on anomaly data.



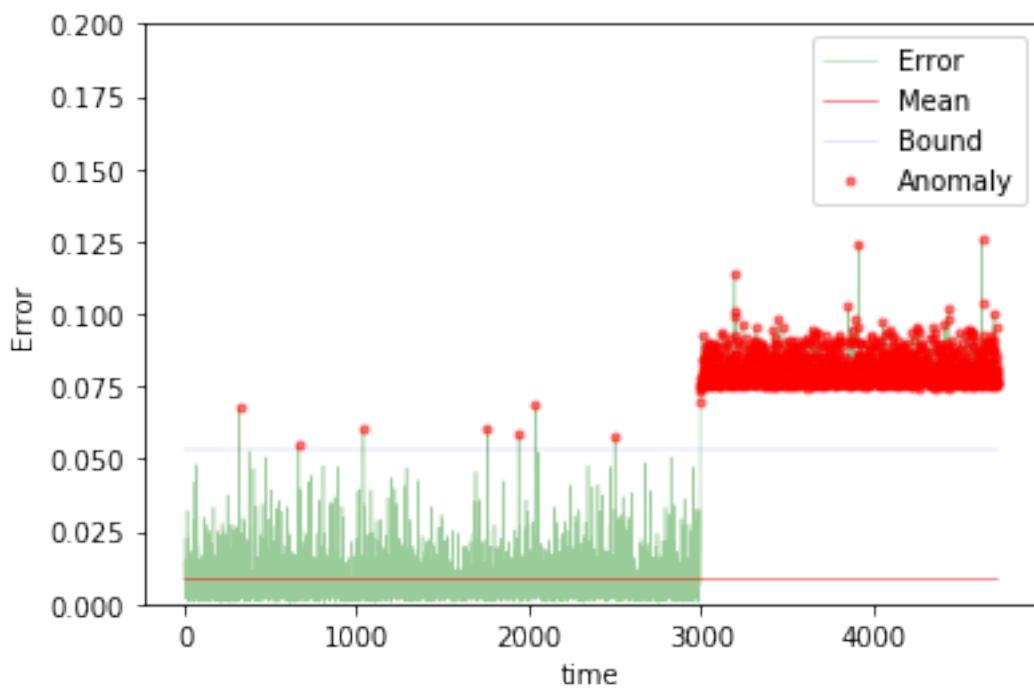
The mean error for nn1_5_anomaly_ is 0.010115106072708246 for length 4724
Testing on different app data.



The mean error for nn1_5_diff_app_ is 0.06552537284211804 for length 4724
Testing on App change synthetic data.



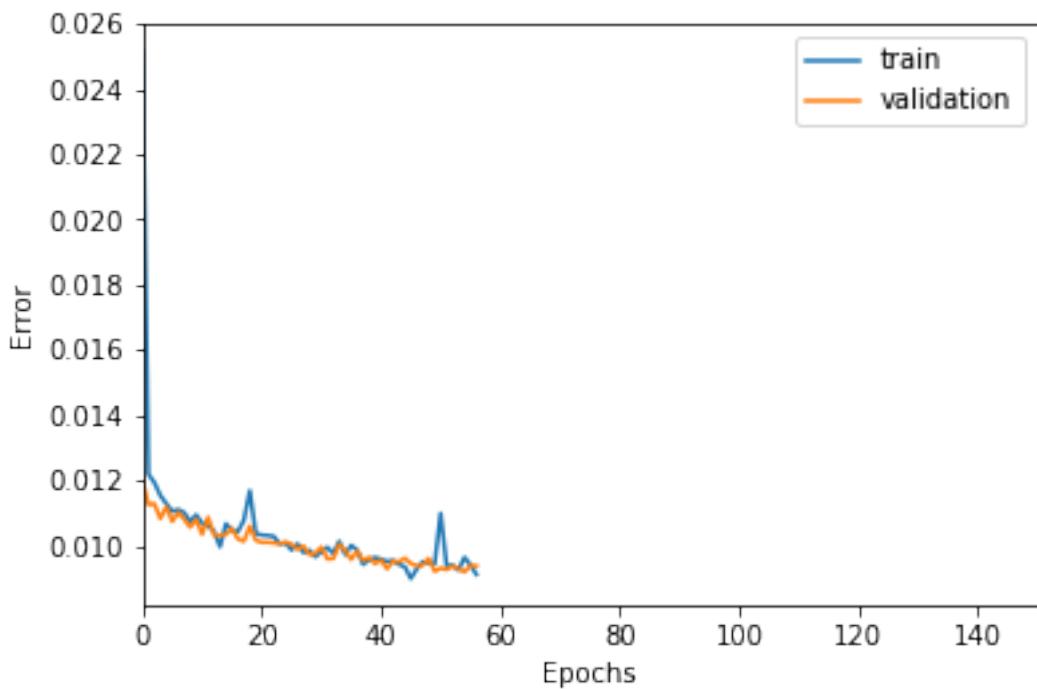
The mean error for nn1_5_app_change_ is 0.023304989109165168 for length 4724
Testing on Net flood synthetic data.



```
The mean error for nn1_5_net_flood_ is 0.03468598574181329 for length 4724  
=====
```

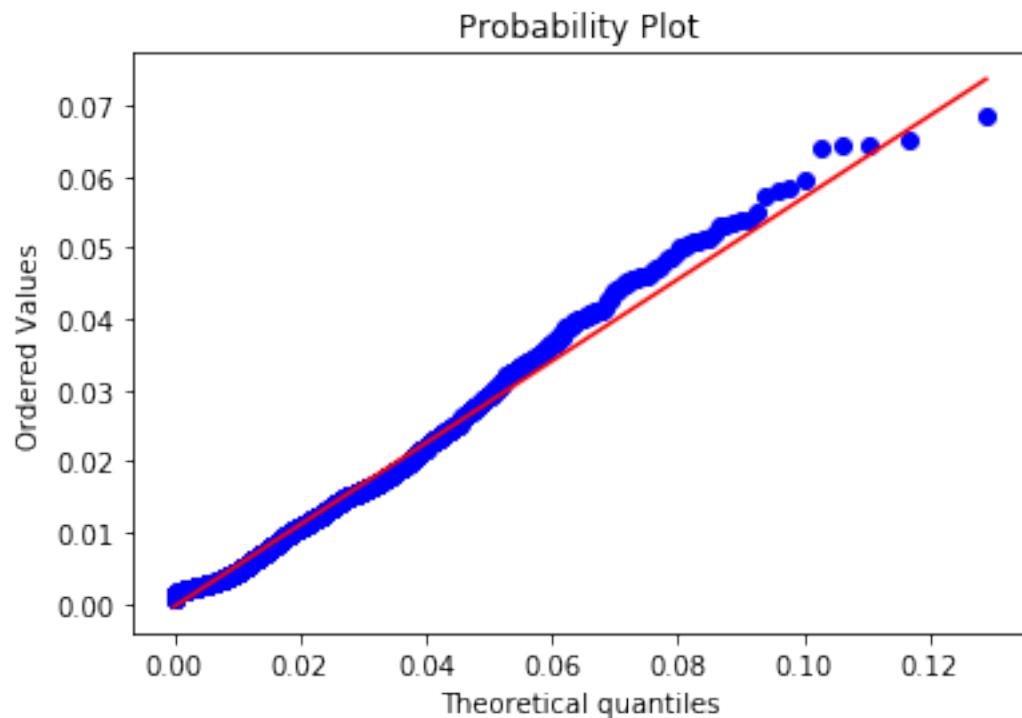
10 steps

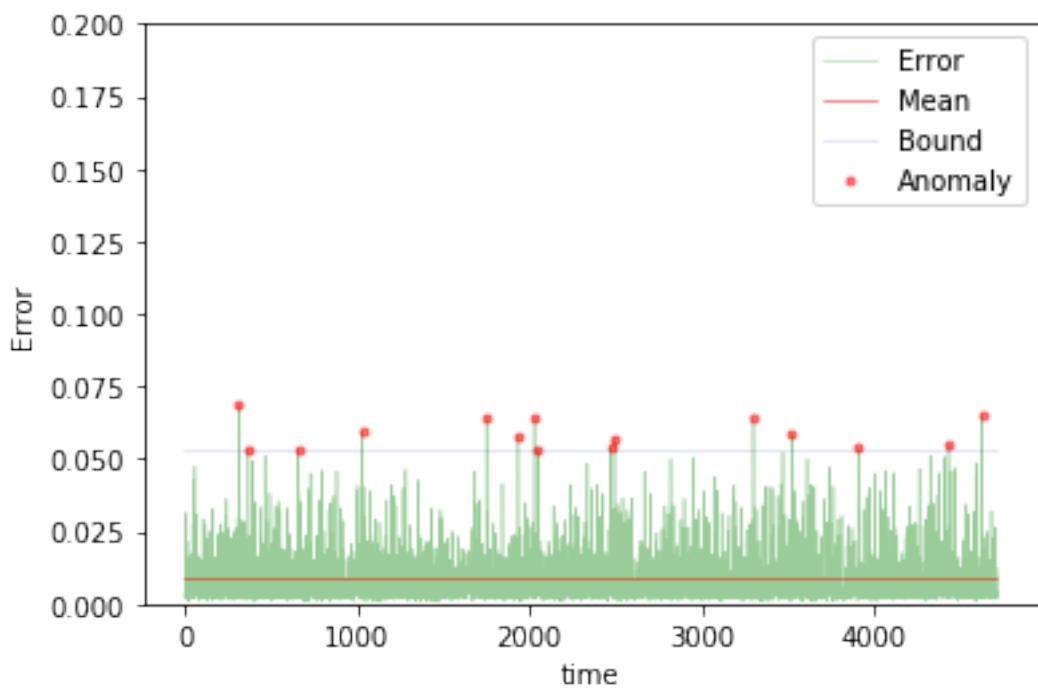
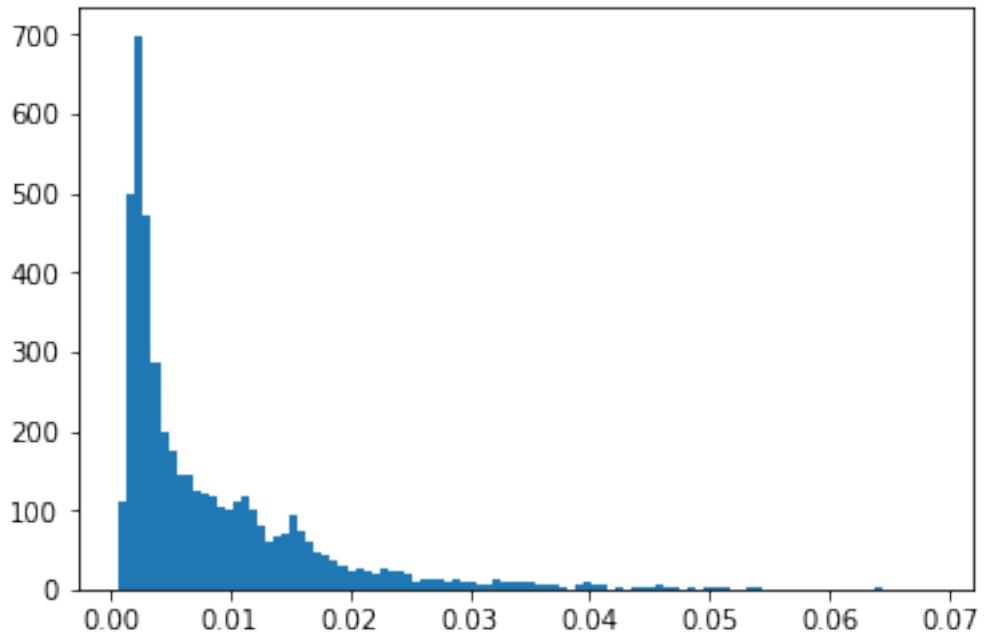
```
In [72]: TIMESTEPS = 10  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS)  
vgen = flat_generator(val_X, TIMESTEPS)  
name = "nn1_10"  
  
In [73]: input_layer = Input(shape=(TIMESTEPS*DIM,))  
hidden = Dense(100, activation='relu')(input_layer)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [74]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [75]: train(model, tgen, vgen, name=name)  
test(model, name=name, window=TIMESTEPS)
```



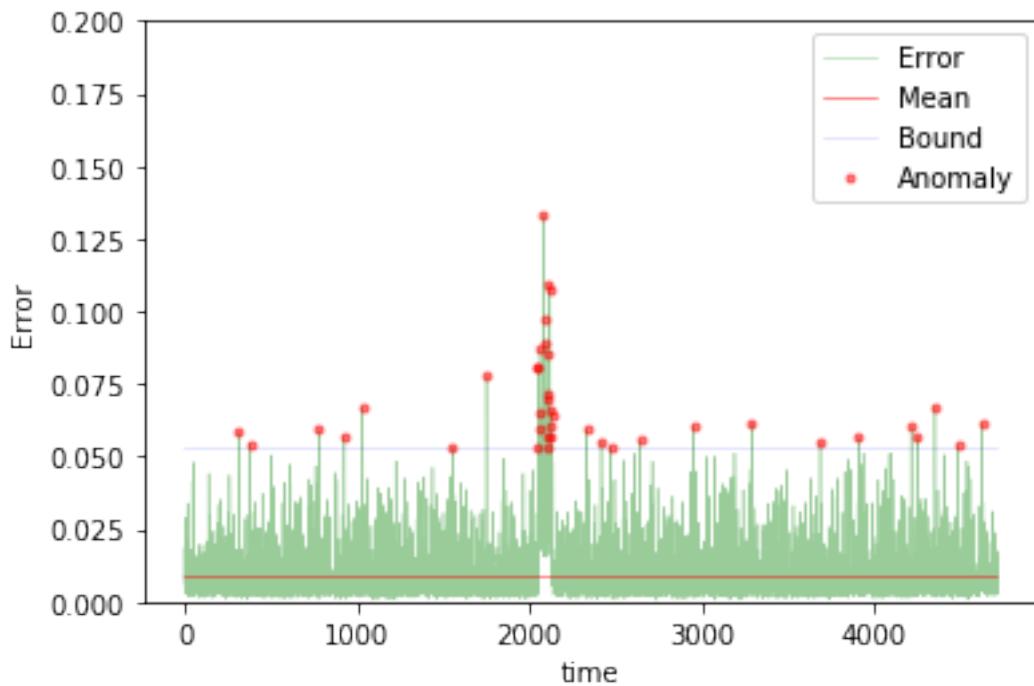
```
Training loss for final epoch is 0.009128186049405485
Validation loss for final epoch is 0.009387759334524163
----- Beginning tests for nn1_10 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

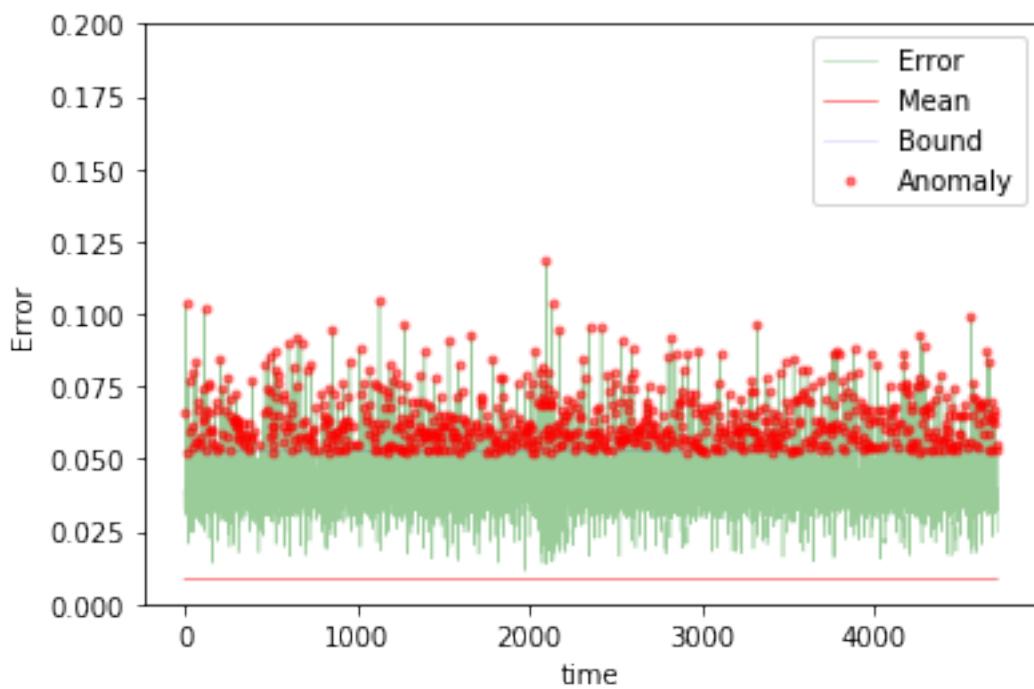




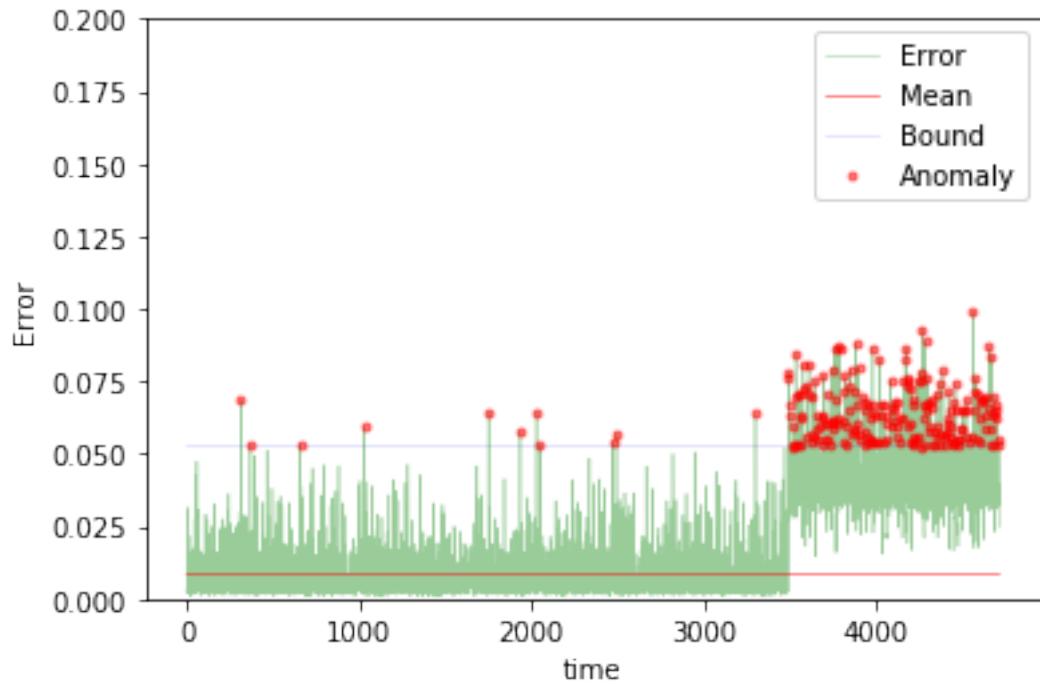
The mean error for nn1_10_normal_ is 0.008601511358921607 for length 4719
Testing on anomaly data.



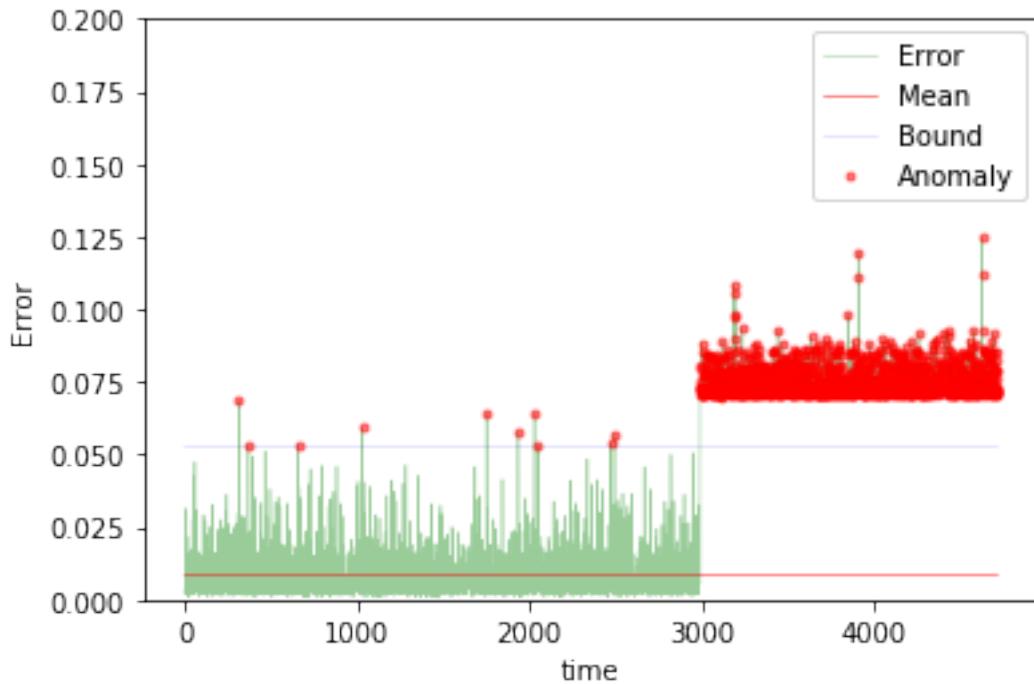
The mean error for nn1_10_anomaly_ is 0.010277102970022566 for length 4719
Testing on different app data.



The mean error for nn1_10_diff_app_ is 0.042218556996753184 for length 4719
Testing on App change synthetic data.



The mean error for nn1_10_app_change_ is 0.017332887368182892 for length 4719
Testing on Net flood synthetic data.



```
The mean error for nn1_10_net_flood_ is 0.0329497881962391 for length 4719
=====
```

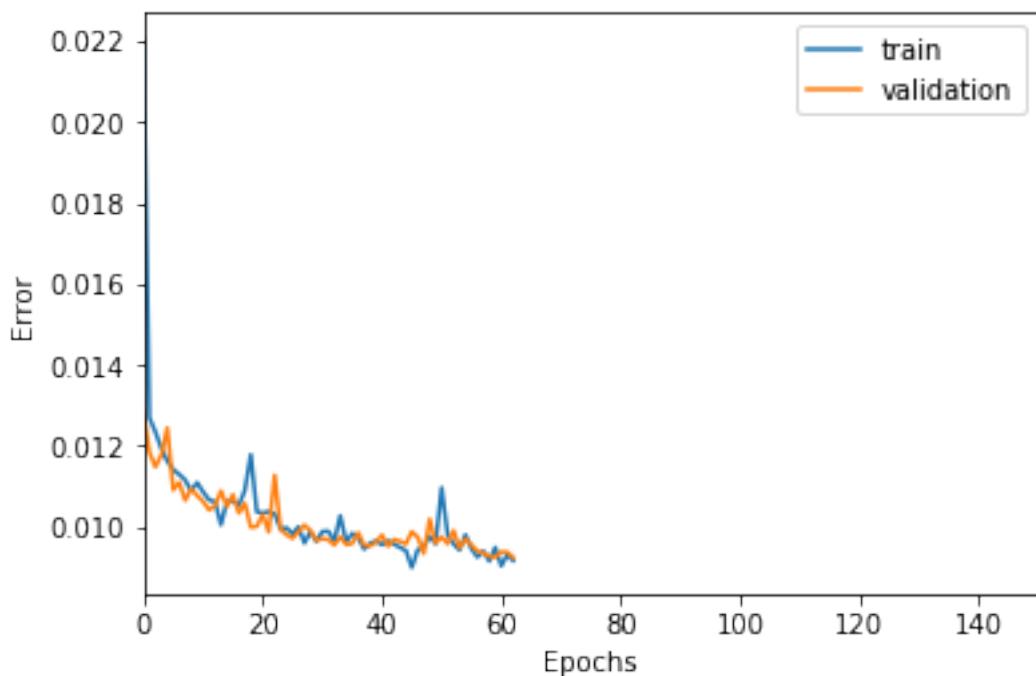
20 steps

```
In [76]: TIMESTEPS = 20
        DIM = 29
        tgen = flat_generator(X, TIMESTEPS)
        vgen = flat_generator(val_X, TIMESTEPS)
        name = "nn1_20"

In [77]: input_layer = Input(shape=(TIMESTEPS*DIM,))
        hidden = Dense(100,activation='relu')(input_layer)
        output = Dense(DIM, activation='sigmoid')(hidden)

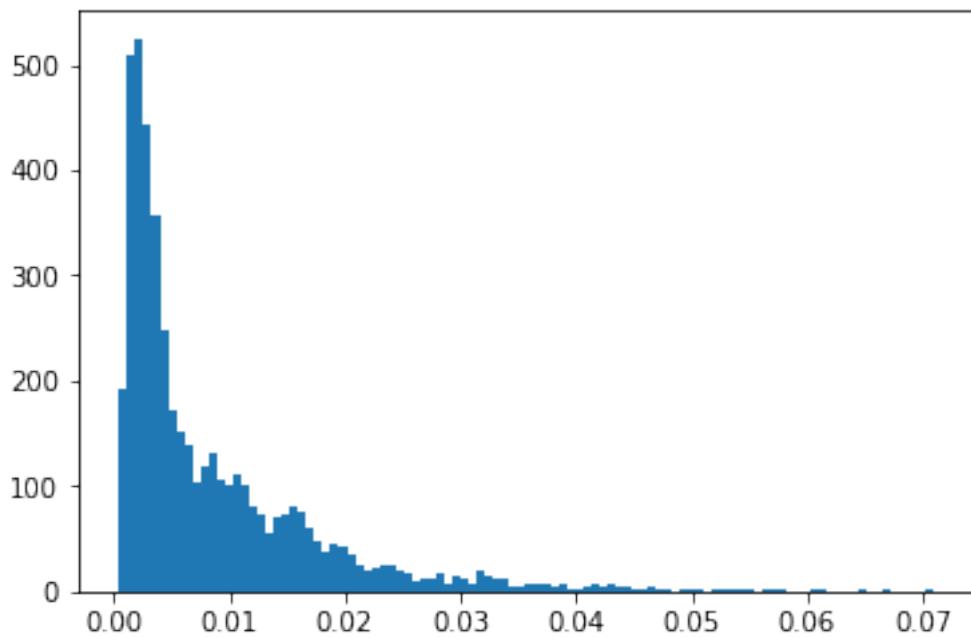
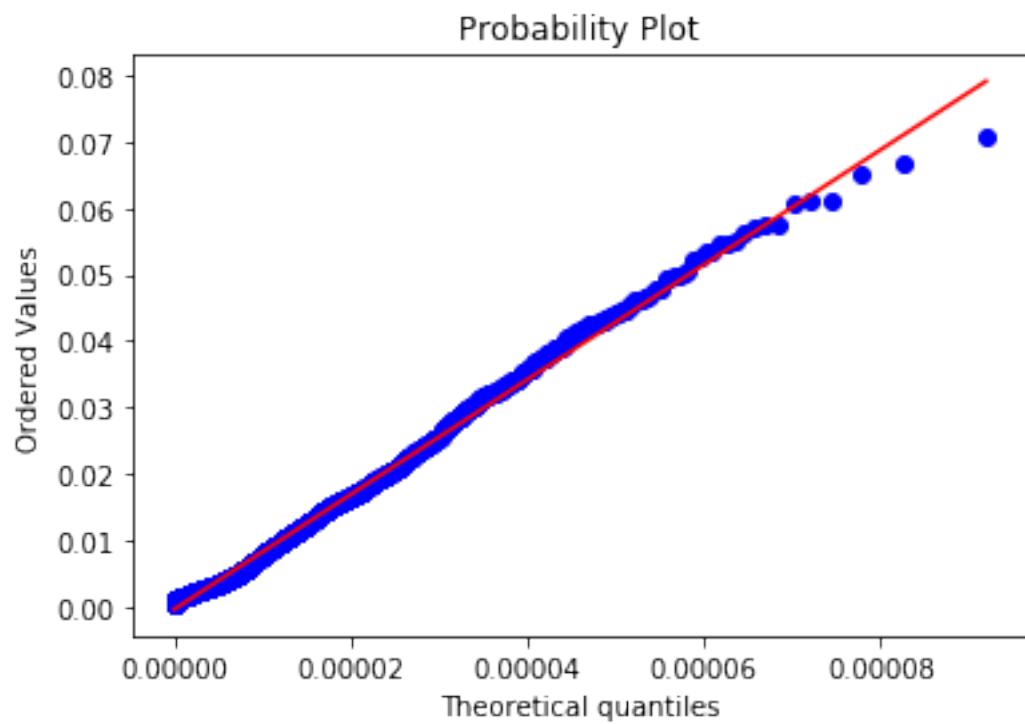
In [78]: model = Model(input_layer, output)
        model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

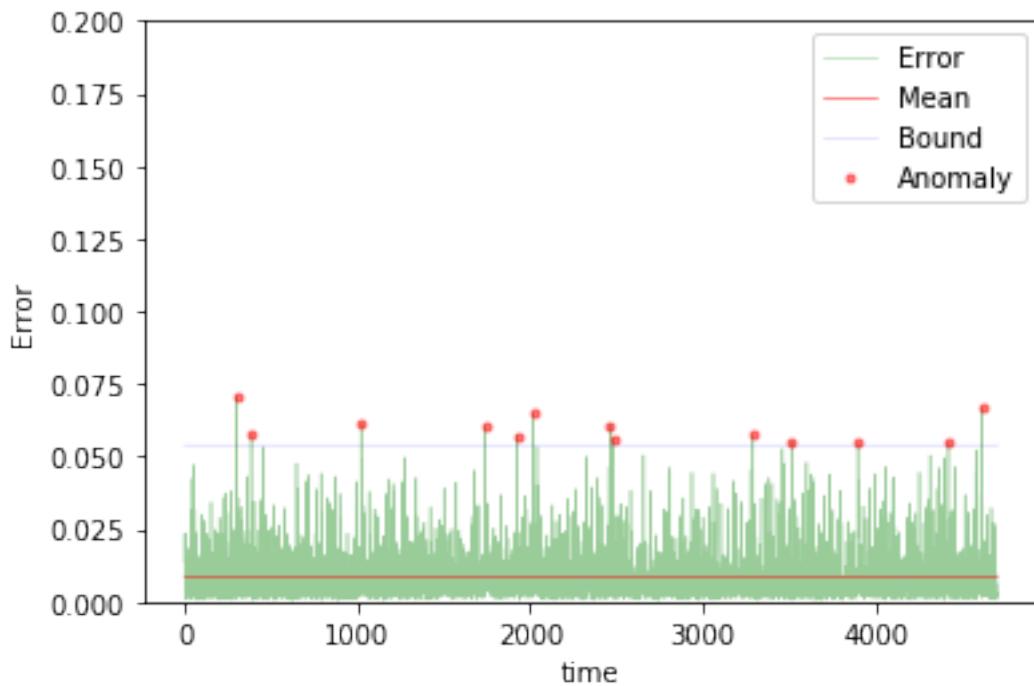
In [79]: train(model, tgen, vgen, name=name)
        test(model, name=name, window=TIMESTEPS)
```



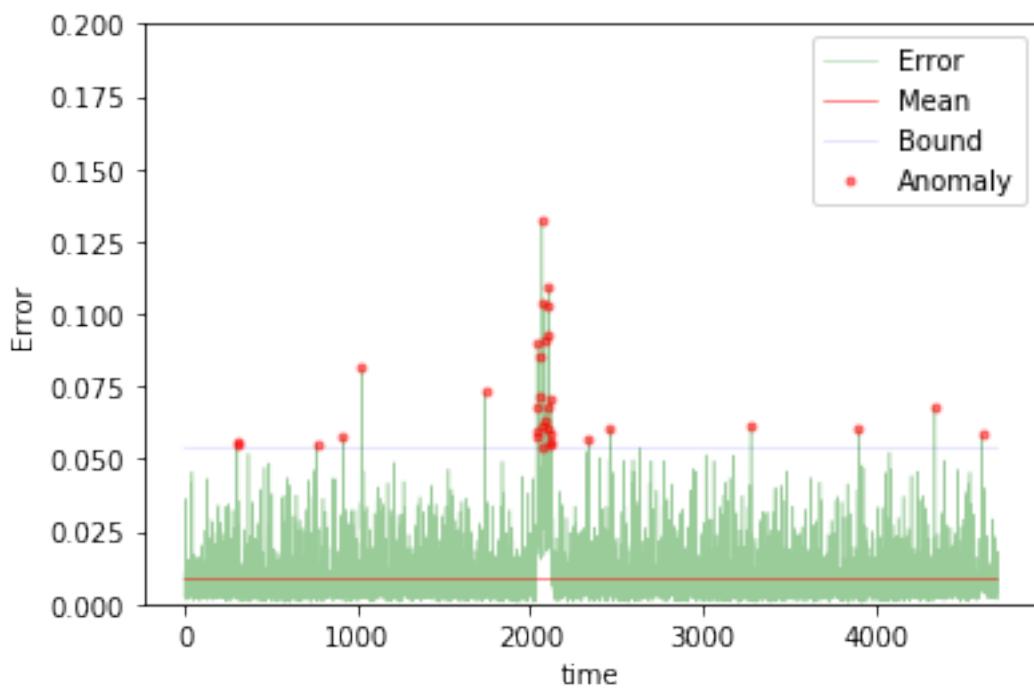
```
Training loss for final epoch is 0.00917332671827171
Validation loss for final epoch is 0.009249395199934953
----- Beginning tests for nn1_20 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
    warnings.warn(msg, RuntimeWarning)
```

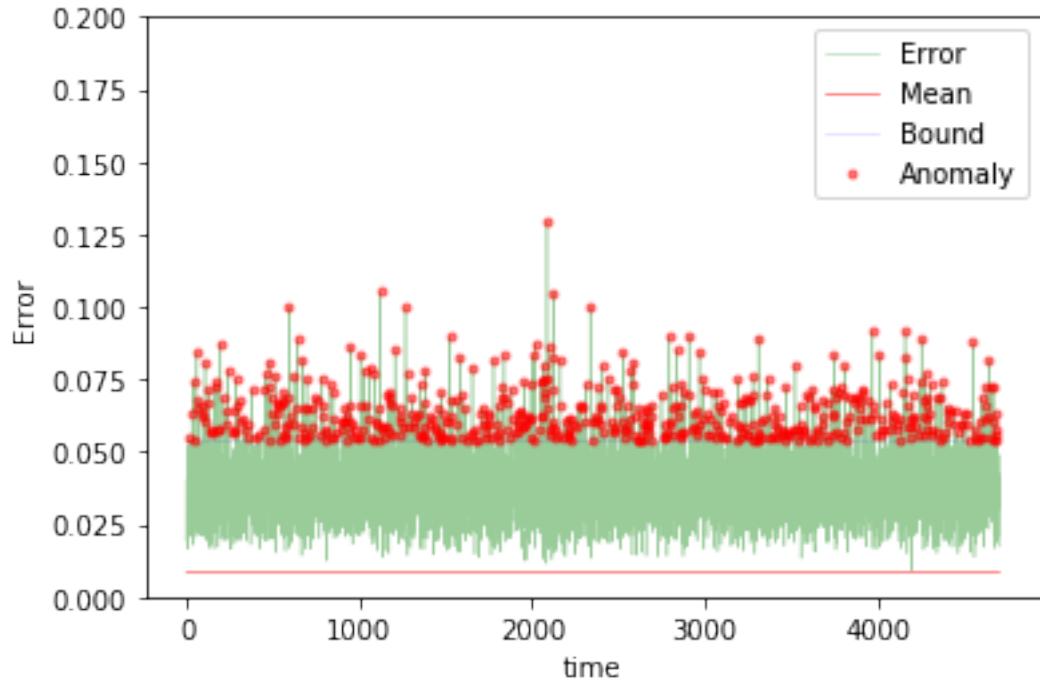




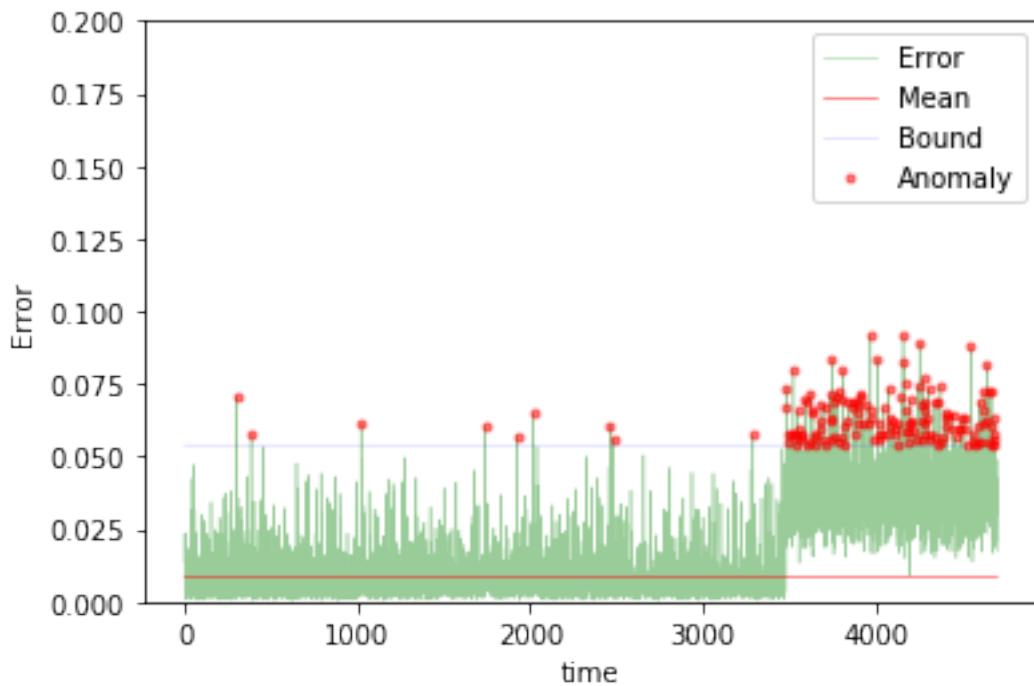
The mean error for nn1_20_normal_ is 0.008669323679901612 for length 4709
Testing on anomaly data.



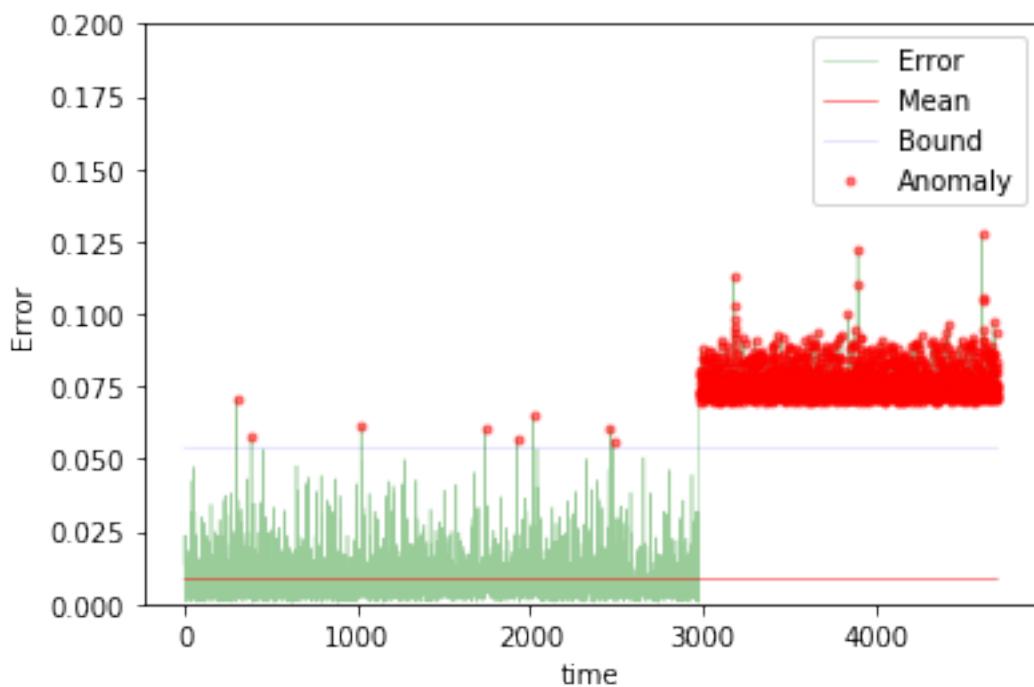
The mean error for nn1_20_anomaly_ is 0.009950267834319713 for length 4709
Testing on different app data.



The mean error for nn1_20_diff_app_ is 0.03861702077729091 for length 4709
Testing on App change synthetic data.



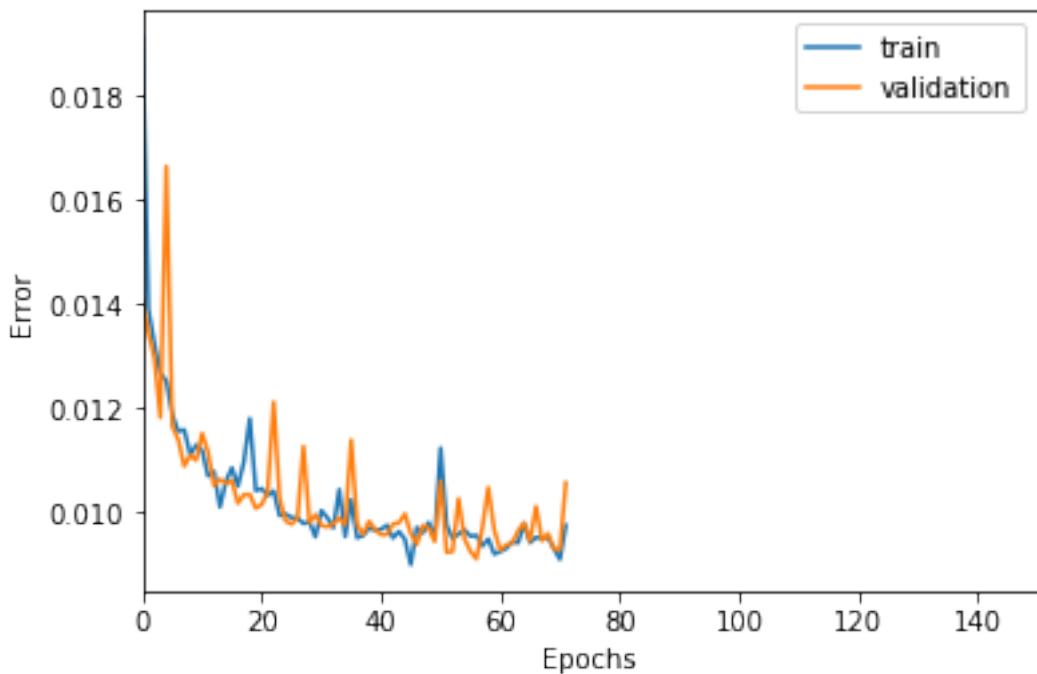
The mean error for nn1_20_app_change_ is 0.016415182120188065 for length 4709
Testing on Net flood synthetic data.



```
The mean error for nn1_20_net_flood_ is 0.033571041730044555 for length 4709  
=====
```

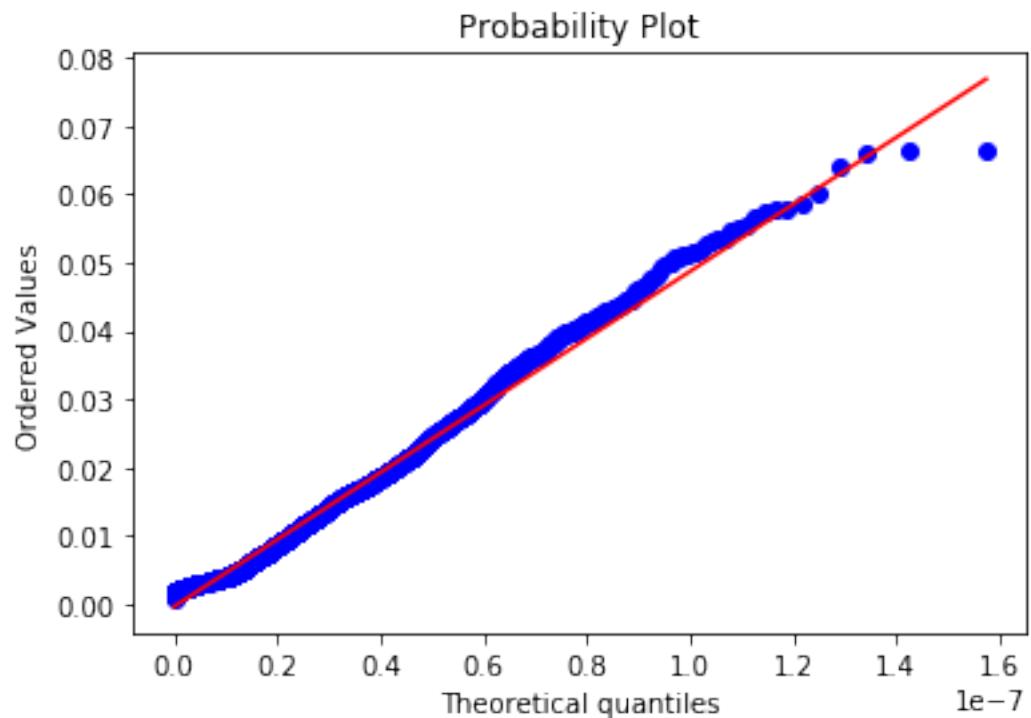
50 steps

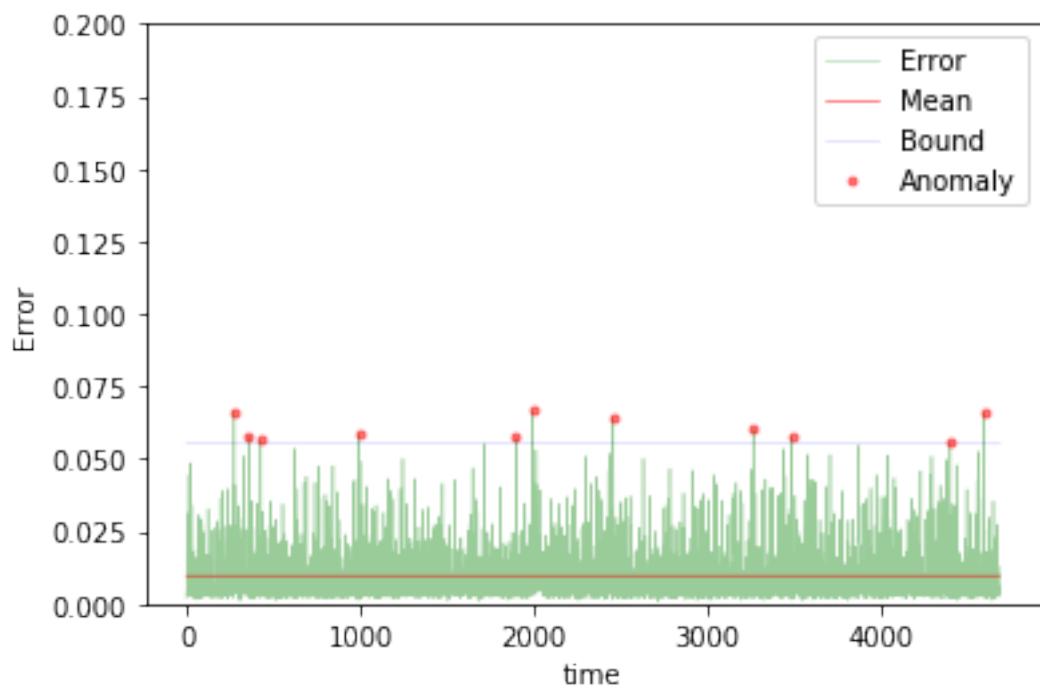
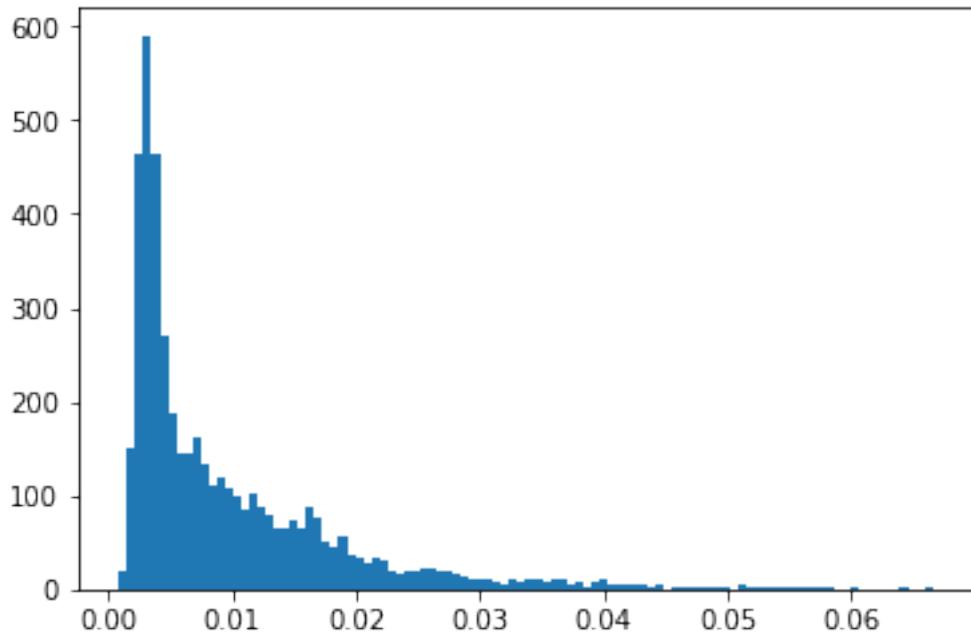
```
In [80]: TIMESTEPS = 50  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS)  
vgen = flat_generator(val_X, TIMESTEPS)  
name = "nn1_50"  
  
In [81]: input_layer = Input(shape=(TIMESTEPS*DIM,))  
hidden = Dense(100,activation='relu')(input_layer)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [82]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [83]: train(model, tgen, vgen, name=name)  
test(model, name=name, window=TIMESTEPS)
```



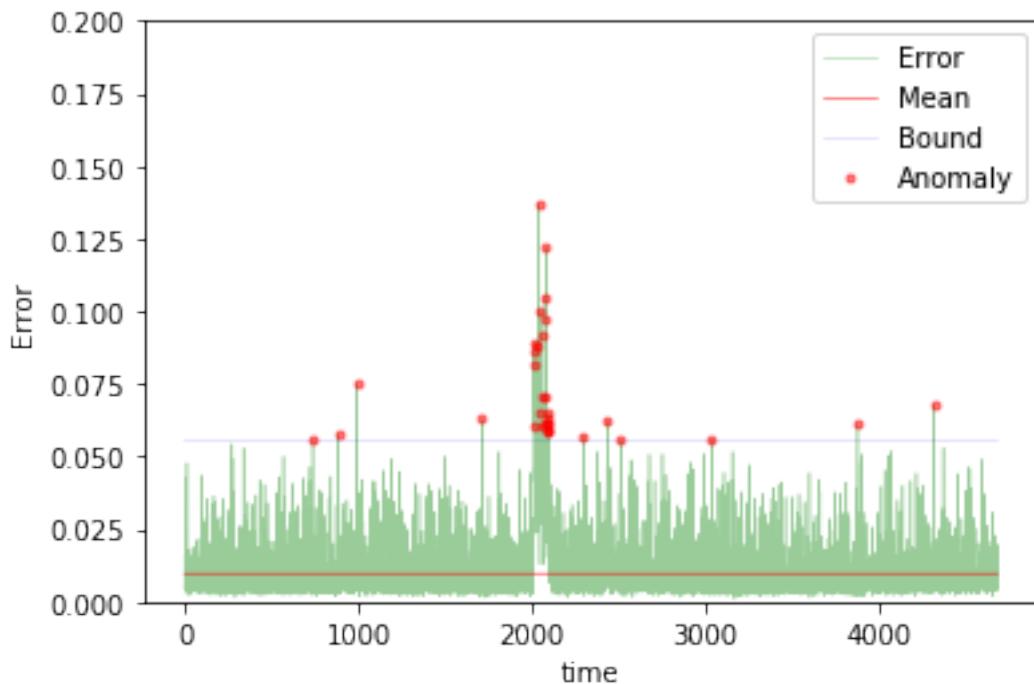
```
Training loss for final epoch is 0.009748656613635831
Validation loss for final epoch is 0.010563848153222352
----- Beginning tests for nn1_50 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

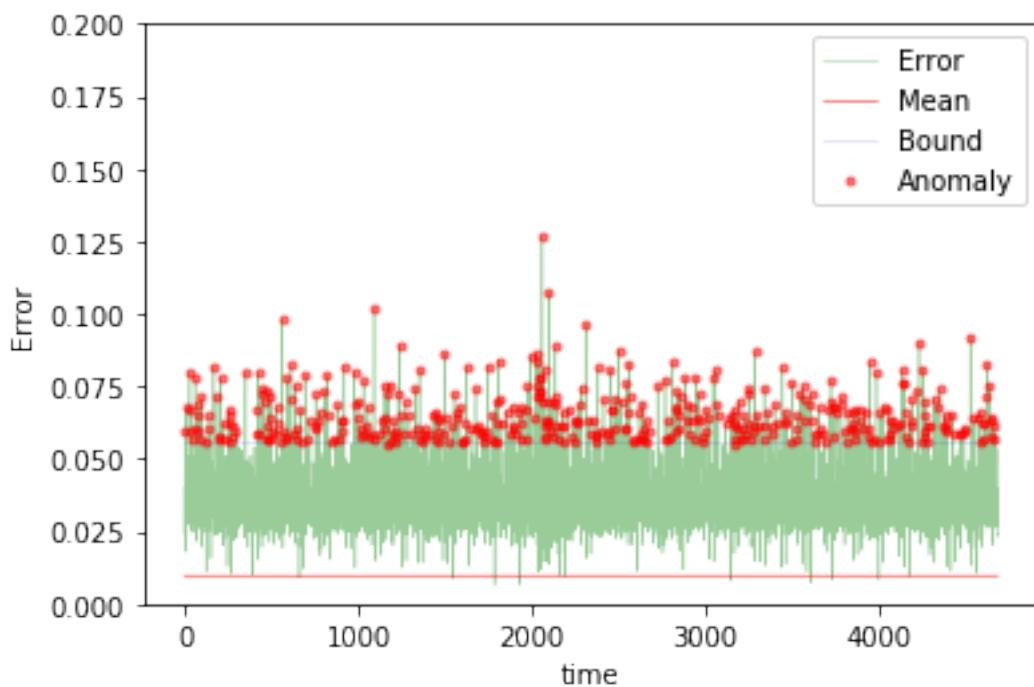




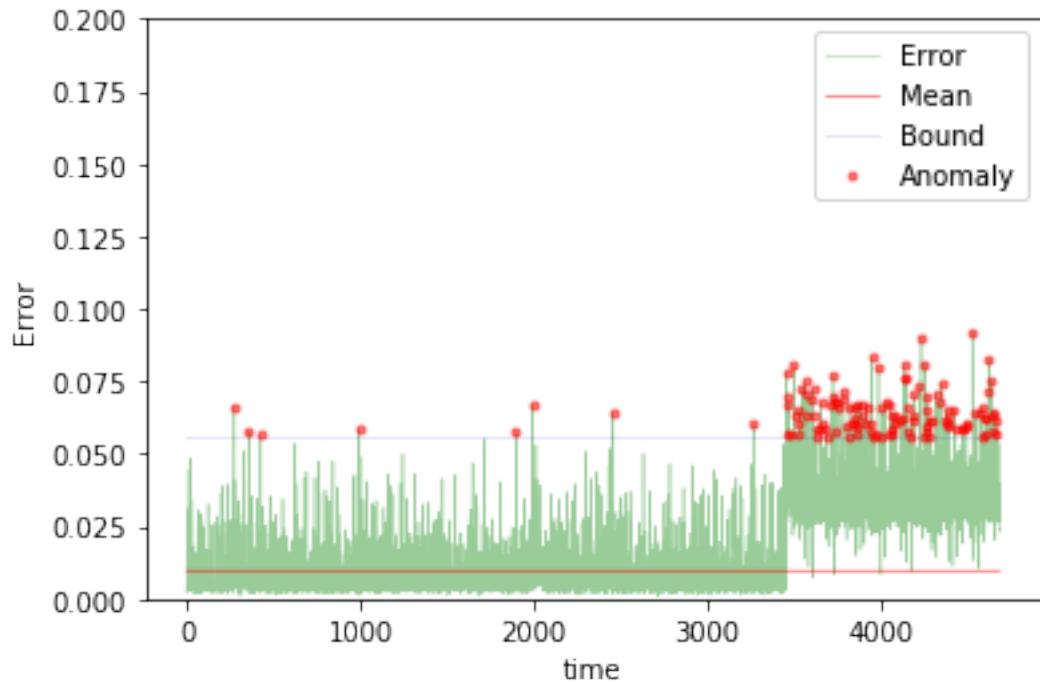
The mean error for nn1_50_normal_ is 0.009757616763219706 for length 4679
Testing on anomaly data.



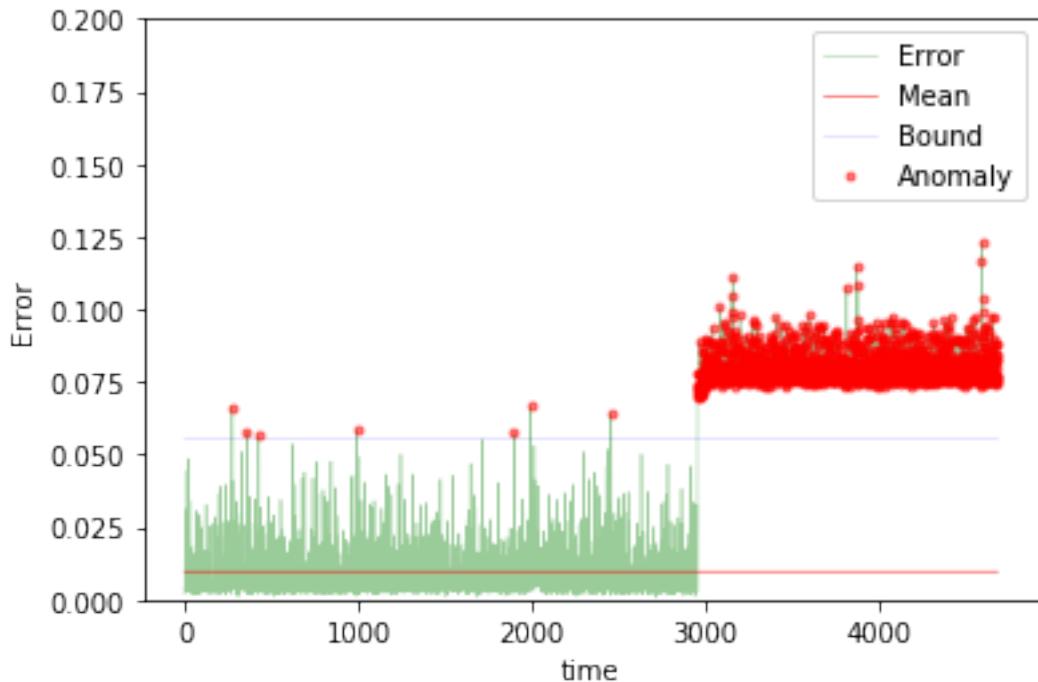
The mean error for nn1_50_anomaly_ is 0.011505548336418026 for length 4679
Testing on different app data.



The mean error for nn1_50_diff_app_ is 0.03694164105008707 for length 4679
Testing on App change synthetic data.



The mean error for nn1_50_app_change_ is 0.01688040184476267 for length 4679
Testing on Net flood synthetic data.



```
The mean error for nn1_50_net_flood_ is 0.03583563477026742 for length 4679
=====
```

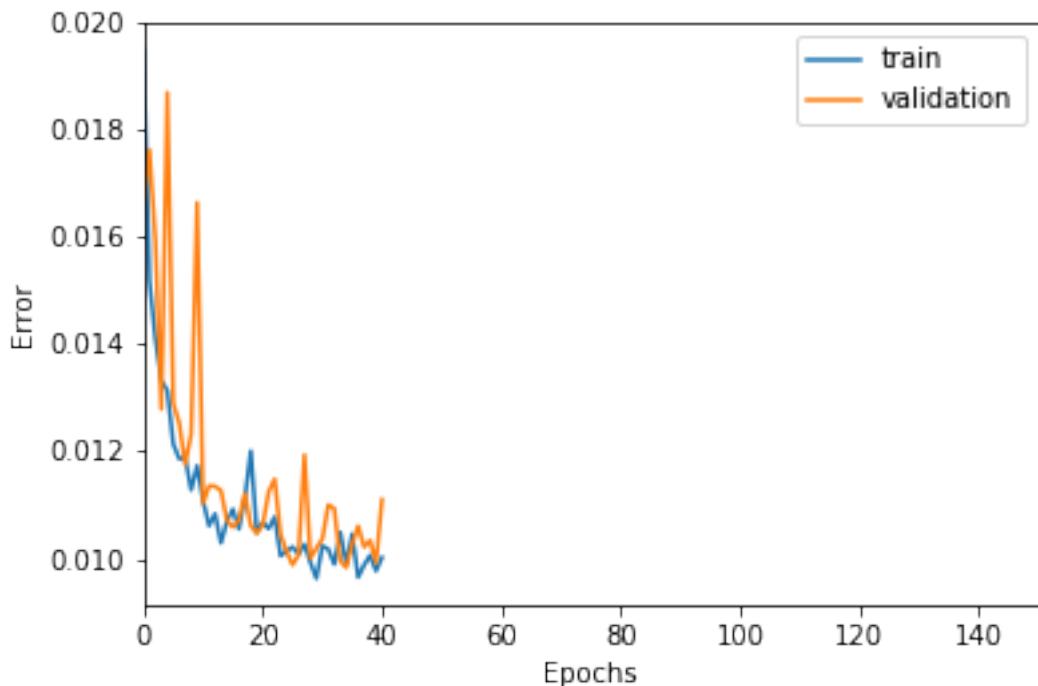
100 steps

```
In [84]: TIMESTEPS = 100
        DIM = 29
        tgen = flat_generator(X, TIMESTEPS)
        vgen = flat_generator(val_X, TIMESTEPS)
        name = "nn1_100"

In [85]: input_layer = Input(shape=(TIMESTEPS*DIM,))
        hidden = Dense(100,activation='relu')(input_layer)
        output = Dense(DIM, activation='sigmoid')(hidden)

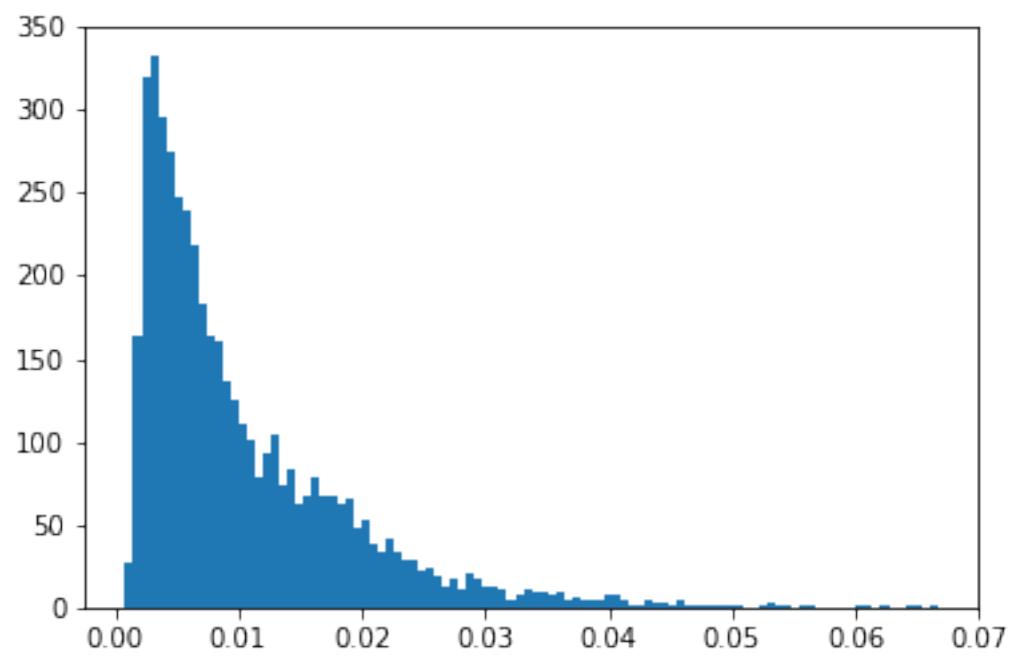
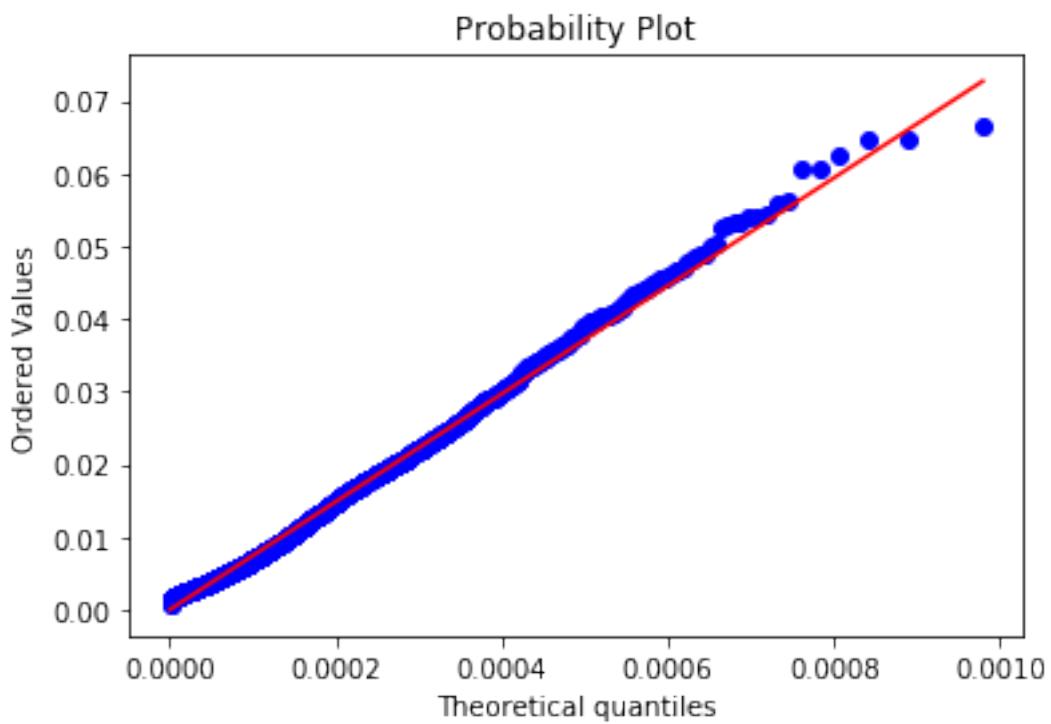
In [86]: model = Model(input_layer, output)
        model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

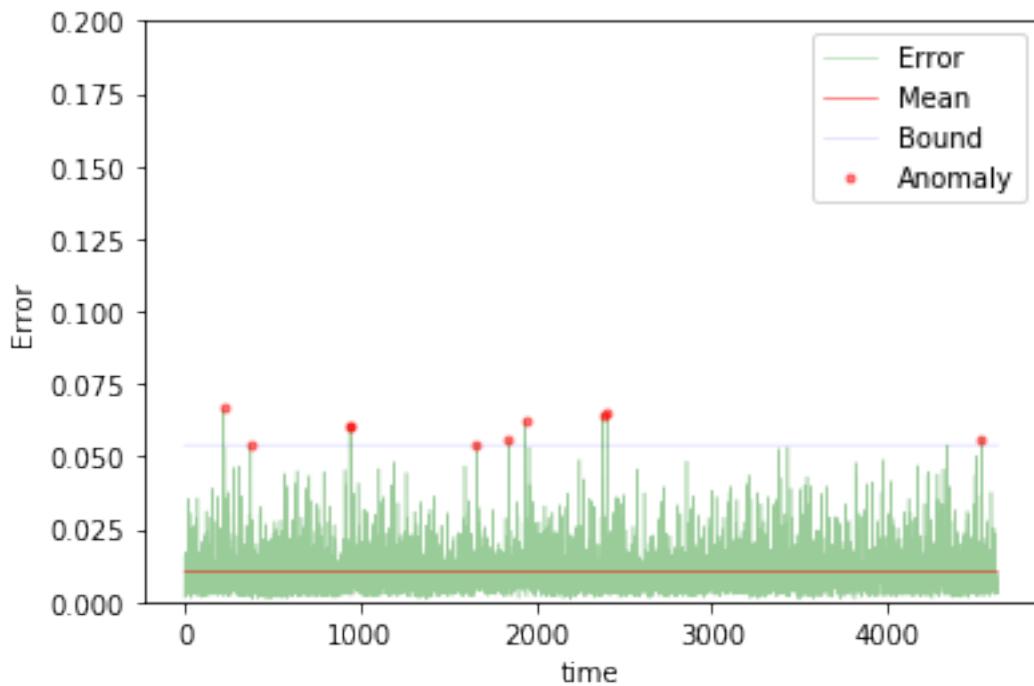
In [87]: train(model, tgen, vgen, name=name)
        test(model, name=name, window=TIMESTEPS)
```



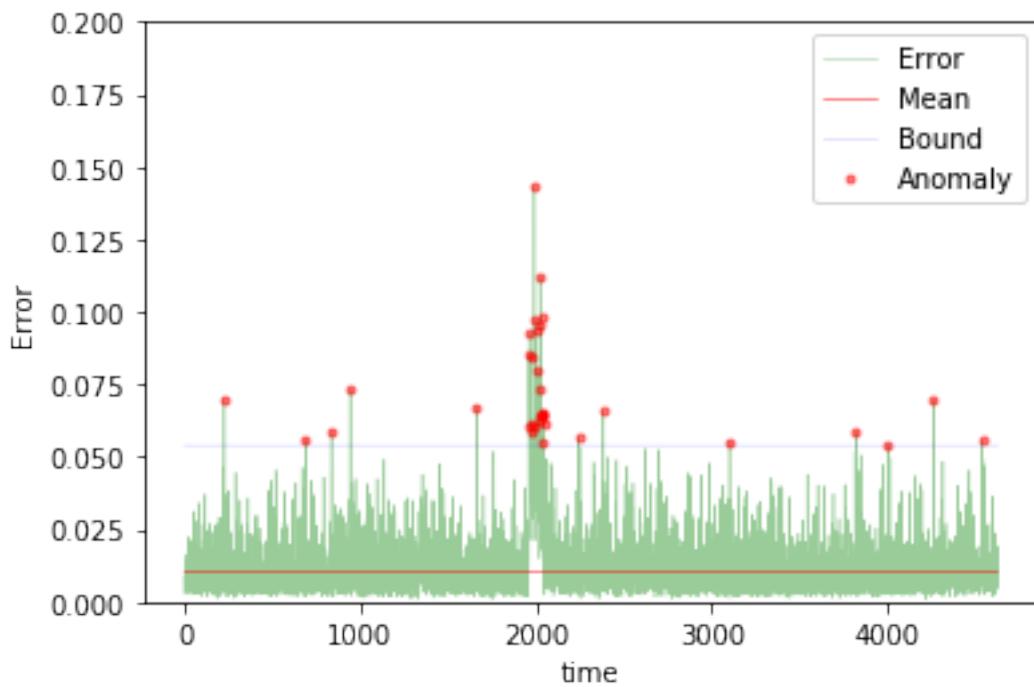
```
Training loss for final epoch is 0.010021210052305832
Validation loss for final epoch is 0.01110392516083084
----- Beginning tests for nn1_100 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarr
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

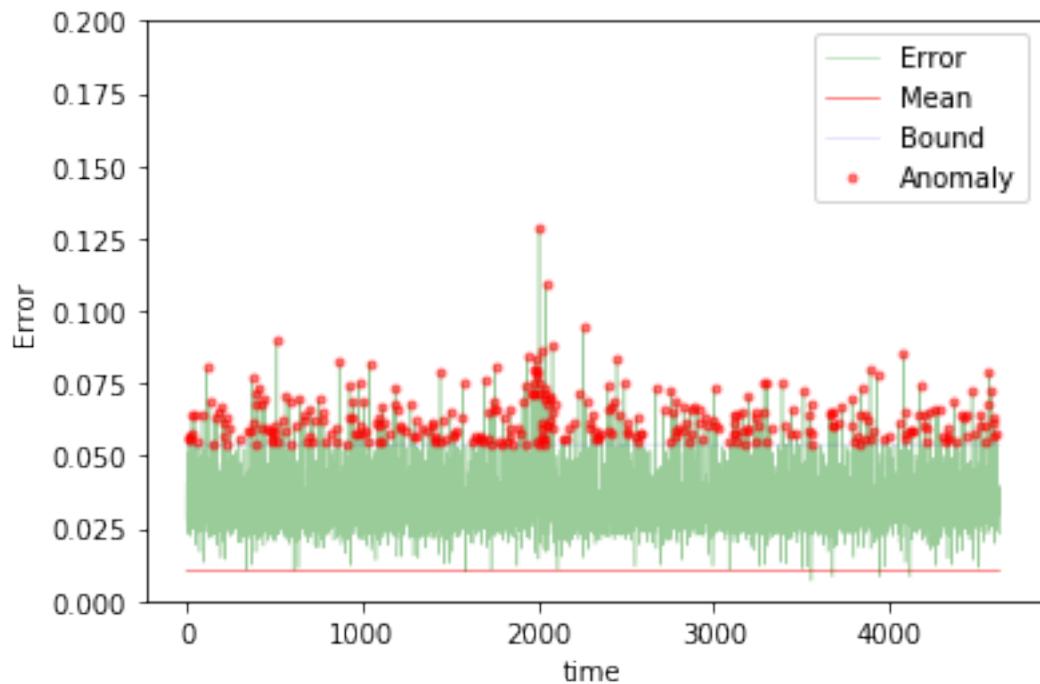




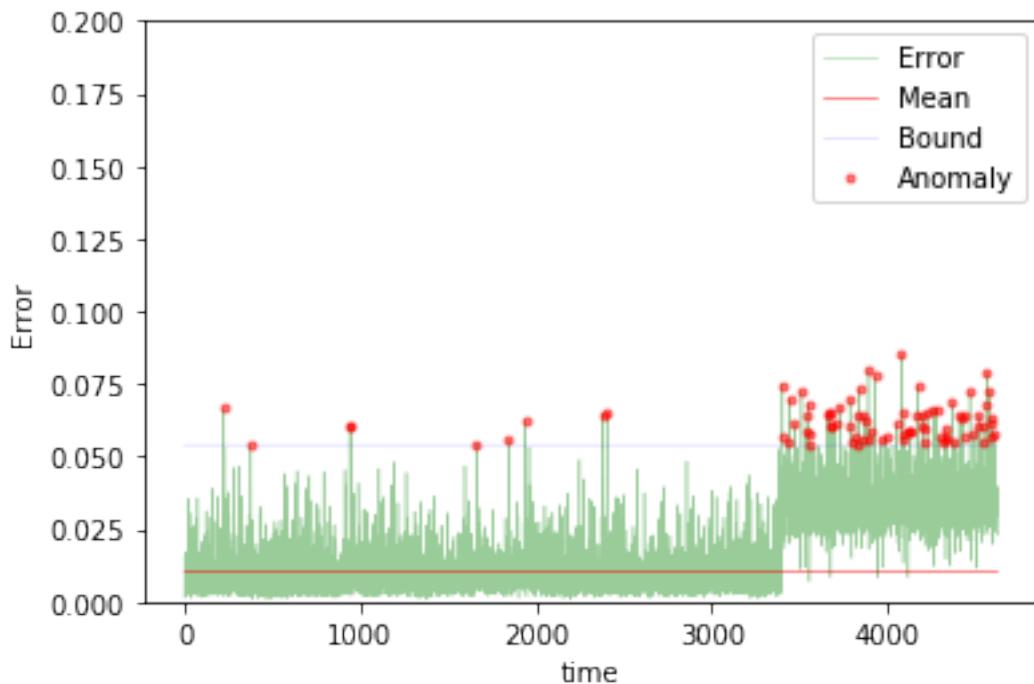
The mean error for nn1_100_normal_ is 0.010395604270698555 for length 4629
Testing on anomaly data.



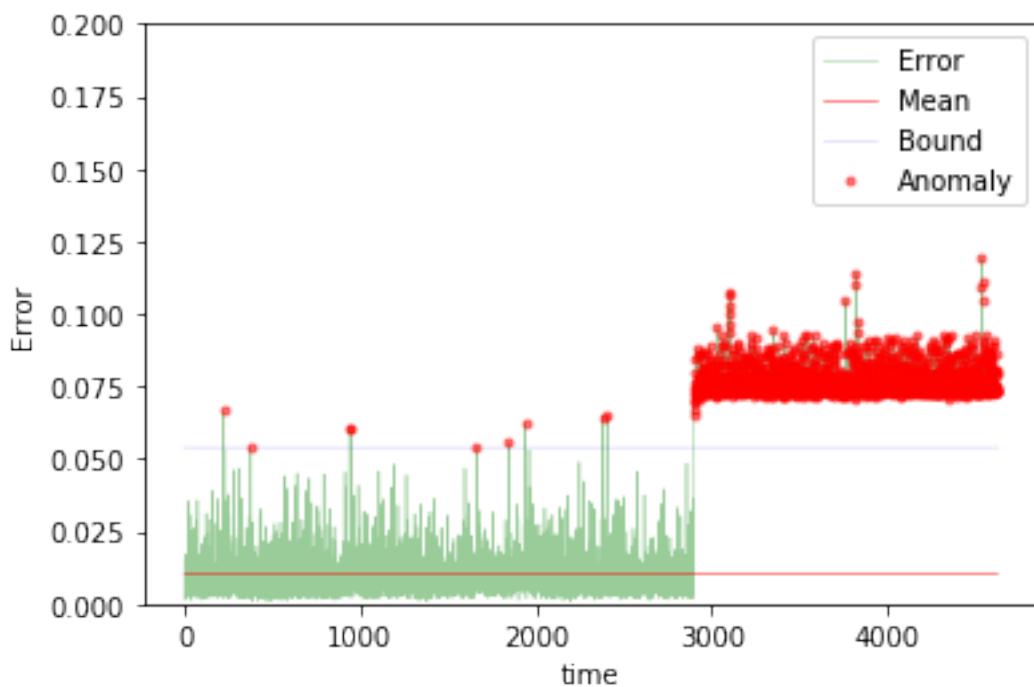
The mean error for nn1_100_anomaly_ is 0.012097168775885543 for length 4629
Testing on different app data.



The mean error for nn1_100_diff_app_ is 0.034712964129766856 for length 4629
Testing on App change synthetic data.



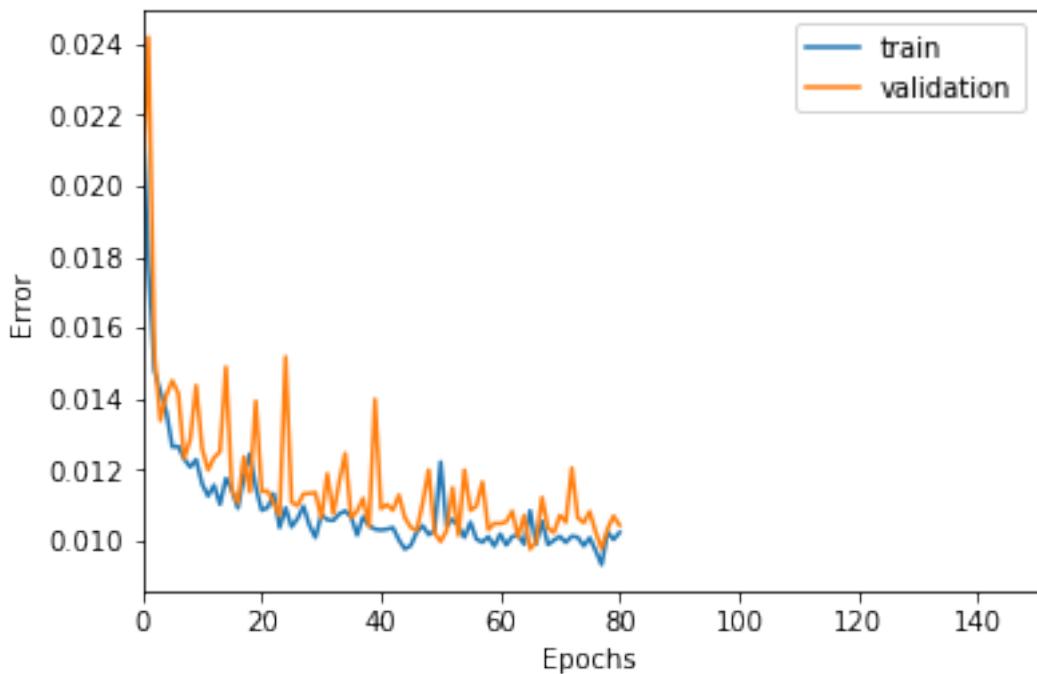
The mean error for nn1_100_app_change_ is 0.016752388218204725 for length 4629
Testing on Net flood synthetic data.



```
The mean error for nn1_100_net_flood_ is 0.035430818537116375 for length 4629  
=====
```

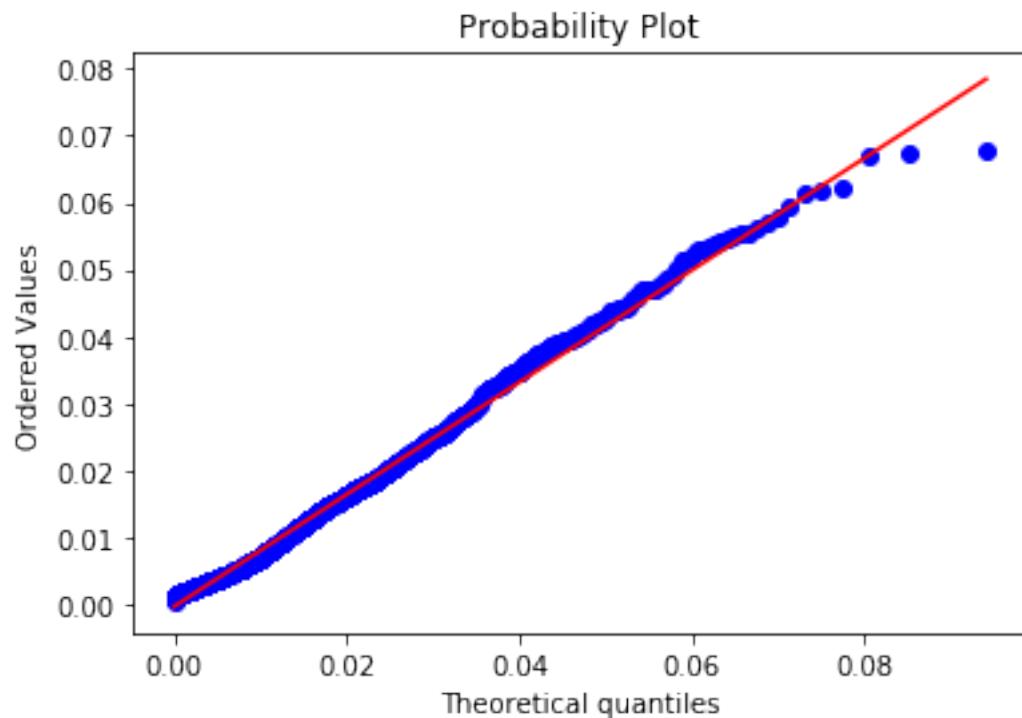
200 steps

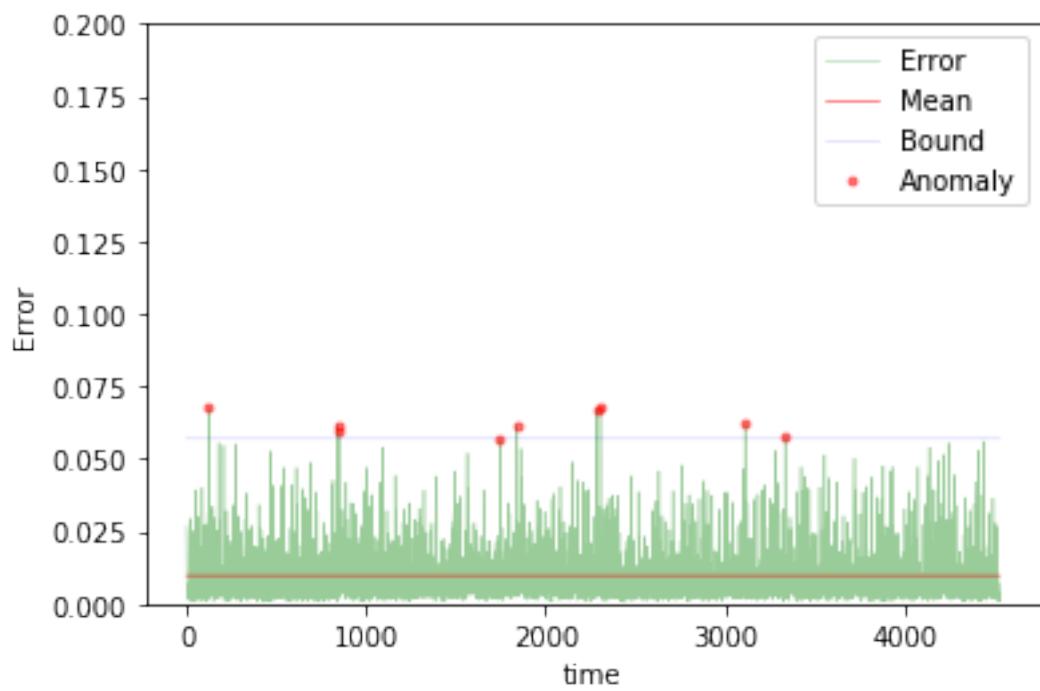
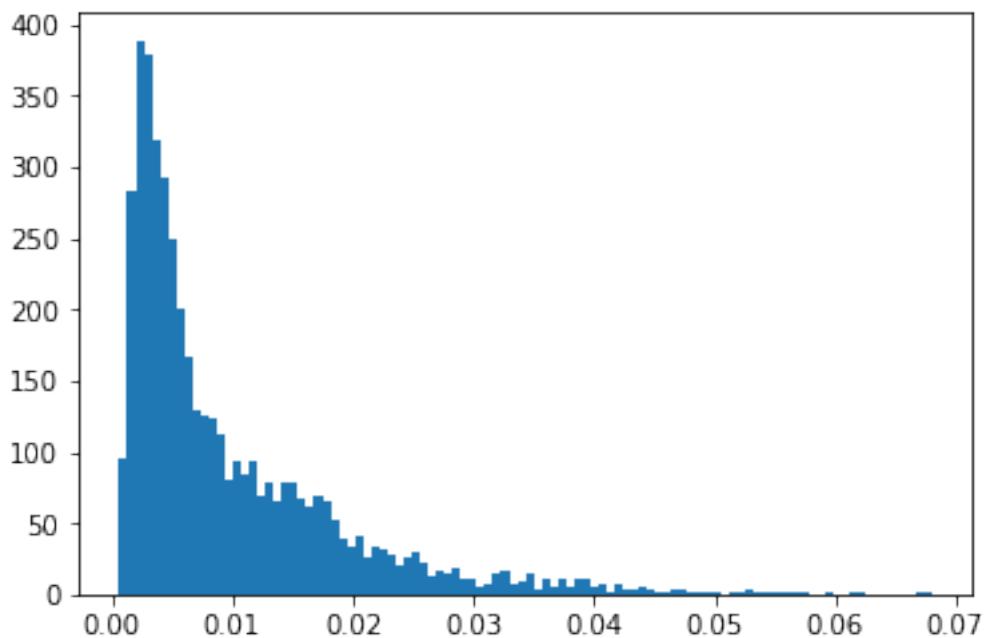
```
In [88]: TIMESTEPS = 200  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS)  
vgen = flat_generator(val_X, TIMESTEPS)  
name = "nn1_200"  
  
In [89]: input_layer = Input(shape=(TIMESTEPS*DIM,))  
hidden = Dense(100,activation='relu')(input_layer)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [90]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [91]: train(model, tgen, vgen, name=name)  
test(model, name=name, window=TIMESTEPS)
```



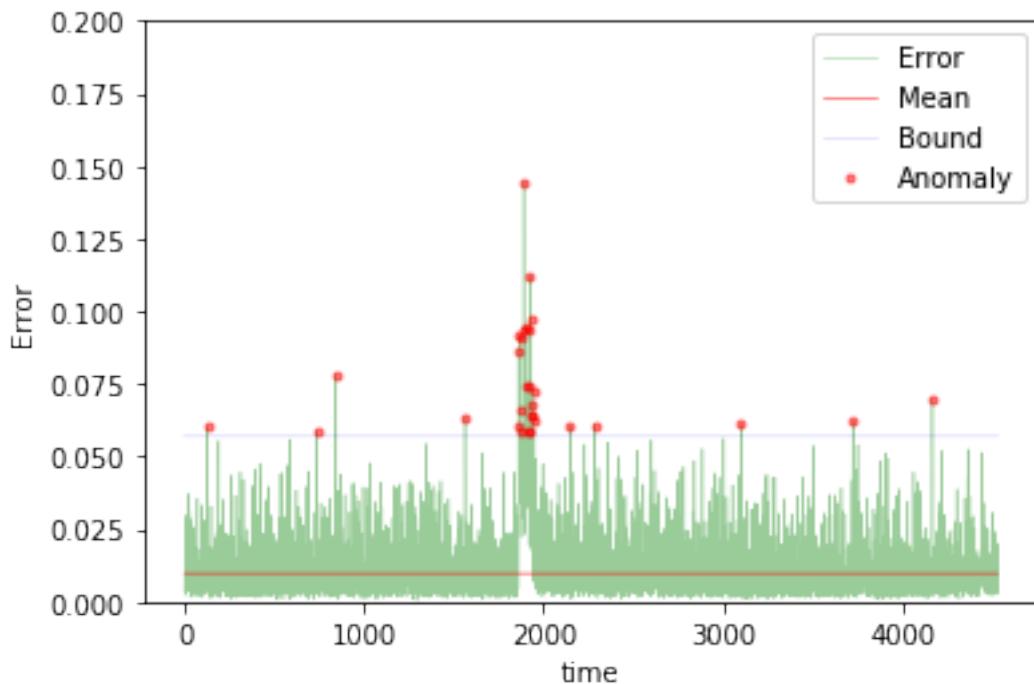
```
Training loss for final epoch is 0.010239744153455832
Validation loss for final epoch is 0.010444162765052169
----- Beginning tests for nn1_200 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

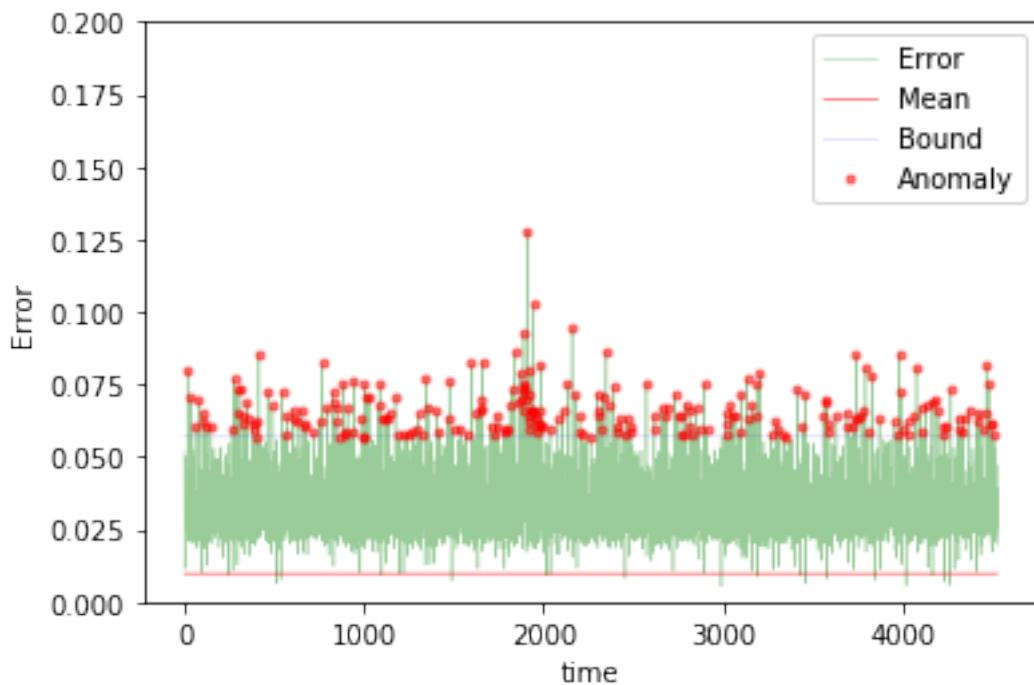




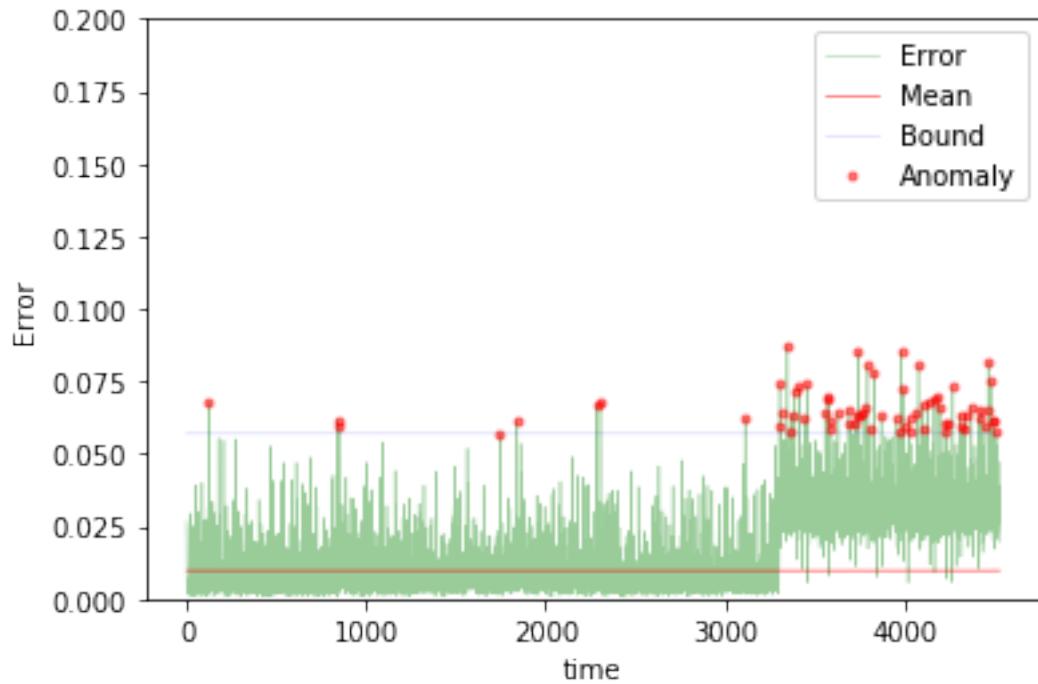
The mean error for nn1_200_normal_ is 0.009832392805333318 for length 4529
Testing on anomaly data.



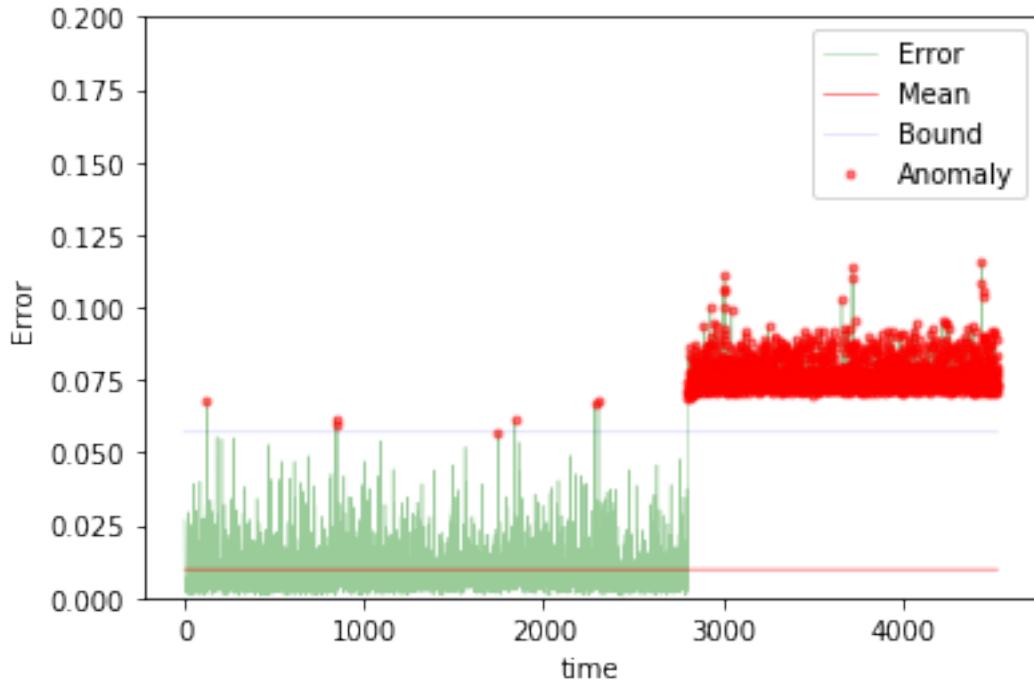
The mean error for nn1_200_anomaly_ is 0.011814727407014107 for length 4529
Testing on different app data.



The mean error for nn1_200_diff_app_ is 0.033453162335244035 for length 4529
Testing on App change synthetic data.



The mean error for nn1_200_app_change_ is 0.016223124870744874 for length 4529
Testing on Net flood synthetic data.



```
The mean error for nn1_200_net_flood_ is 0.03558964818773853 for length 4529
=====
```

2.1.3 NN with 2 hidden layers

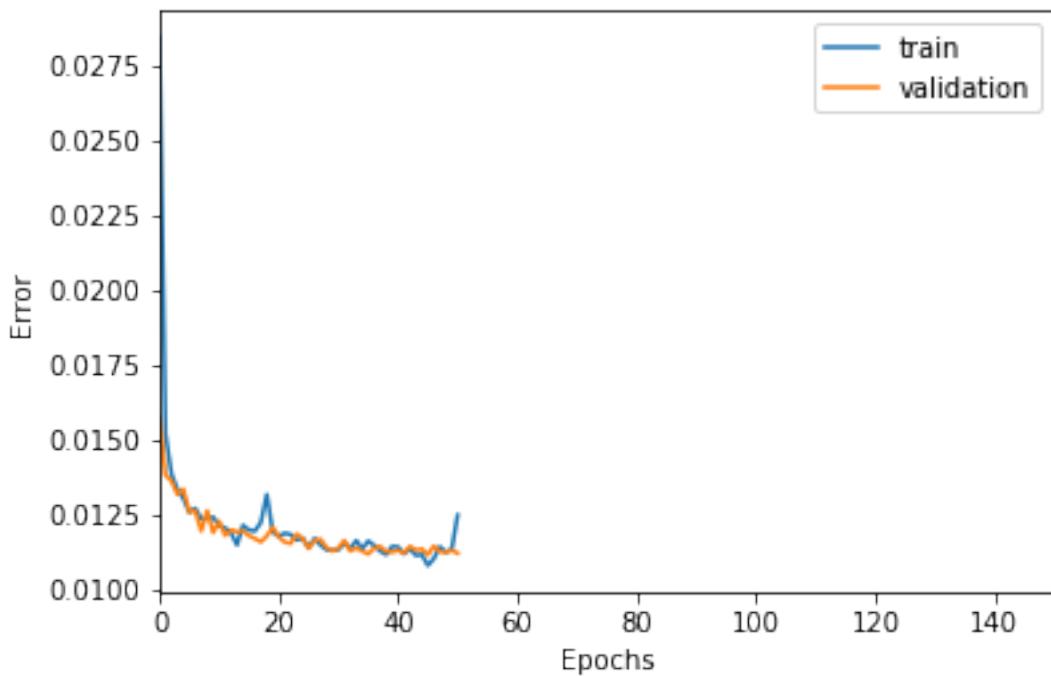
2 steps

```
In [92]: TIMESTEPS = 2
        DIM = 29
        tgen = flat_generator(X, TIMESTEPS)
        vgen = flat_generator(val_X, TIMESTEPS)
        name = "nn2_2"

In [93]: 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)

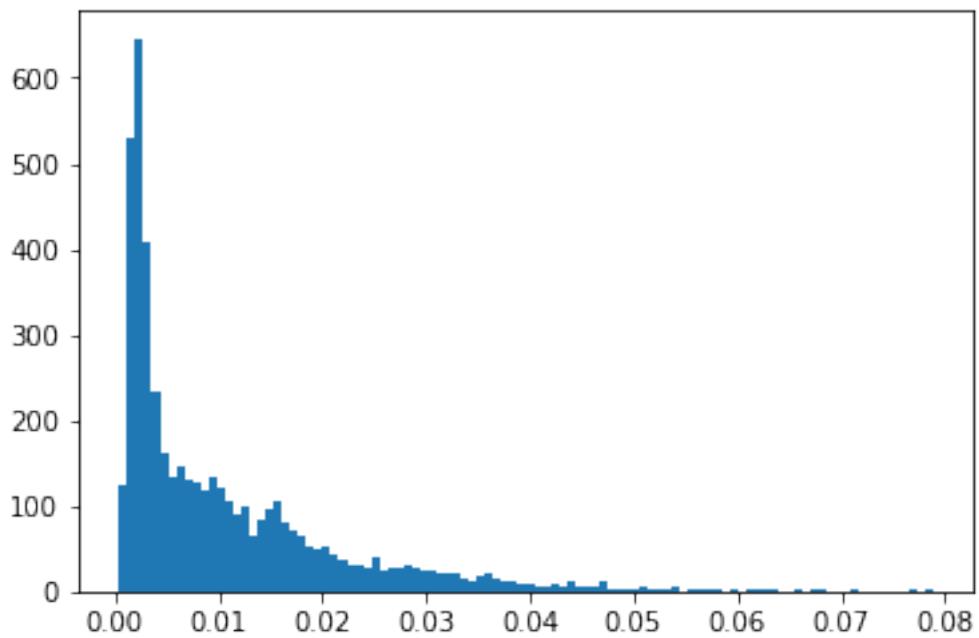
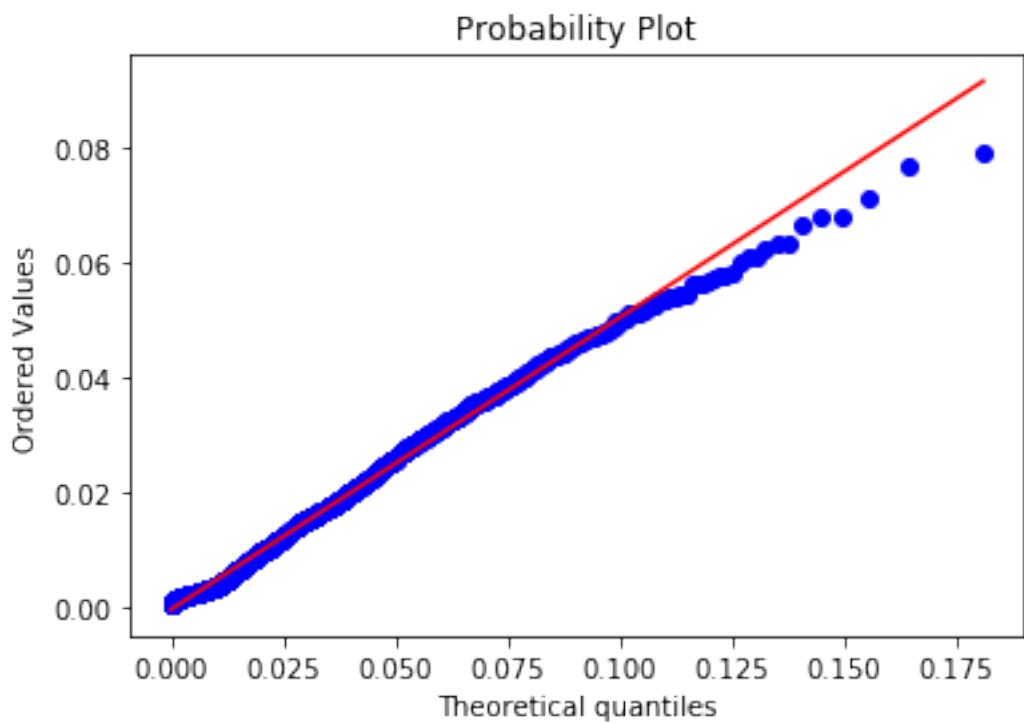
In [94]: model = Model(input_layer, output)
        model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])
```

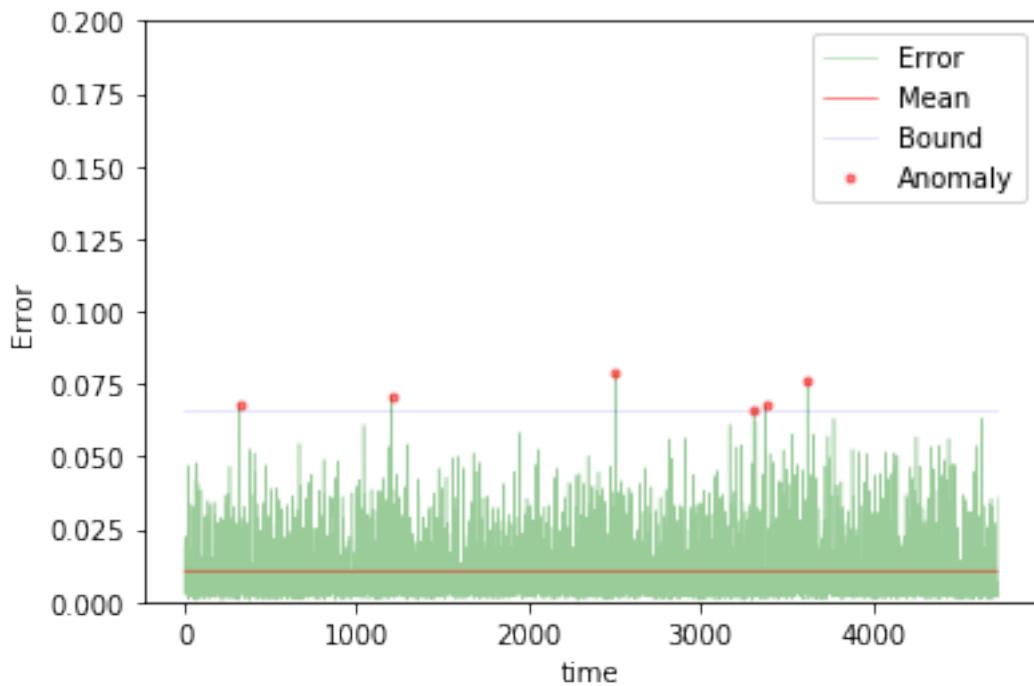
```
In [95]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```



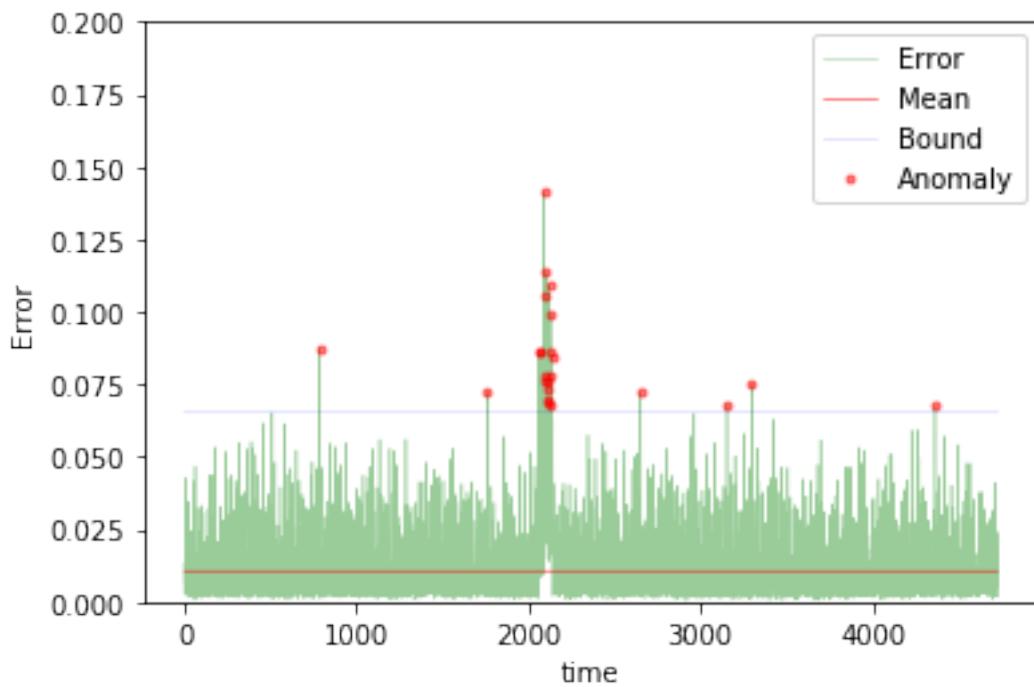
```
Training loss for final epoch is 0.01249816711503081
Validation loss for final epoch is 0.011226942741428502
----- Beginning tests for nn2_2 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
    warnings.warn(msg, RuntimeWarning)
```

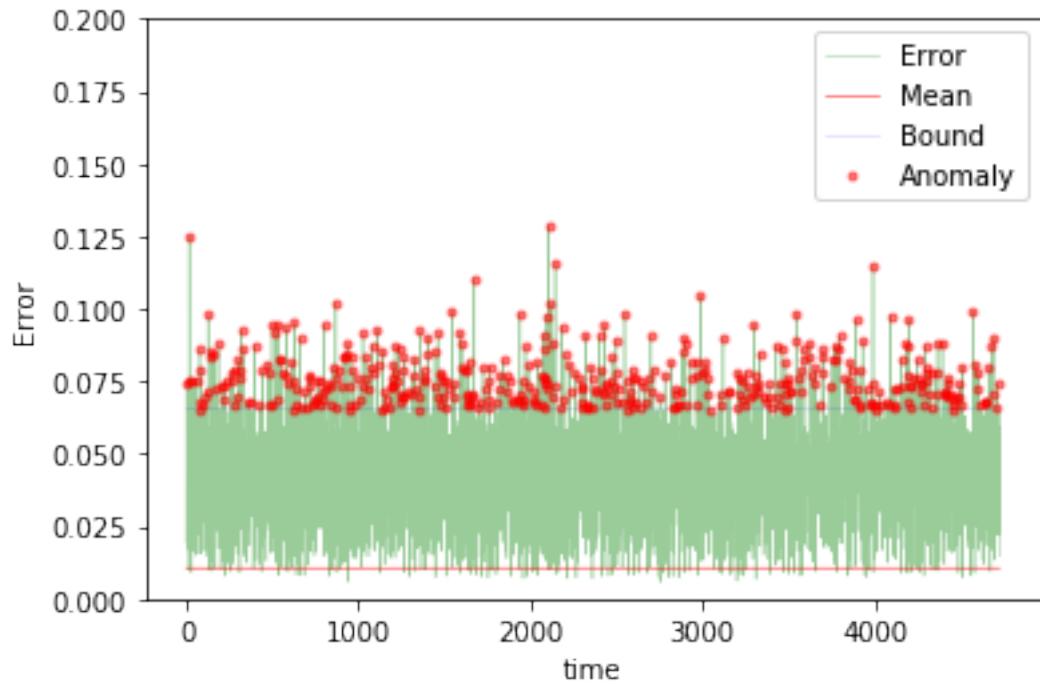




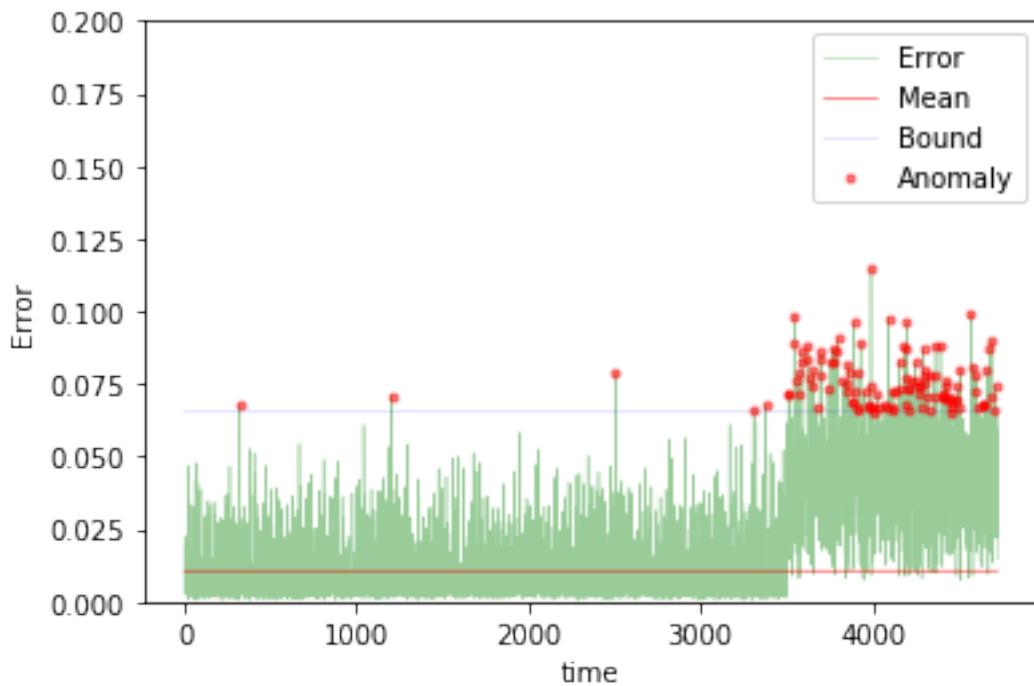
The mean error for nn2_2_normal_ is 0.010673421023006655 for length 4727
Testing on anomaly data.



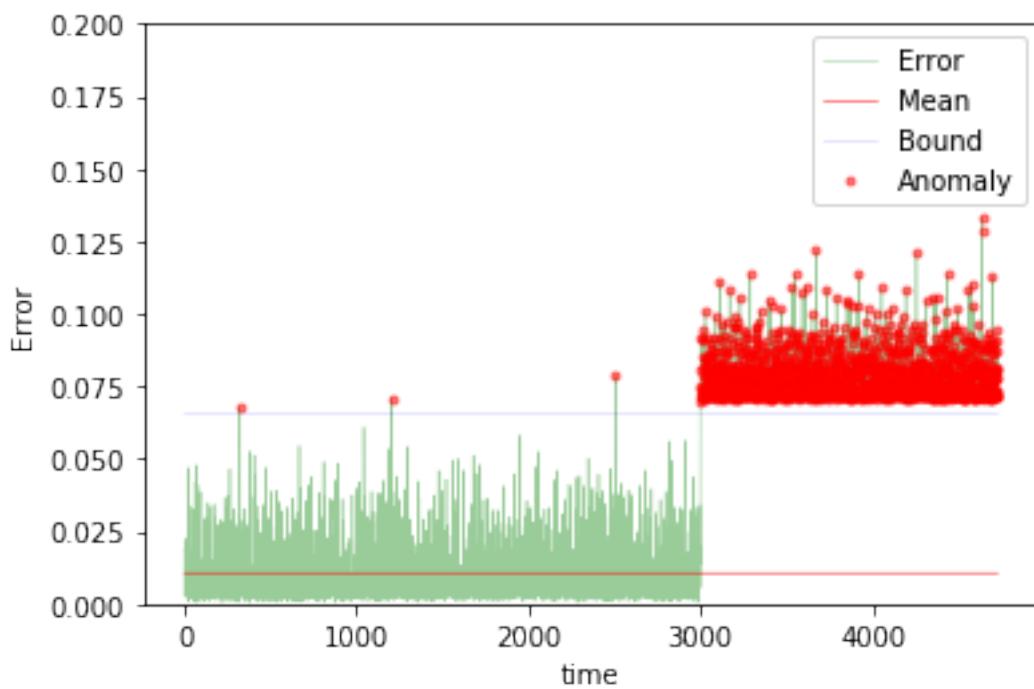
The mean error for nn2_2_anomaly_ is 0.012025300497217731 for length 4727
Testing on different app data.



The mean error for nn2_2_diff_app_ is 0.04356586919832851 for length 4727
Testing on App change synthetic data.



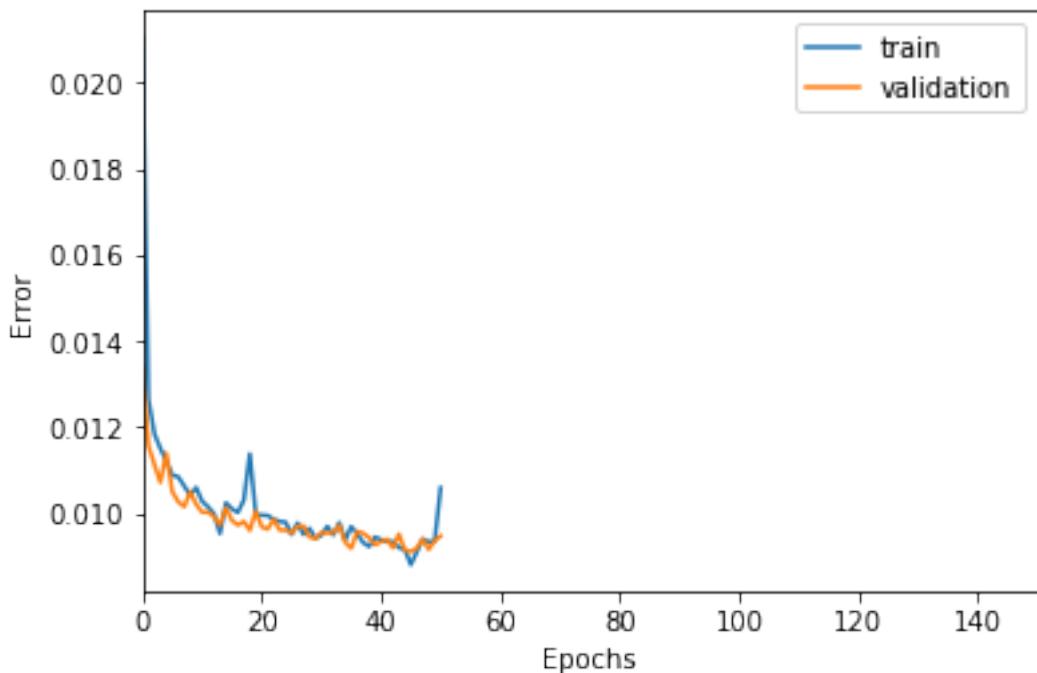
The mean error for nn2_2_app_change_ is 0.019209475143002065 for length 4727
Testing on Net flood synthetic data.



```
The mean error for nn2_2_net_flood_ is 0.0353840810297696 for length 4727  
=====
```

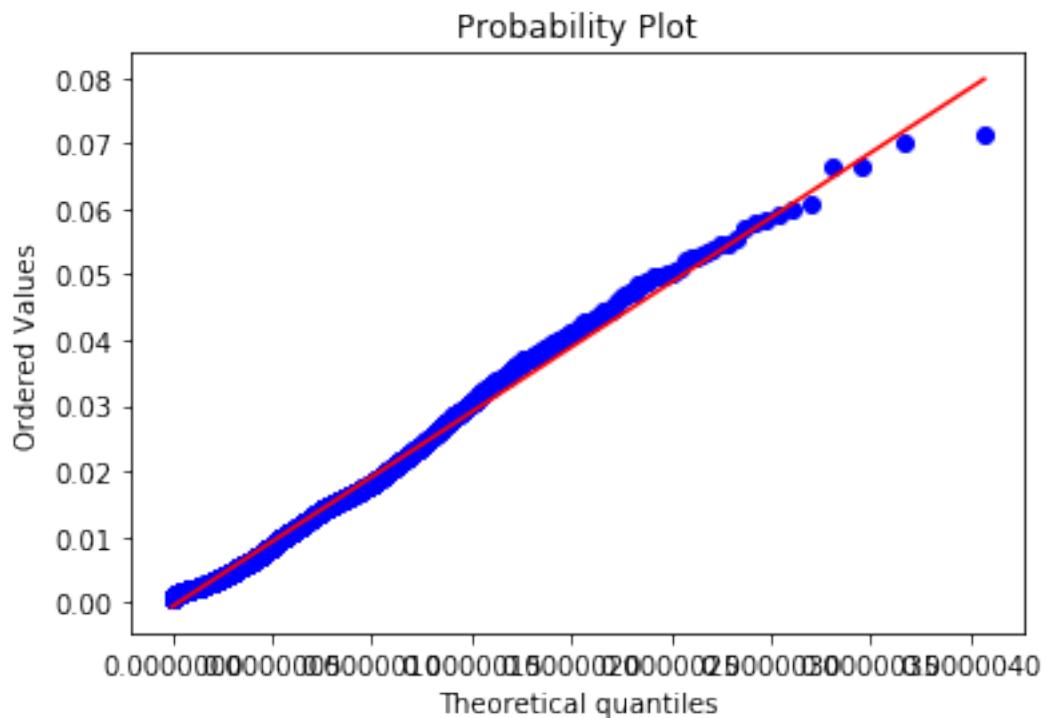
5 steps

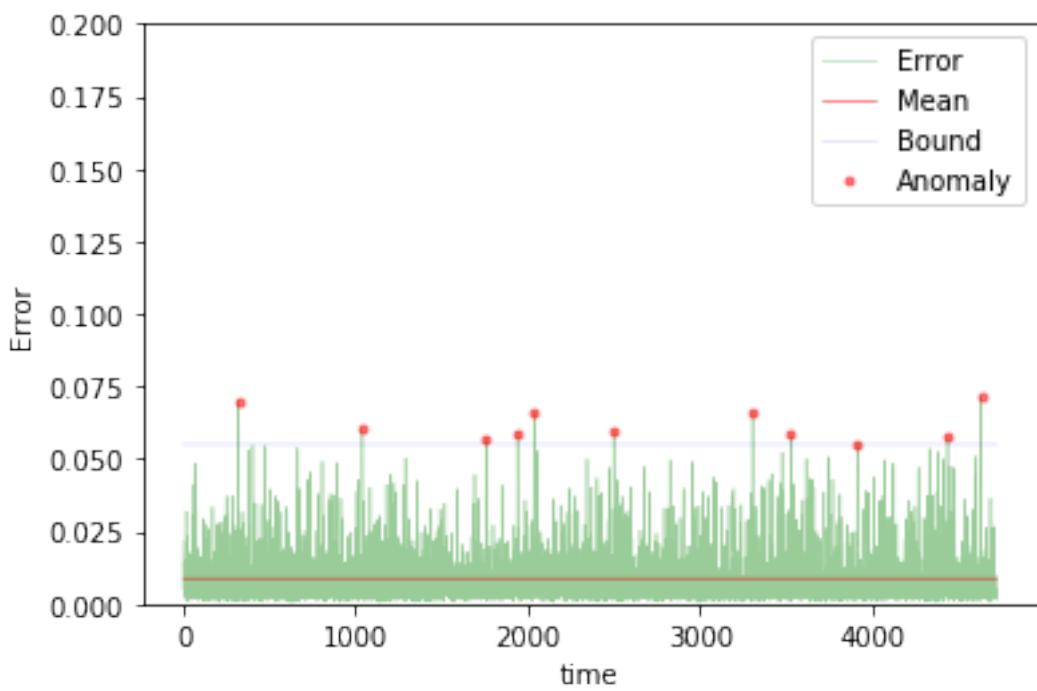
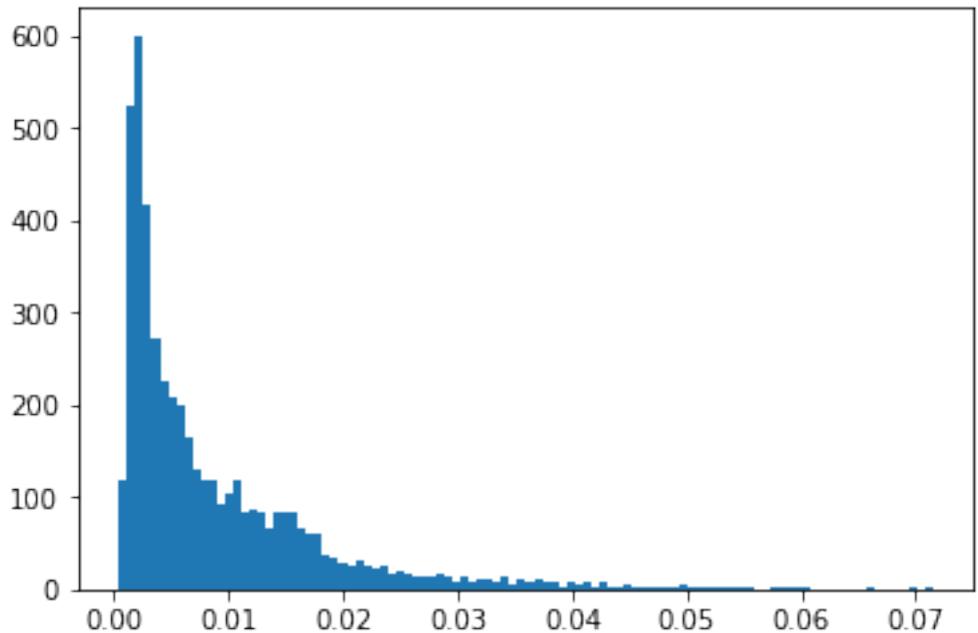
```
In [96]: TIMESTEPS = 5  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS)  
vgen = flat_generator(val_X, TIMESTEPS)  
name = "nn2_5"  
  
In [97]: input_layer = Input(shape=(TIMESTEPS*DIM,))  
hidden = Dense(500, activation='relu')(input_layer)  
hidden = Dense(100, activation='relu')(hidden)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [98]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [99]: train(model, tgen, vgen, name=name)  
test(model, name=name, window=TIMESTEPS)
```



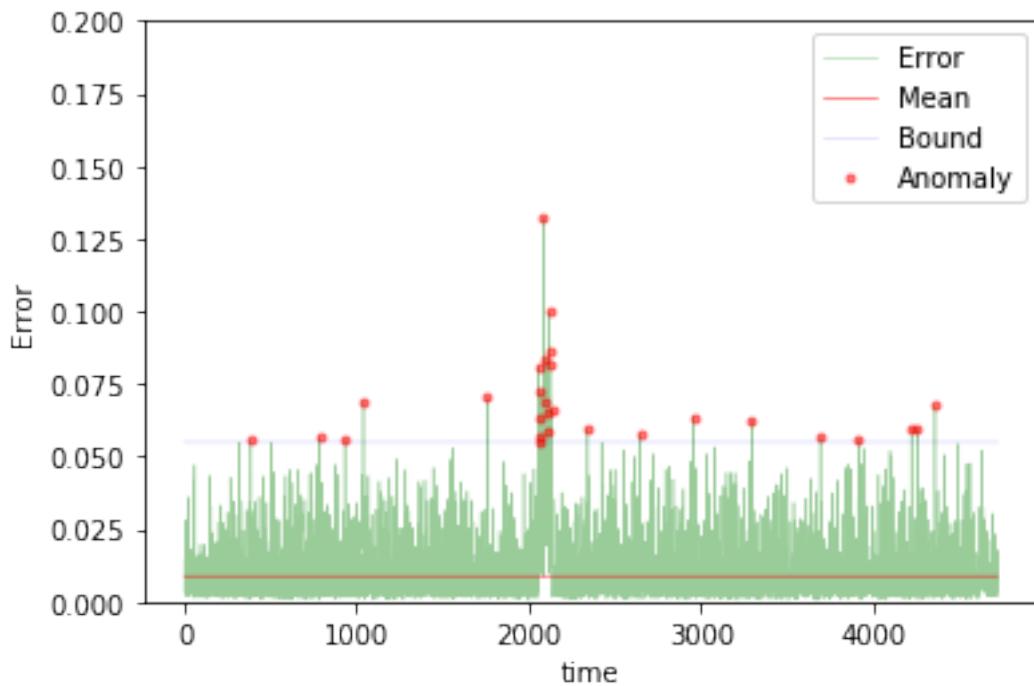
```
Training loss for final epoch is 0.0105977627459215
Validation loss for final epoch is 0.009476703165797516
----- Beginning tests for nn2_5 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

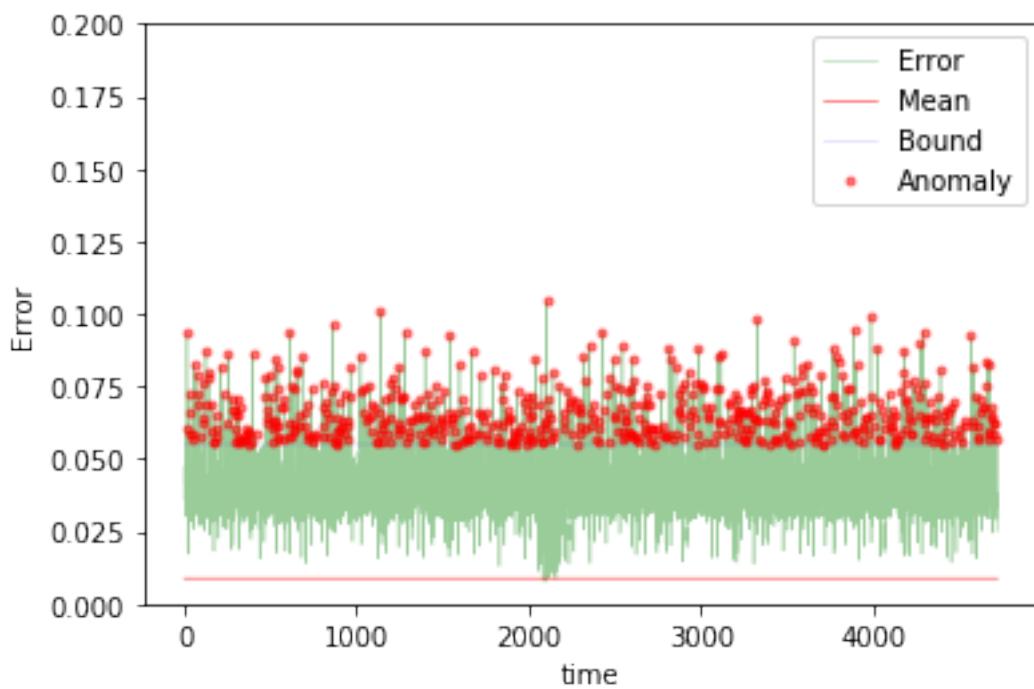




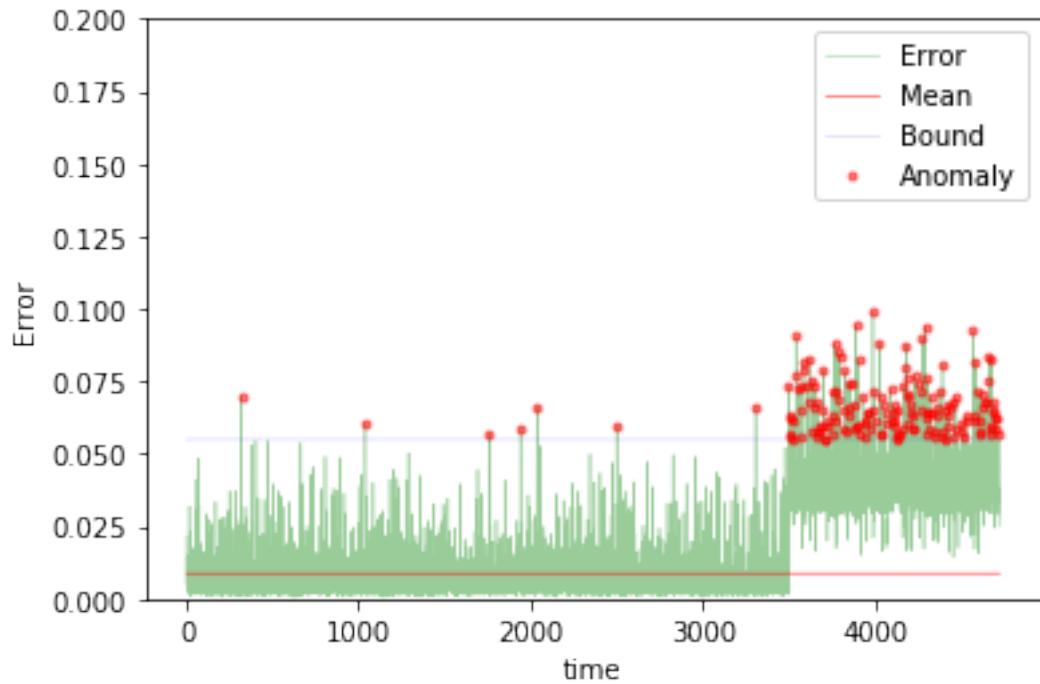
The mean error for nn2_5_normal_ is 0.008862563671550644 for length 4724
Testing on anomaly data.



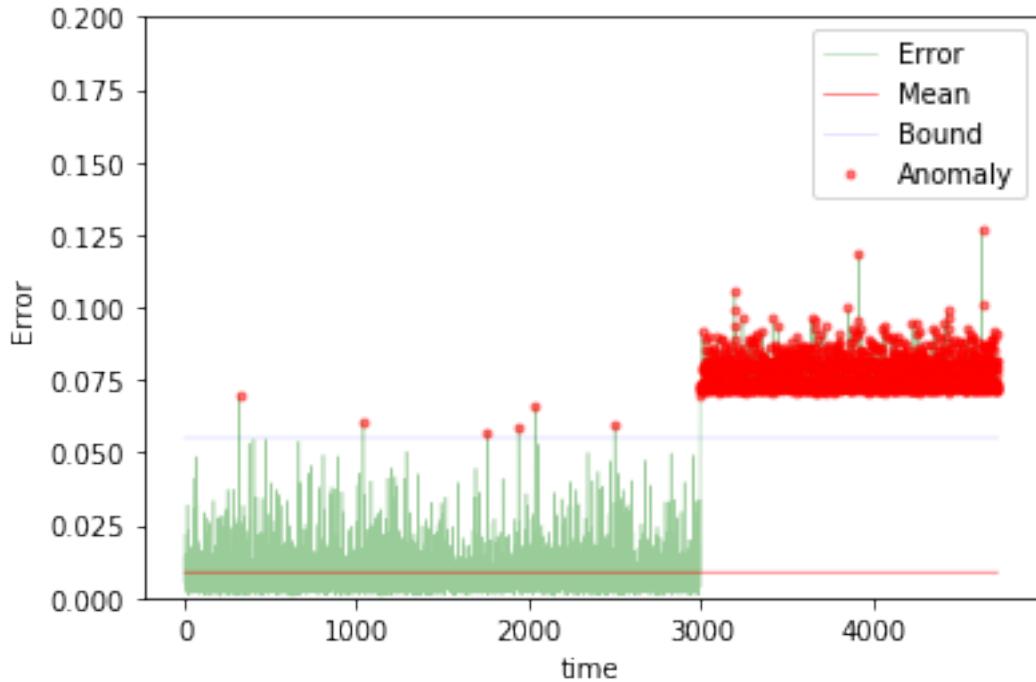
The mean error for nn2_5_anomaly_ is 0.01020333839115681 for length 4724
Testing on different app data.



The mean error for nn2_5_diff_app_ is 0.04078151195838591 for length 4724
Testing on App change synthetic data.



The mean error for nn2_5_app_change_ is 0.017206626618824574 for length 4724
Testing on Net flood synthetic data.



The mean error for nn2_5_net_flood_ is 0.033707567056049936 for length 4724
=====

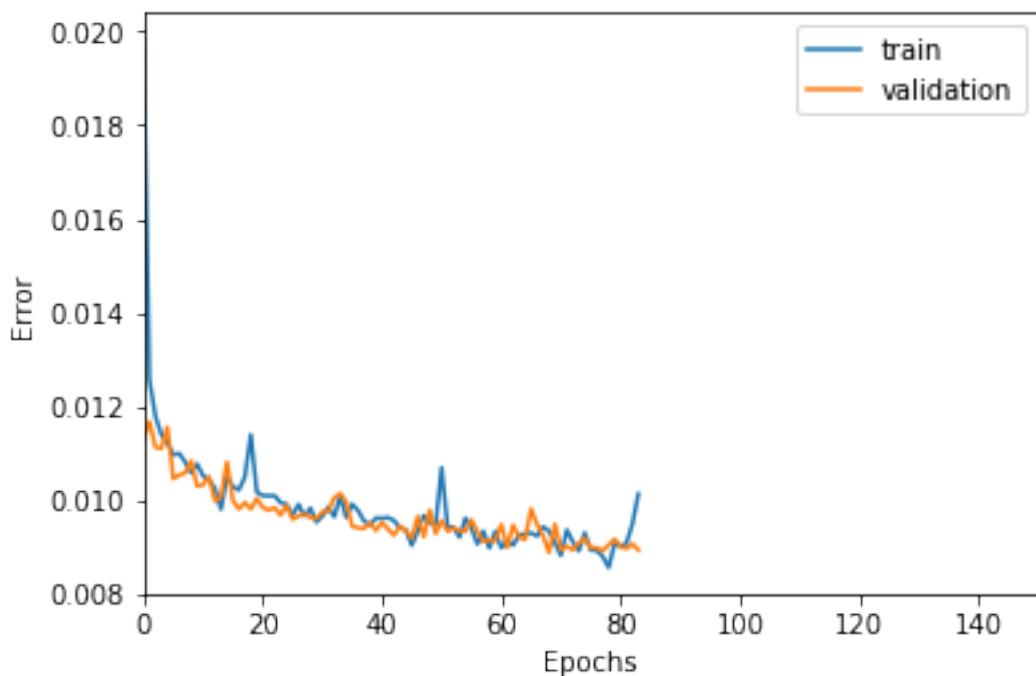
10 steps

```
In [100]: TIMESTEPS = 10
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS)
          vgen = flat_generator(val_X, TIMESTEPS)
          name = "nn2_10"

In [101]: 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)

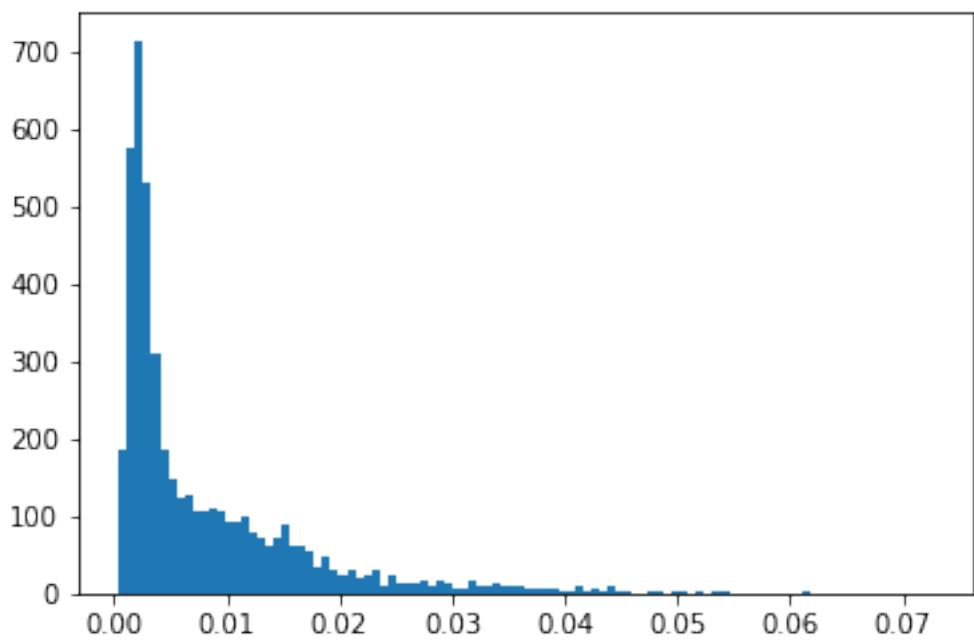
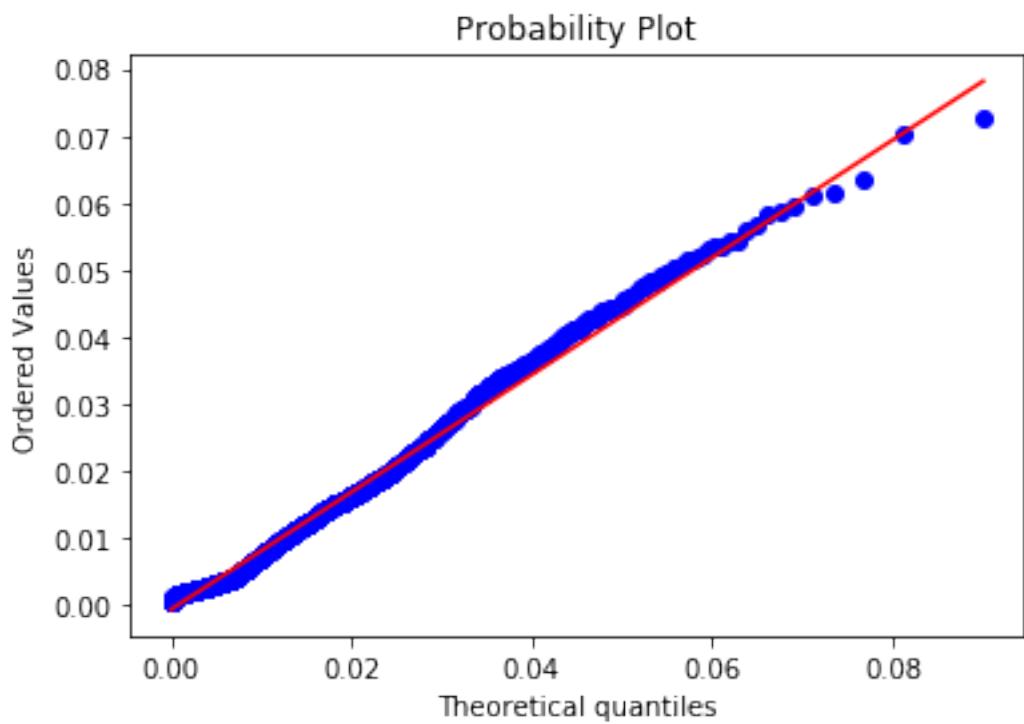
In [102]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

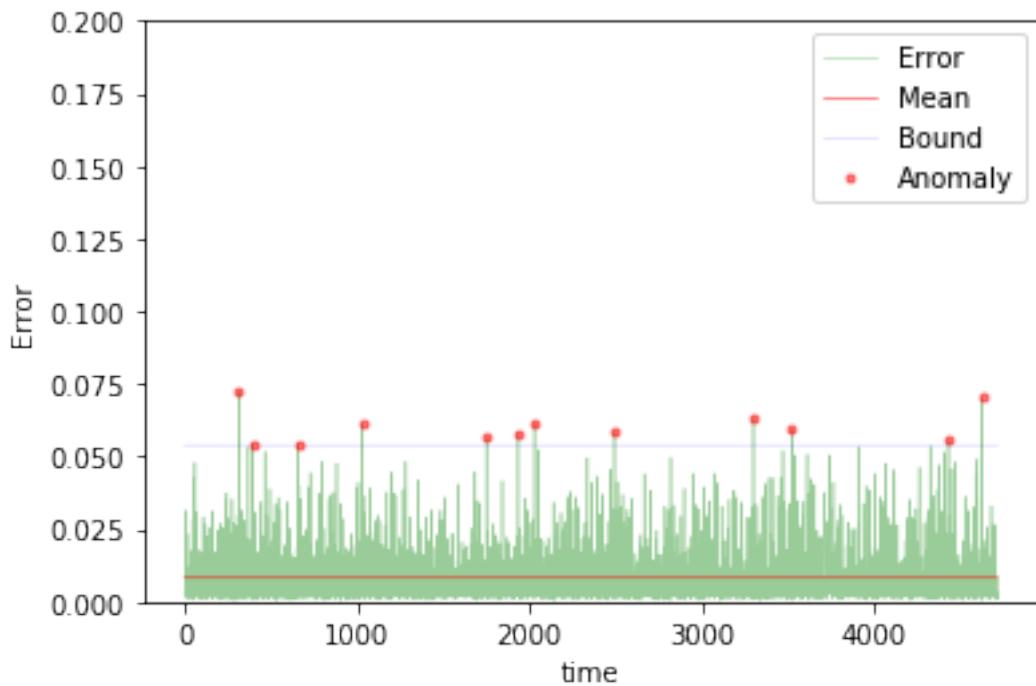
In [103]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```



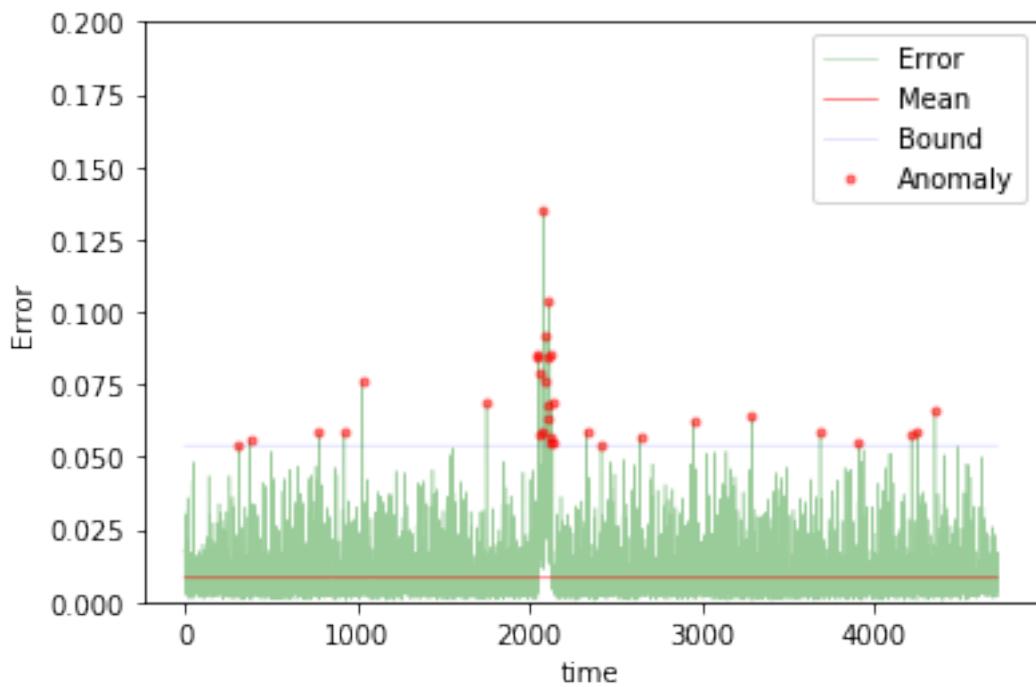
```
Training loss for final epoch is 0.010144109691958874
Validation loss for final epoch is 0.008945434572990052
----- Beginning tests for nn2_10 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
    warnings.warn(msg, RuntimeWarning)
```

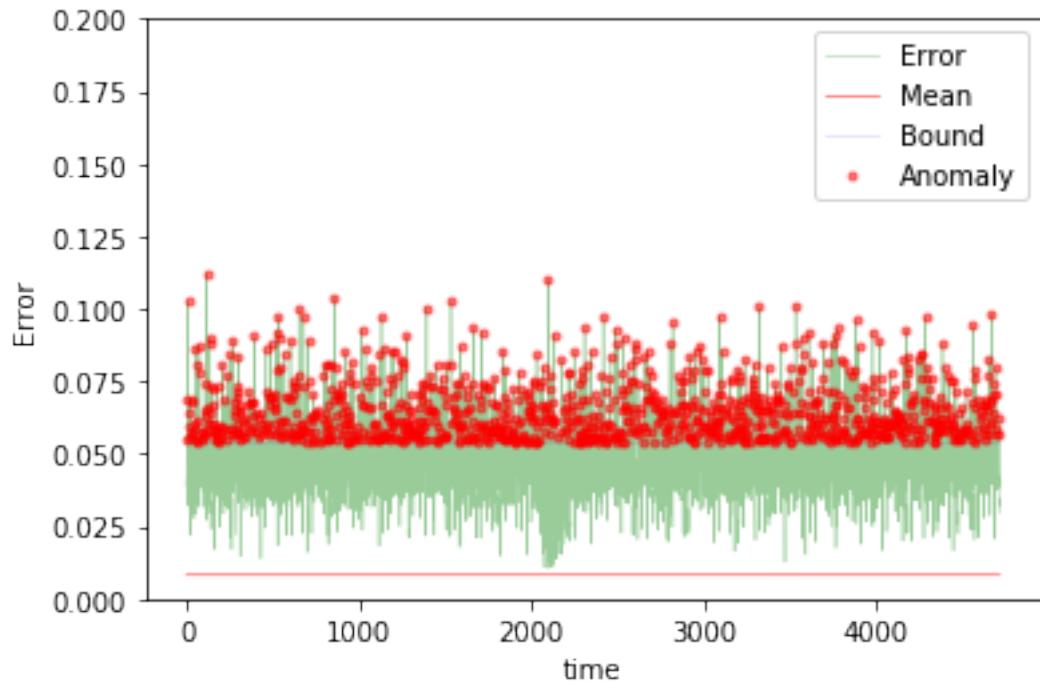




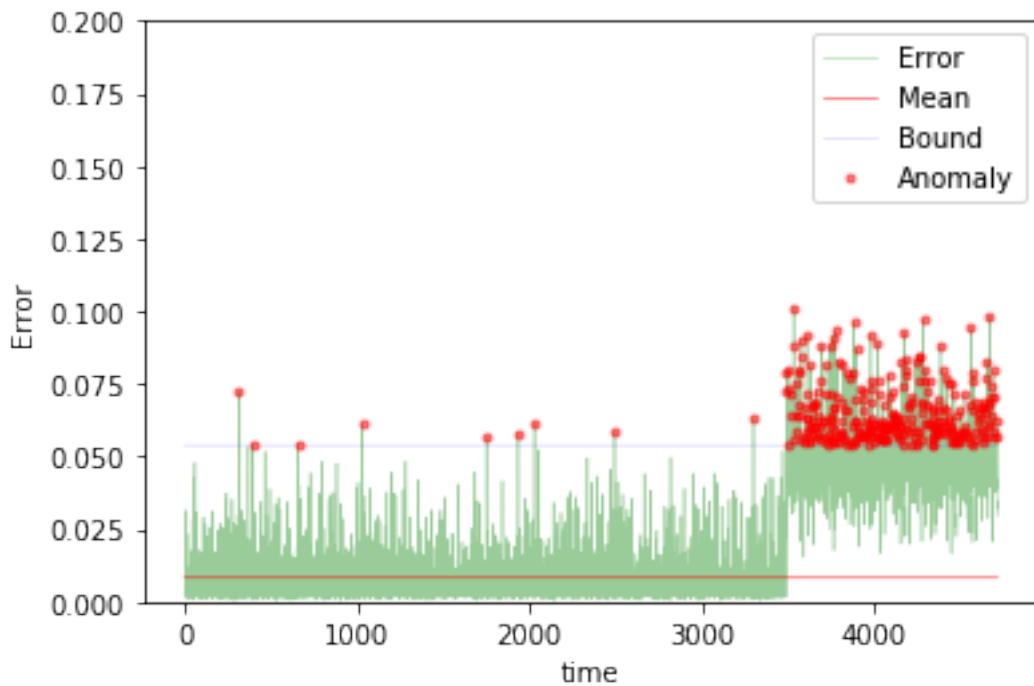
The mean error for nn2_10_normal_ is 0.008279752746813241 for length 4719
Testing on anomaly data.



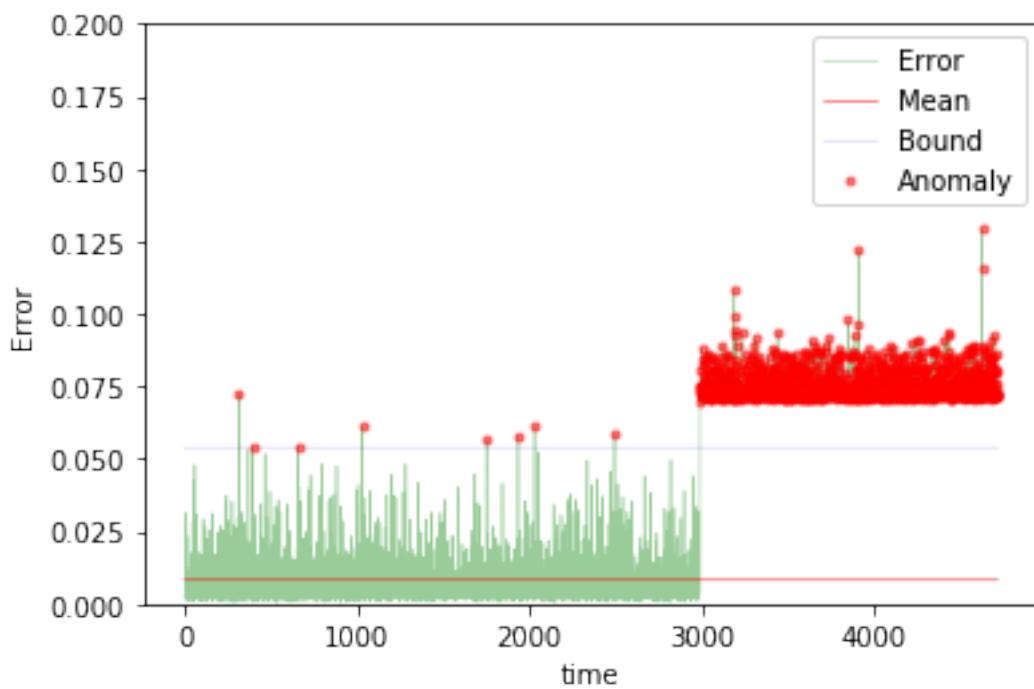
The mean error for nn2_10_anomaly_ is 0.009974929146334095 for length 4719
Testing on different app data.



The mean error for nn2_10_diff_app_ is 0.047099055417432166 for length 4719
Testing on App change synthetic data.



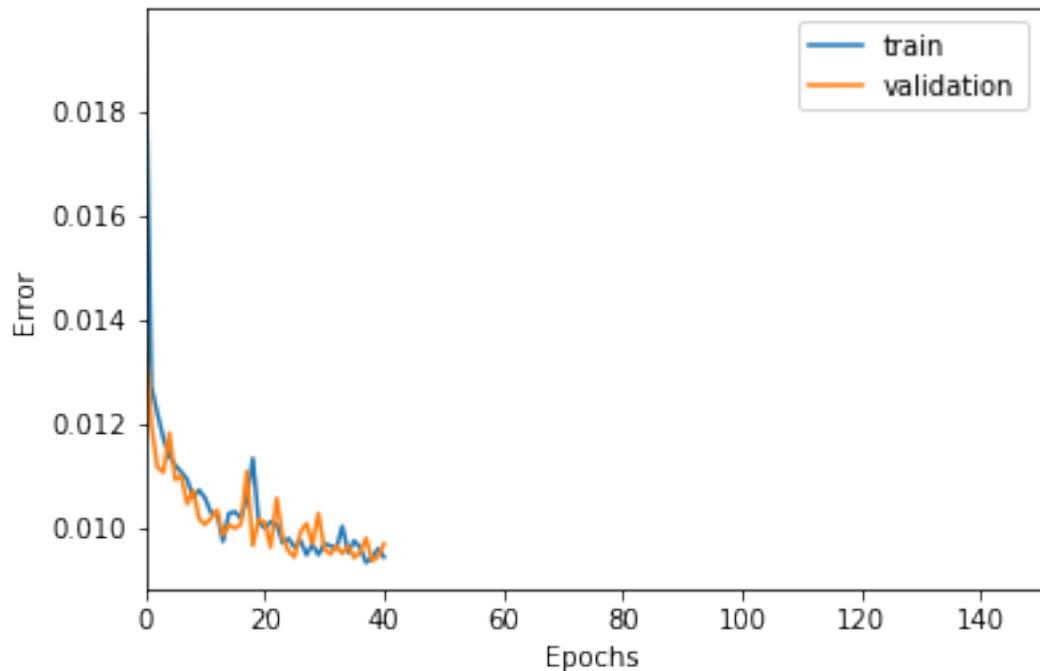
The mean error for nn2_10_app_change_ is 0.018451027013878282 for length 4719
Testing on Net flood synthetic data.



```
The mean error for nn2_10_net_flood_ is 0.032788656535287106 for length 4719  
=====
```

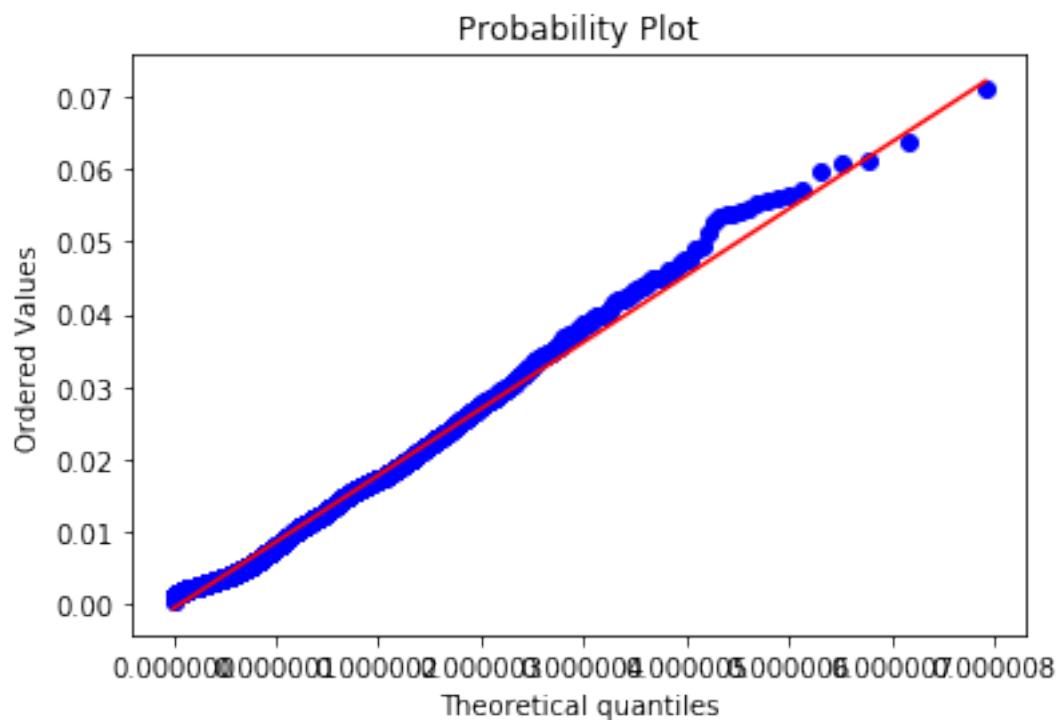
20 steps

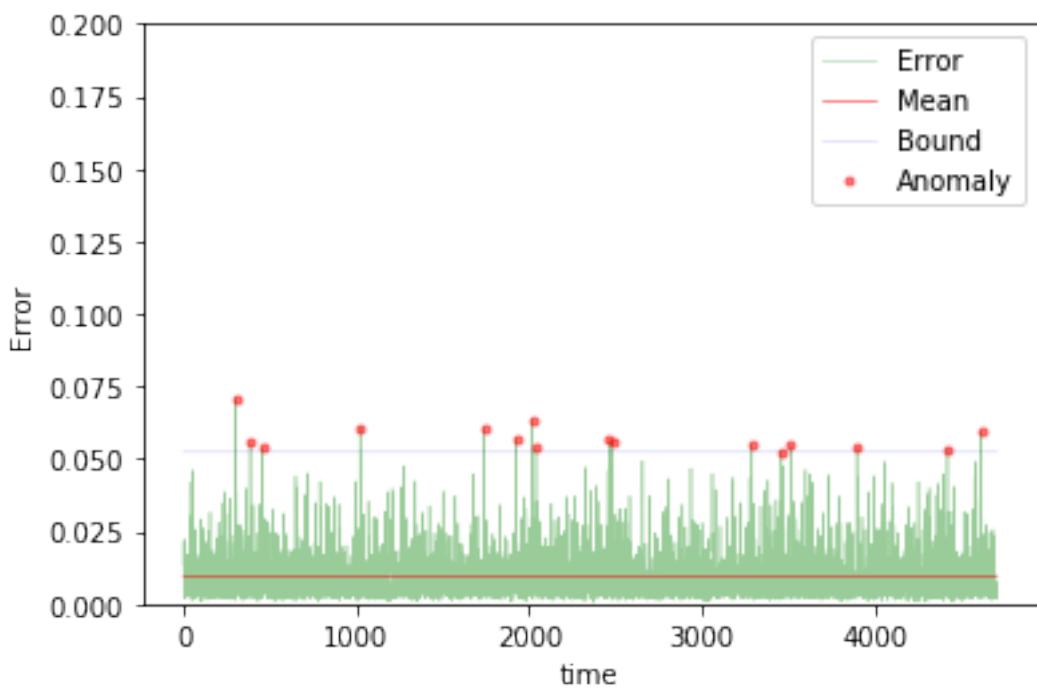
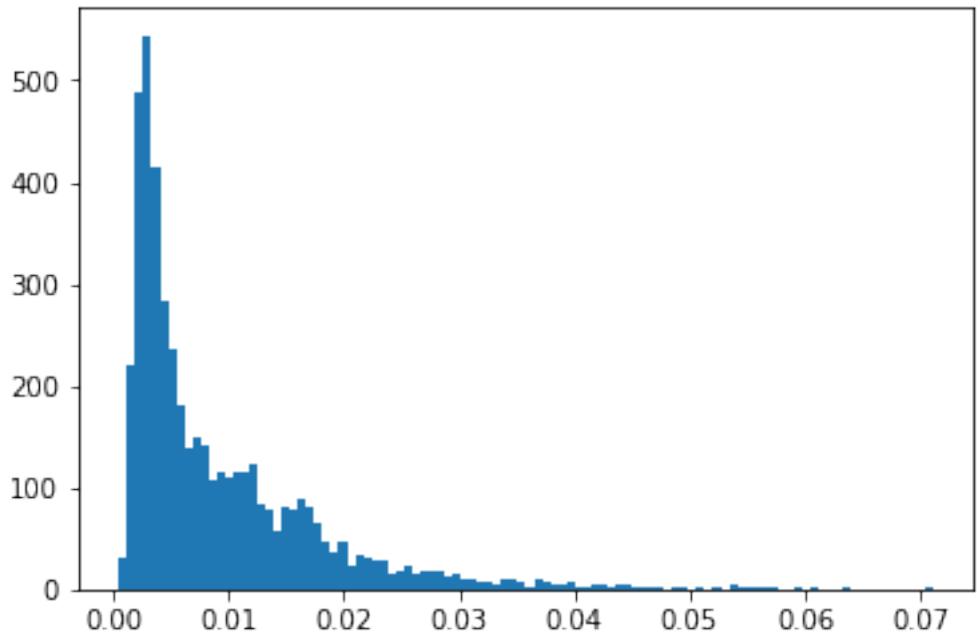
```
In [104]: TIMESTEPS = 20  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS)  
vgen = flat_generator(val_X, TIMESTEPS)  
name = "nn2_20"  
  
In [105]: input_layer = Input(shape=(TIMESTEPS*DIM,))  
hidden = Dense(500, activation='relu')(input_layer)  
hidden = Dense(100, activation='relu')(hidden)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [106]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [107]: train(model, tgen, vgen, name=name)  
test(model, name=name, window=TIMESTEPS)
```



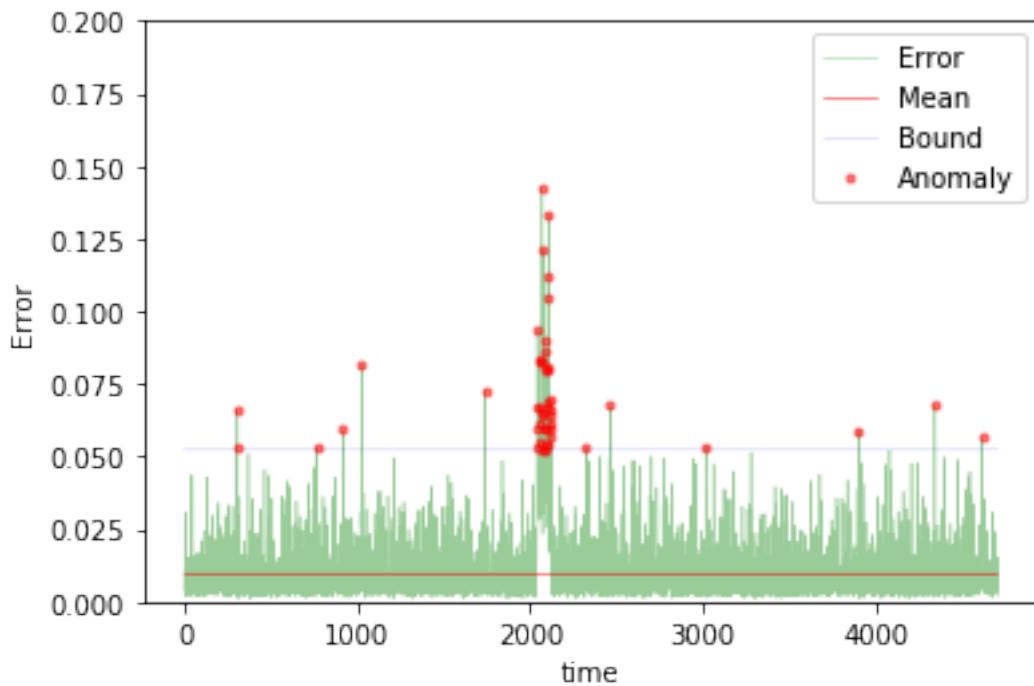
```
Training loss for final epoch is 0.00942148942372296
Validation loss for final epoch is 0.0096727455095388
----- Beginning tests for nn2_20 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

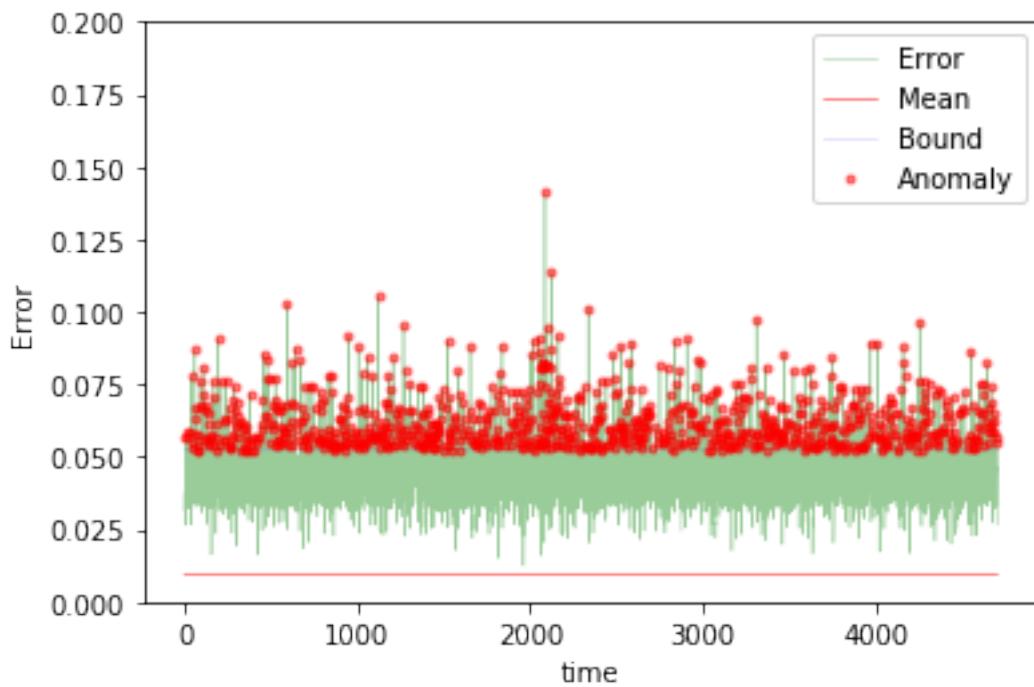




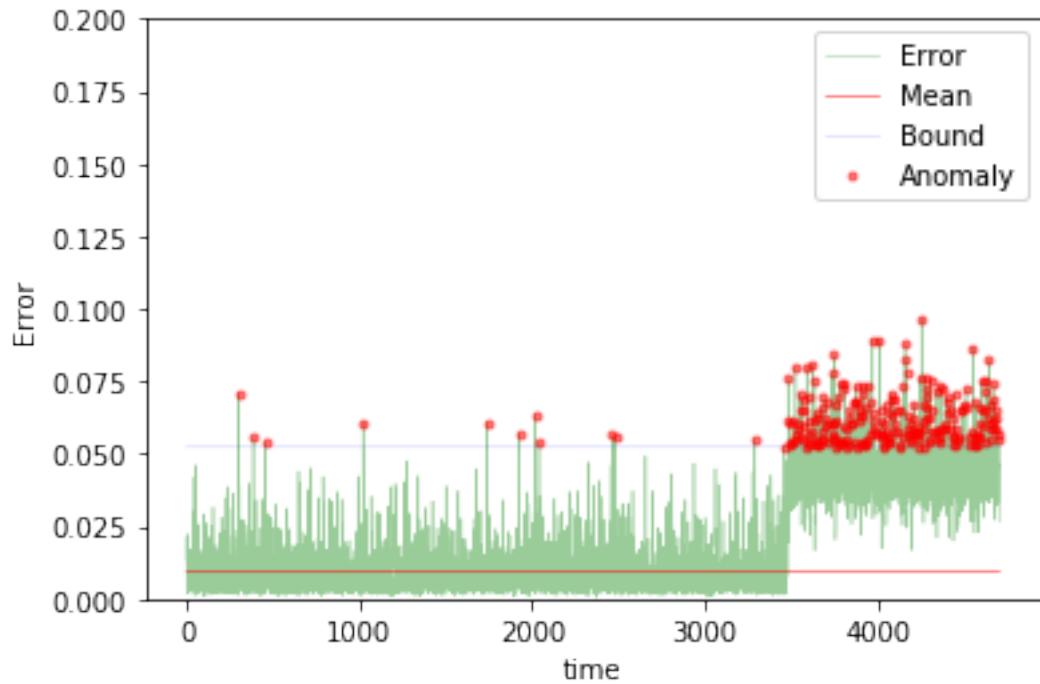
The mean error for nn2_20_normal_ is 0.009294964793695654 for length 4709
Testing on anomaly data.



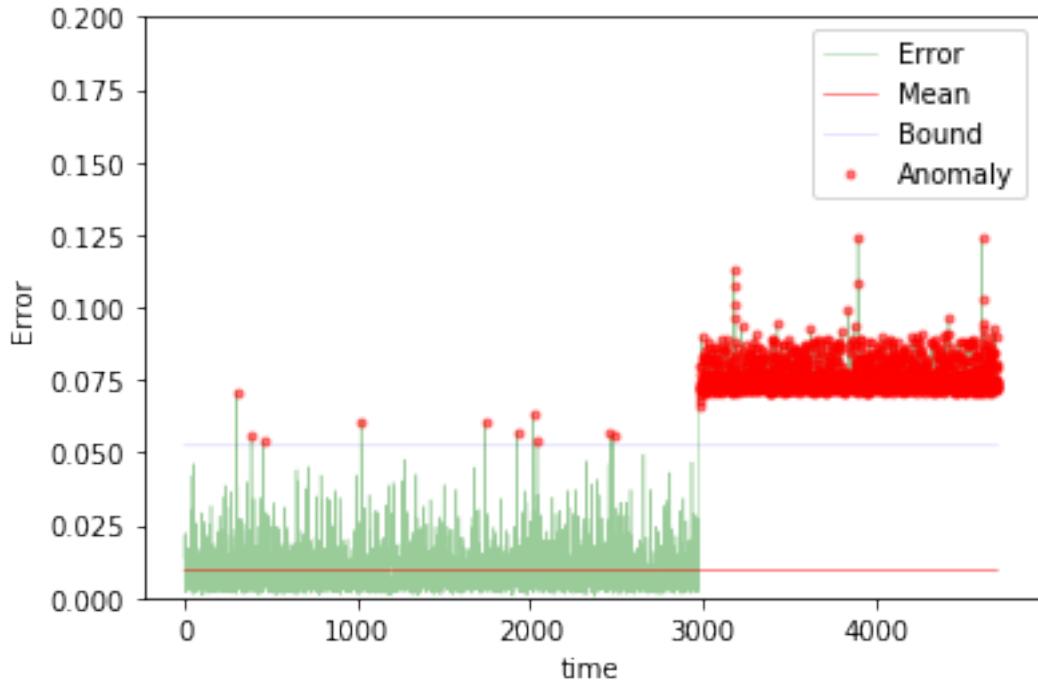
The mean error for nn2_20_anomaly_ is 0.010569869342364122 for length 4709
Testing on different app data.



The mean error for nn2_20_diff_app_ is 0.043839995523119006 for length 4709
Testing on App change synthetic data.



The mean error for nn2_20_app_change_ is 0.01820904889937924 for length 4709
Testing on Net flood synthetic data.



```
The mean error for nn2_20_net_flood_ is 0.03384518606434765 for length 4709
=====
```

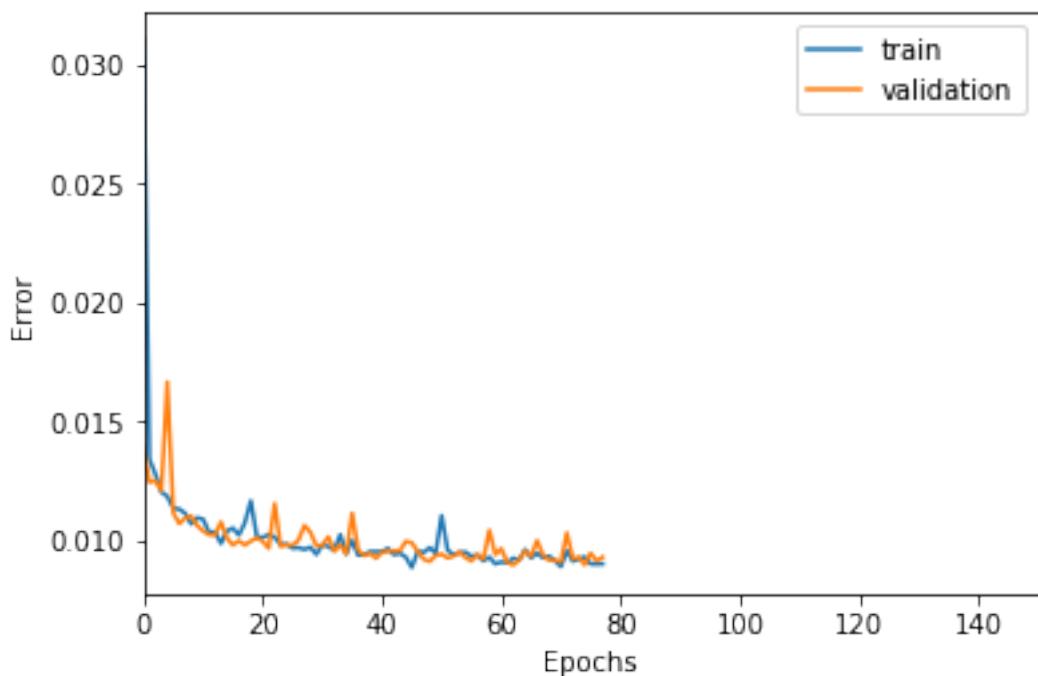
50 steps

```
In [108]: TIMESTEPS = 50
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS)
          vgen = flat_generator(val_X, TIMESTEPS)
          name = "nn2_50"

In [109]: 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)

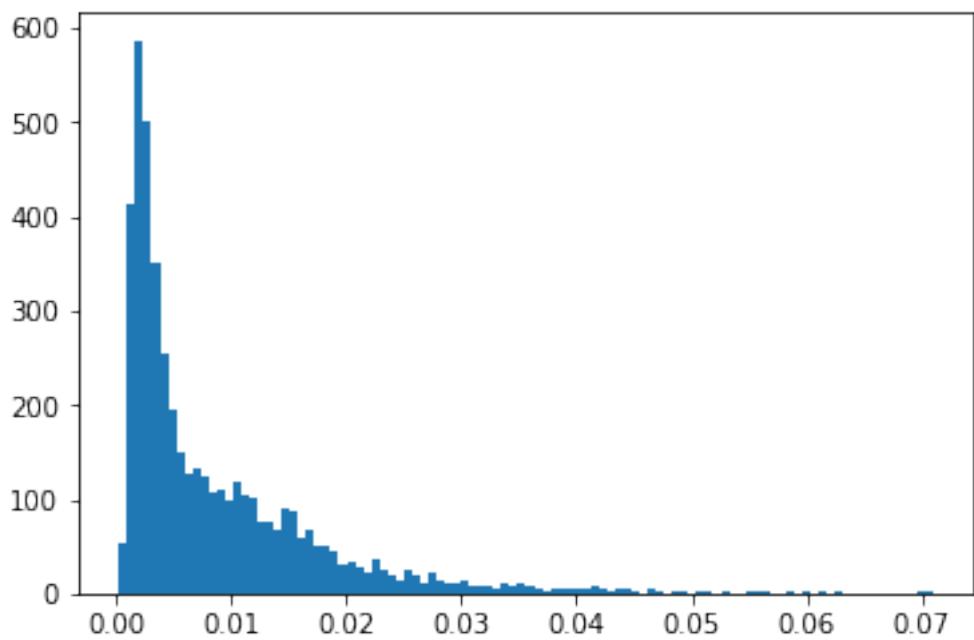
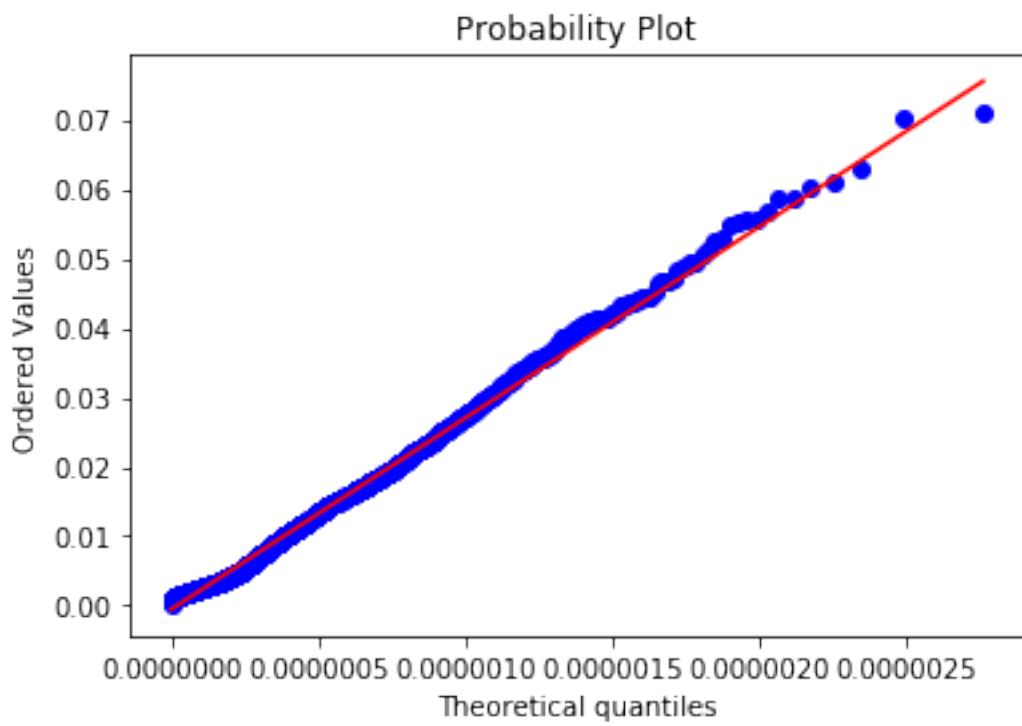
In [110]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

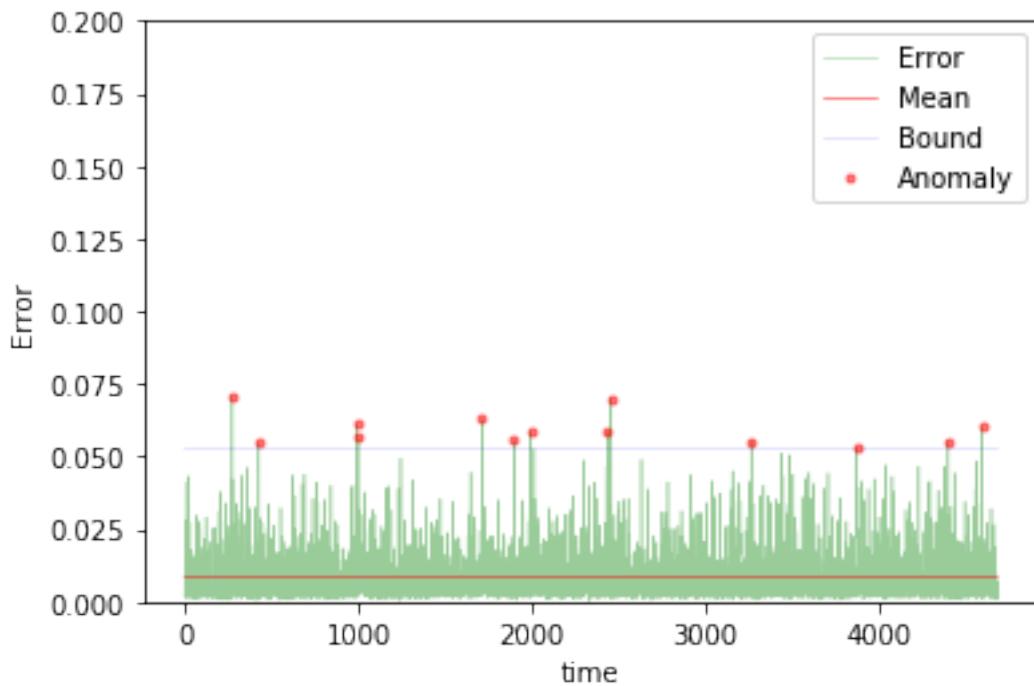
In [111]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```



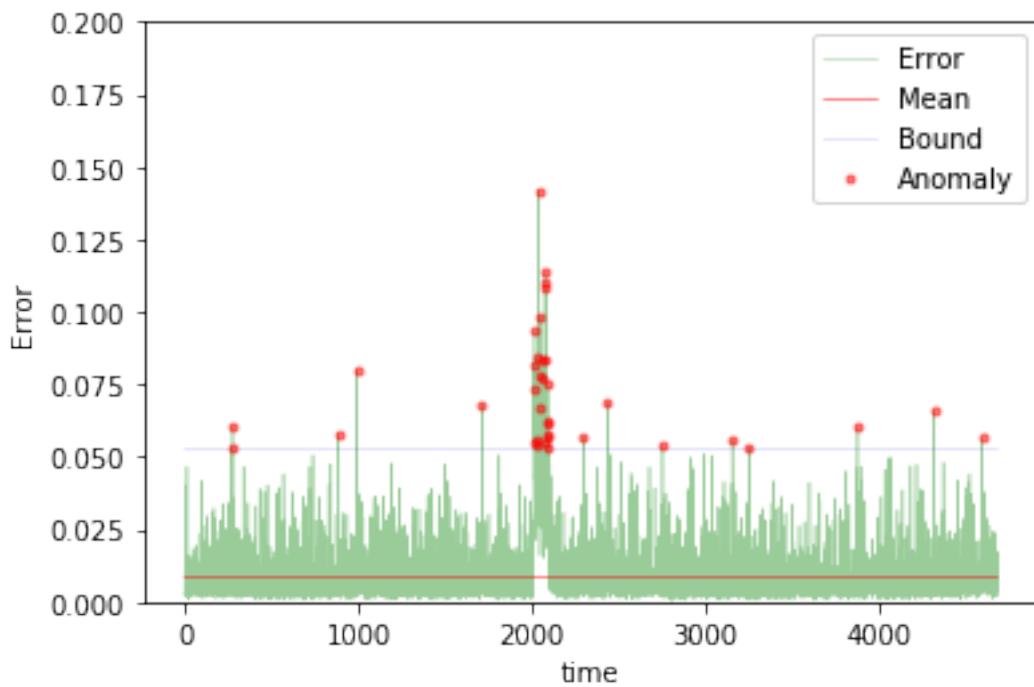
```
Training loss for final epoch is 0.00904349858139176
Validation loss for final epoch is 0.009326373887713998
----- Beginning tests for nn2_50 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

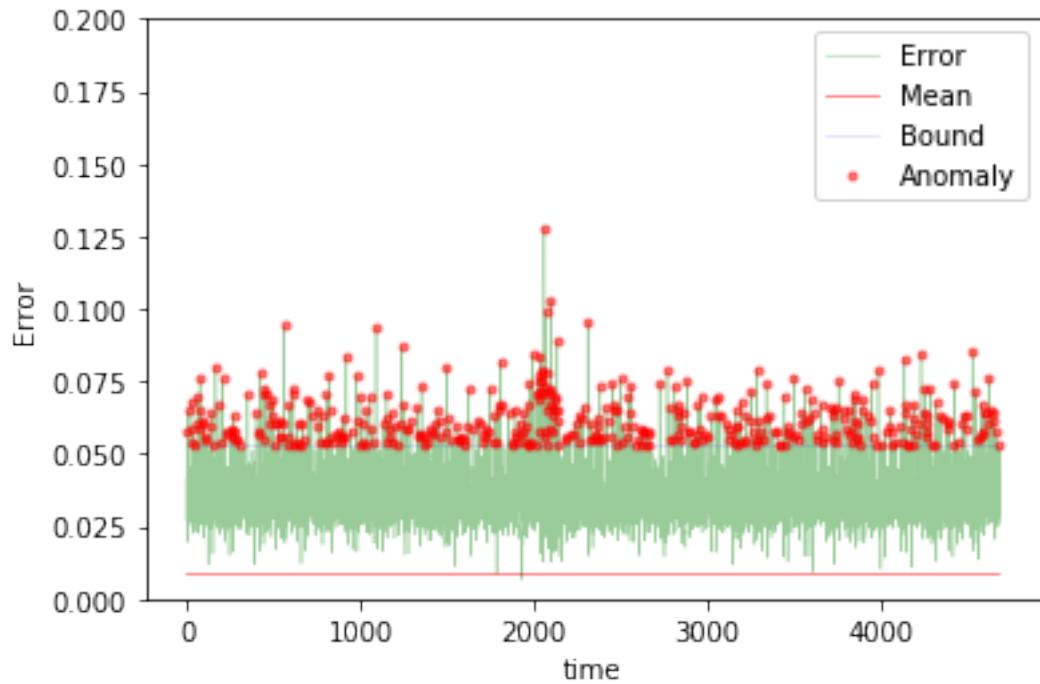




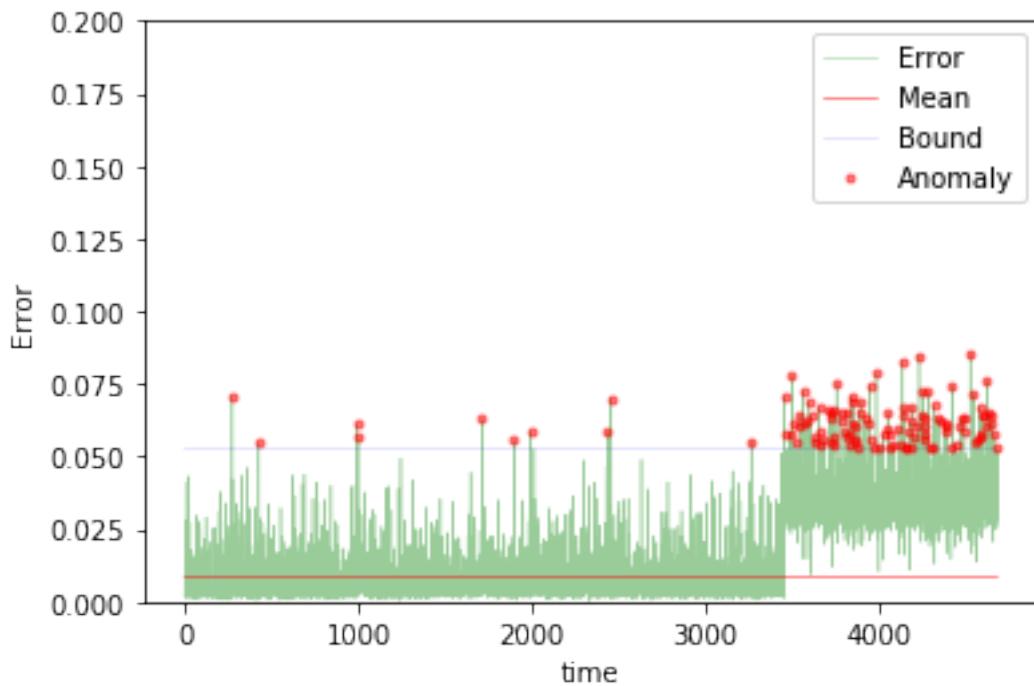
The mean error for nn2_50_normal_ is 0.008713309442990052 for length 4679
Testing on anomaly data.



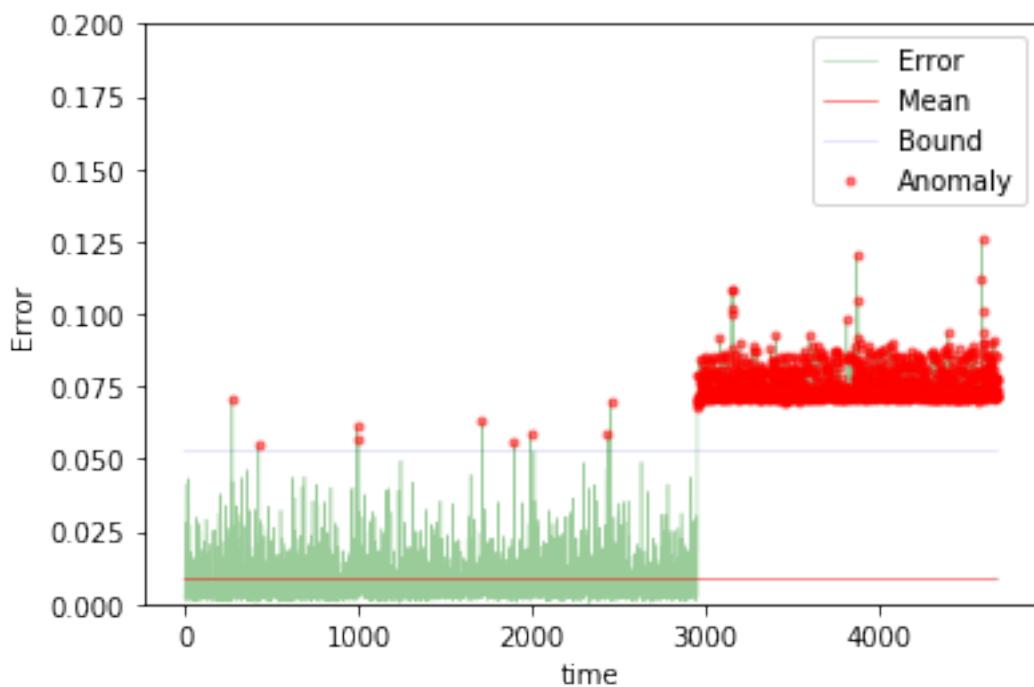
The mean error for nn2_50_anomaly_ is 0.010112070644929012 for length 4679
Testing on different app data.



The mean error for nn2_50_diff_app_ is 0.03601312358574939 for length 4679
Testing on App change synthetic data.



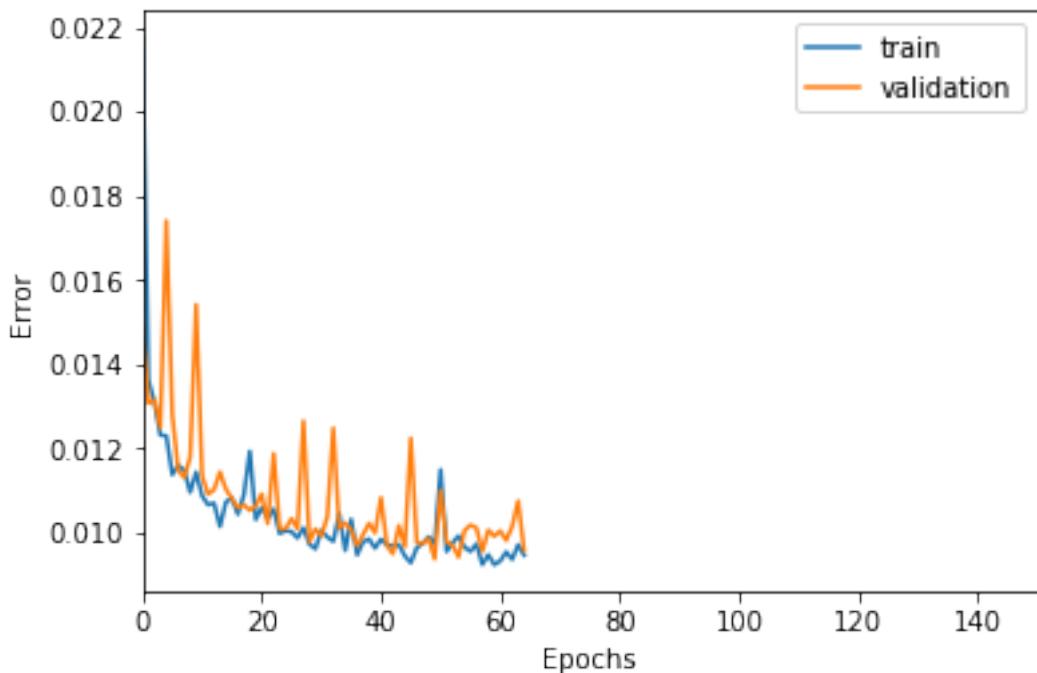
The mean error for nn2_50_app_change_ is 0.0158088237072686 for length 4679
Testing on Net flood synthetic data.



```
The mean error for nn2_50_net_flood_ is 0.03320967037378566 for length 4679  
=====
```

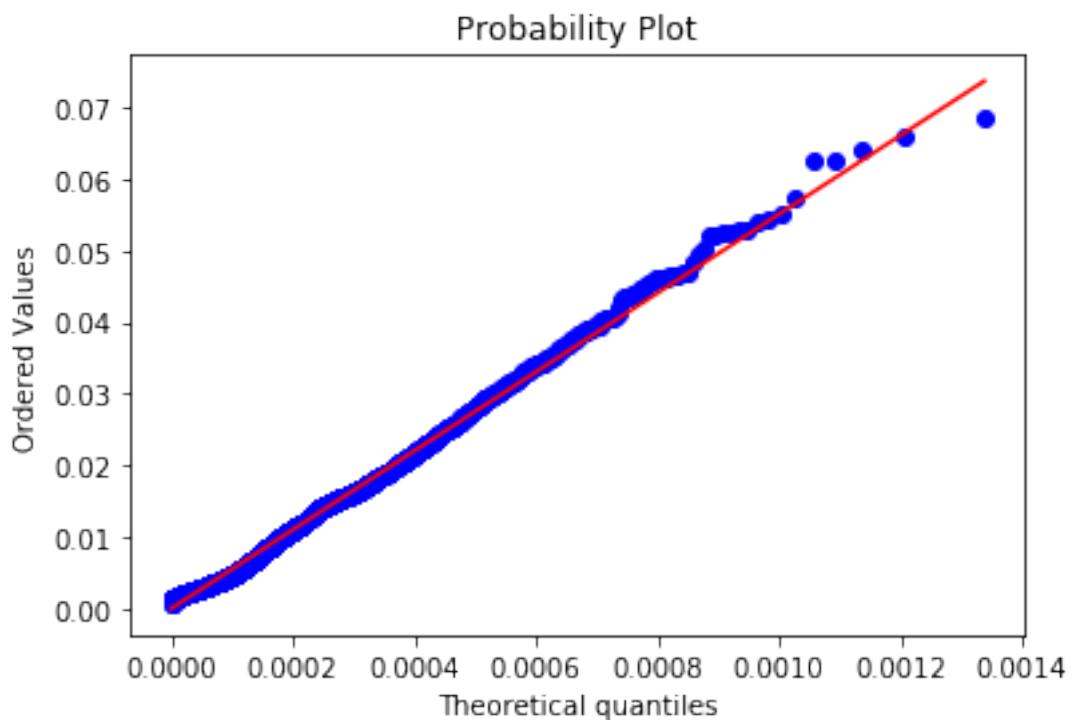
100 steps

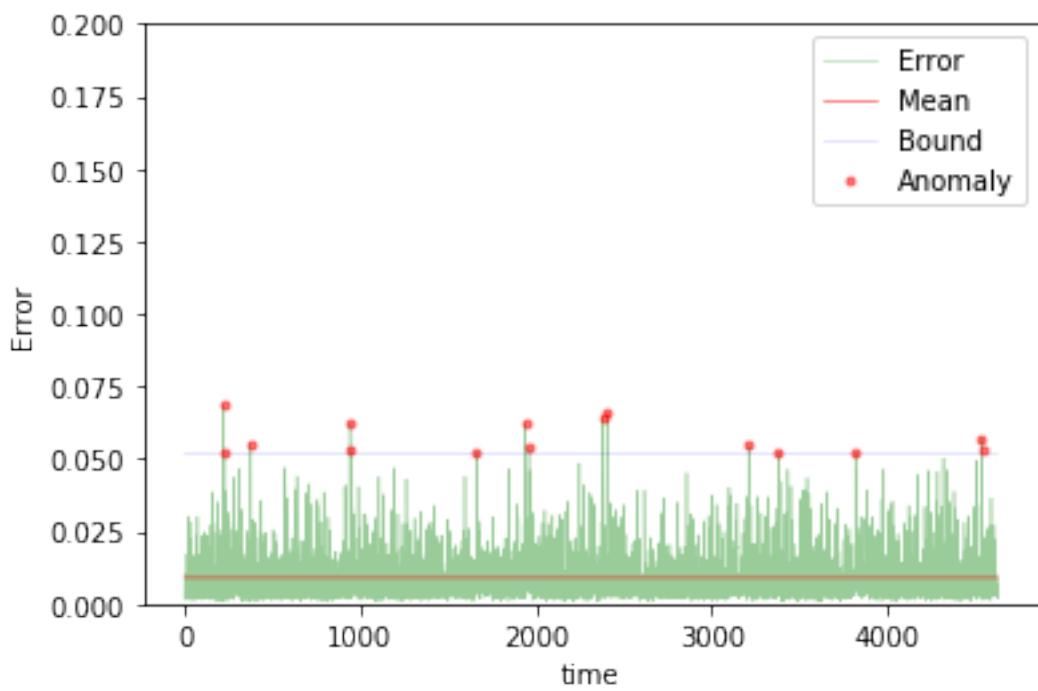
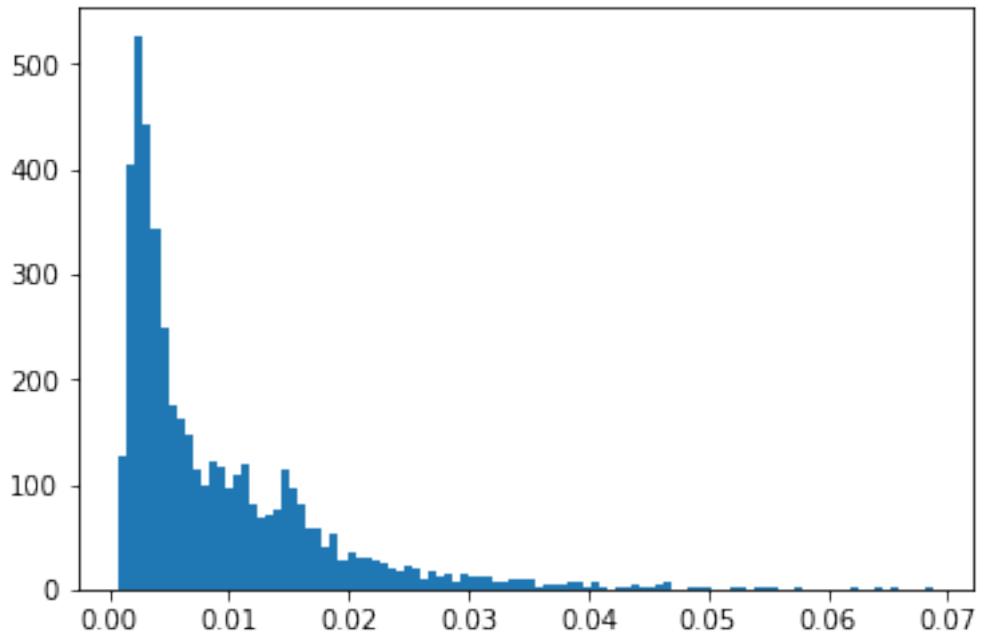
```
In [112]: TIMESTEPS = 100  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS)  
vgen = flat_generator(val_X, TIMESTEPS)  
name = "nn2_100"  
  
In [113]: input_layer = Input(shape=(TIMESTEPS*DIM,))  
hidden = Dense(500, activation='relu')(input_layer)  
hidden = Dense(100, activation='relu')(hidden)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [114]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [115]: train(model, tgen, vgen, name=name)  
test(model, name=name, window=TIMESTEPS)
```



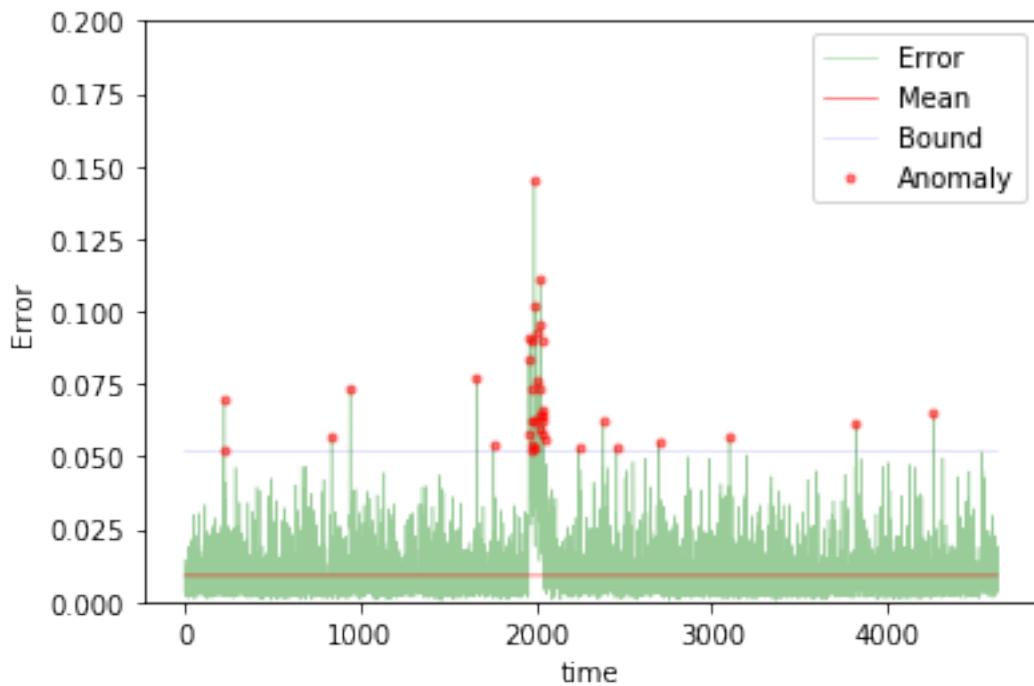
```
Training loss for final epoch is 0.009472169831278734
Validation loss for final epoch is 0.009582036448526196
----- Beginning tests for nn2_100 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

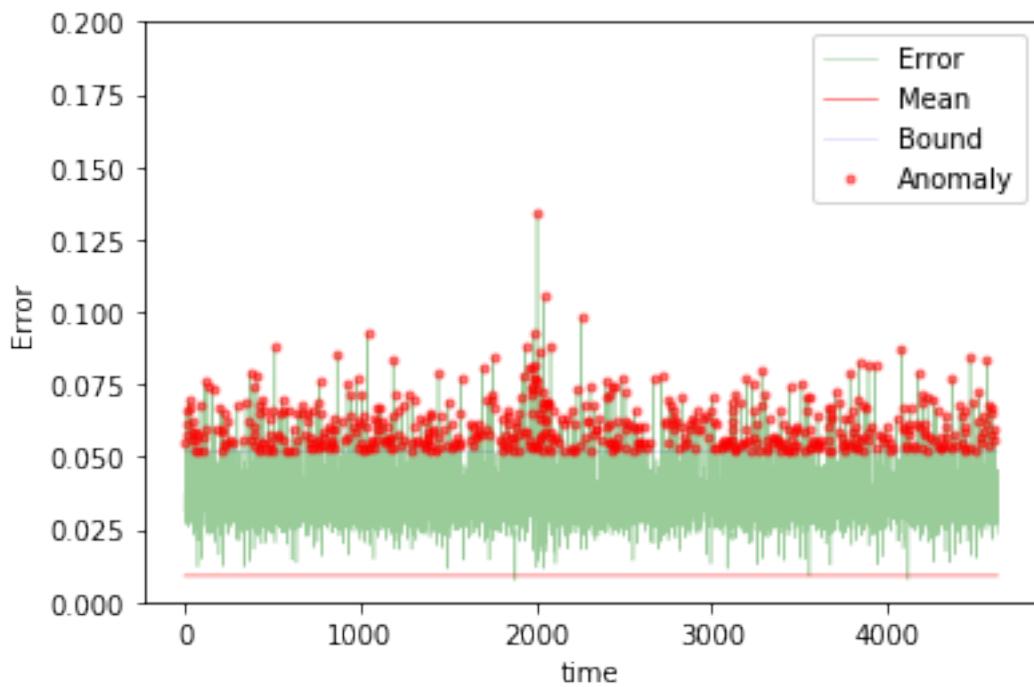




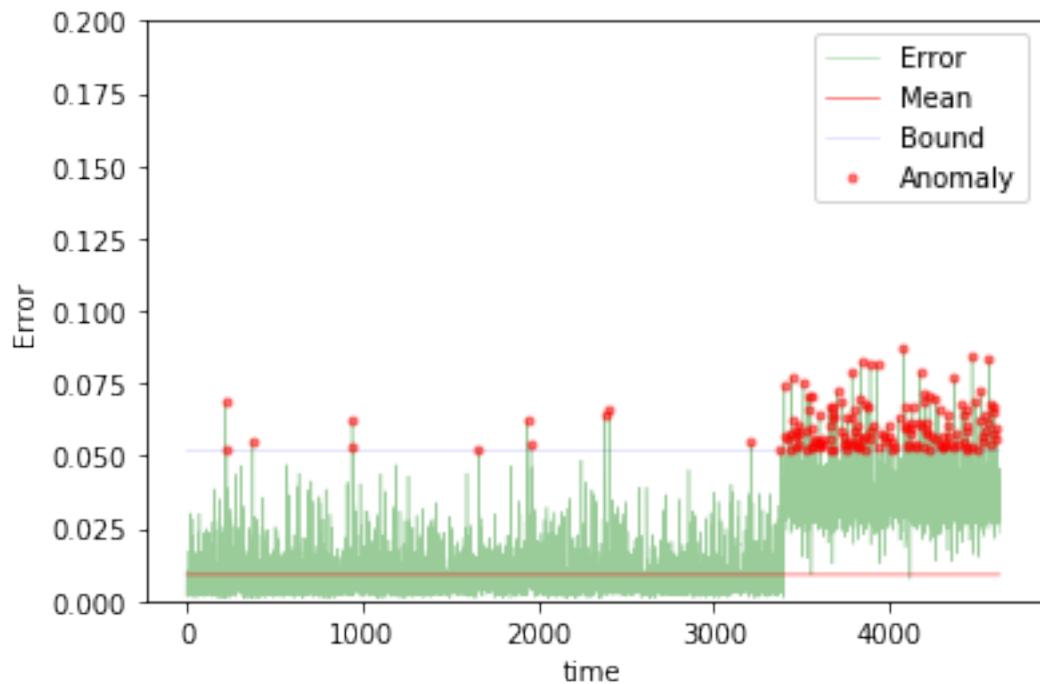
The mean error for nn2_100_normal_ is 0.009075670634473721 for length 4629
Testing on anomaly data.



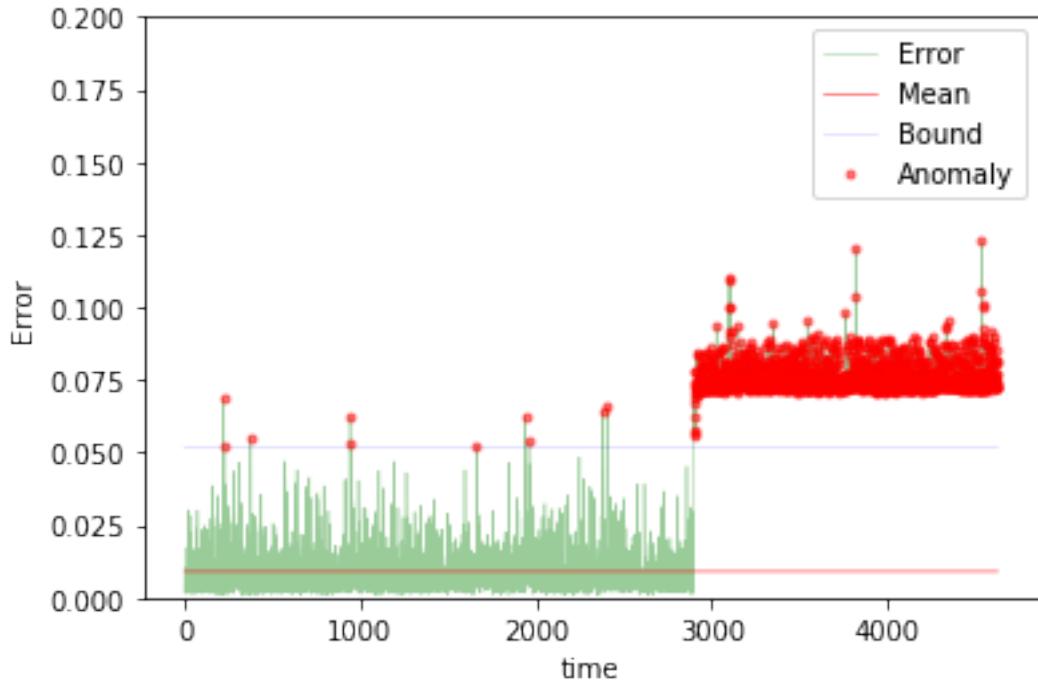
The mean error for nn2_100_anomaly_ is 0.01037805618670891 for length 4629
Testing on different app data.



The mean error for nn2_100_diff_app_ is 0.03721779011405672 for length 4629
Testing on App change synthetic data.



The mean error for nn2_100_app_change_ is 0.016467125552377438 for length 4629
Testing on Net flood synthetic data.



```
The mean error for nn2_100_net_flood_ is 0.03412920049712802 for length 4629
=====
```

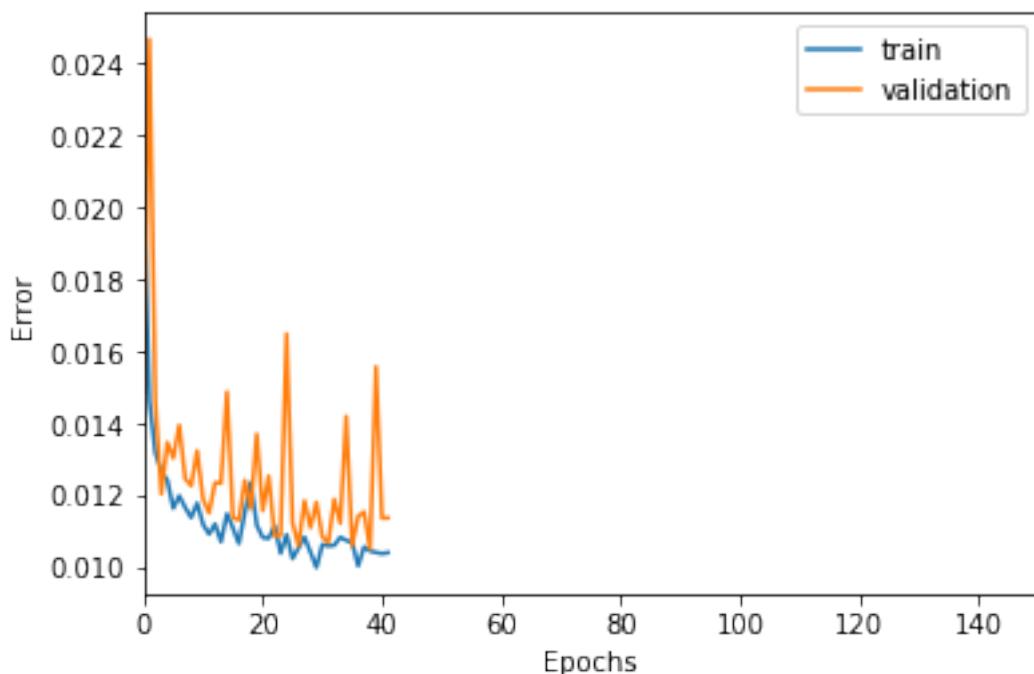
200 steps

```
In [116]: TIMESTEPS = 200
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS)
          vgen = flat_generator(val_X, TIMESTEPS)
          name = "nn2_200"

In [117]: 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)

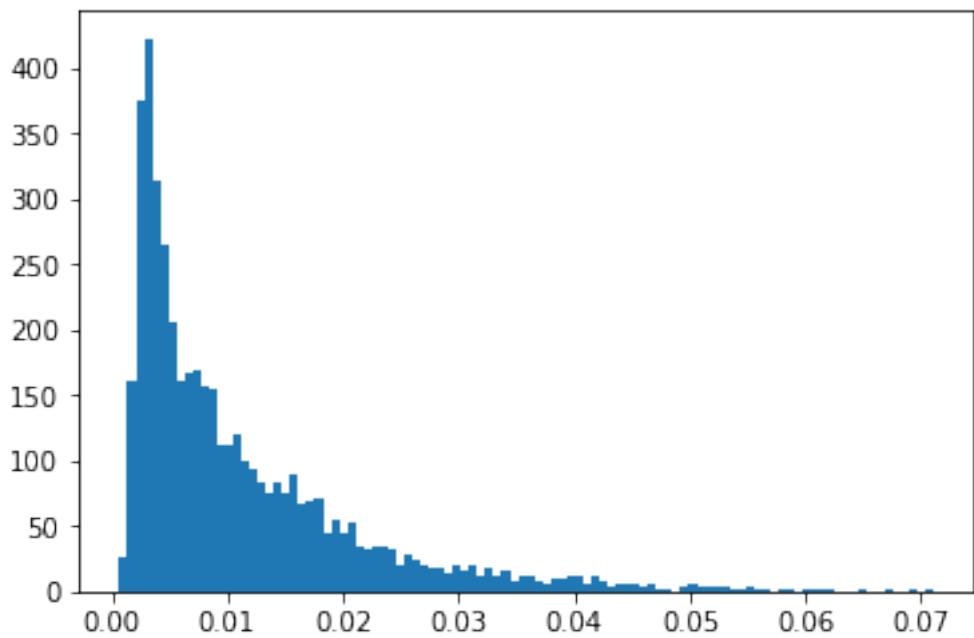
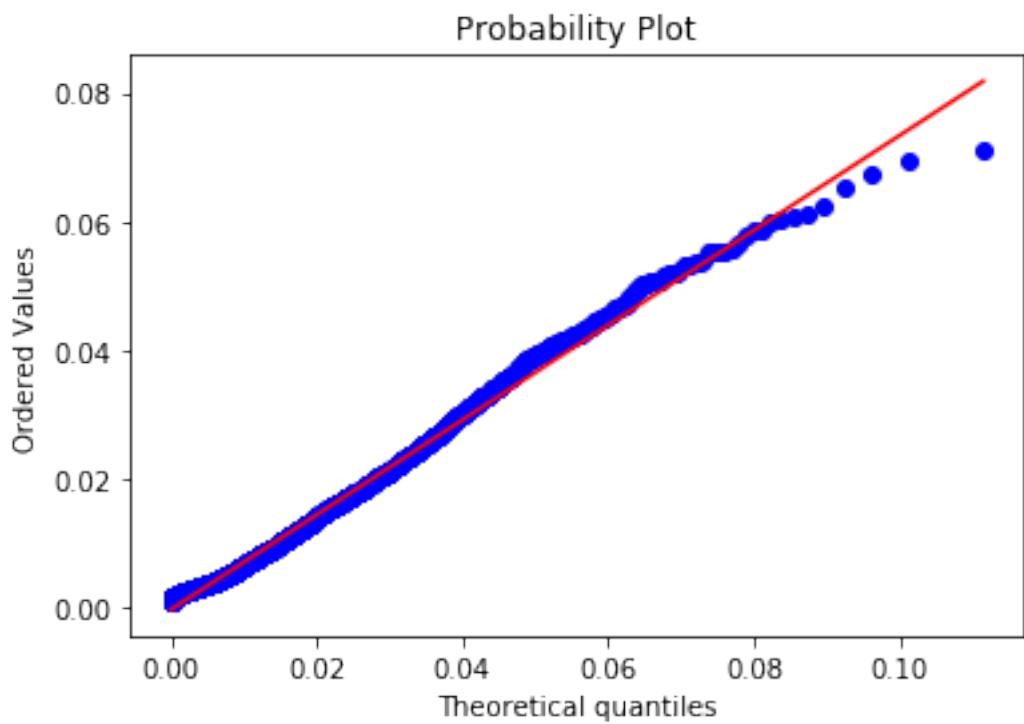
In [118]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

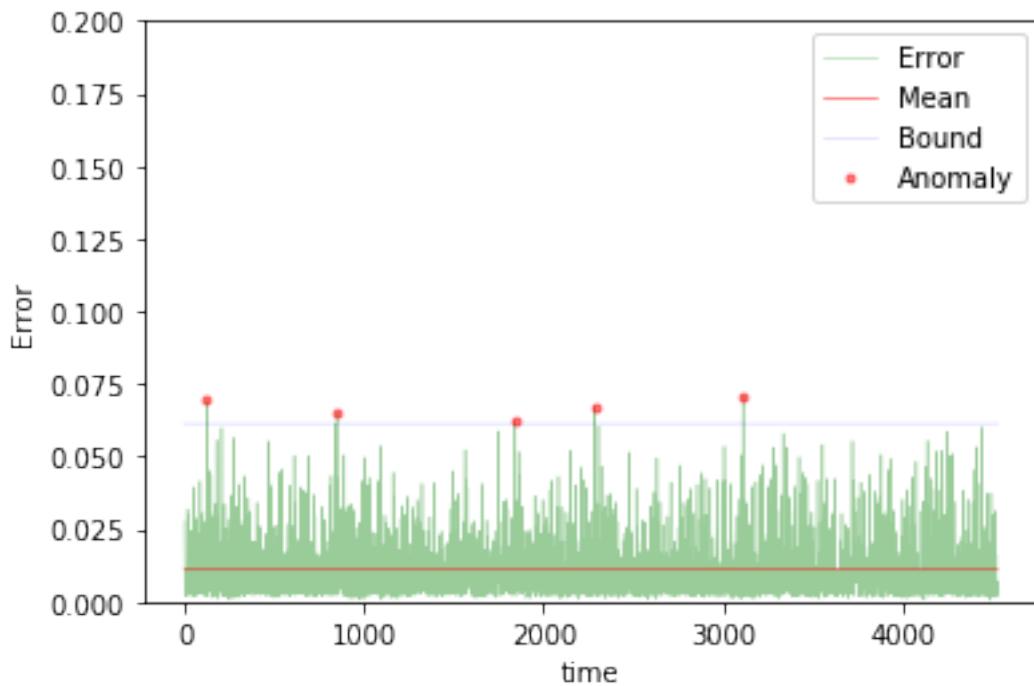
In [119]: train(model, tgen, vgen, name=name)
          test(model, name=name, window=TIMESTEPS)
```



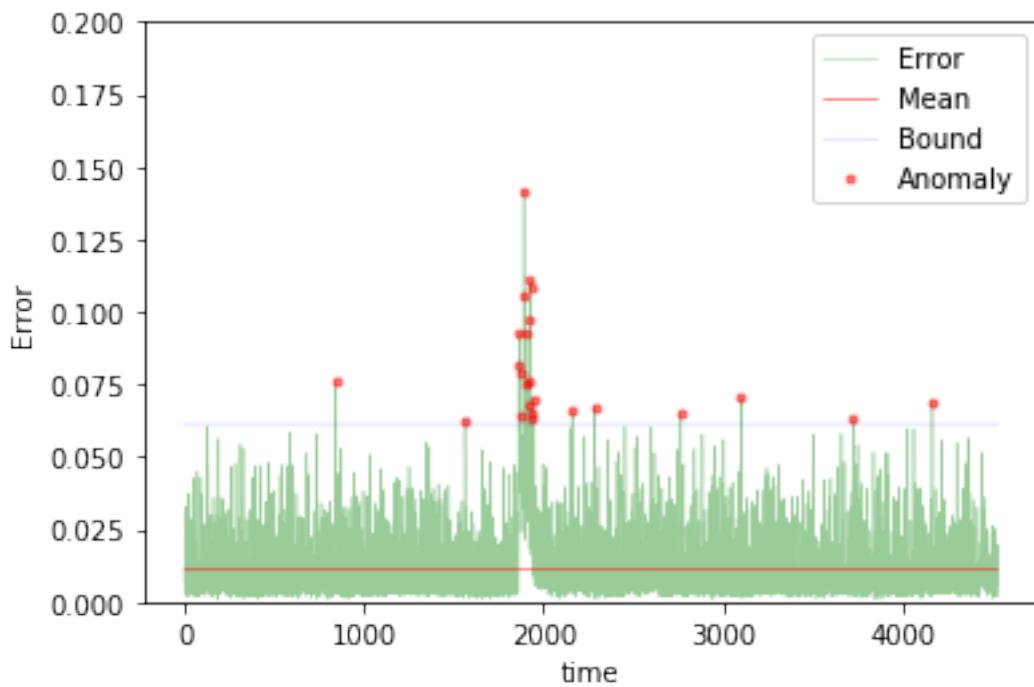
```
Training loss for final epoch is 0.010424807147239335
Validation loss for final epoch is 0.01139011222589761
----- Beginning tests for nn2_200 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
    warnings.warn(msg, RuntimeWarning)
```

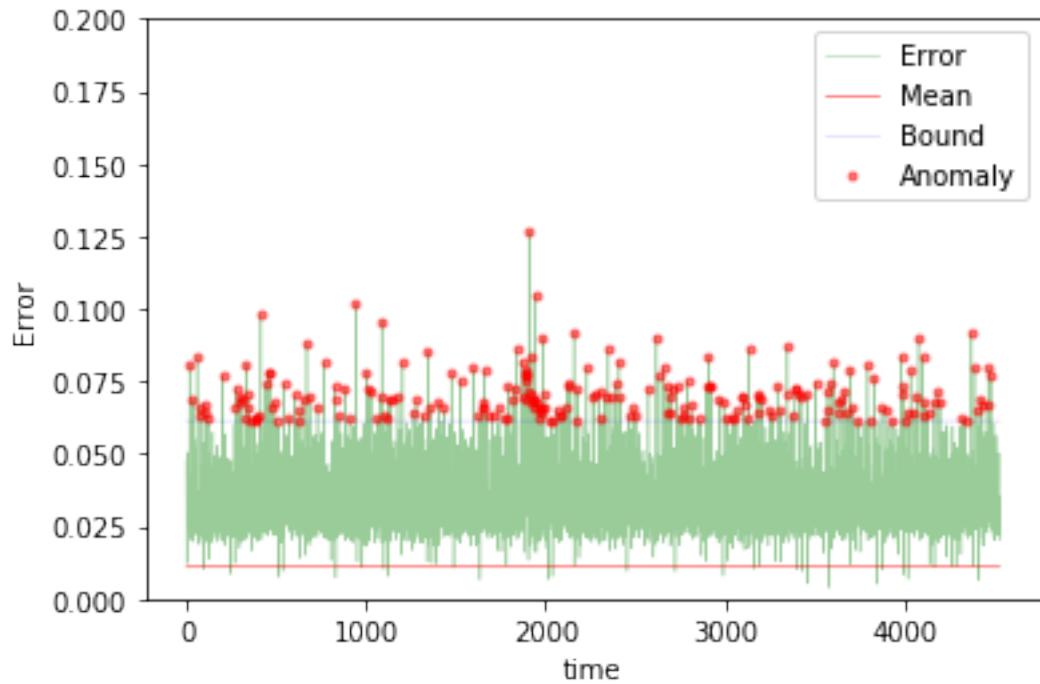




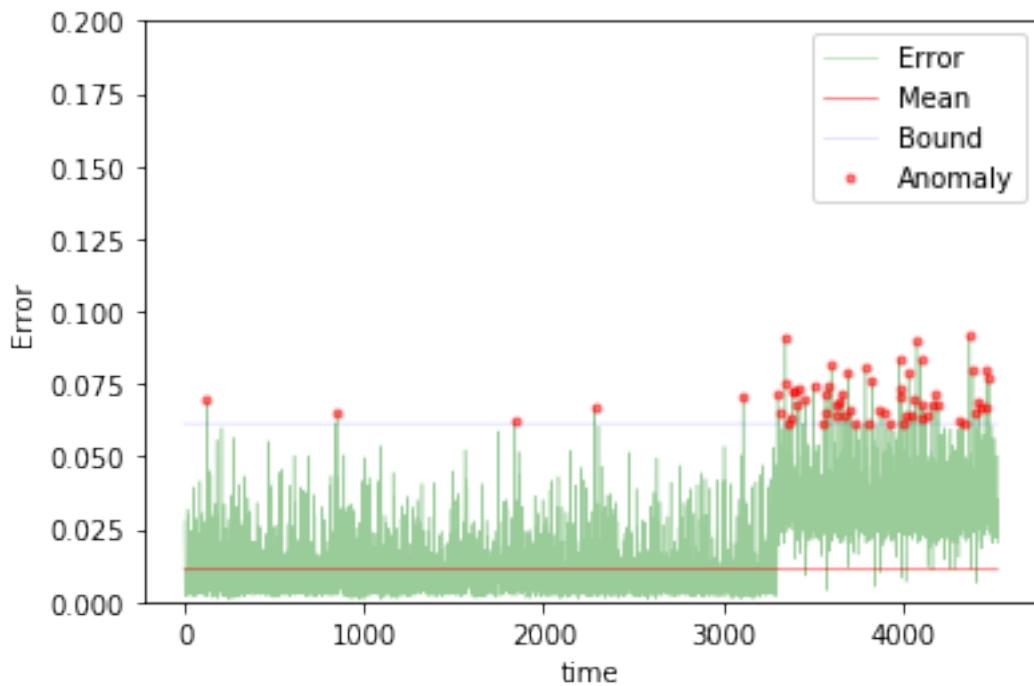
The mean error for nn2_200_normal_ is 0.01105813031832391 for length 4529
Testing on anomaly data.



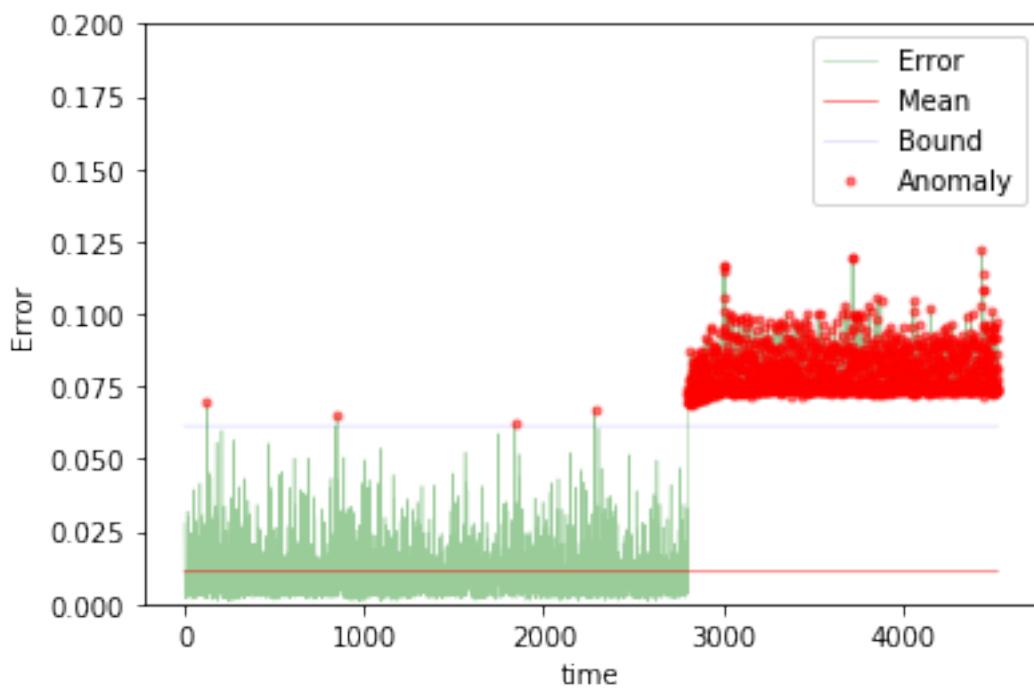
The mean error for nn2_200_anomaly_ is 0.012740844408391787 for length 4529
Testing on different app data.



The mean error for nn2_200_diff_app_ is 0.03482717507961881 for length 4529
Testing on App change synthetic data.



The mean error for nn2_200_app_change_ is 0.017575750117863227 for length 4529
Testing on Net flood synthetic data.

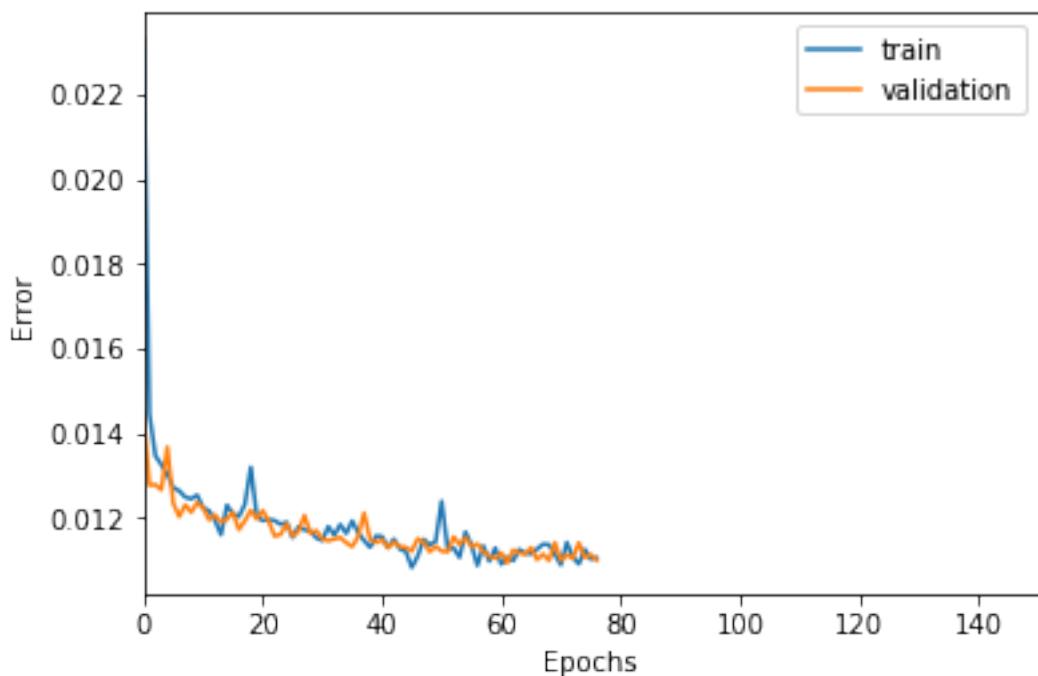


```
The mean error for nn2_200_net_flood_ is 0.037113847123650685 for length 4529  
=====
```

2.1.4 NN with 3 hidden layers

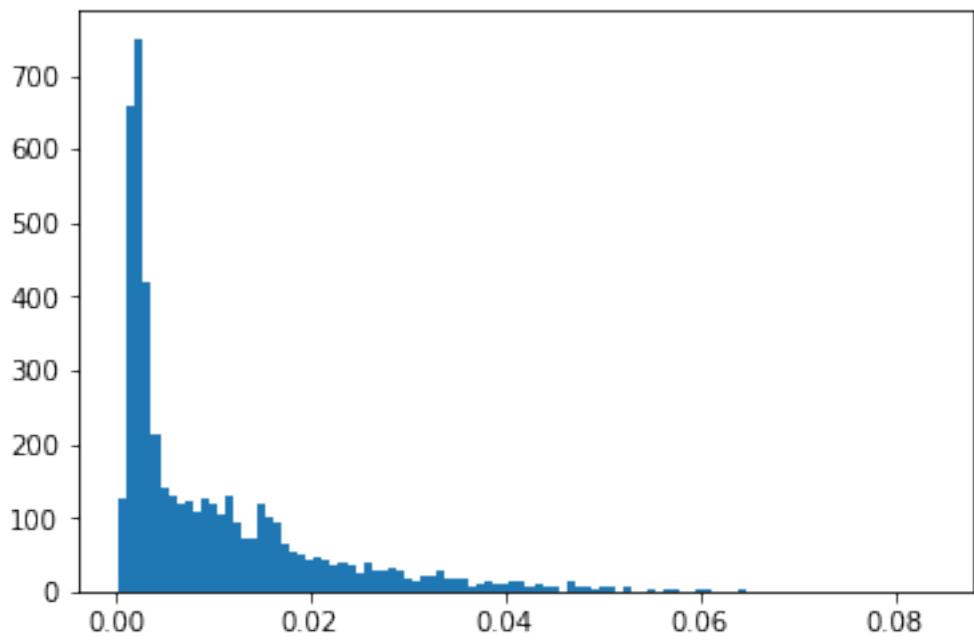
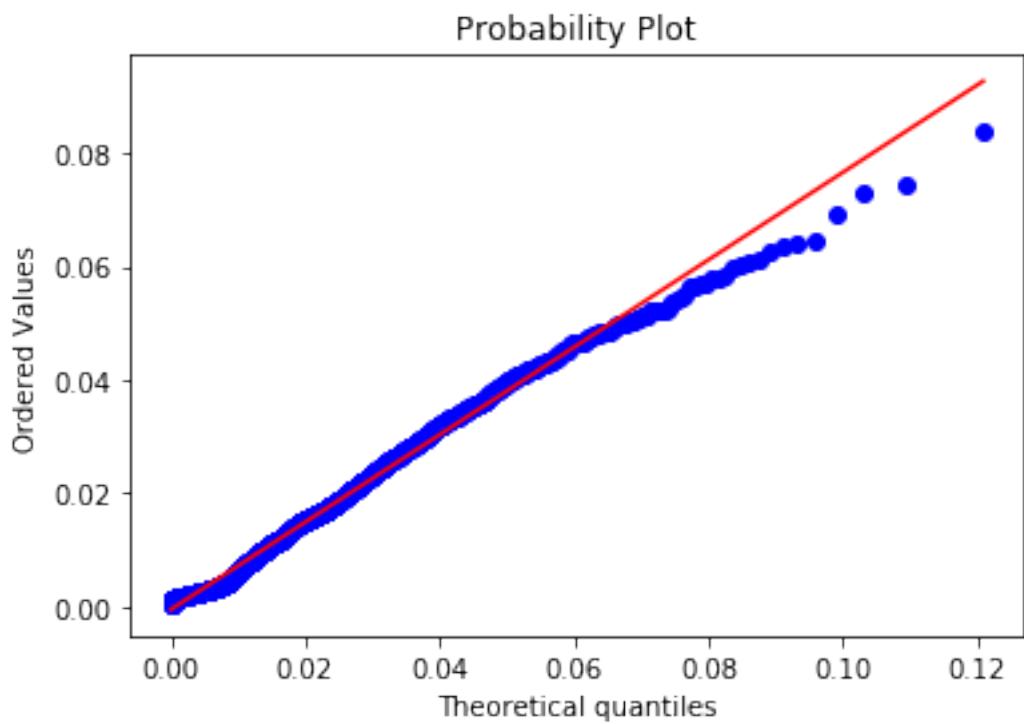
2 steps

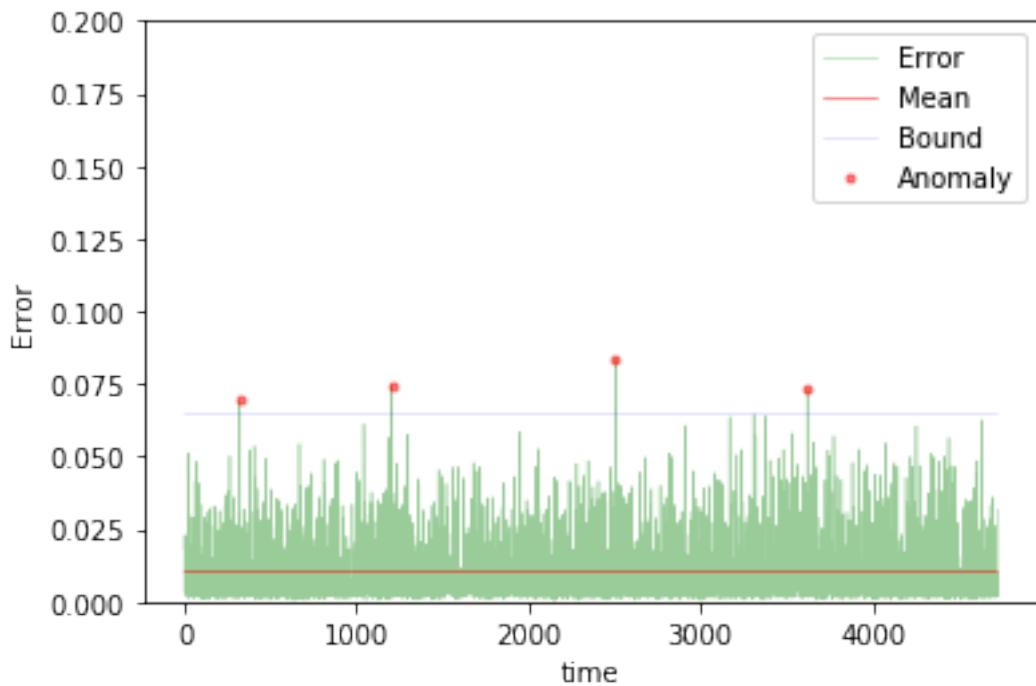
```
In [120]: TIMESTEPS = 2  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS)  
vgen = flat_generator(val_X, TIMESTEPS)  
name = "nn3_2"  
  
In [121]: input_layer = Input(shape=(TIMESTEPS*DIM,))  
hidden = Dense(1000, activation='relu')(input_layer)  
hidden = Dense(500, activation='relu')(hidden)  
hidden = Dense(100, activation='relu')(hidden)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [122]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [123]: train(model, tgen, vgen, name=name)  
test(model, name=name, window=TIMESTEPS)
```



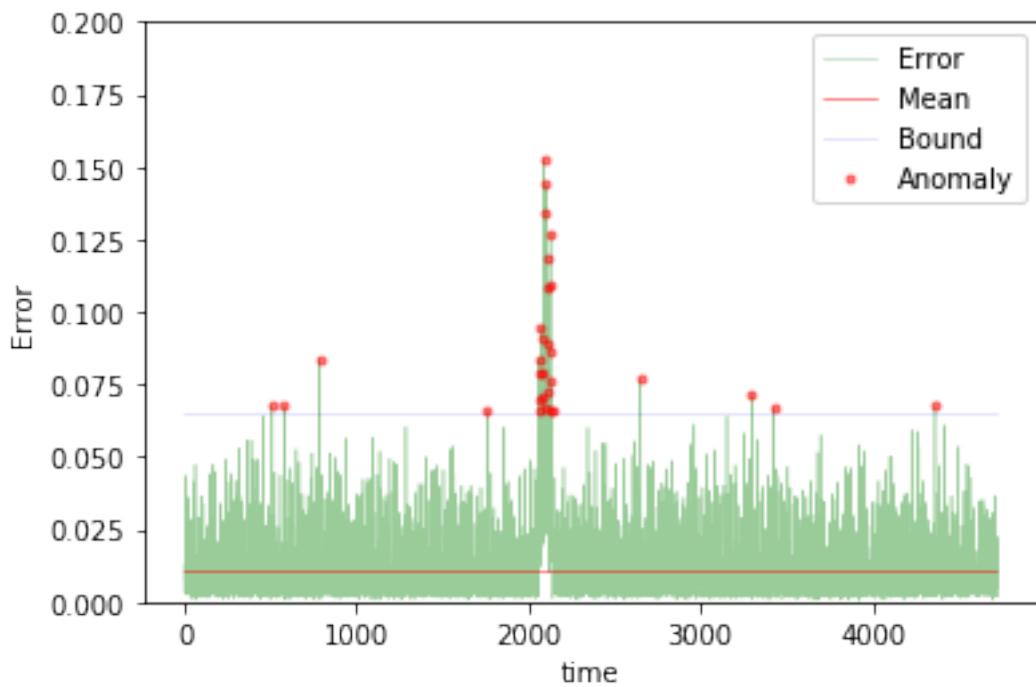
```
Training loss for final epoch is 0.011059373940341174
Validation loss for final epoch is 0.011006242301897146
----- Beginning tests for nn3_2 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
    warnings.warn(msg, RuntimeWarning)
```

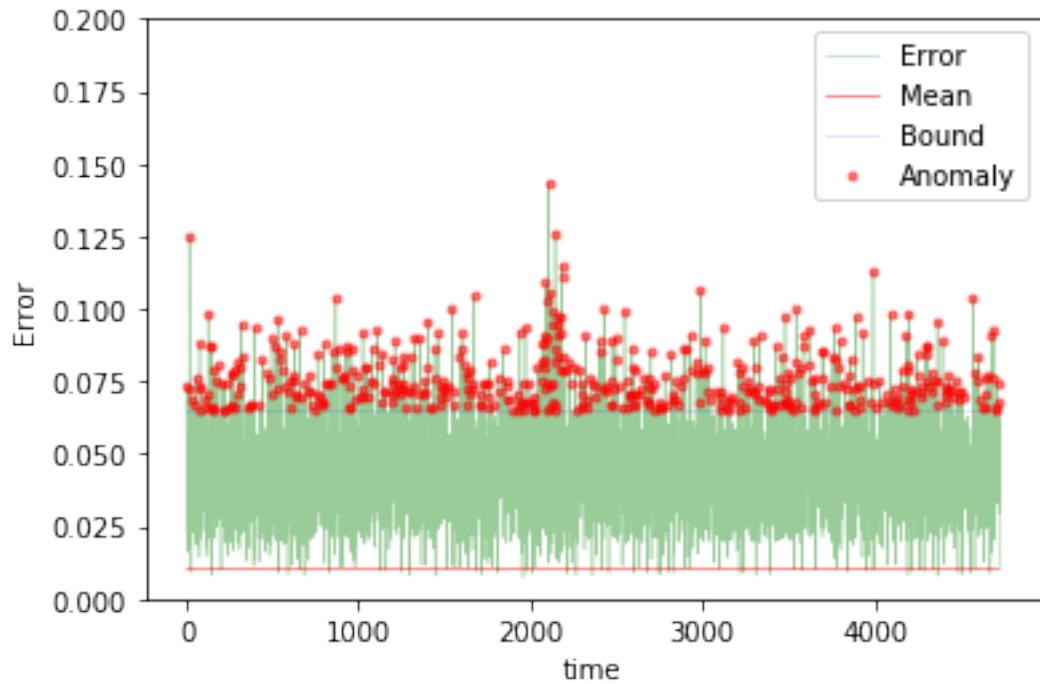




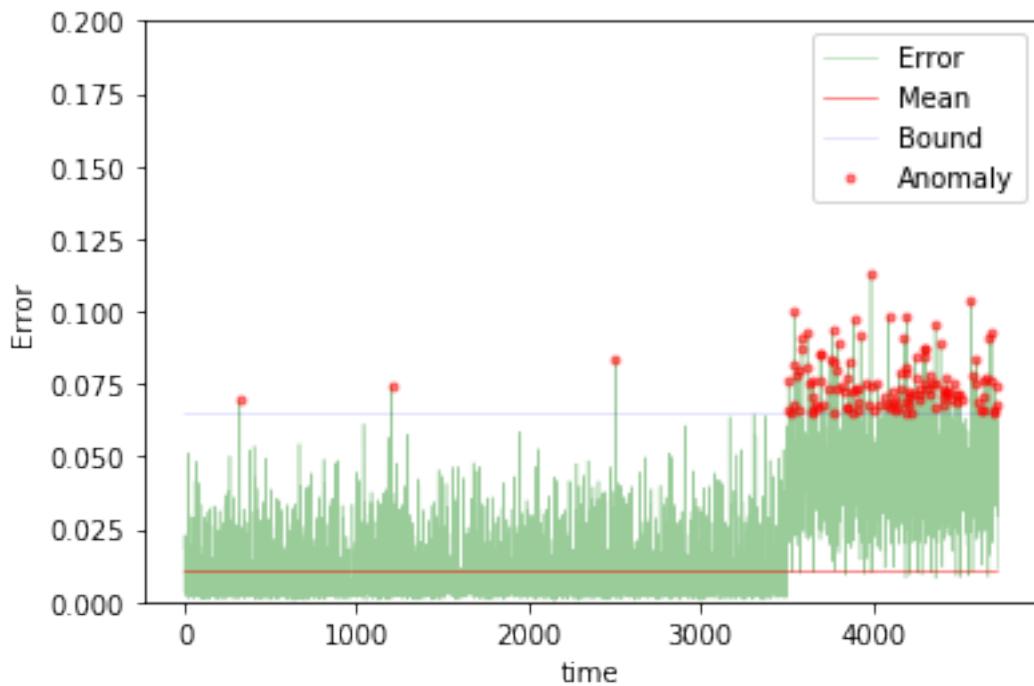
The mean error for nn3_2_normal_ is 0.01025794616037788 for length 4727
Testing on anomaly data.



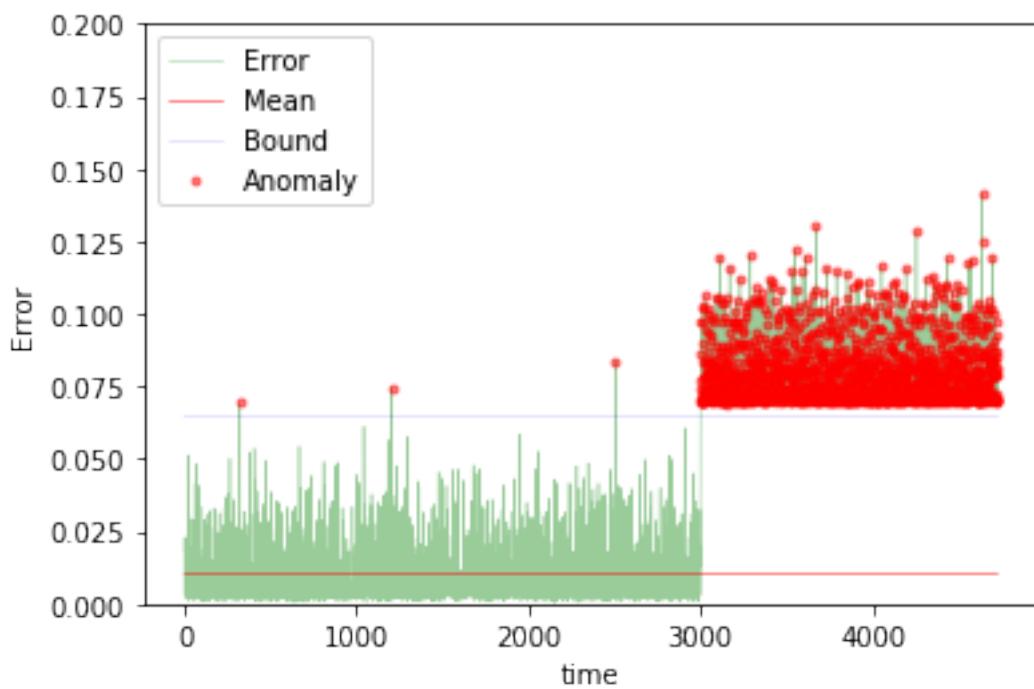
The mean error for nn3_2_anomaly_ is 0.012053205802177525 for length 4727
Testing on different app data.



The mean error for nn3_2_diff_app_ is 0.044996678170551835 for length 4727
Testing on App change synthetic data.



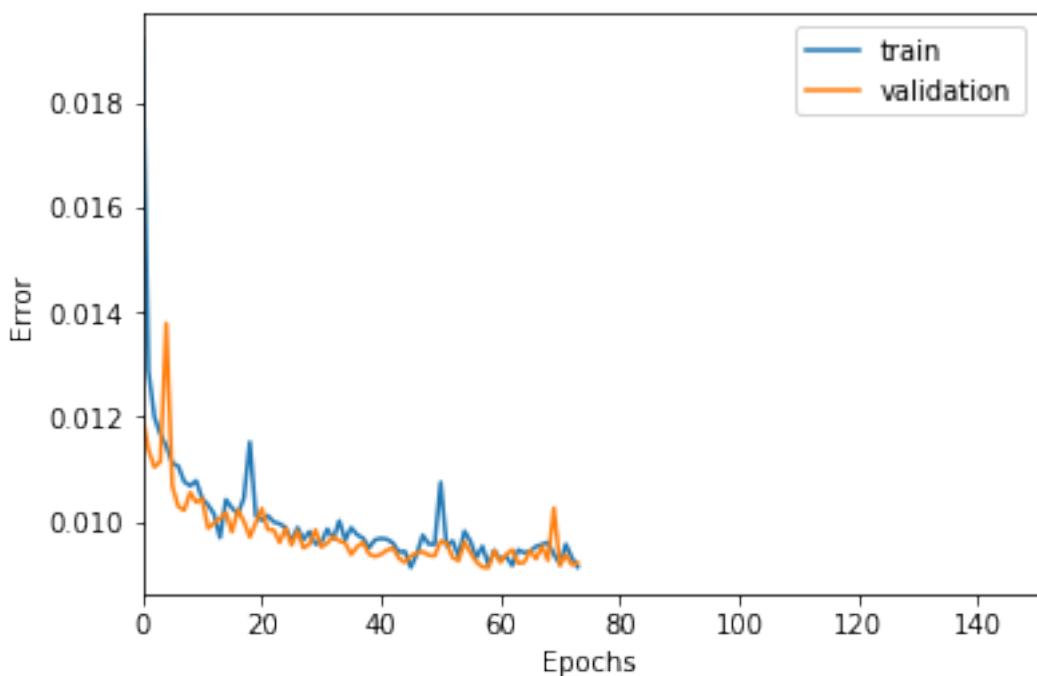
The mean error for nn3_2_app_change_ is 0.01920204245148244 for length 4727
Testing on Net flood synthetic data.



```
The mean error for nn3_2_net_flood_ is 0.03551656370363009 for length 4727  
=====
```

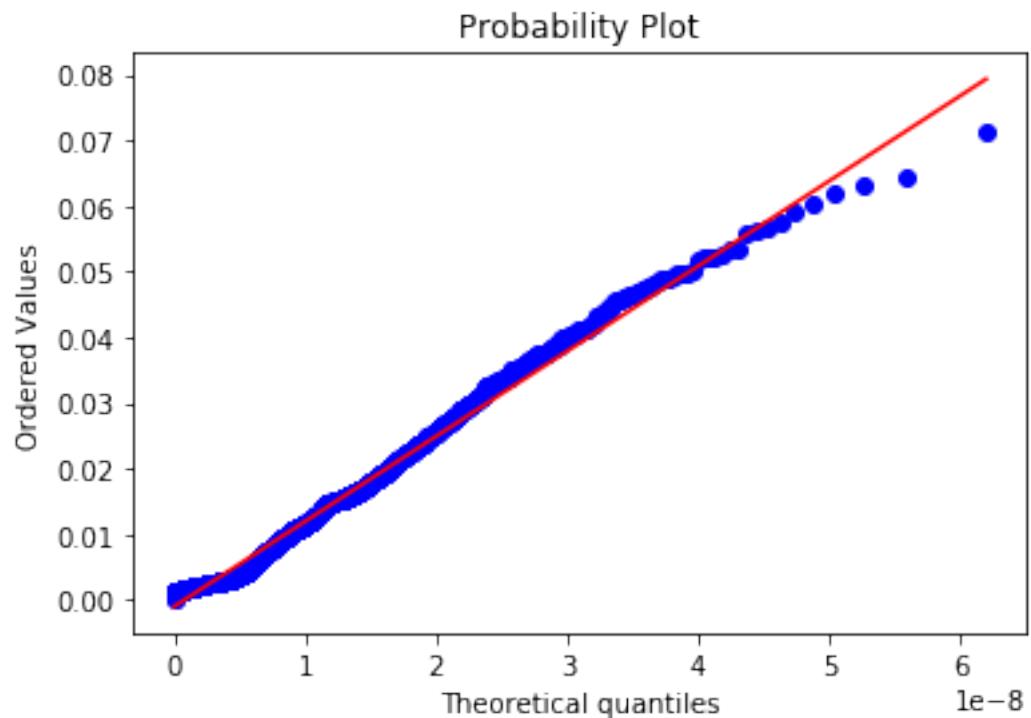
5 steps

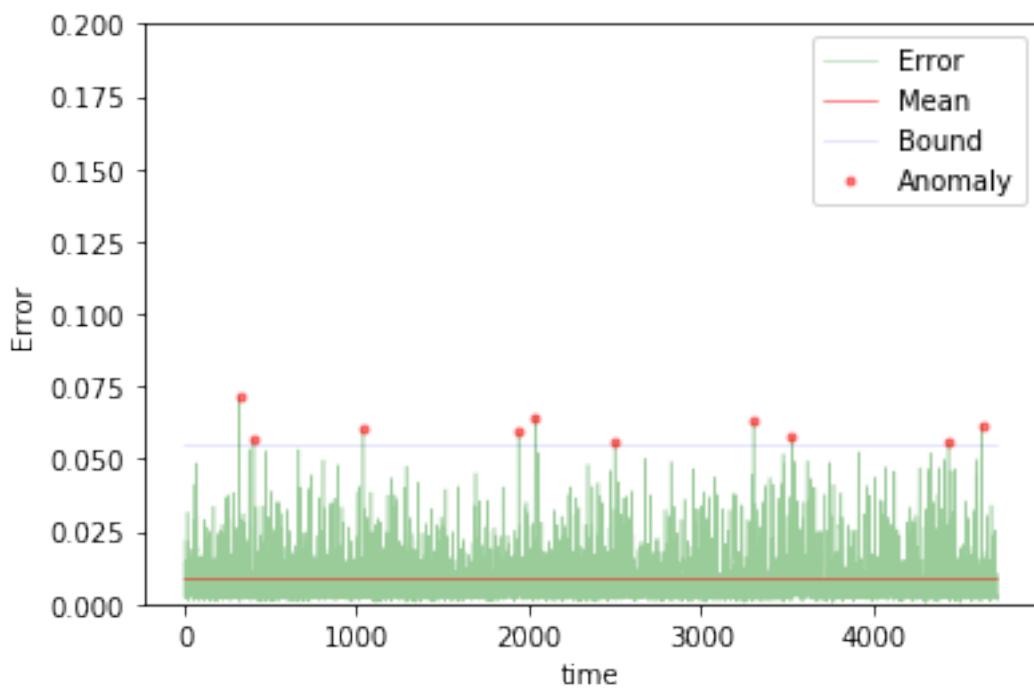
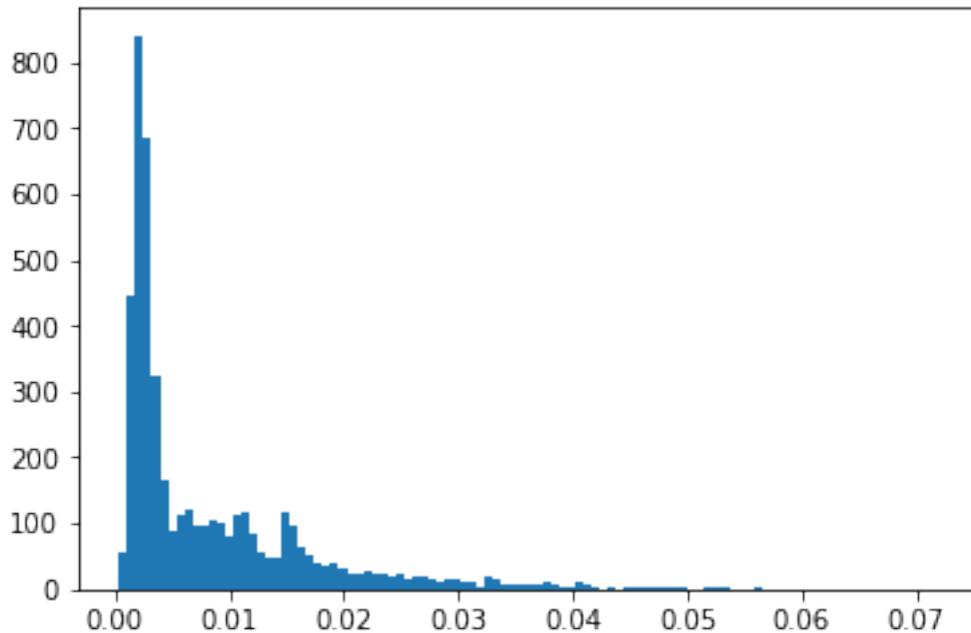
```
In [124]: TIMESTEPS = 5  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS)  
vgen = flat_generator(val_X, TIMESTEPS)  
name = "nn3_5"  
  
In [125]: input_layer = Input(shape=(TIMESTEPS*DIM,))  
hidden = Dense(1000, activation='relu')(input_layer)  
hidden = Dense(500, activation='relu')(hidden)  
hidden = Dense(100, activation='relu')(hidden)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [126]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [127]: train(model, tgen, vgen, name=name)  
test(model, name=name, window=TIMESTEPS)
```



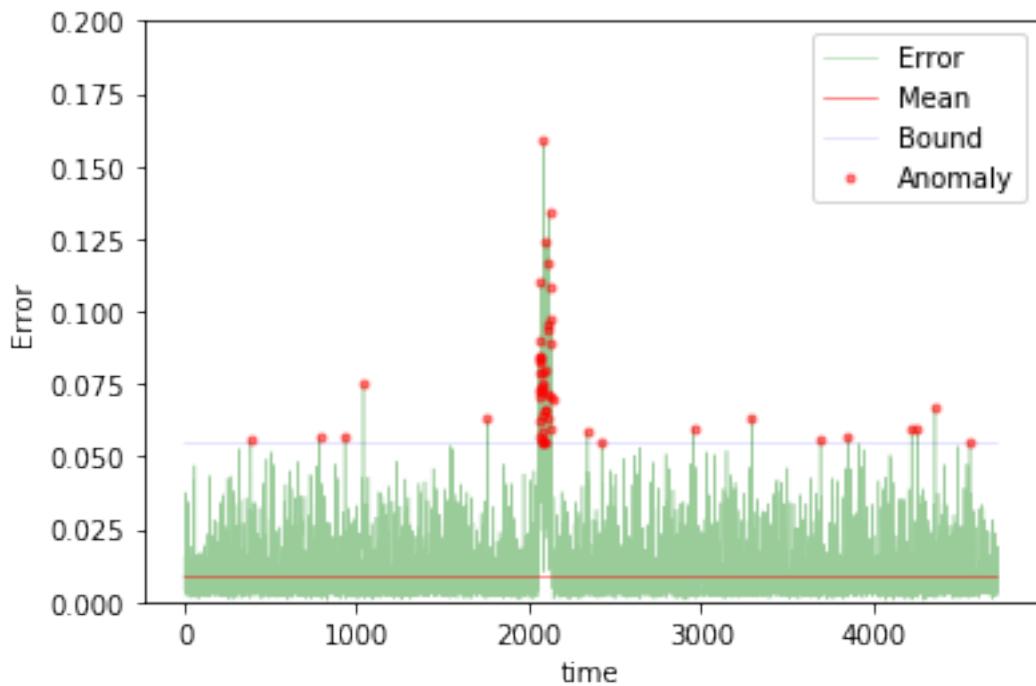
```
Training loss for final epoch is 0.009125758837384637
Validation loss for final epoch is 0.00921126921416726
----- Beginning tests for nn3_5 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

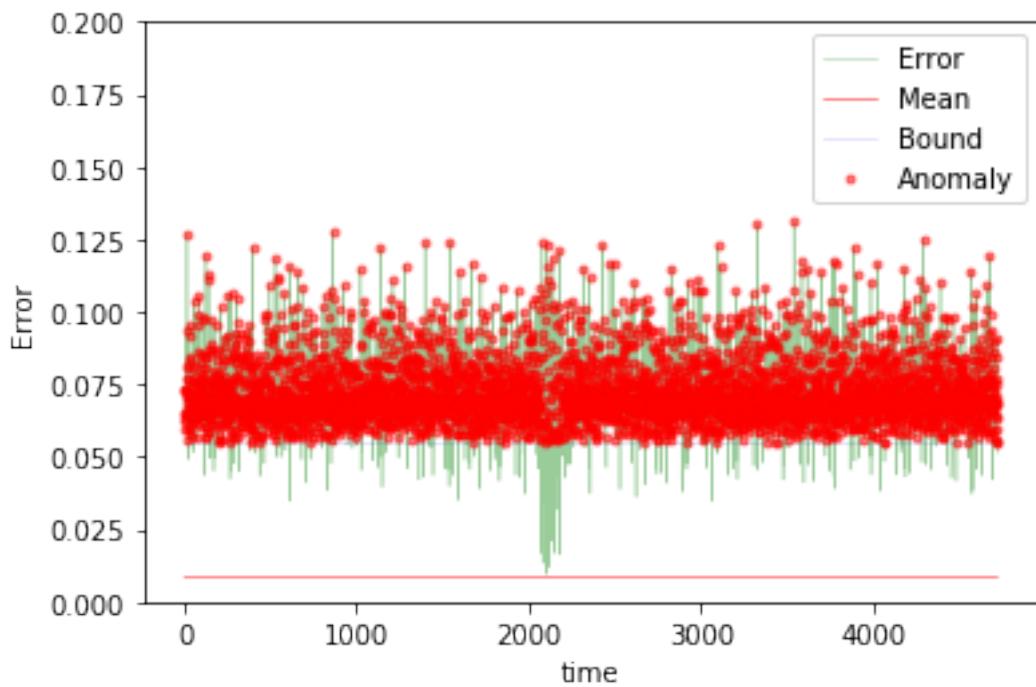




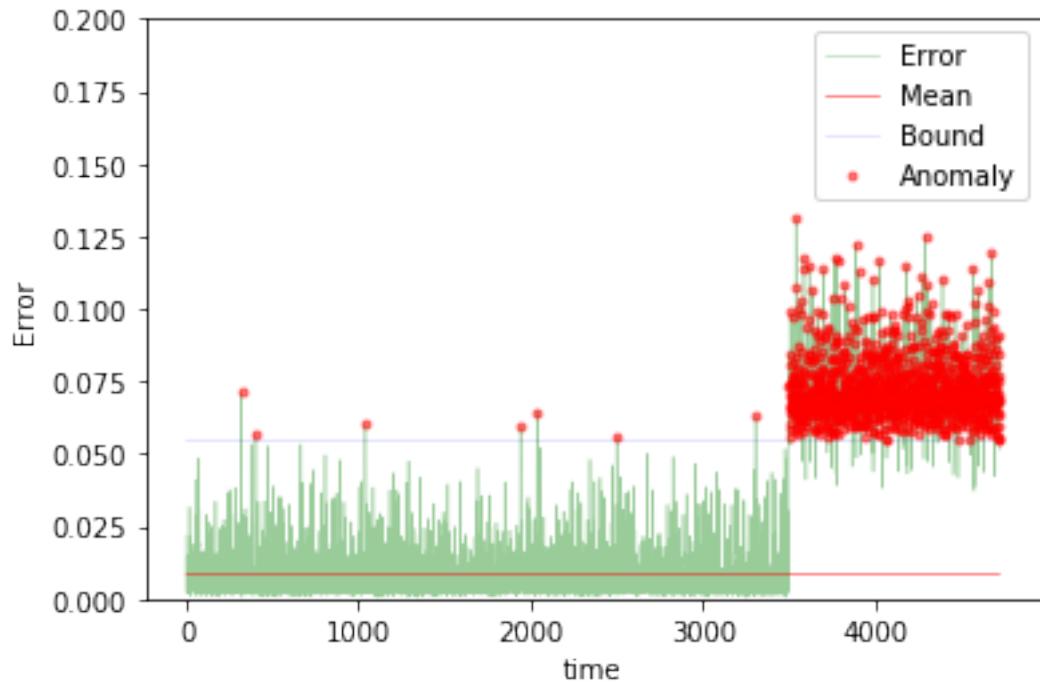
The mean error for nn3_5_normal_ is 0.00846975995108755 for length 4724
Testing on anomaly data.



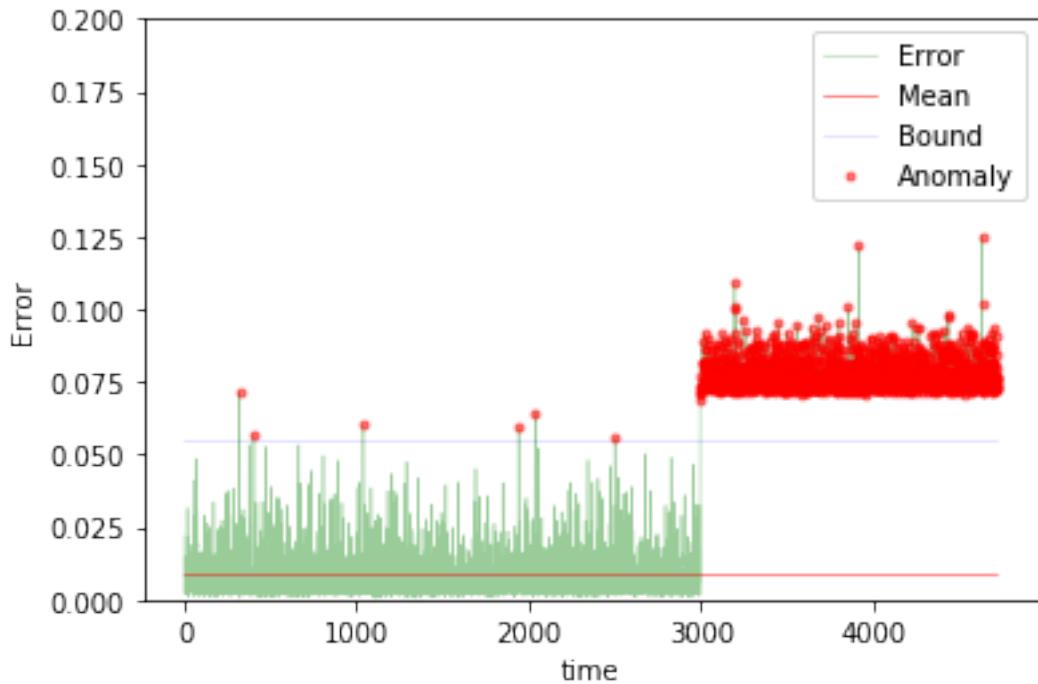
The mean error for nn3_5_anomaly_ is 0.010560166138242058 for length 4724
Testing on different app data.



The mean error for nn3_5_diff_app_ is 0.07130214365147336 for length 4724
Testing on App change synthetic data.



The mean error for nn3_5_app_change_ is 0.024824622131389123 for length 4724
Testing on Net flood synthetic data.



```
The mean error for nn3_5_net_flood_ is 0.03376375814969376 for length 4724
=====
```

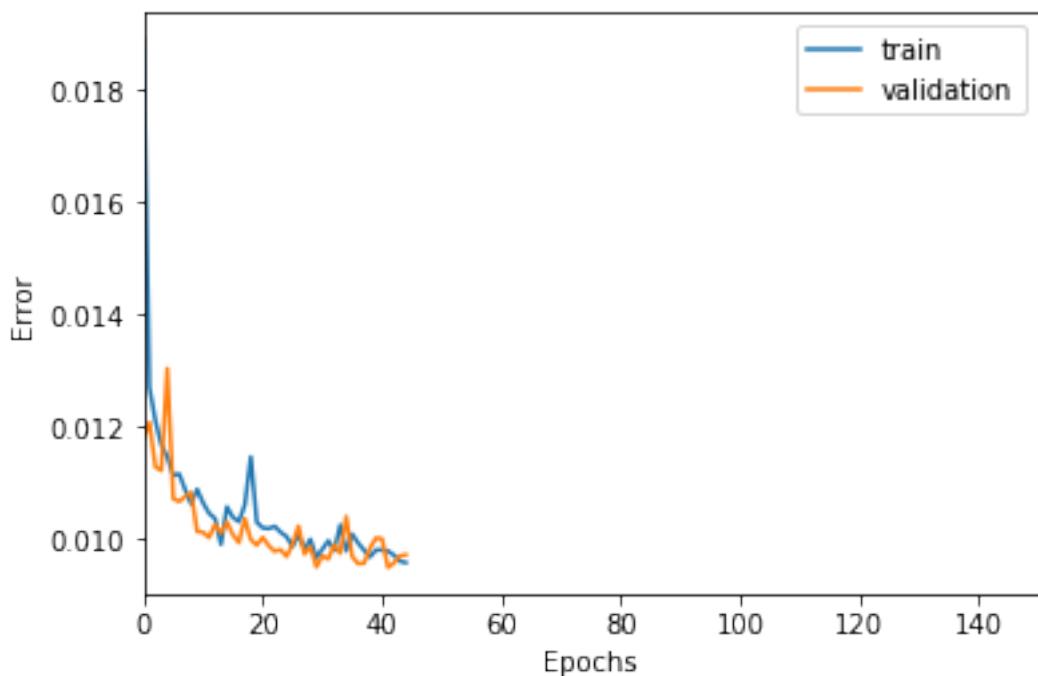
10 steps

```
In [128]: TIMESTEPS = 10
DIM = 29
tgen = flat_generator(X, TIMESTEPS)
vgen = flat_generator(val_X, TIMESTEPS)
name = "nn3_10"

In [129]: 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)

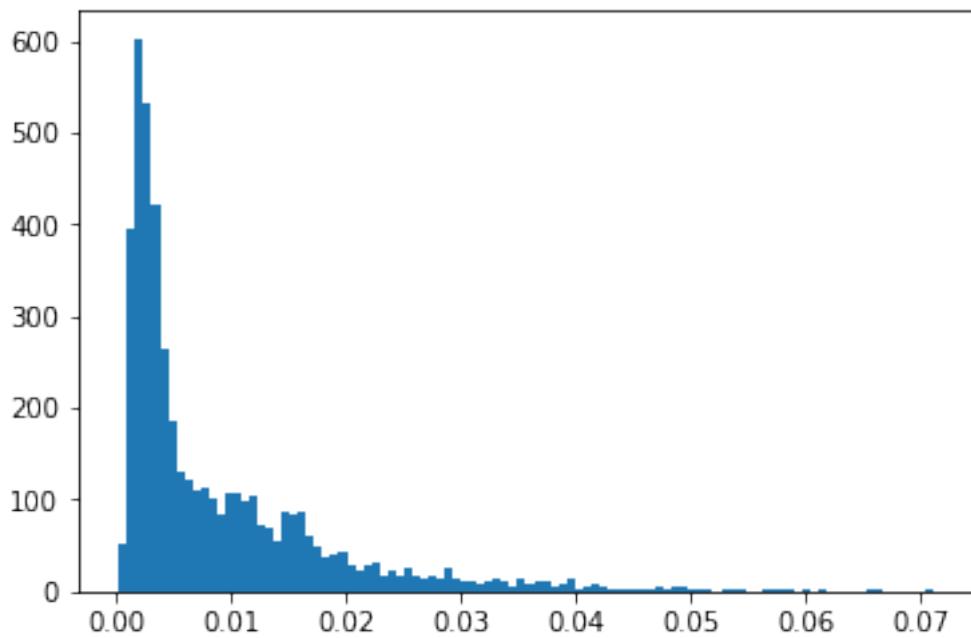
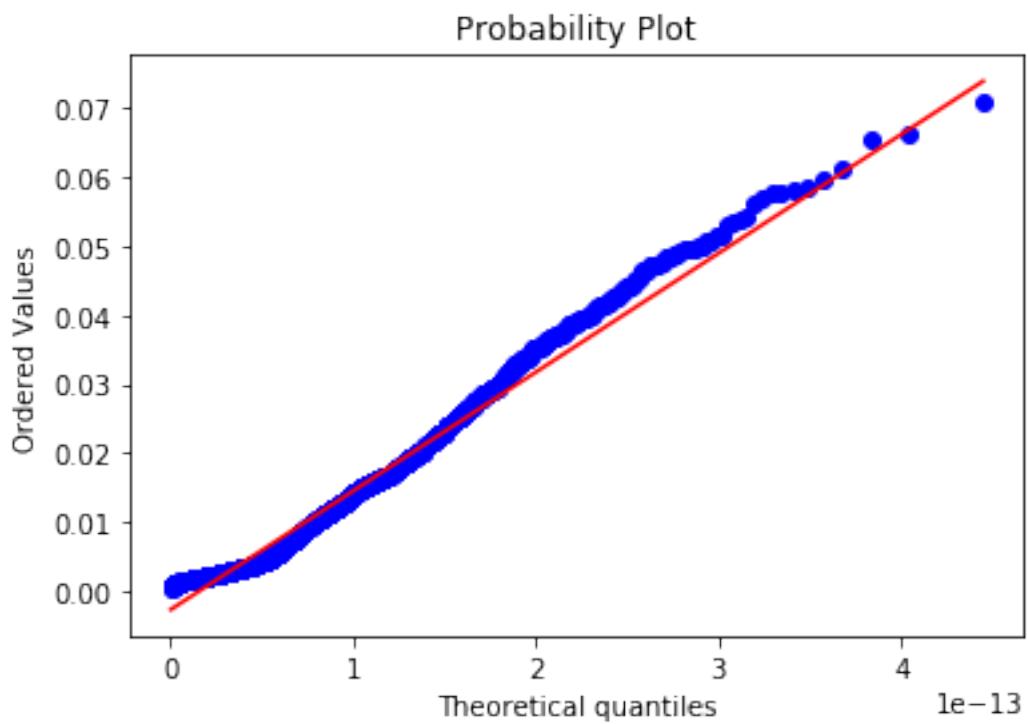
In [130]: model = Model(input_layer, output)
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

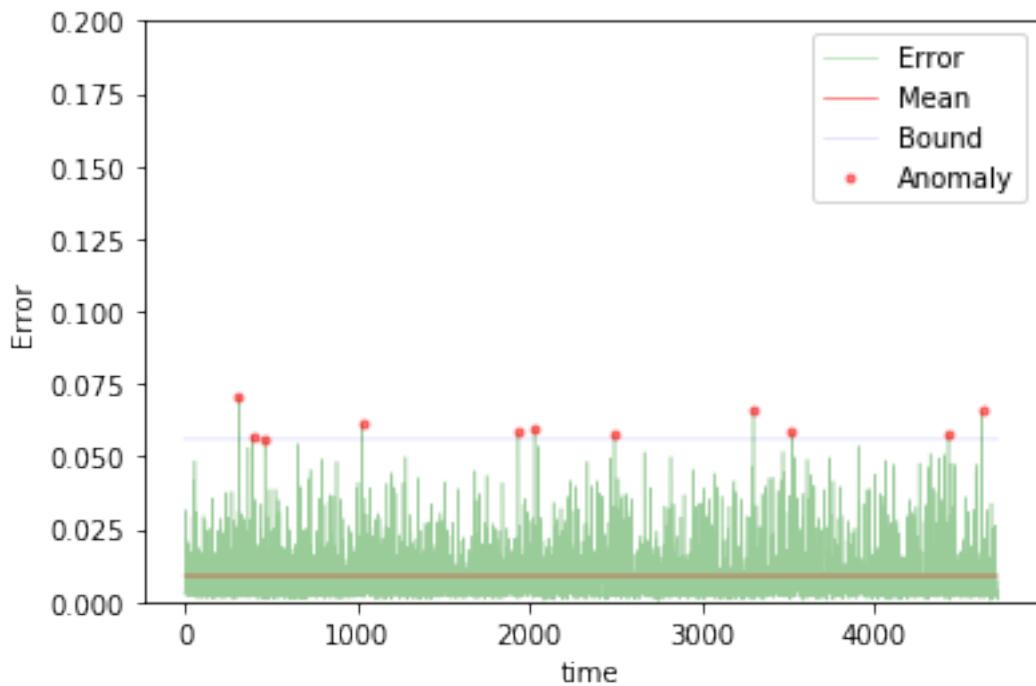
In [131]: train(model, tgen, vgen, name=name)
test(model, name=name, window=TIMESTEPS)
```



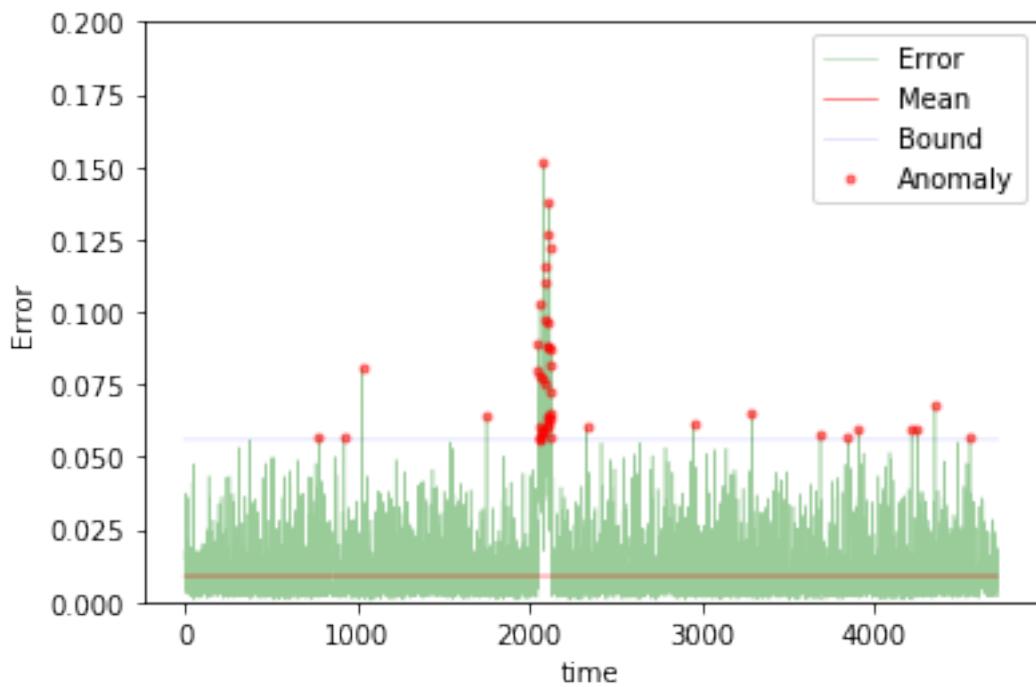
```
Training loss for final epoch is 0.009585862448555417
Validation loss for final epoch is 0.009726240959134885
----- Beginning tests for nn3_10 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
    warnings.warn(msg, RuntimeWarning)
```

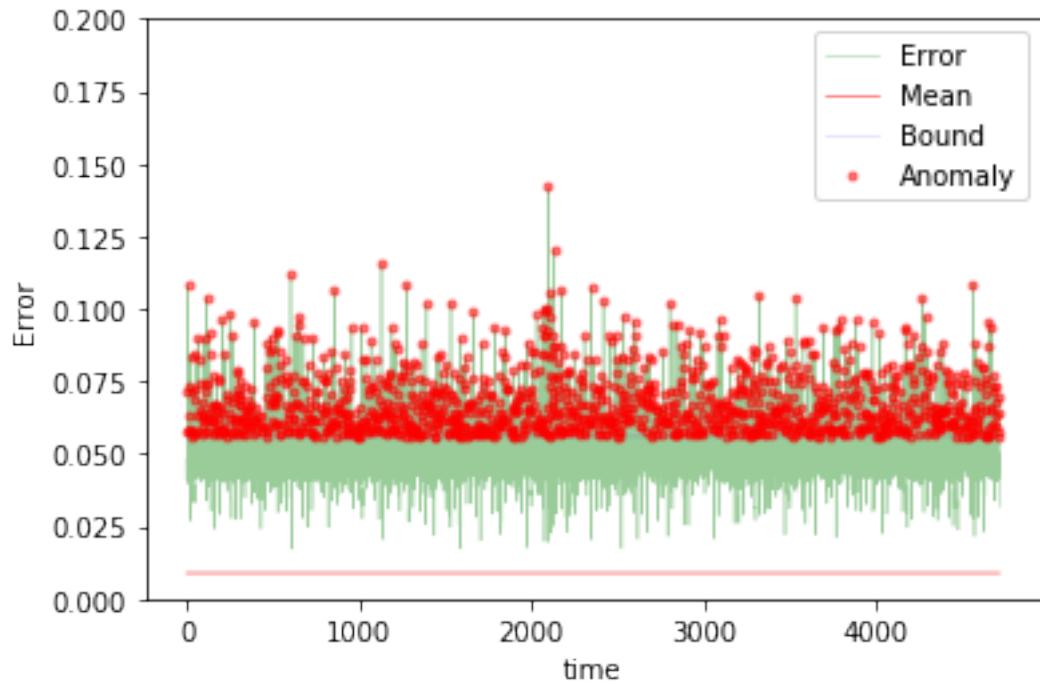




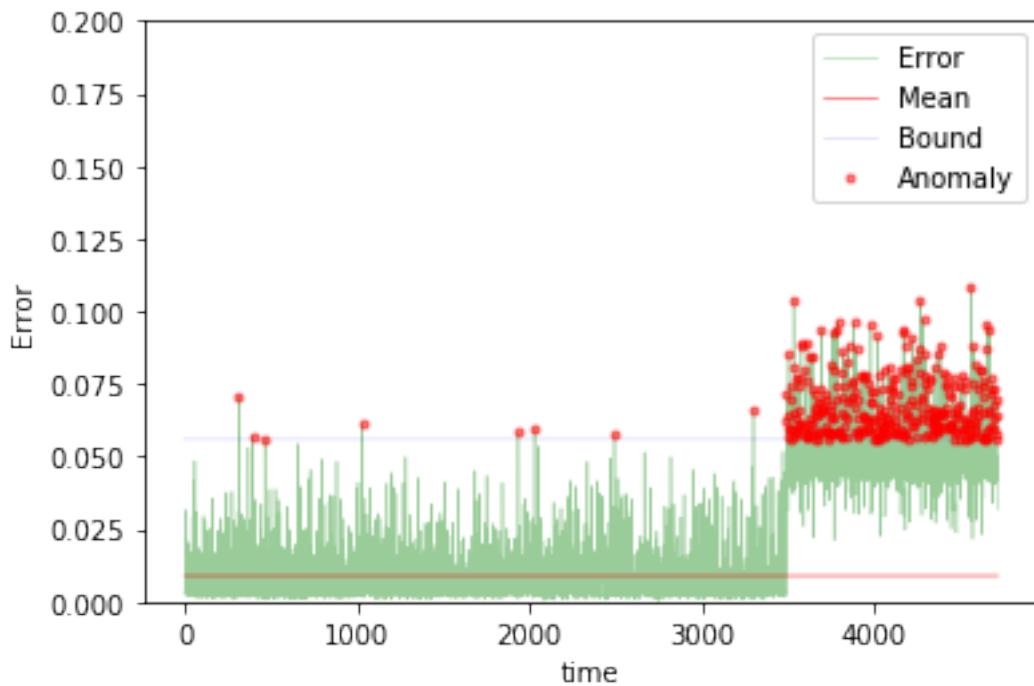
The mean error for nn3_10_normal_ is 0.008989971208853868 for length 4719
Testing on anomaly data.



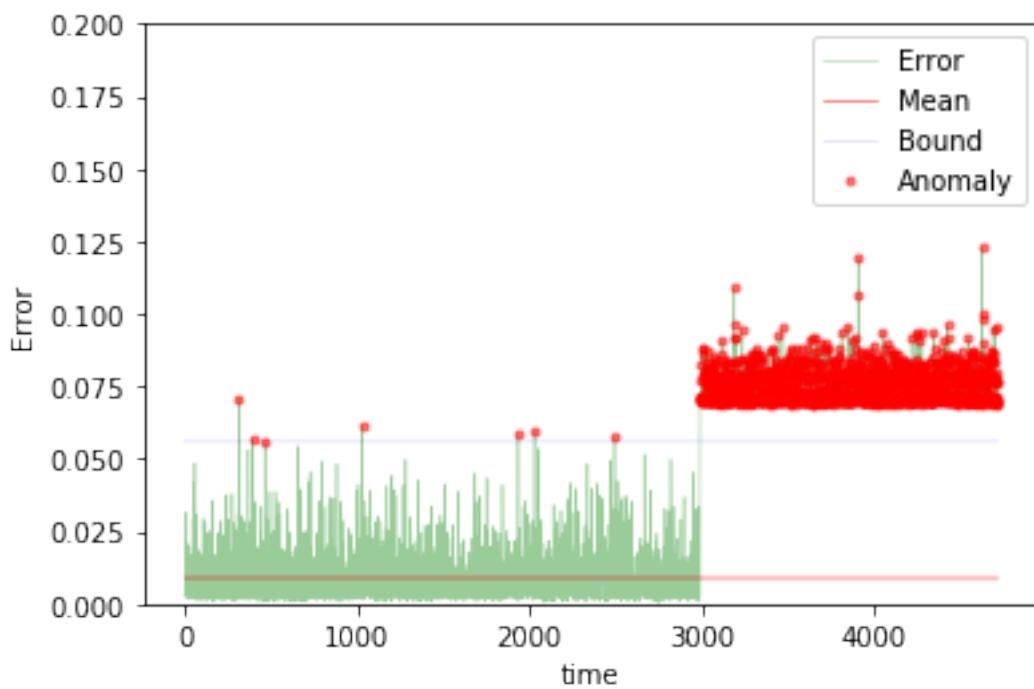
The mean error for nn3_10_anomaly_ is 0.011051465749595462 for length 4719
Testing on different app data.



The mean error for nn3_10_diff_app_ is 0.05155939495051457 for length 4719
Testing on App change synthetic data.



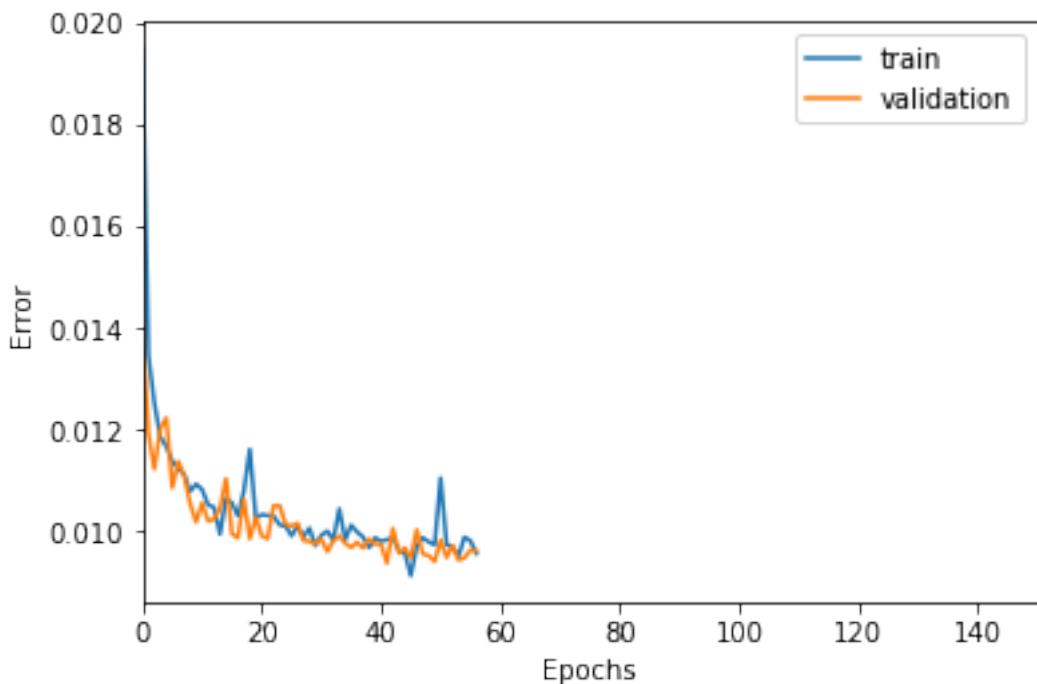
The mean error for nn3_10_app_change_ is 0.019967867350391844 for length 4719
Testing on Net flood synthetic data.



```
The mean error for nn3_10_net_flood_ is 0.033070026791400566 for length 4719  
=====
```

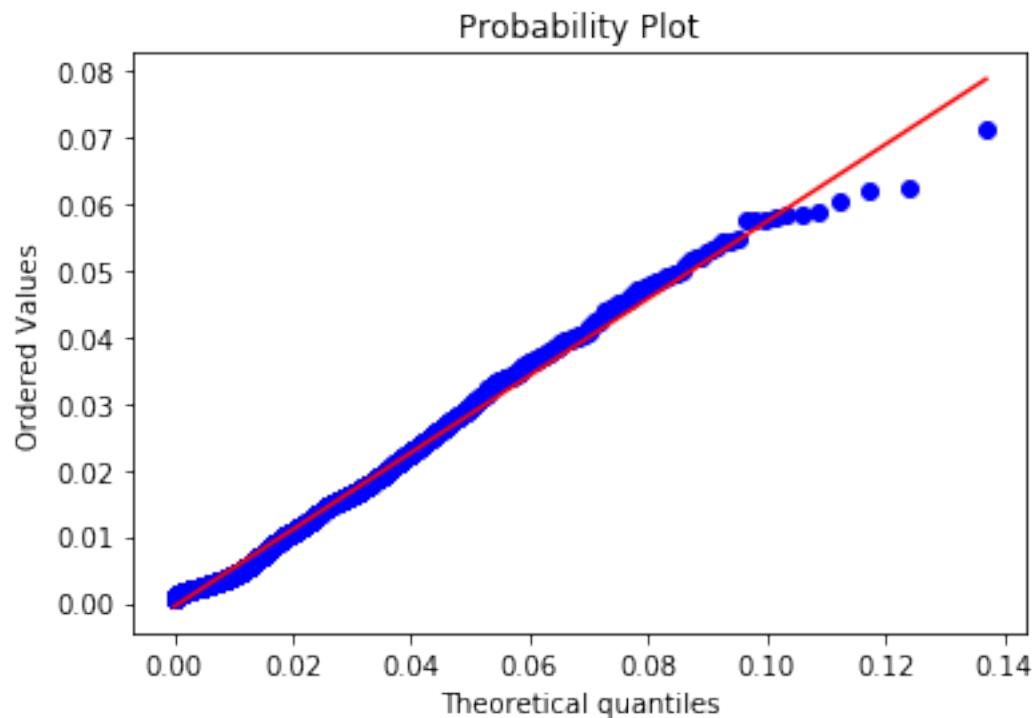
20 steps

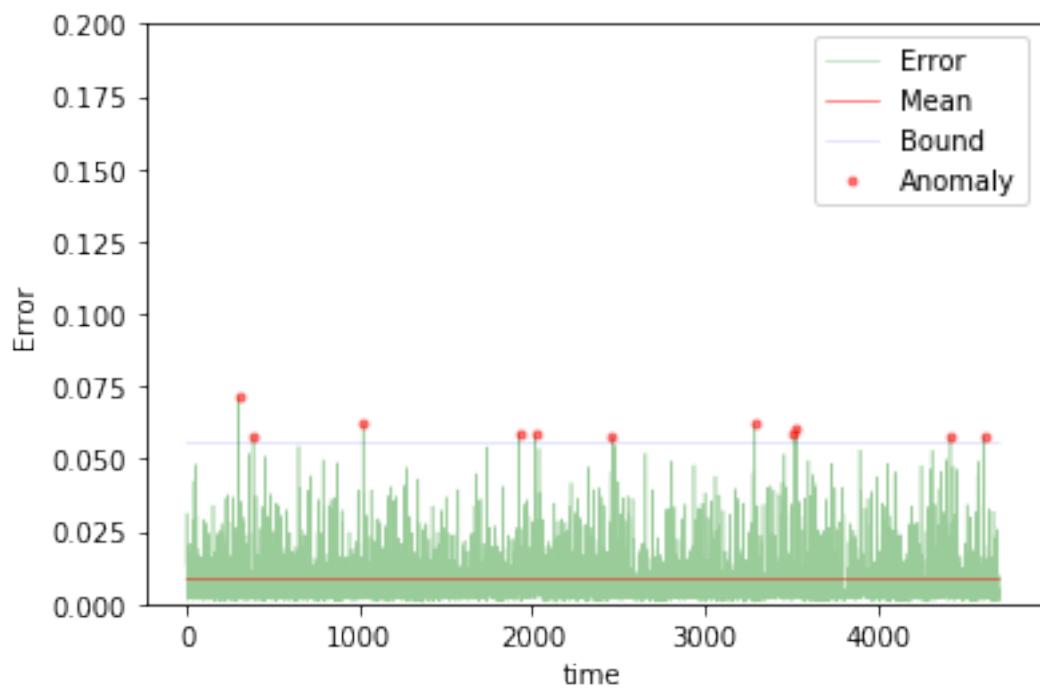
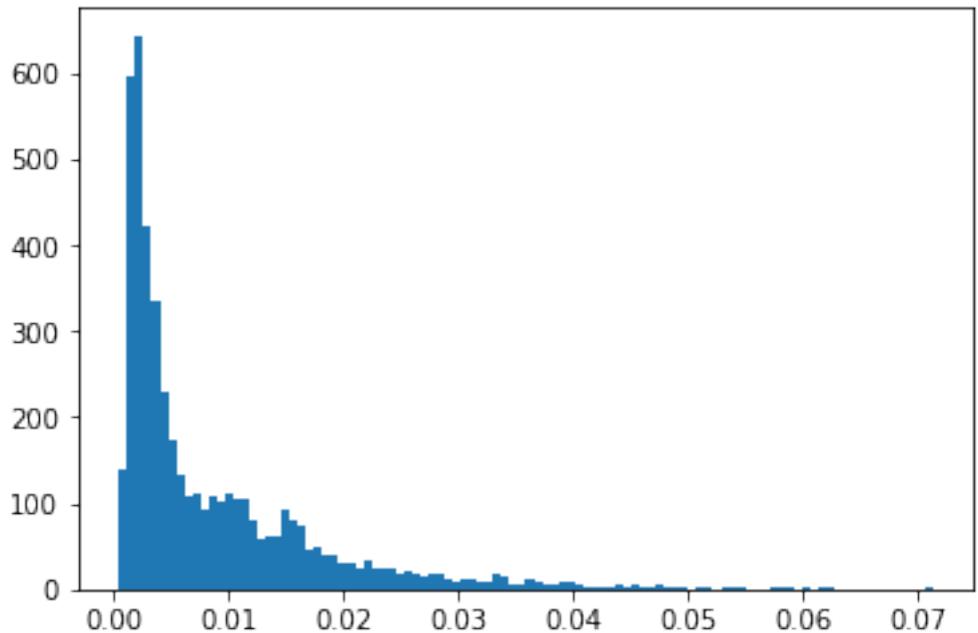
```
In [132]: TIMESTEPS = 20  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS)  
vgen = flat_generator(val_X, TIMESTEPS)  
name = "nn3_20"  
  
In [133]: input_layer = Input(shape=(TIMESTEPS*DIM,))  
hidden = Dense(1000, activation='relu')(input_layer)  
hidden = Dense(500, activation='relu')(hidden)  
hidden = Dense(100, activation='relu')(hidden)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [134]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [135]: train(model, tgen, vgen, name=name)  
test(model, name=name, window=TIMESTEPS)
```



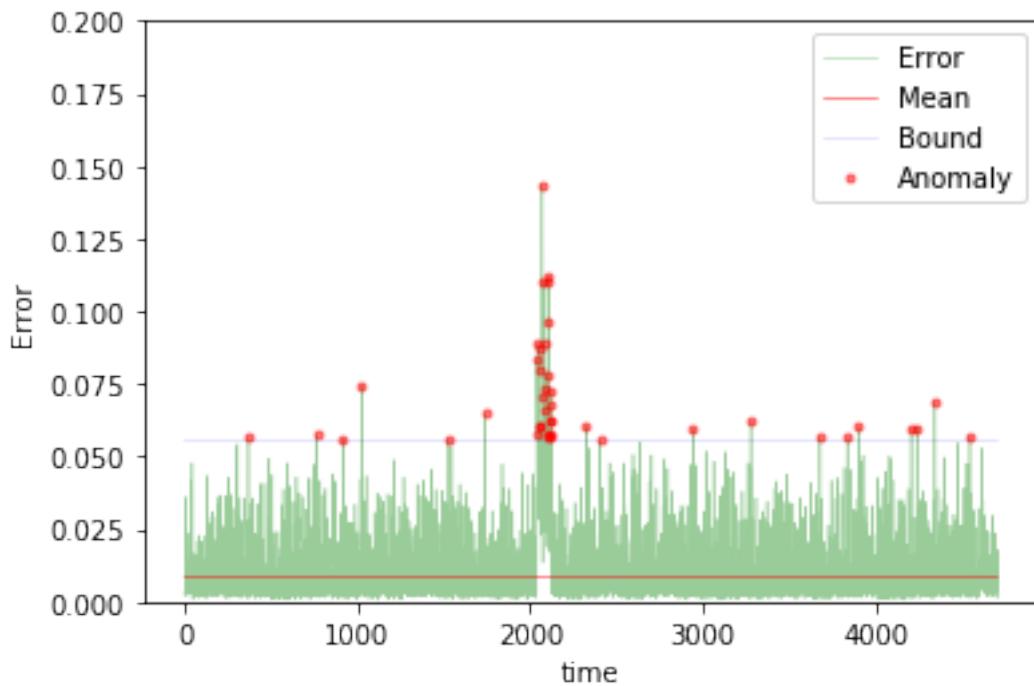
```
Training loss for final epoch is 0.009561208141967655
Validation loss for final epoch is 0.009614539212896489
----- Beginning tests for nn3_20 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

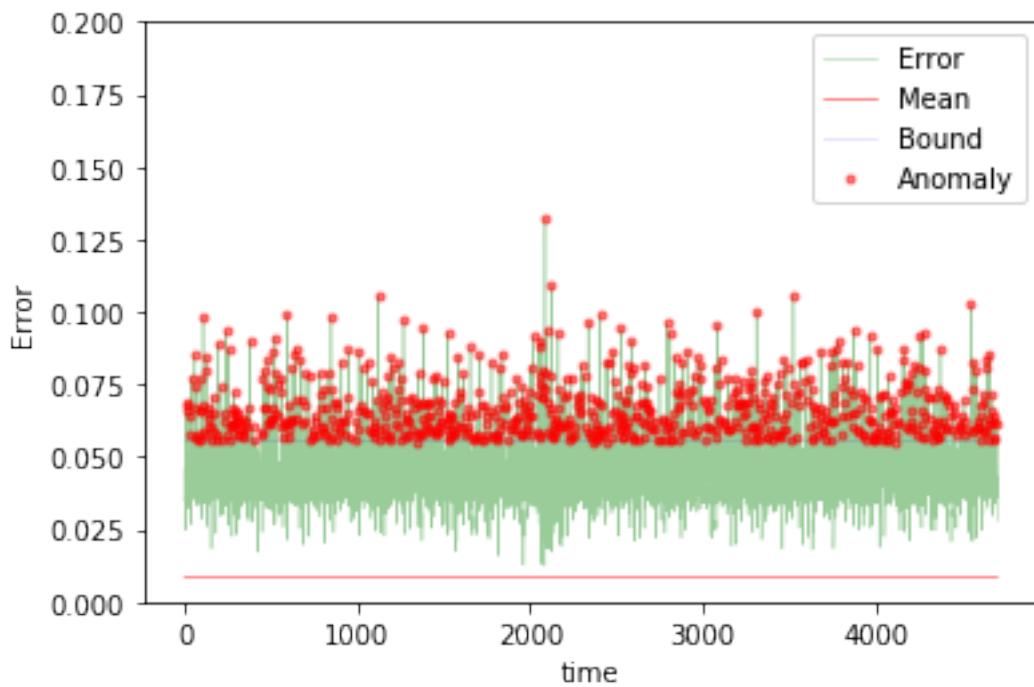




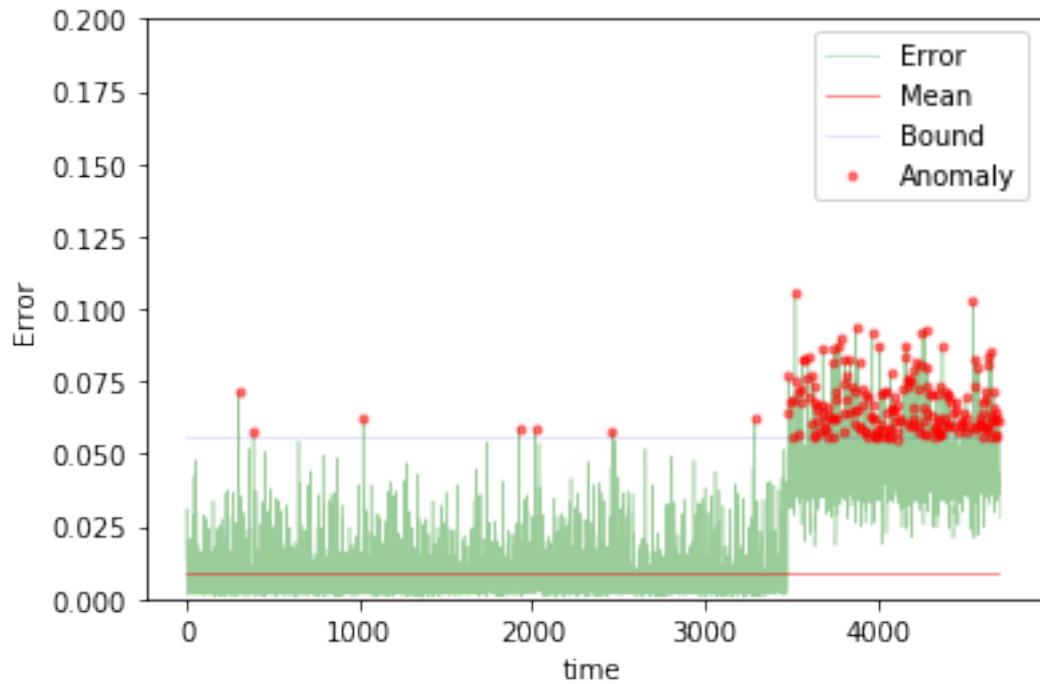
The mean error for nn3_20_normal_ is 0.008760568939827231 for length 4709
Testing on anomaly data.



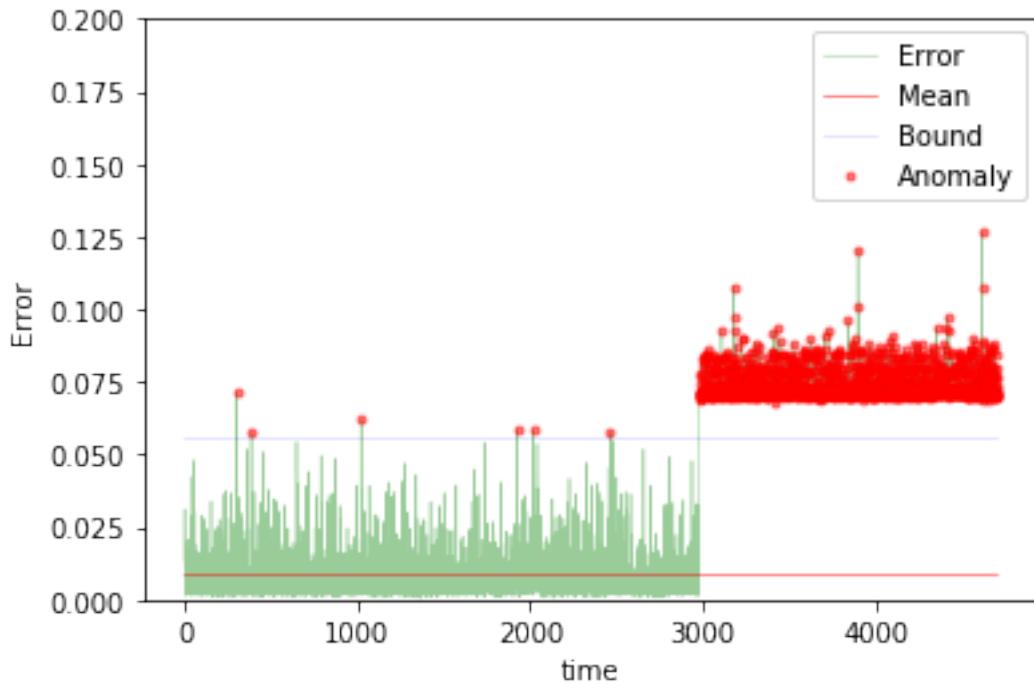
The mean error for nn3_20_anomaly_ is 0.010702195711009109 for length 4709
Testing on different app data.



The mean error for nn3_20_diff_app_ is 0.04490480851034667 for length 4709
Testing on App change synthetic data.



The mean error for nn3_20_app_change_ is 0.01818021868591191 for length 4709
Testing on Net flood synthetic data.



```
The mean error for nn3_20_net_flood_ is 0.032826999575025315 for length 4709
=====
```

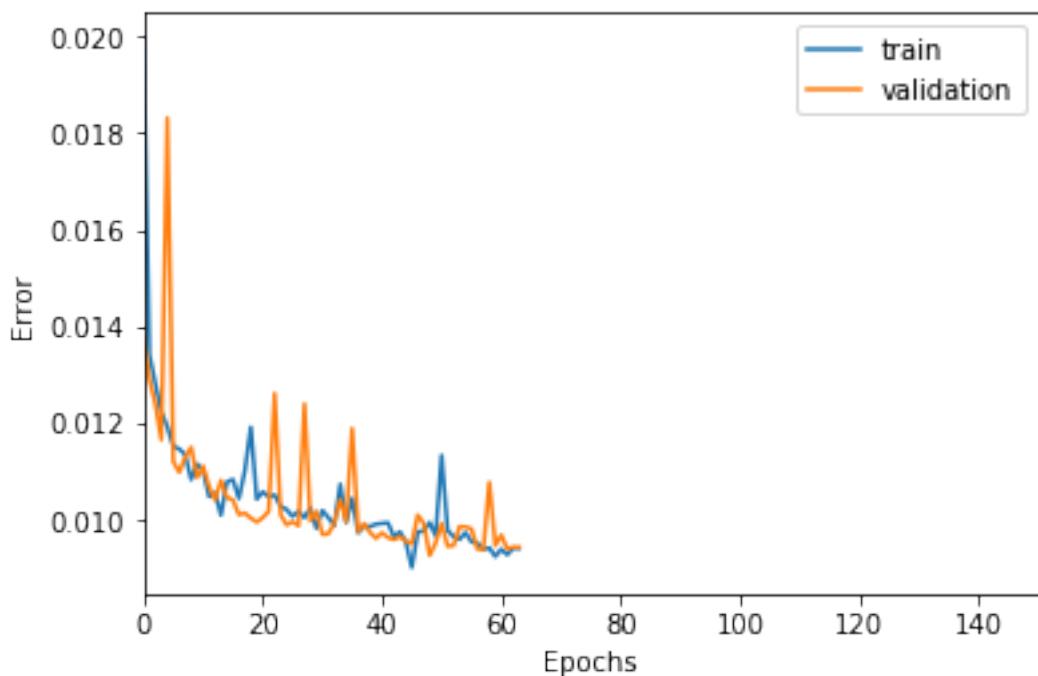
50 steps

```
In [136]: TIMESTEPS = 50
DIM = 29
tgen = flat_generator(X, TIMESTEPS)
vgen = flat_generator(val_X, TIMESTEPS)
name = "nn3_50"

In [137]: 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)

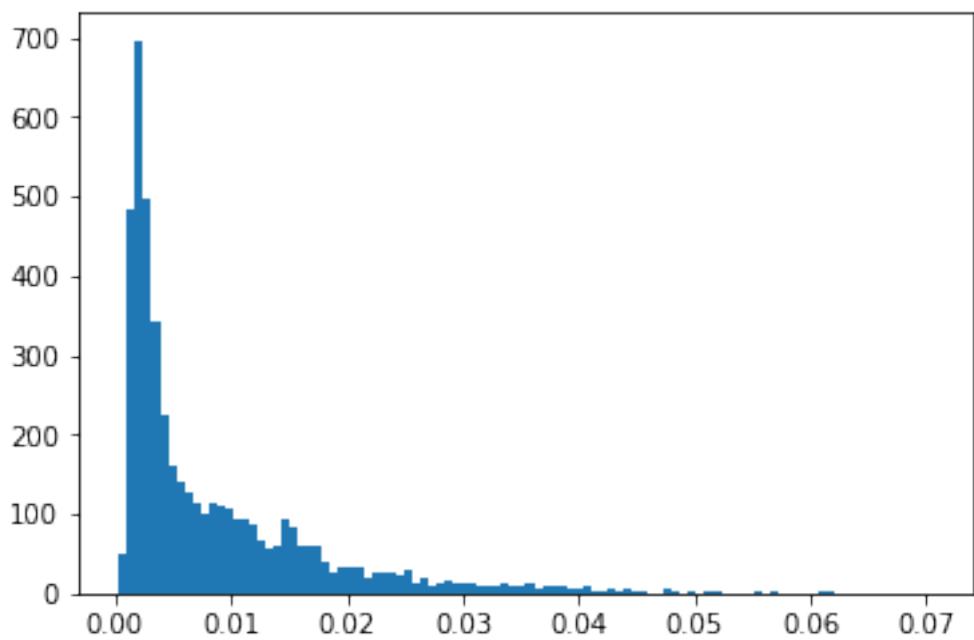
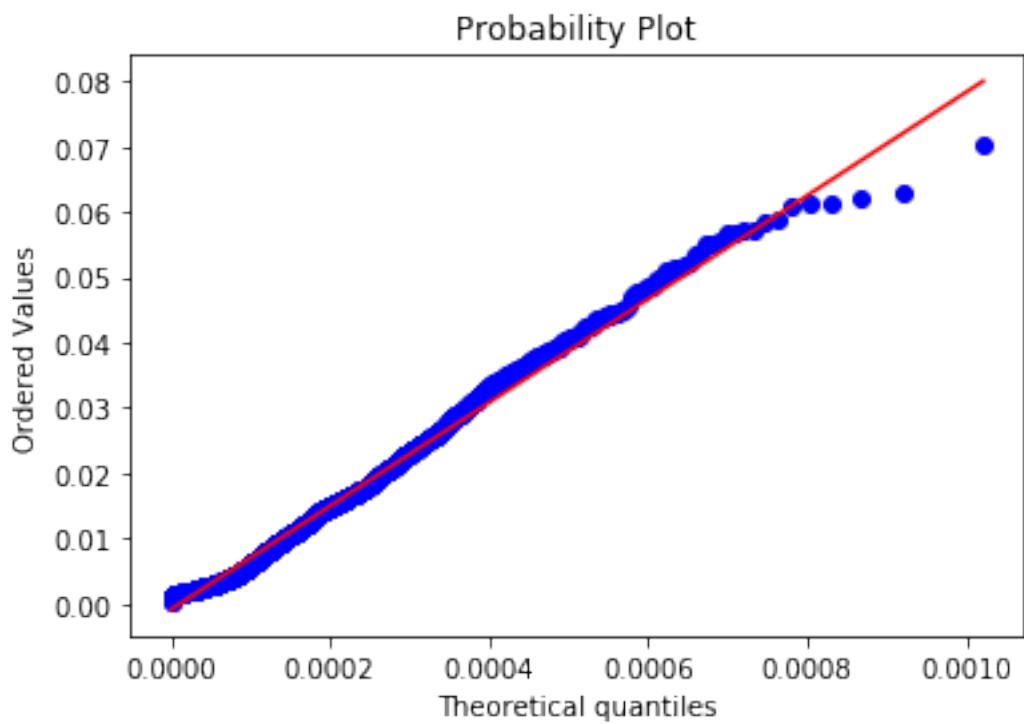
In [138]: model = Model(input_layer, output)
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

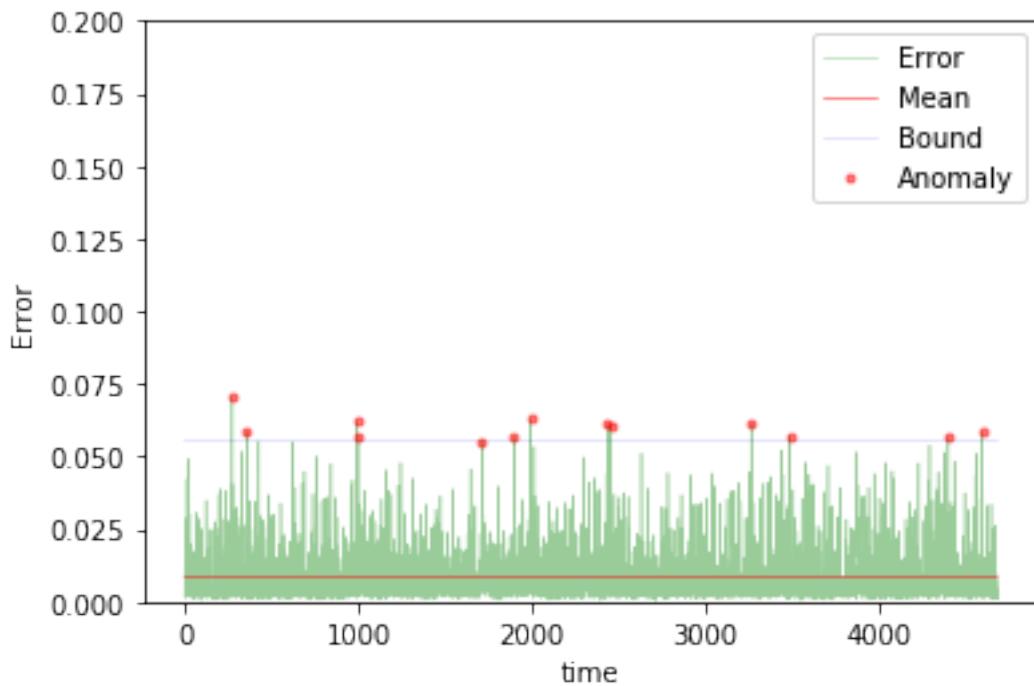
In [139]: train(model, tgen, vgen, name=name)
test(model, name=name, window=TIMESTEPS)
```



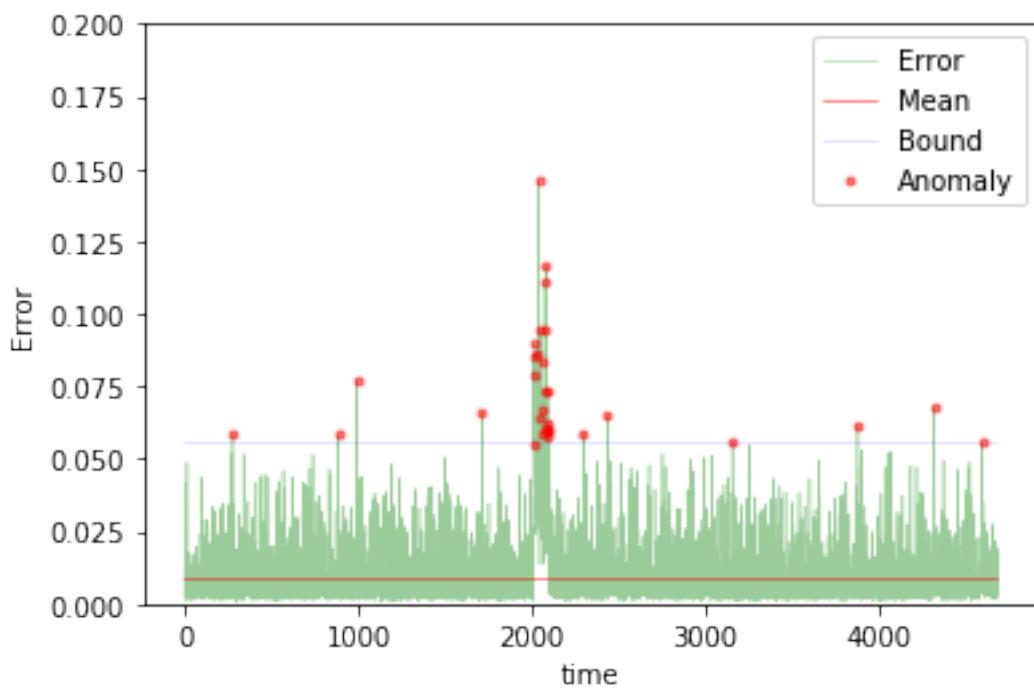
```
Training loss for final epoch is 0.009419451088644564
Validation loss for final epoch is 0.009439013539580628
----- Beginning tests for nn3_50 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
    warnings.warn(msg, RuntimeWarning)
```

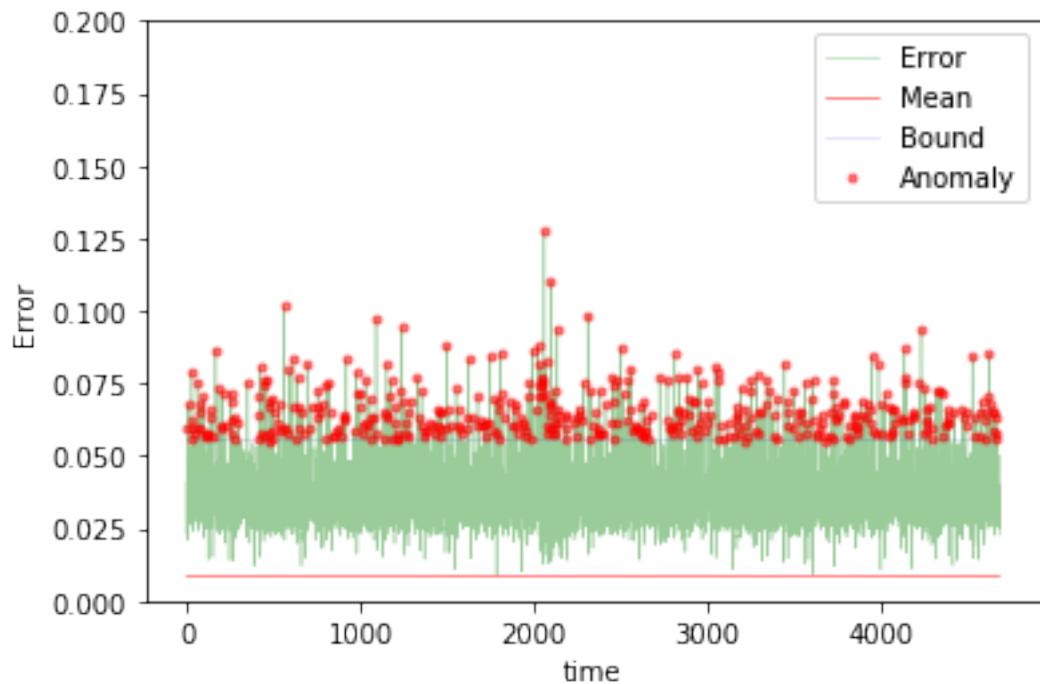




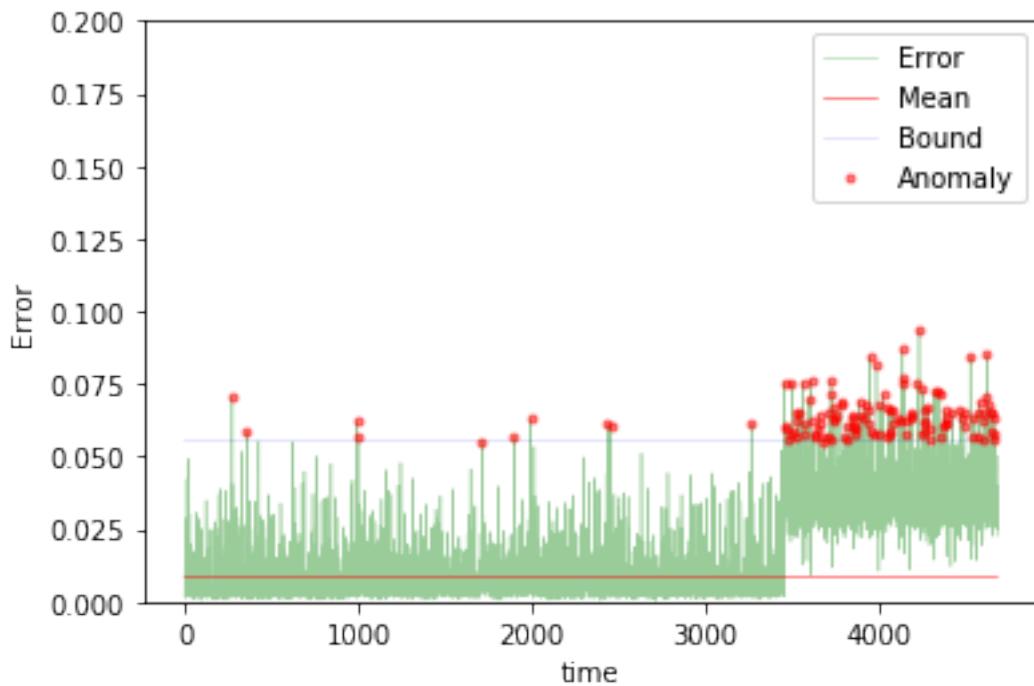
The mean error for nn3_50_normal_ is 0.008738435684353485 for length 4679
Testing on anomaly data.



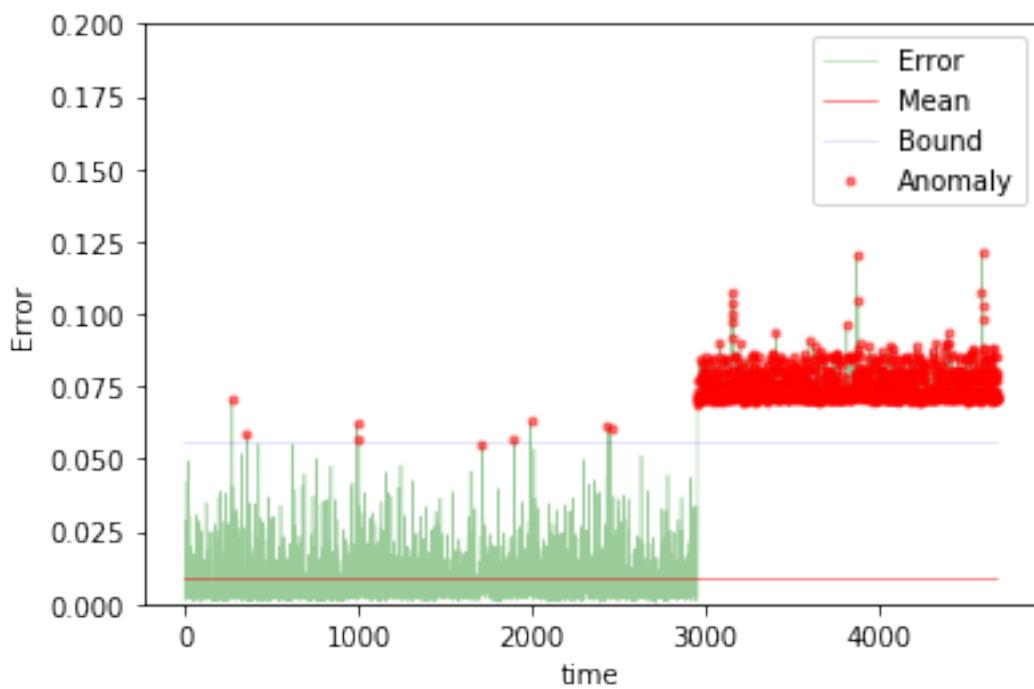
The mean error for nn3_50_anomaly_ is 0.010416667662053142 for length 4679
Testing on different app data.



The mean error for nn3_50_diff_app_ is 0.037398679963817606 for length 4679
Testing on App change synthetic data.



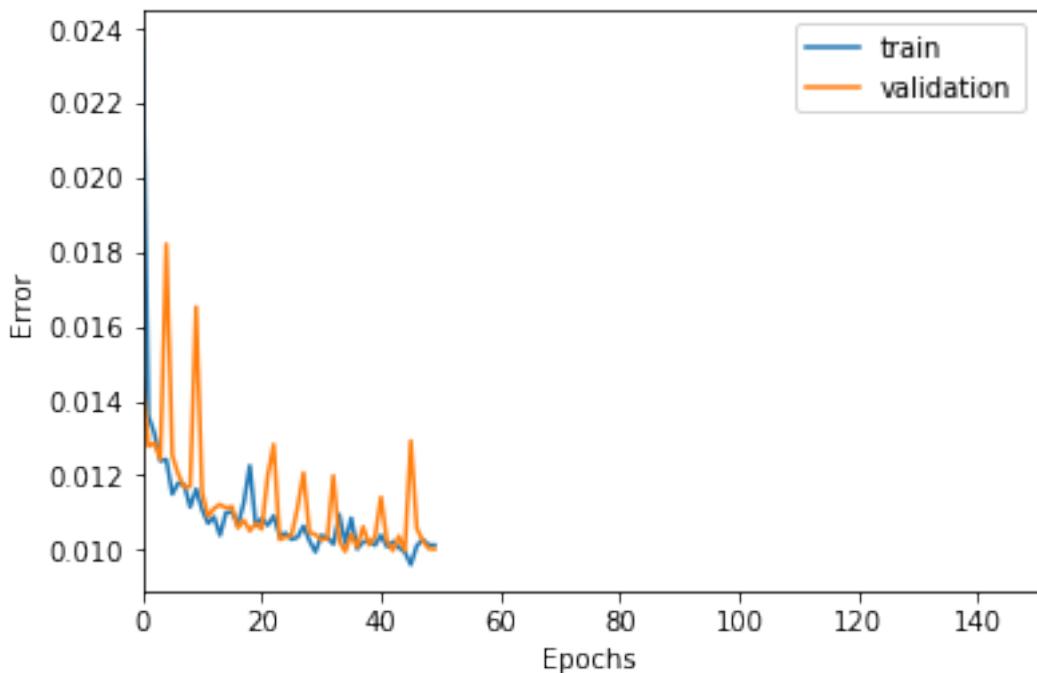
The mean error for nn3_50_app_change_ is 0.0162086306335976 for length 4679
Testing on Net flood synthetic data.



```
The mean error for nn3_50_net_flood_ is 0.0330152840453521 for length 4679  
=====
```

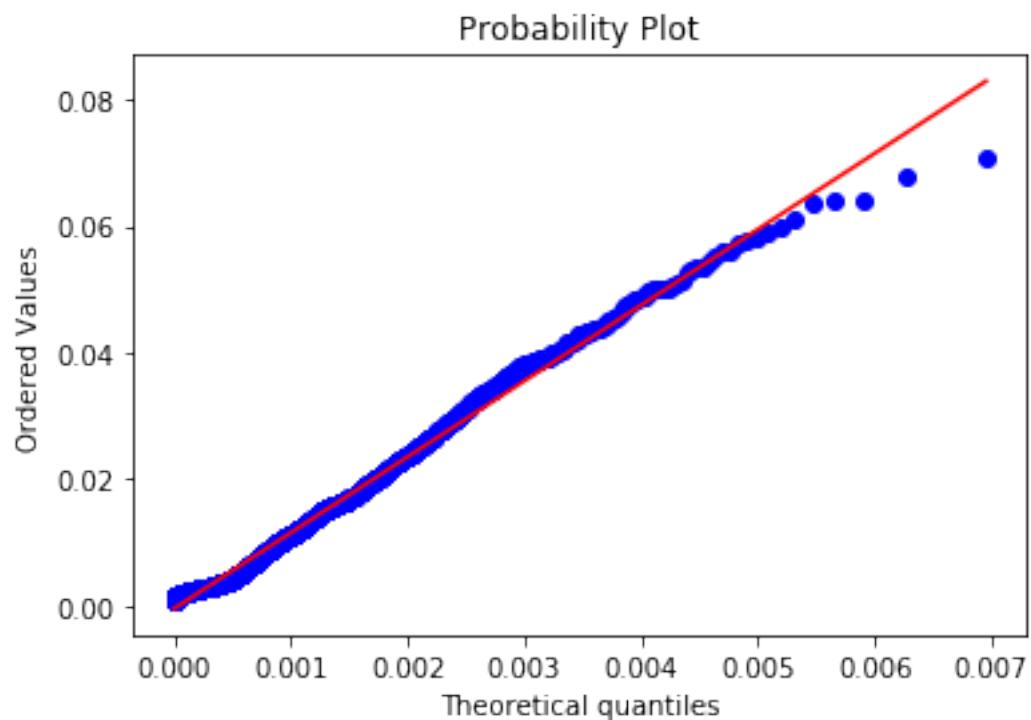
100 steps

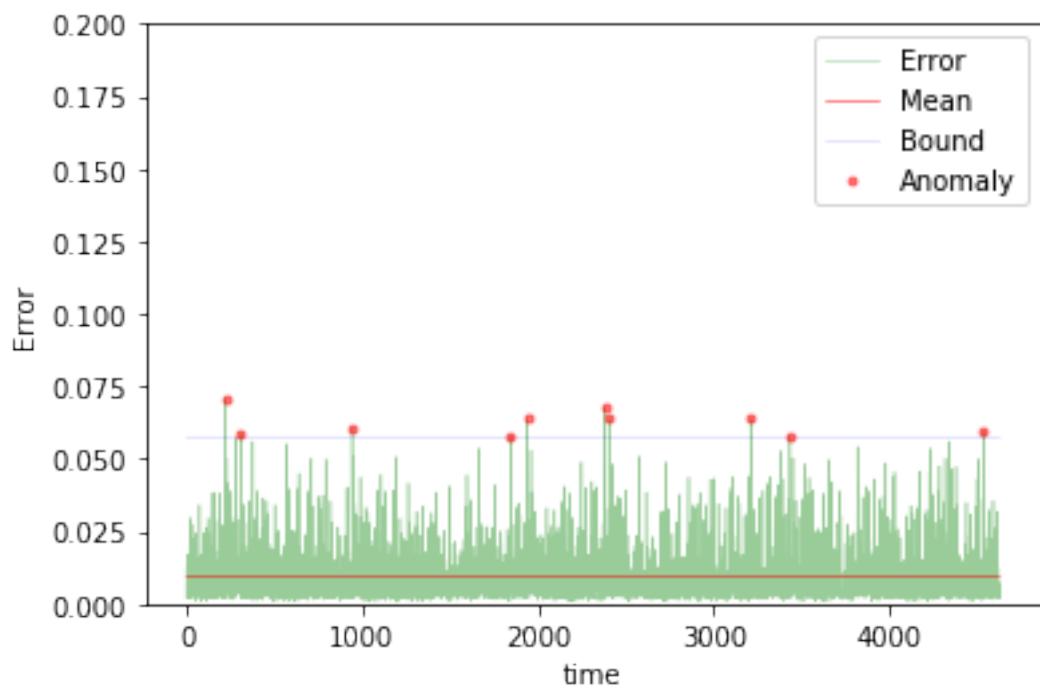
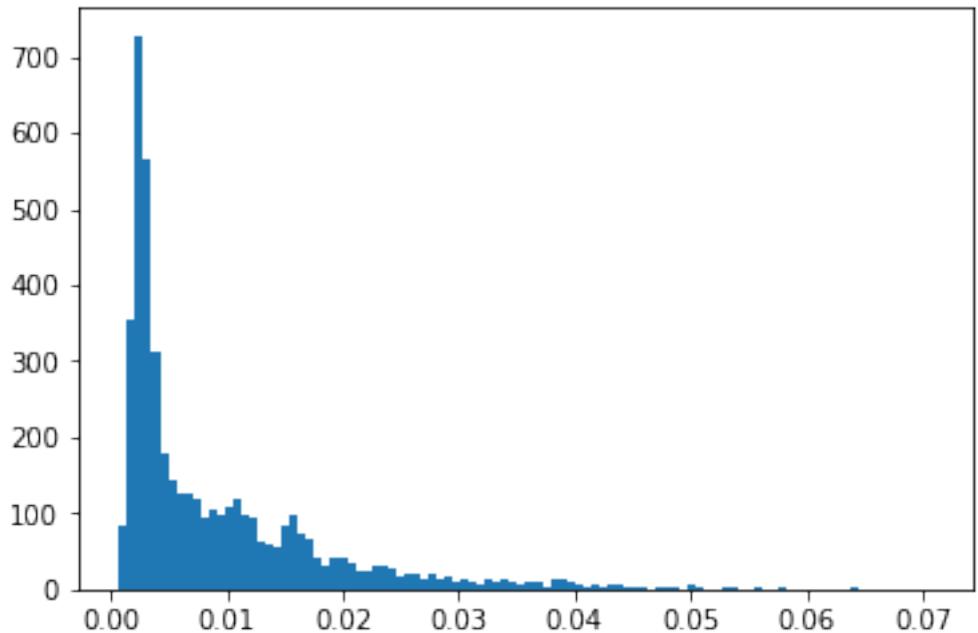
```
In [140]: TIMESTEPS = 100  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS)  
vgen = flat_generator(val_X, TIMESTEPS)  
name = "nn3_100"  
  
In [141]: input_layer = Input(shape=(TIMESTEPS*DIM,))  
hidden = Dense(1000, activation='relu')(input_layer)  
hidden = Dense(500, activation='relu')(hidden)  
hidden = Dense(100, activation='relu')(hidden)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [142]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [143]: train(model, tgen, vgen, name=name)  
test(model, name=name, window=TIMESTEPS)
```



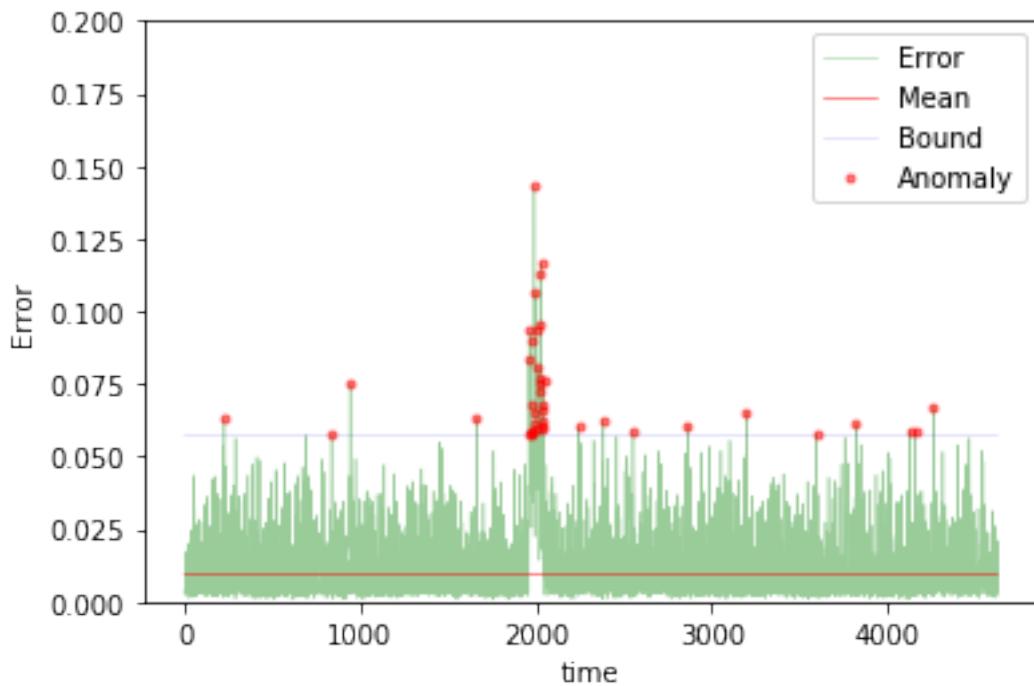
```
Training loss for final epoch is 0.01011661703151185
Validation loss for final epoch is 0.010010033966624178
----- Beginning tests for nn3_100 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

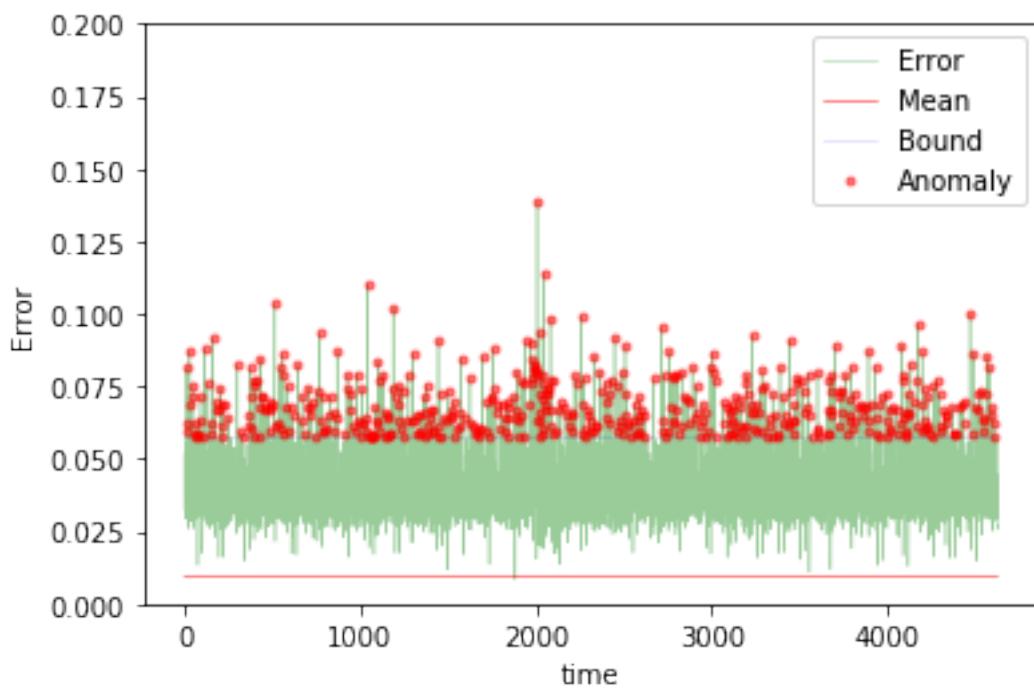




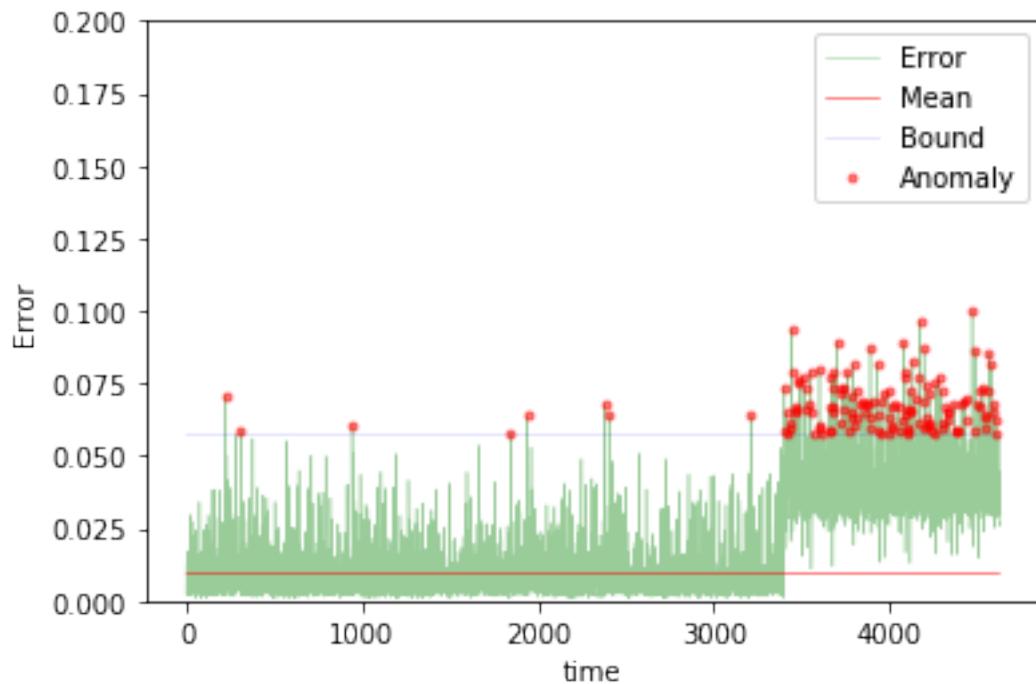
The mean error for nn3_100_normal_ is 0.009437197940212172 for length 4629
Testing on anomaly data.



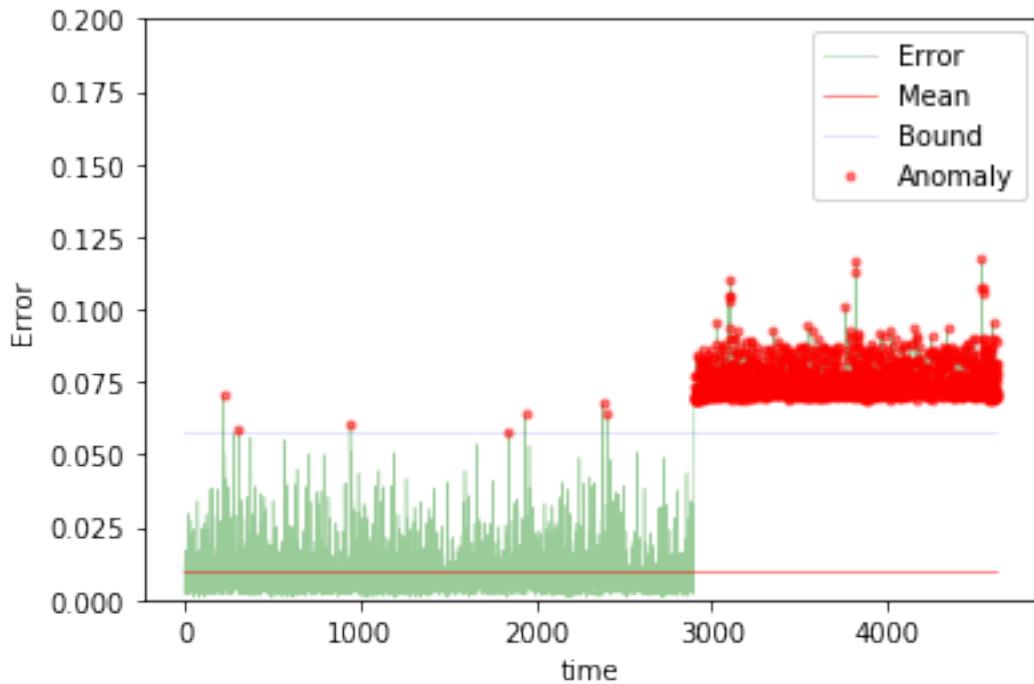
The mean error for nn3_100_anomaly_ is 0.01159210050872634 for length 4629
Testing on different app data.



The mean error for nn3_100_diff_app_ is 0.040634706522613126 for length 4629
Testing on App change synthetic data.



The mean error for nn3_100_app_change_ is 0.017669940622670348 for length 4629
Testing on Net flood synthetic data.



```
The mean error for nn3_100_net_flood_ is 0.03412258322477406 for length 4629
=====
```

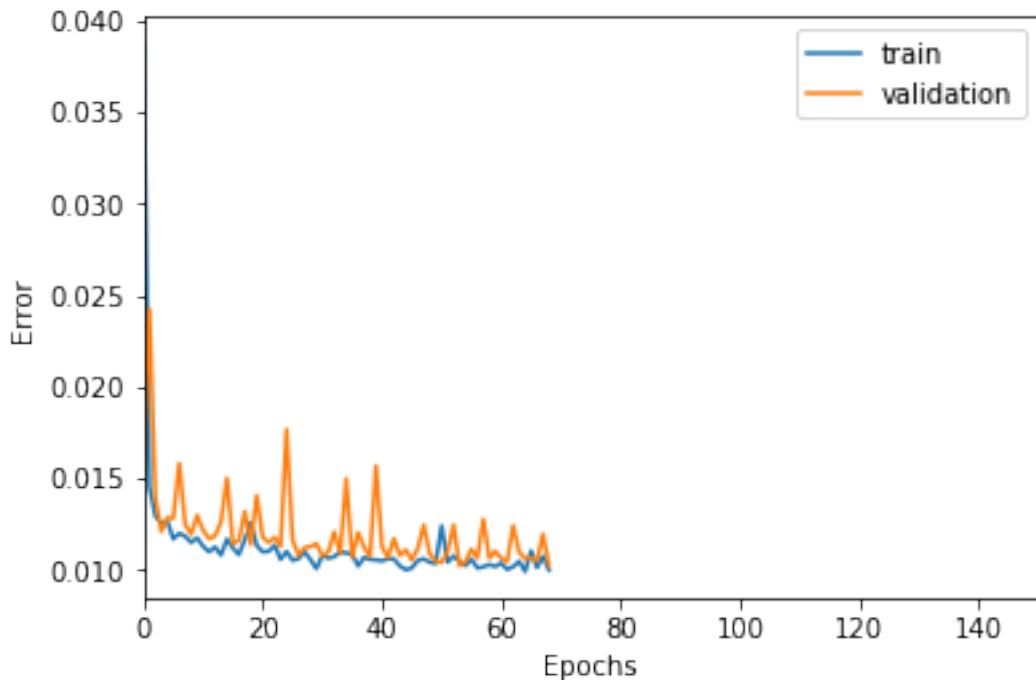
200 steps

```
In [144]: TIMESTEPS = 200
DIM = 29
tgen = flat_generator(X, TIMESTEPS)
vgen = flat_generator(val_X, TIMESTEPS)
name = "nn3_200"

In [145]: 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)

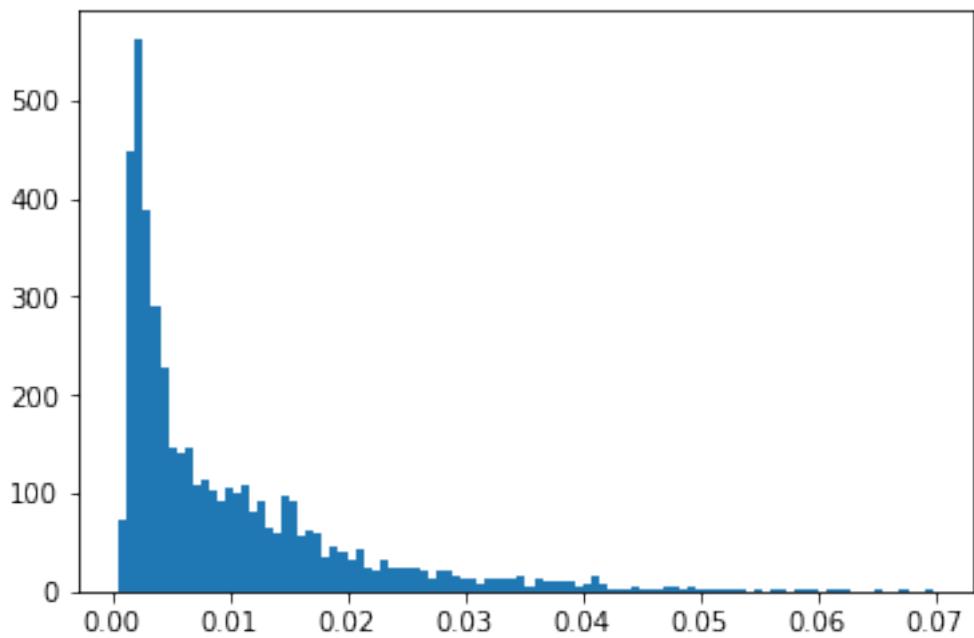
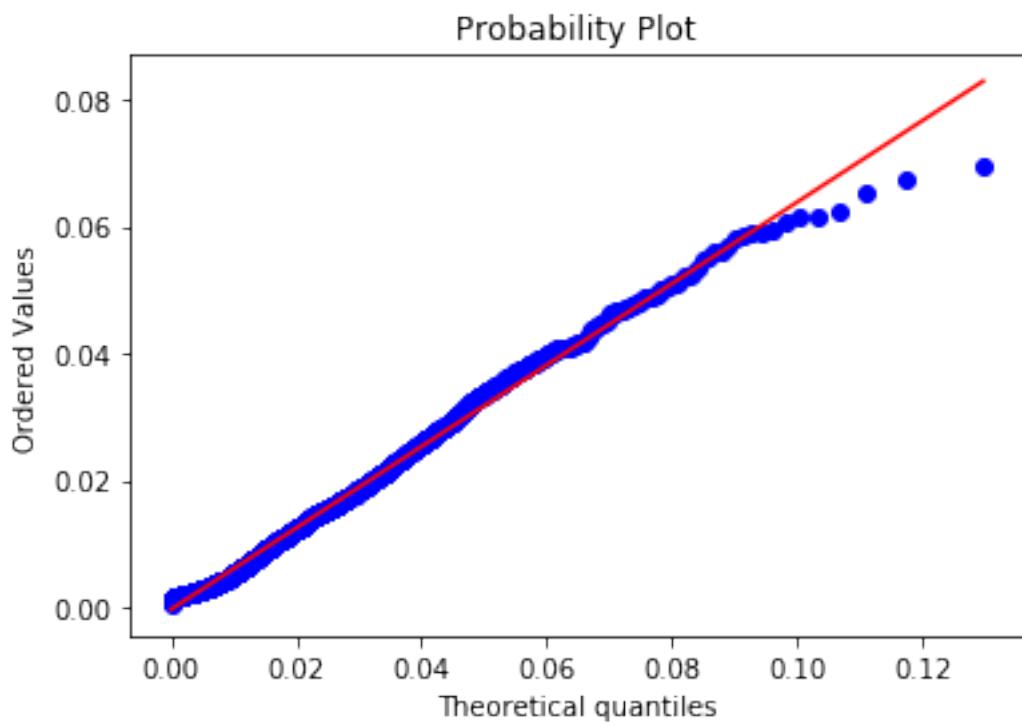
In [146]: model = Model(input_layer, output)
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

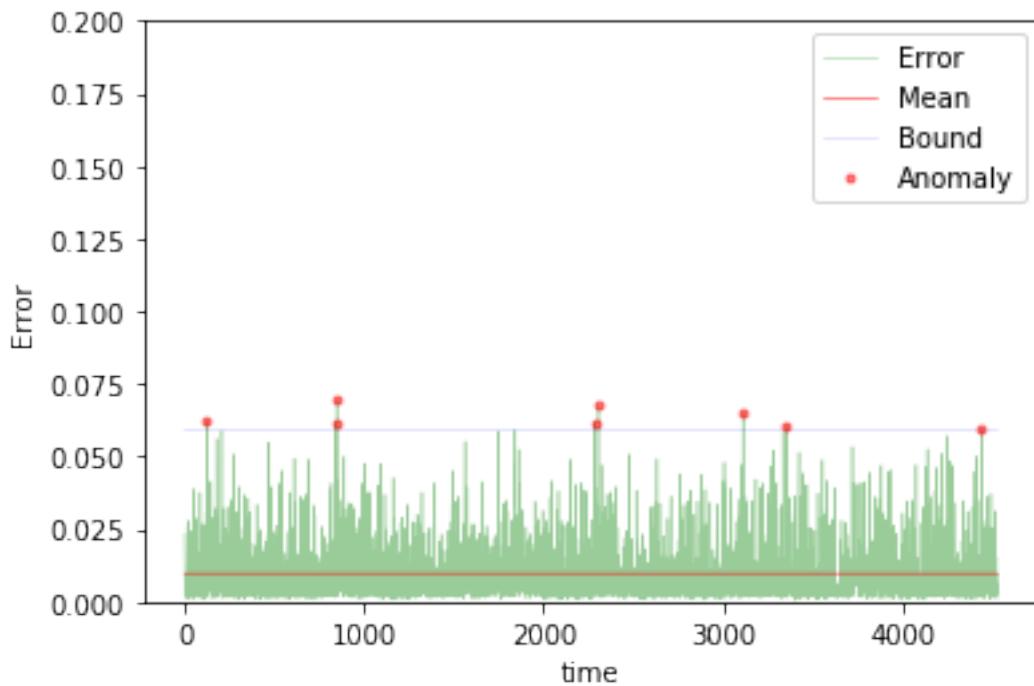
In [147]: train(model, tgen, vgen, name=name)
test(model, name=name, window=TIMESTEPS)
```



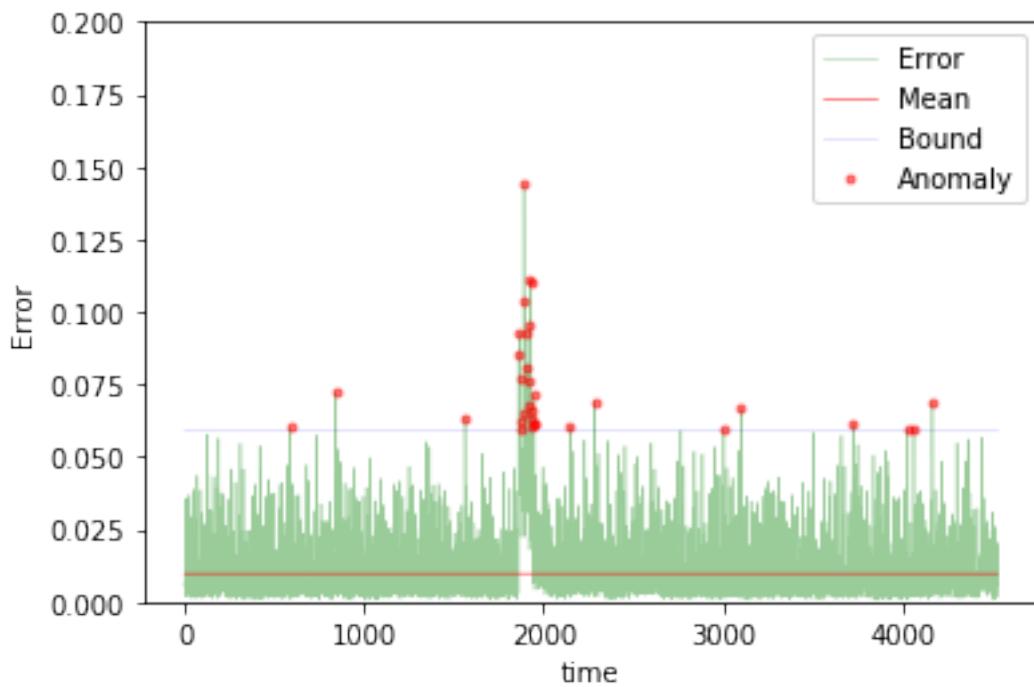
```
Training loss for final epoch is 0.010004253098275512
Validation loss for final epoch is 0.010234042294323445
----- Beginning tests for nn3_200 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

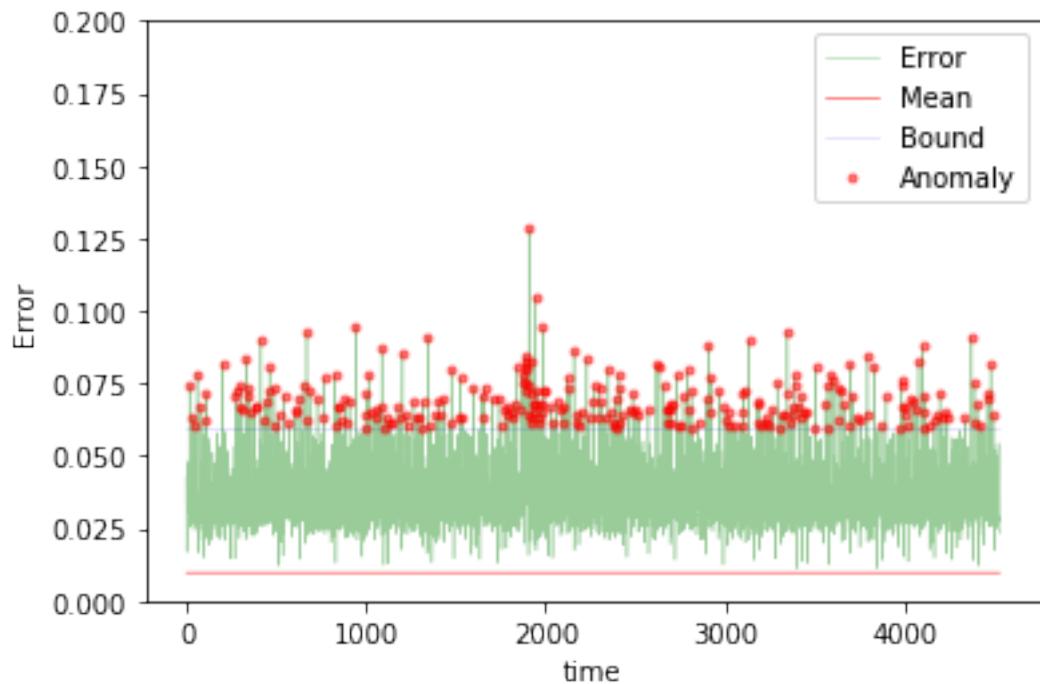




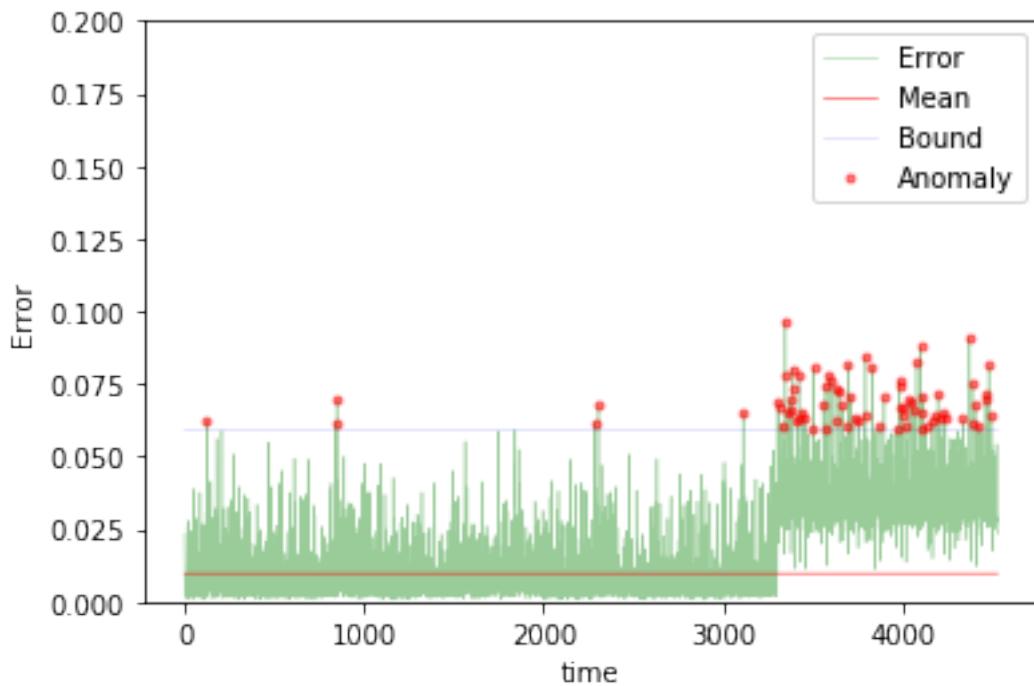
The mean error for nn3_200_normal_ is 0.009802352810755484 for length 4529
Testing on anomaly data.



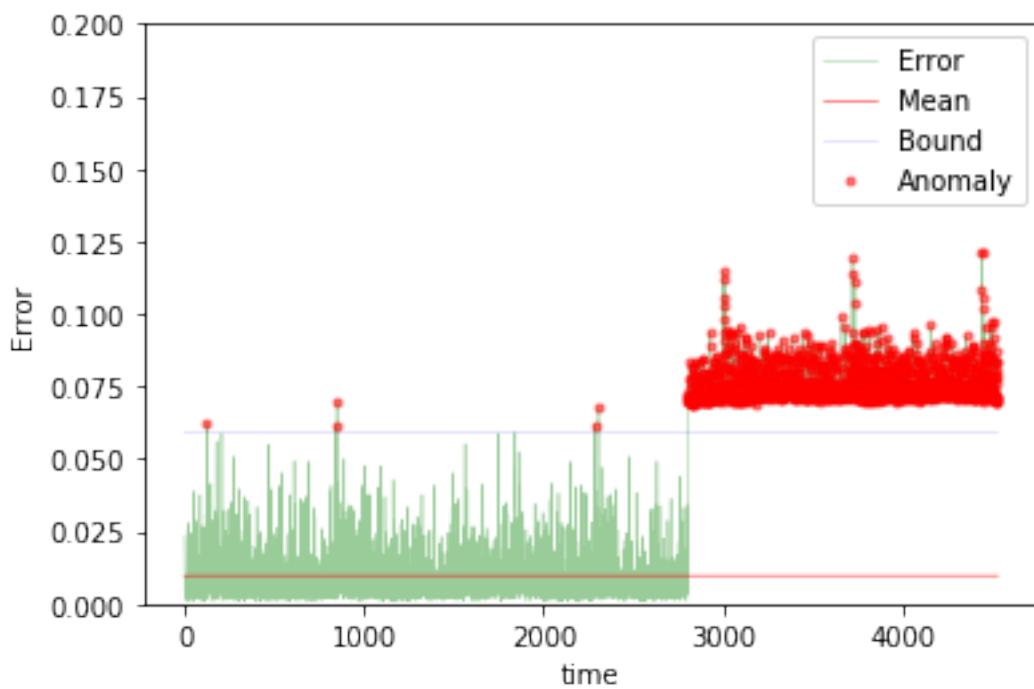
The mean error for nn3_200_anomaly_ is 0.012060567179038651 for length 4529
Testing on different app data.



The mean error for nn3_200_diff_app_ is 0.03713453610766254 for length 4529
Testing on App change synthetic data.



The mean error for nn3_200_app_change_ is 0.01721569976783606 for length 4529
Testing on Net flood synthetic data.

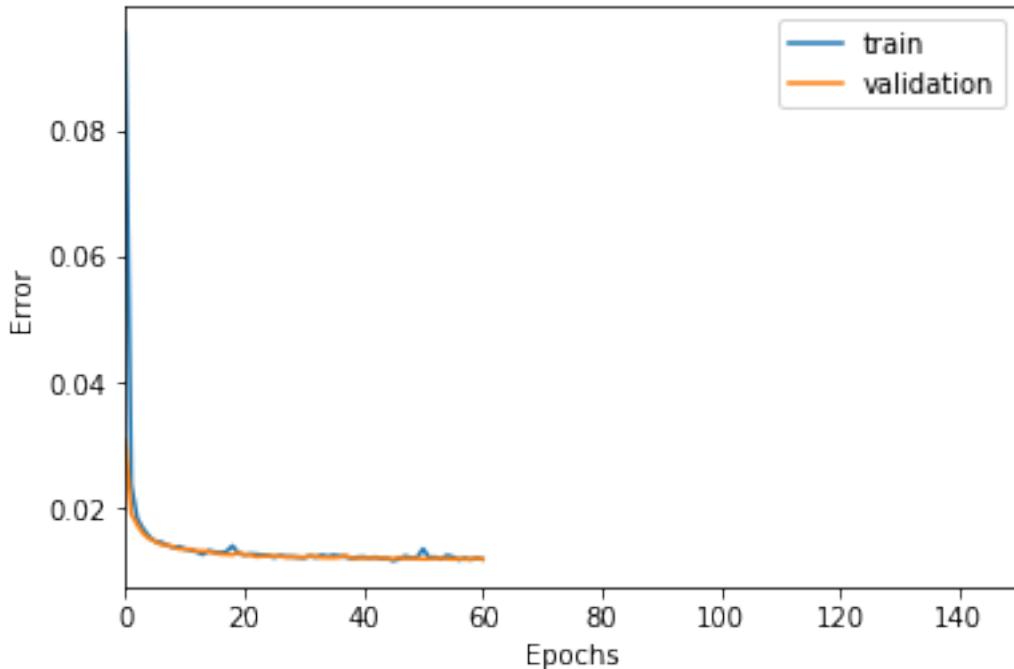


```
The mean error for nn3_200_net_flood_ is 0.03511261143556251 for length 4529  
=====
```

2.1.5 RNN with 1 GRU layers

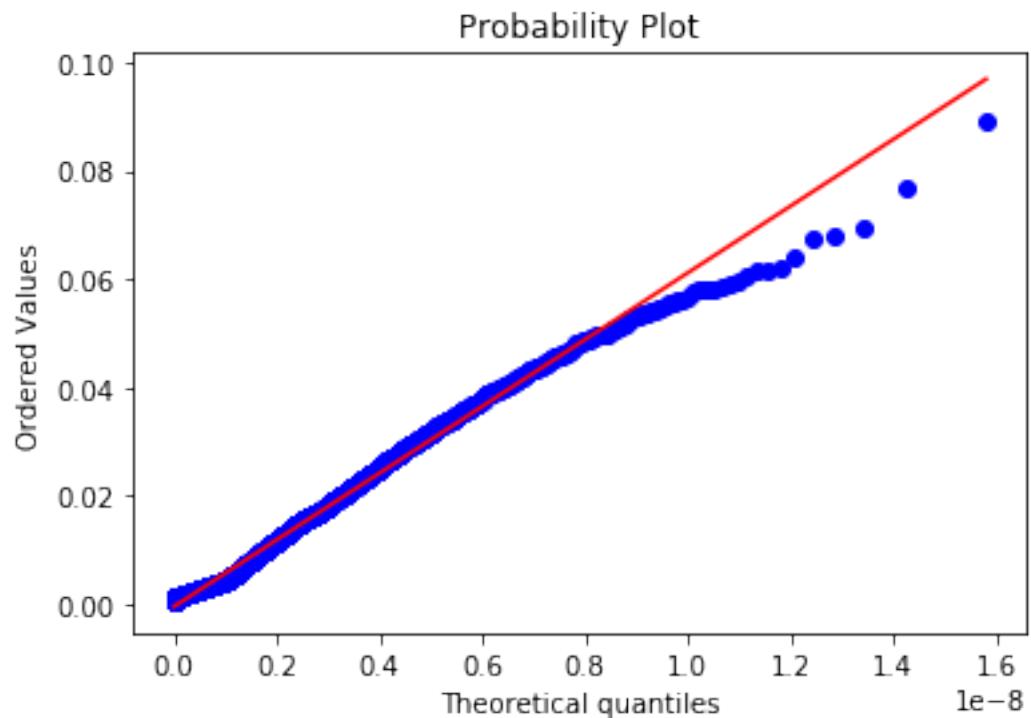
2 steps

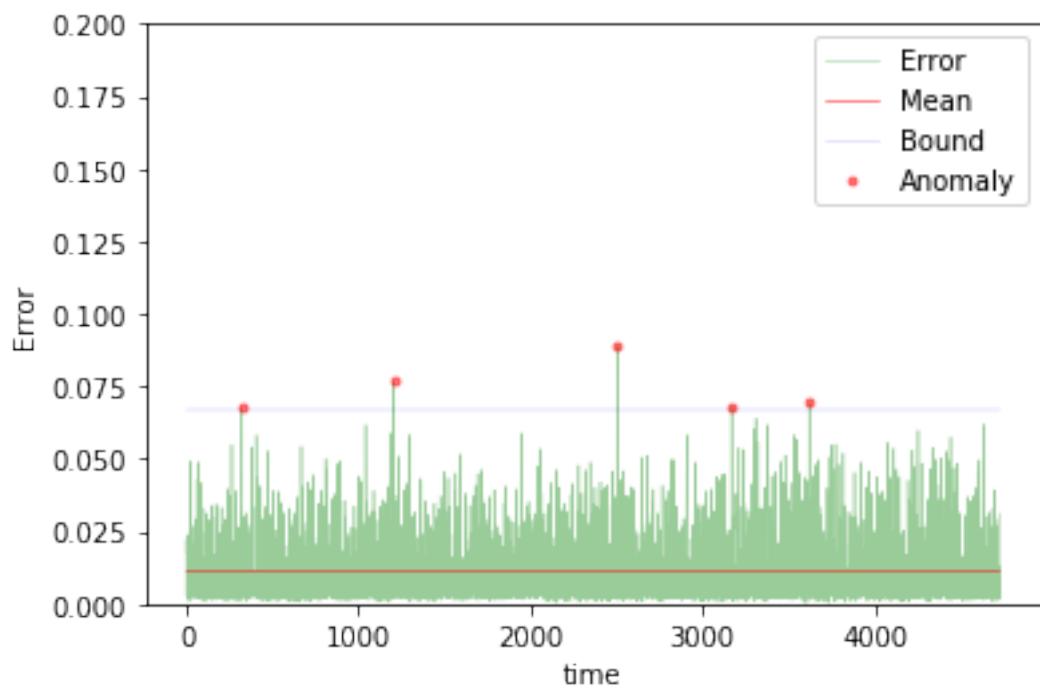
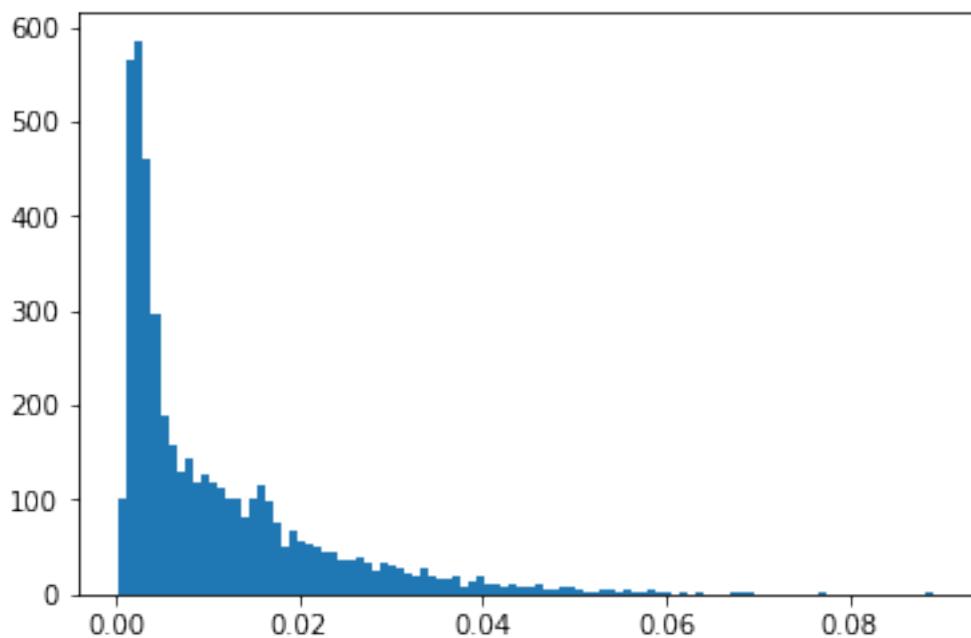
```
In [148]: TIMESTEPS = 2  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS,0)  
vgen = flat_generator(val_X, TIMESTEPS,0)  
name = "gru1_2"  
  
In [149]: input_layer = Input(shape=(TIMESTEPS,DIM))  
hidden = GRU(10, activation='relu')(input_layer)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [150]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [151]: train(model, tgen, vgen, name=name)  
test(model, ravel=0, name=name, window=TIMESTEPS)
```



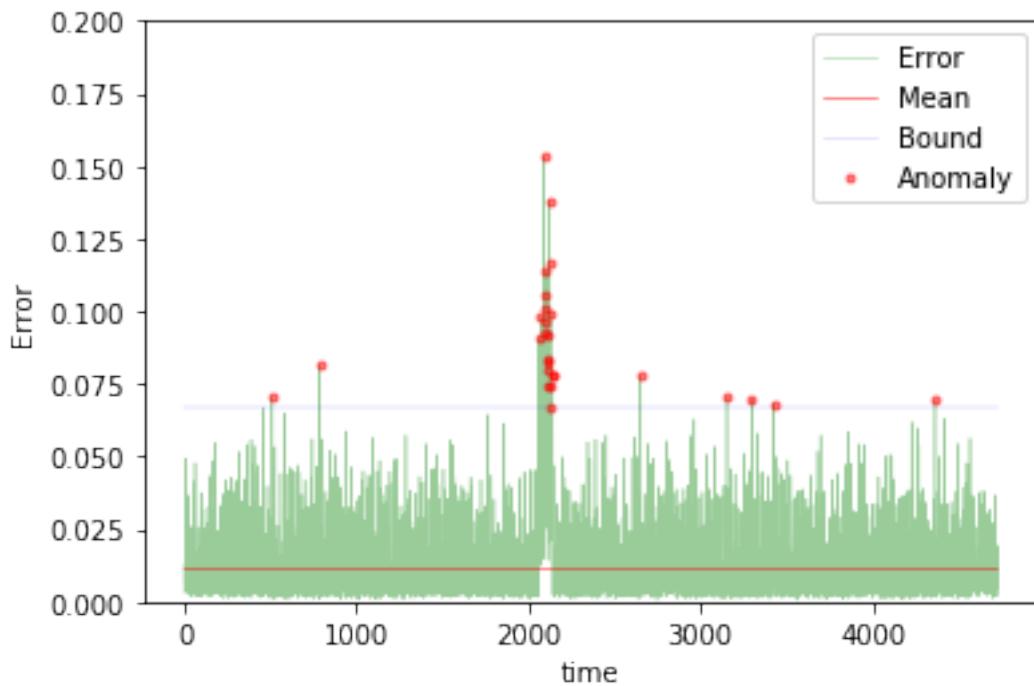
```
Training loss for final epoch is 0.012013664361671545
Validation loss for final epoch is 0.012004661906161345
----- Beginning tests for gru1_2 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

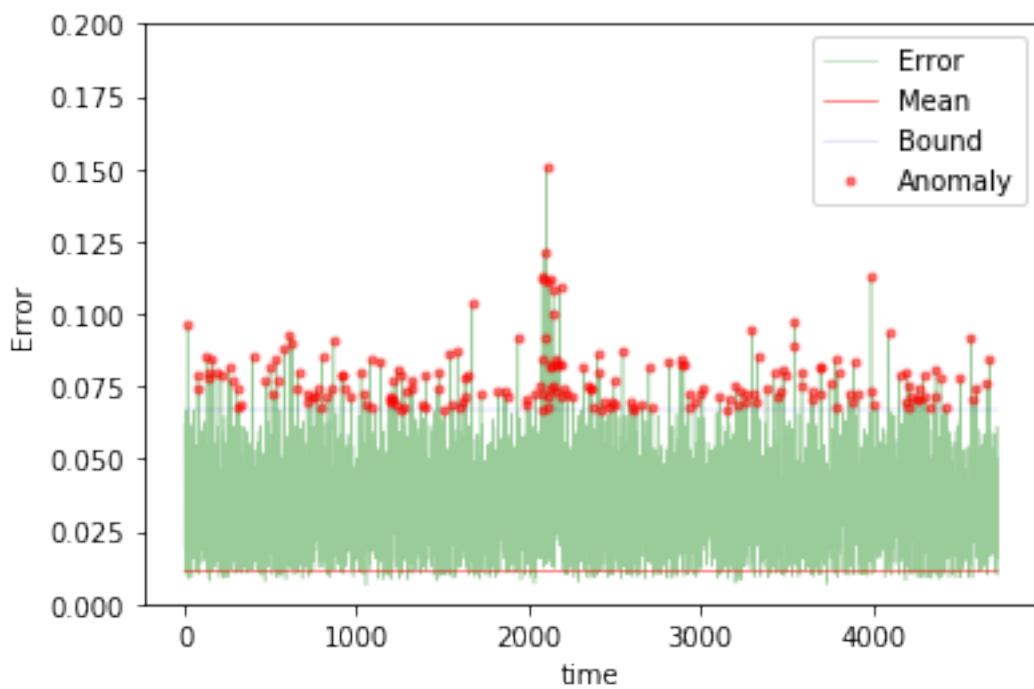




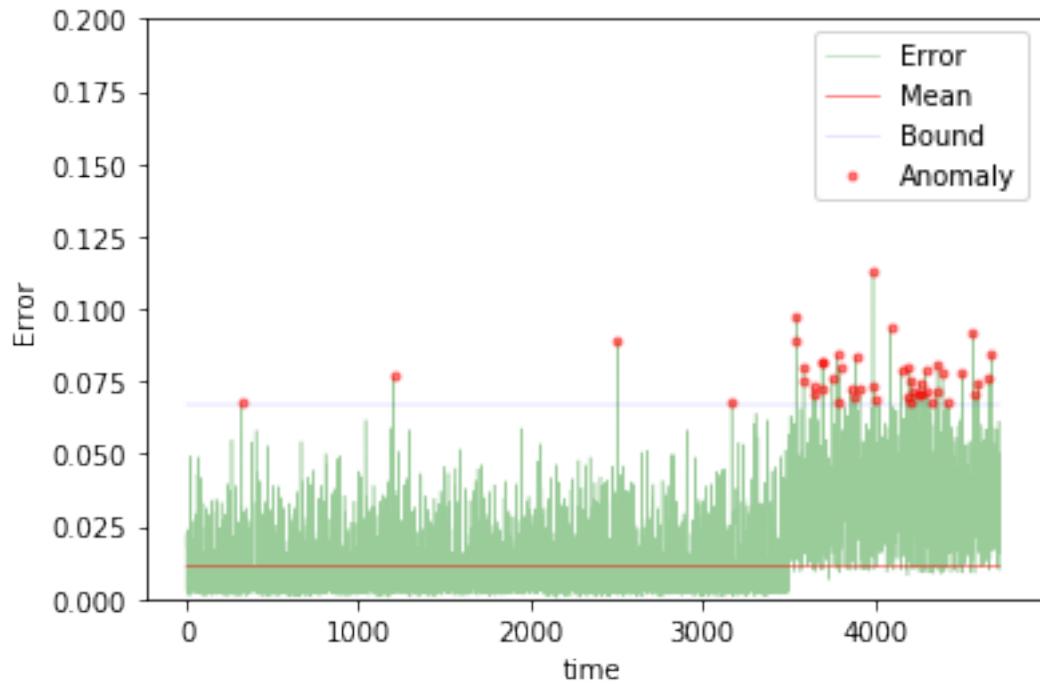
The mean error for gru1_2_normal_ is 0.01108766023548304 for length 4727
Testing on anomaly data.



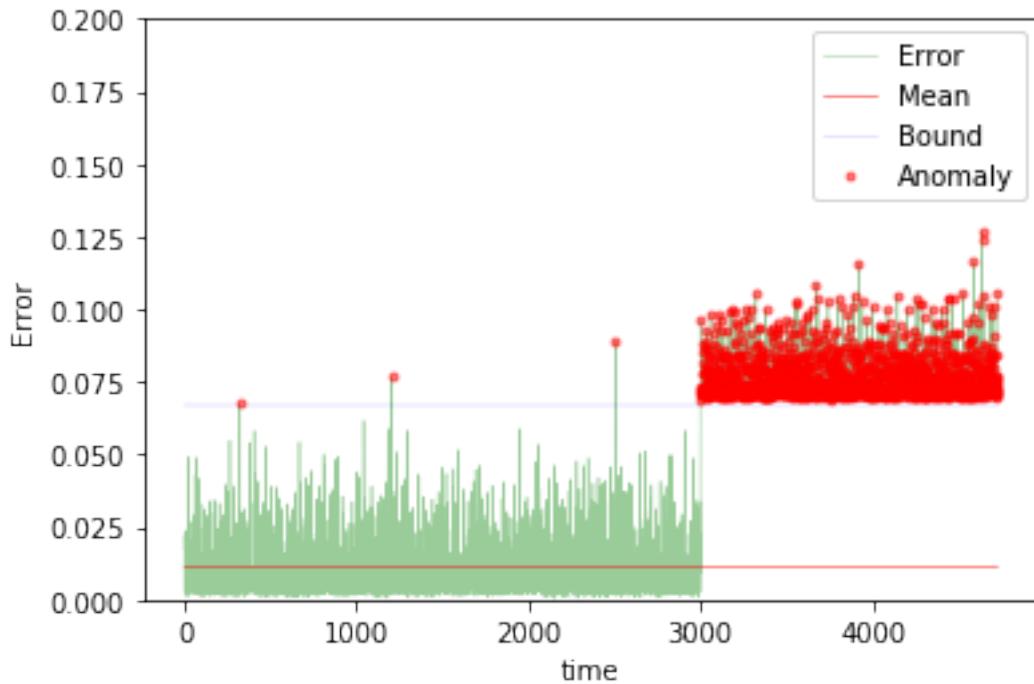
The mean error for gru1_2_anomaly_ is 0.013024567278504815 for length 4727
Testing on different app data.



The mean error for gru1_2_diff_app_ is 0.0350128459736097 for length 4727
Testing on App change synthetic data.



The mean error for gru1_2_app_change_ is 0.01720342401415241 for length 4727
Testing on Net flood synthetic data.



```
The mean error for gru1_2_net_flood_ is 0.034936688893181946 for length 4727
=====
```

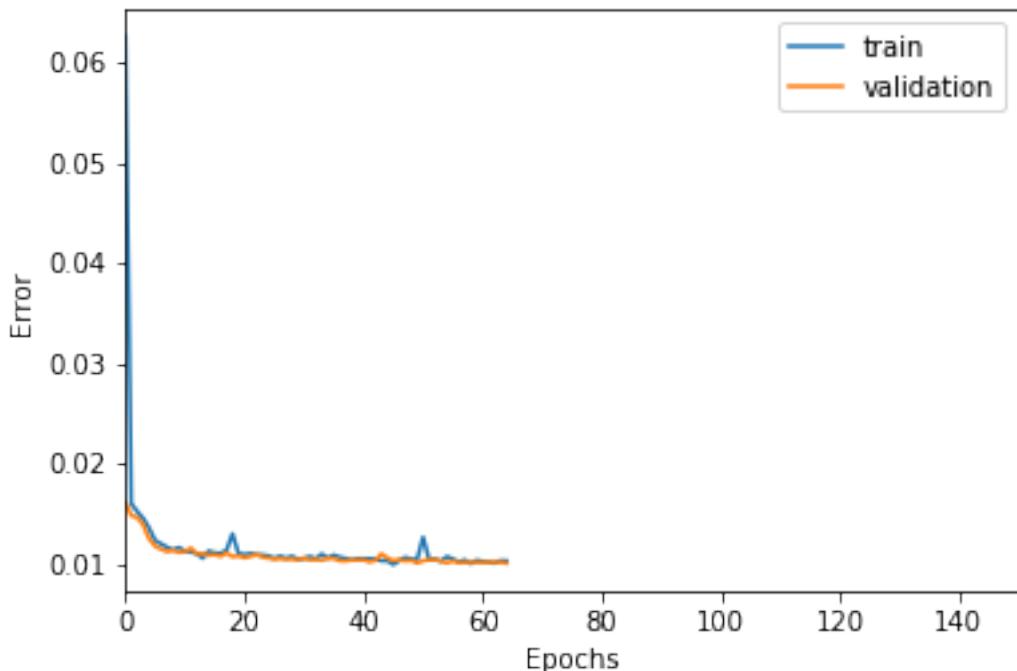
5 steps

```
In [152]: TIMESTEPS = 5
DIM = 29
tgen = flat_generator(X, TIMESTEPS, 0)
vgen = flat_generator(val_X, TIMESTEPS, 0)
name = "gru1_5"

In [153]: input_layer = Input(shape=(TIMESTEPS,DIM))
hidden = GRU(10, activation='relu')(input_layer)
output = Dense(DIM, activation='sigmoid')(hidden)

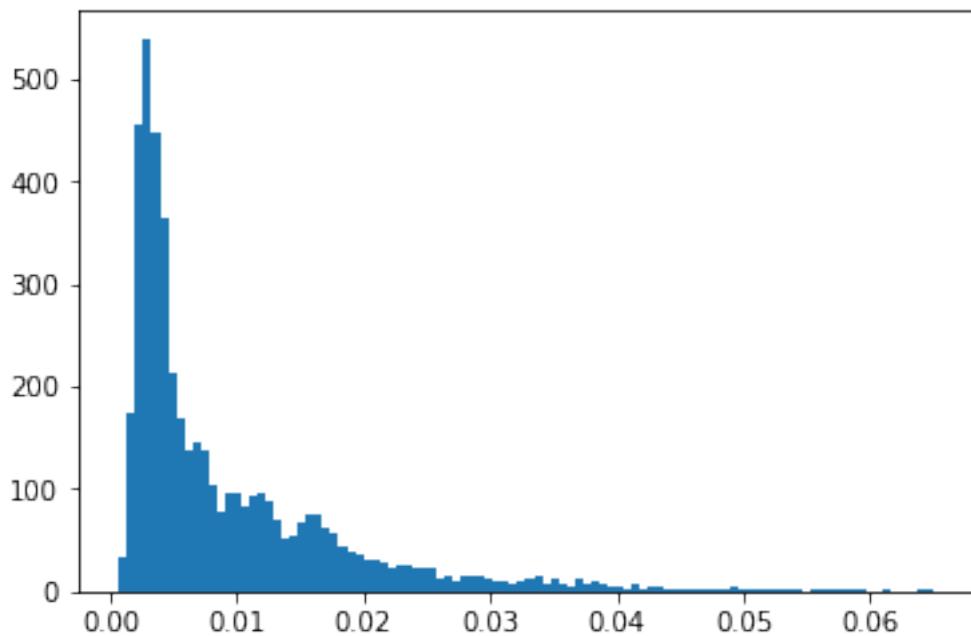
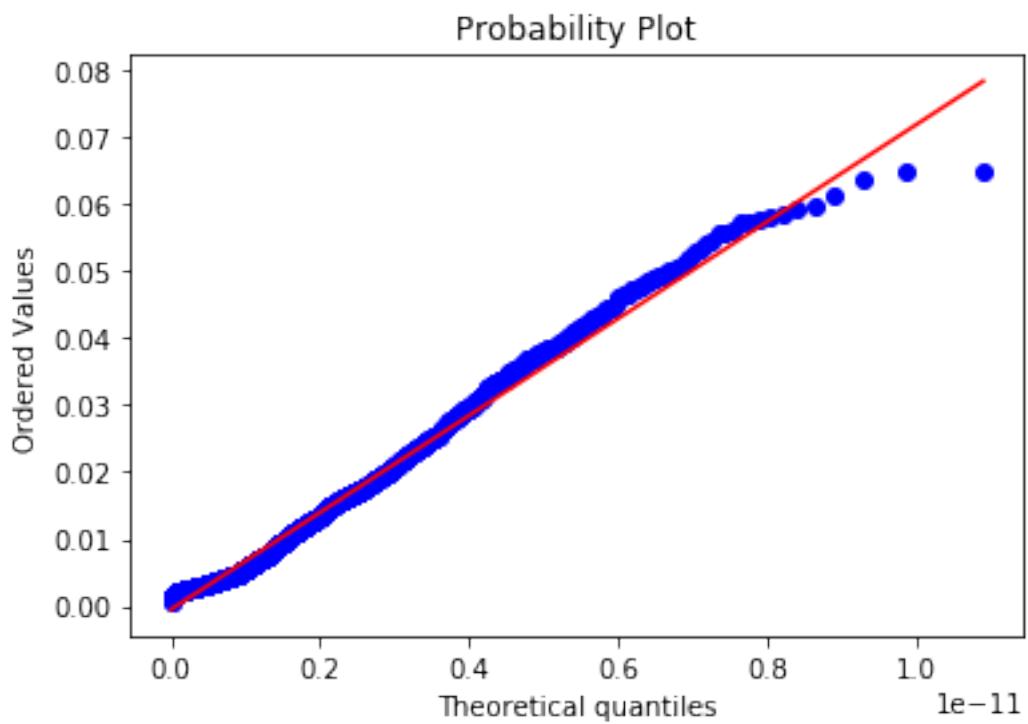
In [154]: model = Model(input_layer, output)
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

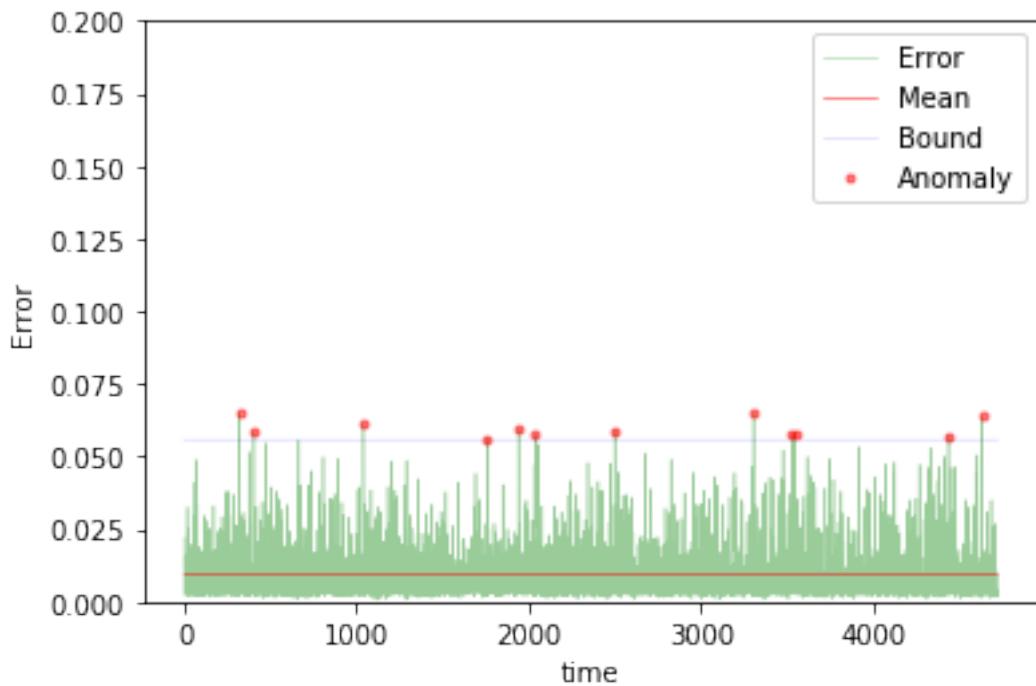
In [155]: train(model, tgen, vgen, name=name)
test(model, ravel=0, name=name, window=TIMESTEPS)
```



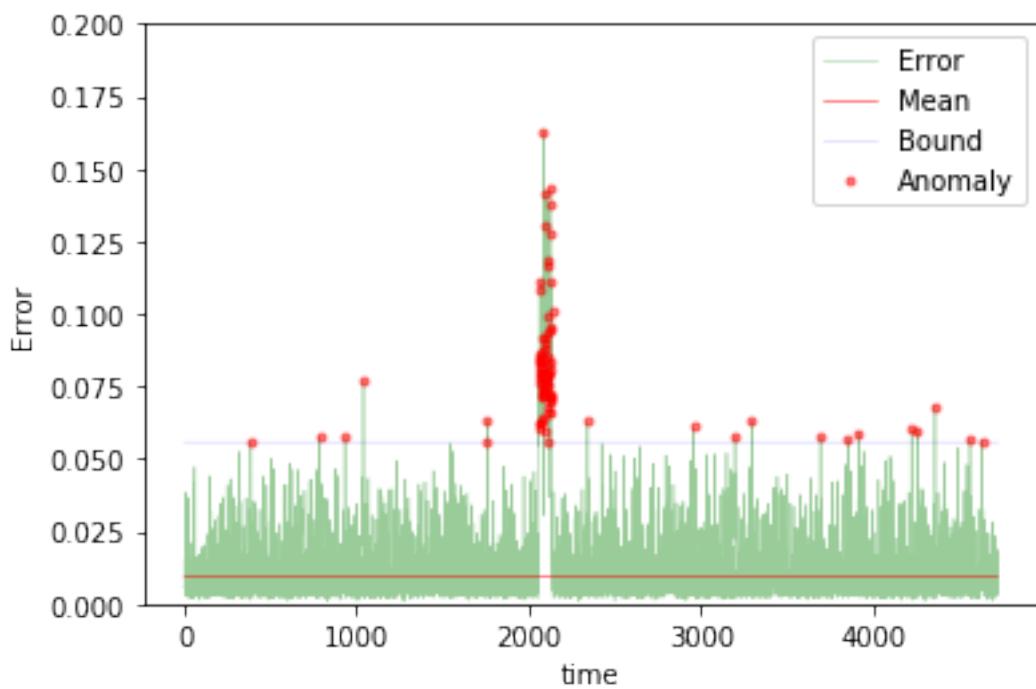
```
Training loss for final epoch is 0.010297565924585797
Validation loss for final epoch is 0.010062597972340883
----- Beginning tests for gru1_5 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWar
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

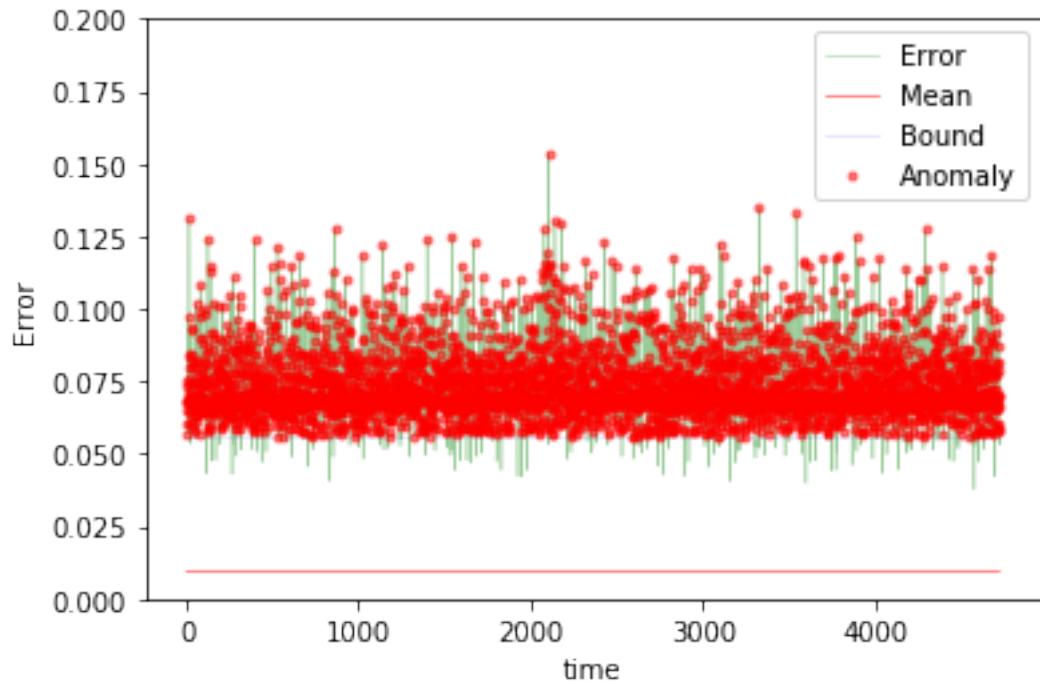




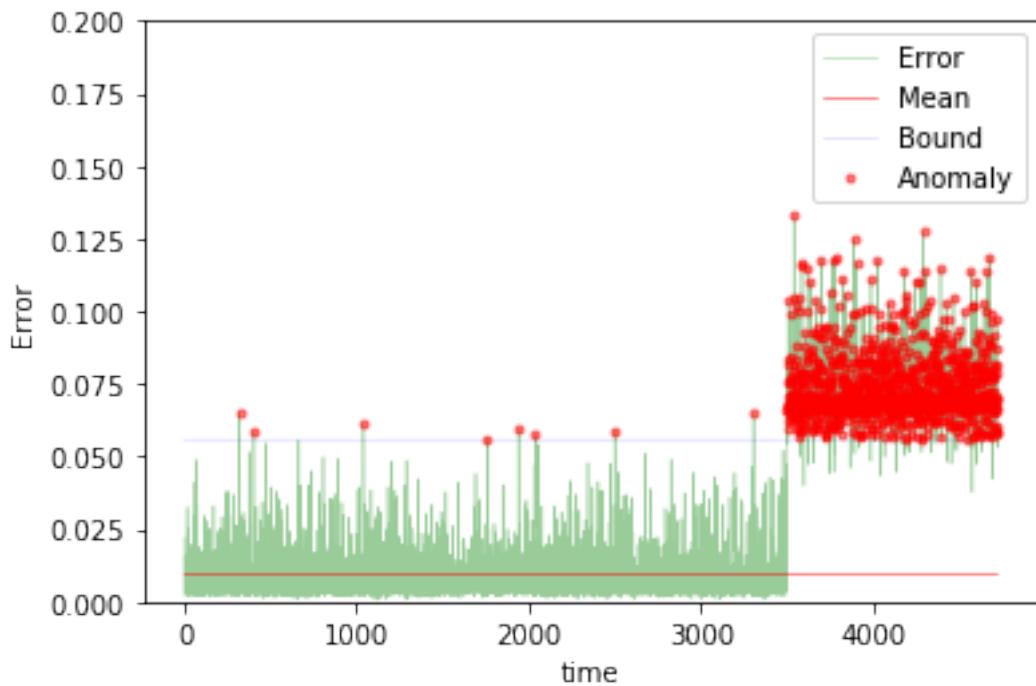
The mean error for gru1_5_normal_ is 0.009539952078709928 for length 4724
Testing on anomaly data.



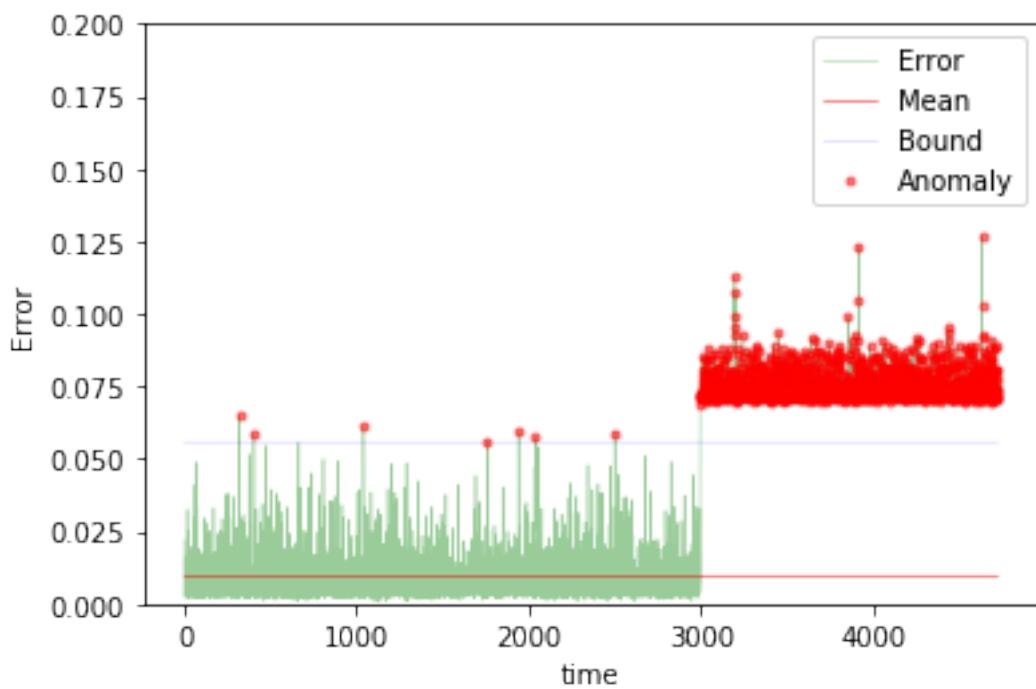
The mean error for gru1_5_anomaly_ is 0.011753185935487997 for length 4724
Testing on different app data.



The mean error for gru1_5_diff_app_ is 0.07317166952625104 for length 4724
Testing on App change synthetic data.



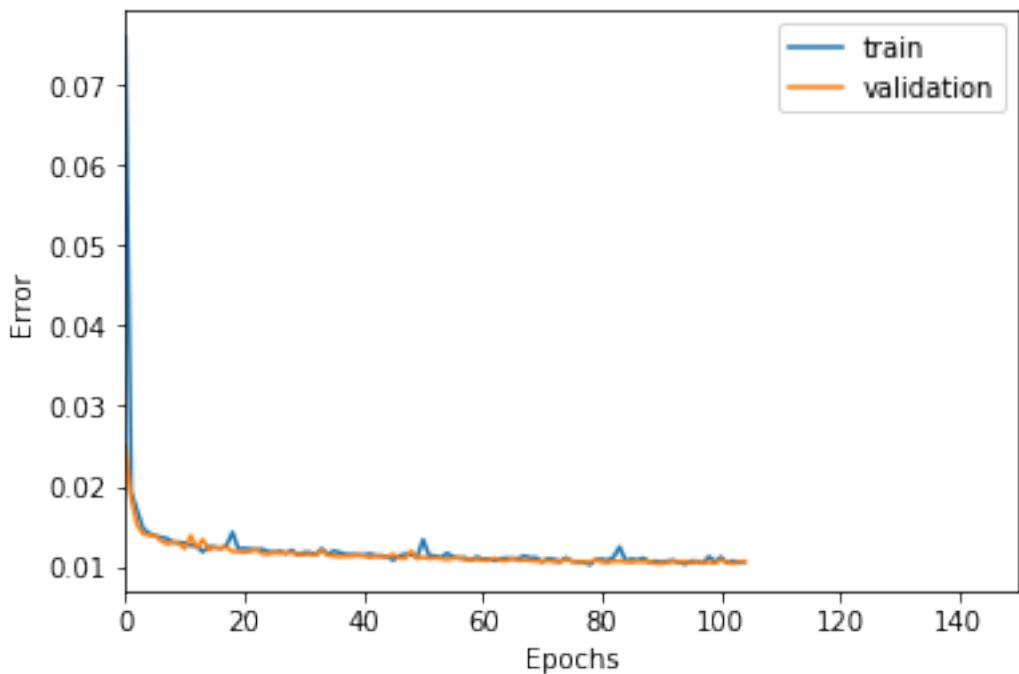
The mean error for gru1_5_app_change_ is 0.02595092763251627 for length 4724
Testing on Net flood synthetic data.



```
The mean error for gru1_5_net_flood_ is 0.03368768629062454 for length 4724  
=====
```

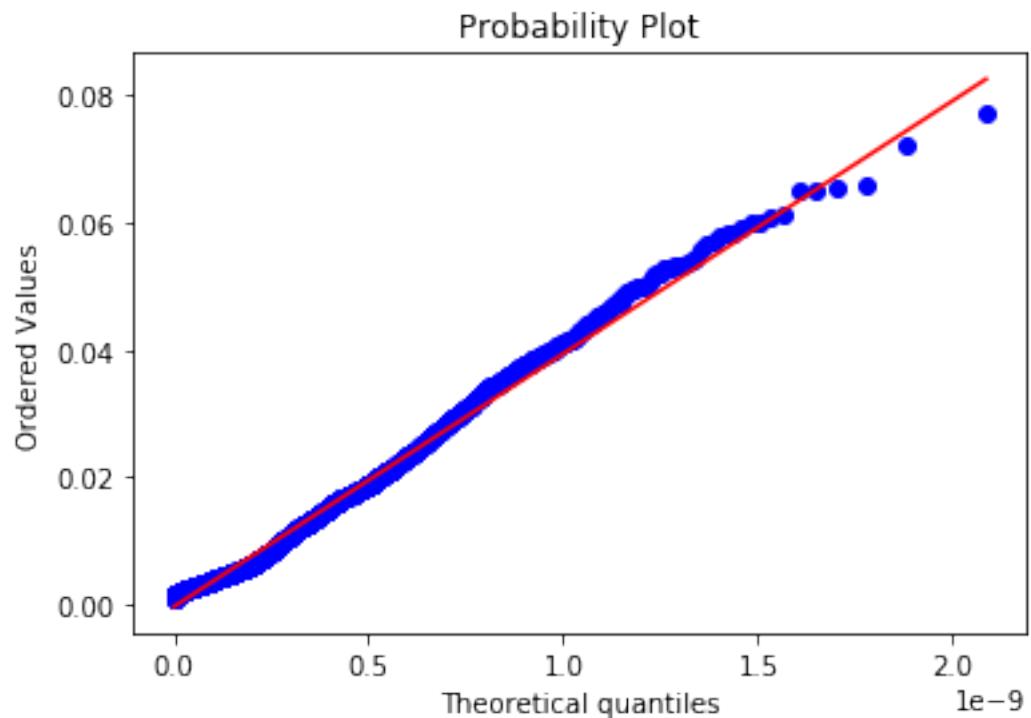
10 steps

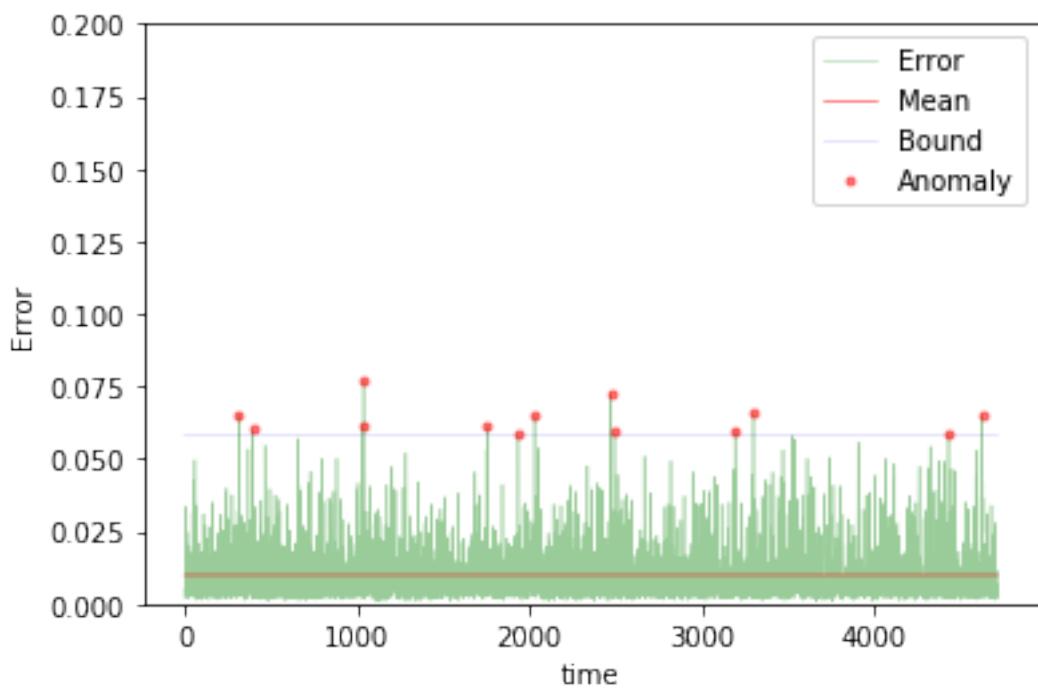
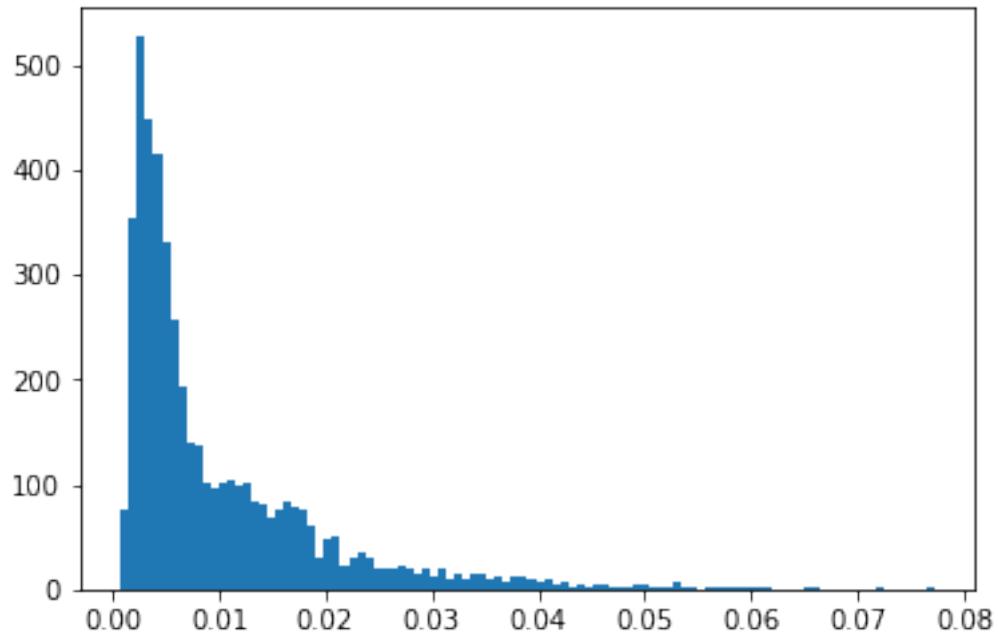
```
In [156]: TIMESTEPS = 10  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS, 0)  
vgen = flat_generator(val_X, TIMESTEPS, 0)  
name = "gru1_10"  
  
In [157]: input_layer = Input(shape=(TIMESTEPS,DIM))  
hidden = GRU(10, activation='relu')(input_layer)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [158]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [159]: train(model, tgen, vgen, name=name)  
test(model, ravel=0, name=name, window=TIMESTEPS)
```



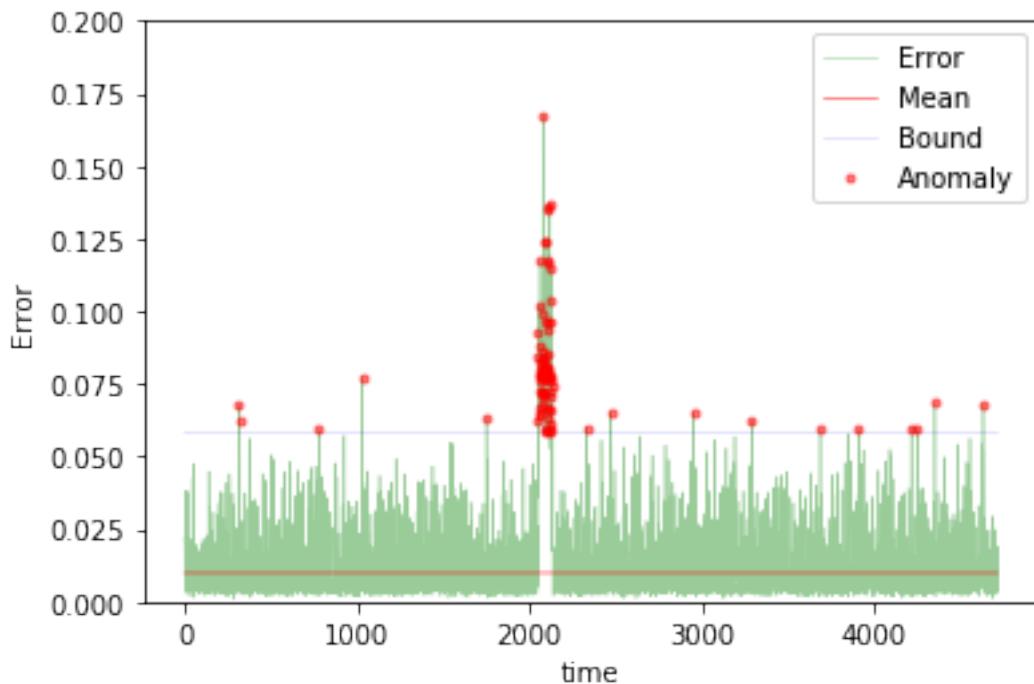
```
Training loss for final epoch is 0.010605071376310662
Validation loss for final epoch is 0.01062025333289057
----- Beginning tests for gru1_10 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

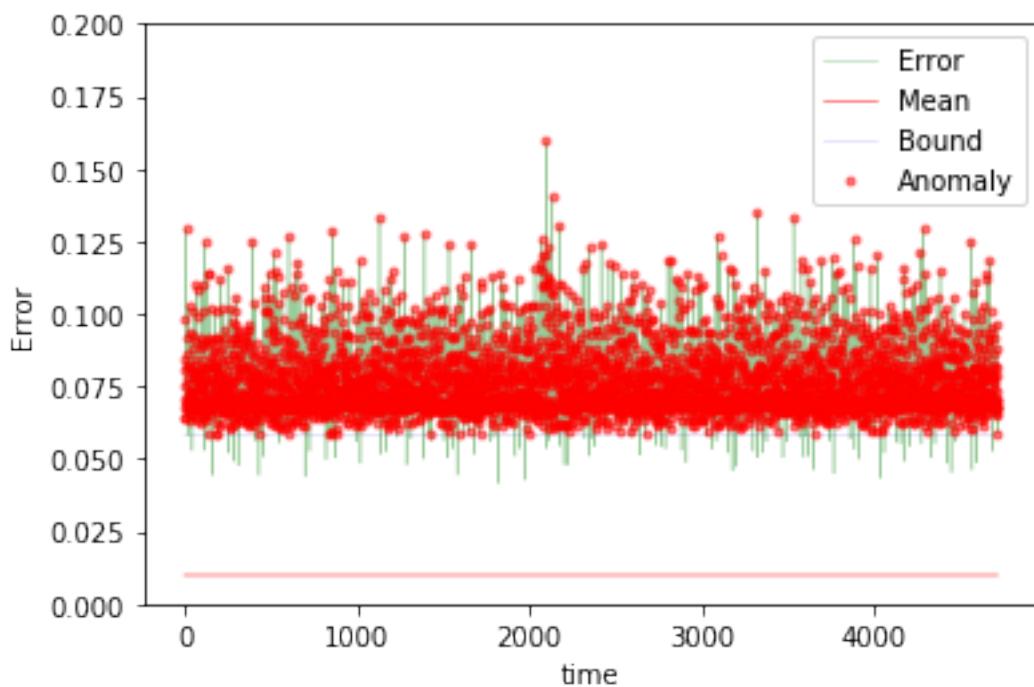




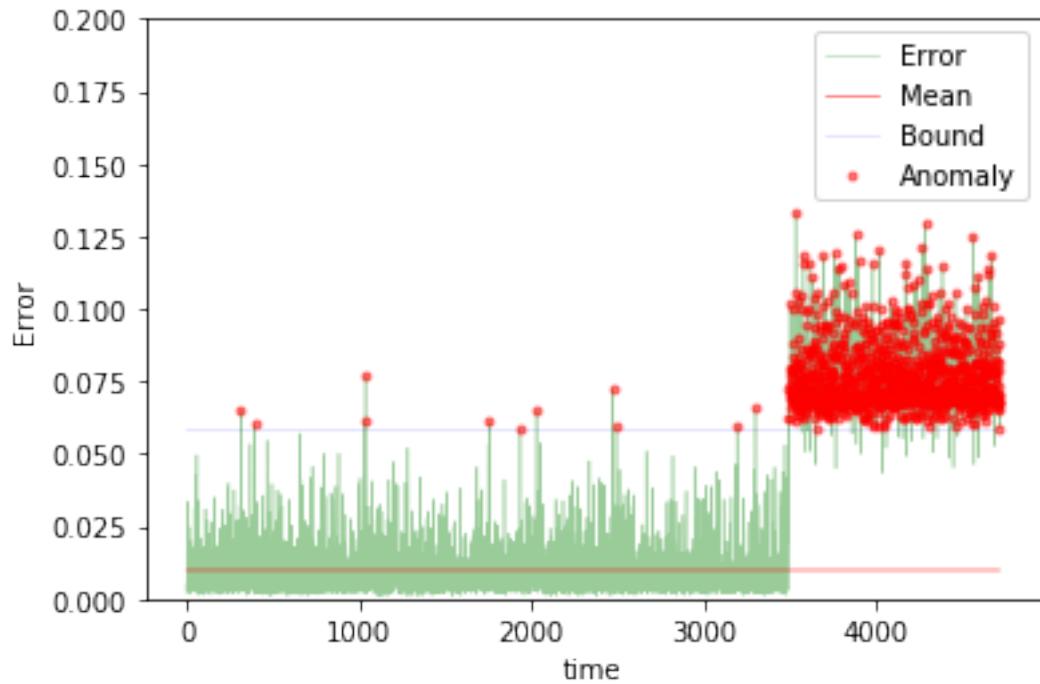
The mean error for gru1_10_normal_ is 0.009950884777217056 for length 4719
Testing on anomaly data.



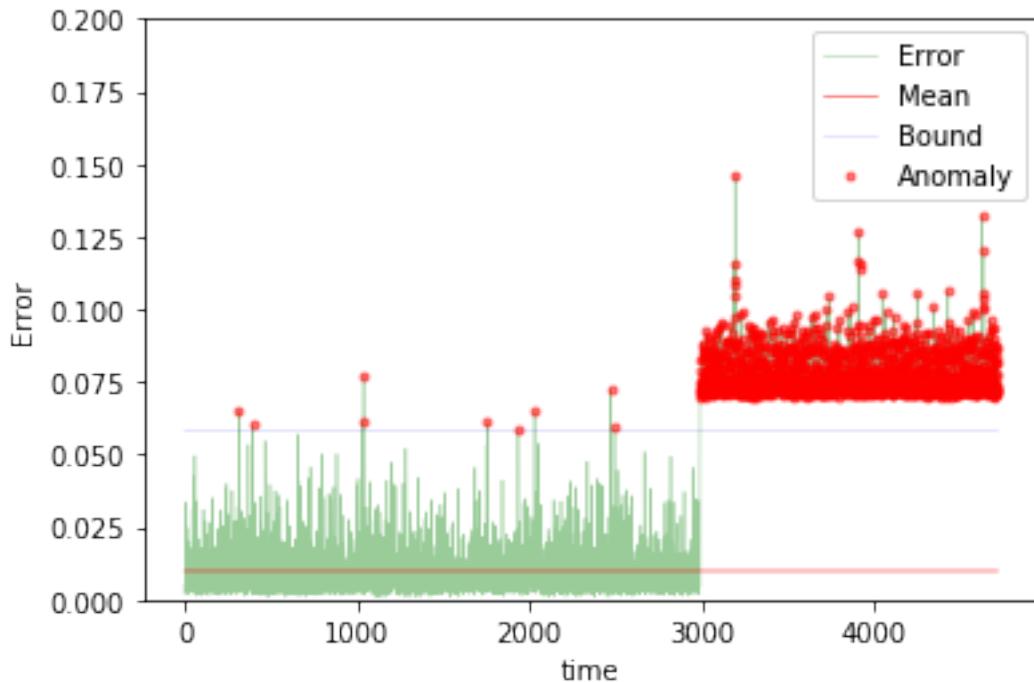
The mean error for gru1_10_anomaly_ is 0.012499780561123911 for length 4719
Testing on different app data.



The mean error for gru1_10_diff_app_ is 0.07601305830216536 for length 4719
Testing on App change synthetic data.



The mean error for gru1_10_app_change_ is 0.026995977861985616 for length 4719
Testing on Net flood synthetic data.



```
The mean error for gru1_10_net_flood_ is 0.034831706679006 for length 4719
=====
```

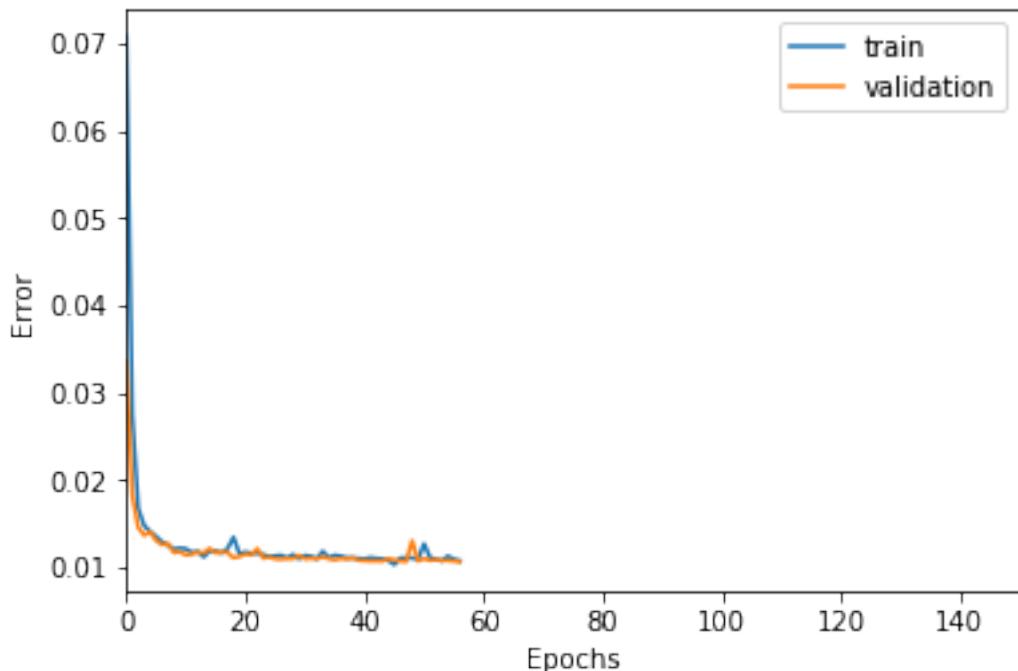
20 steps

```
In [160]: TIMESTEPS = 20
DIM = 29
tgen = flat_generator(X, TIMESTEPS,0)
vgen = flat_generator(val_X, TIMESTEPS,0)
name = "gru1_20"

In [161]: input_layer = Input(shape=(TIMESTEPS,DIM))
hidden = GRU(10, activation='relu')(input_layer)
output = Dense(DIM, activation='sigmoid')(hidden)

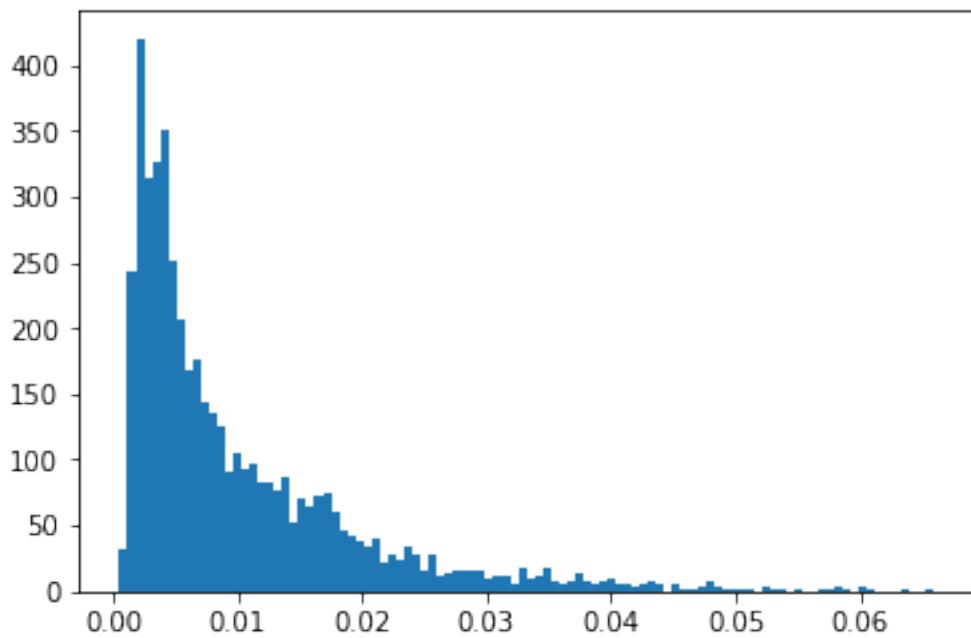
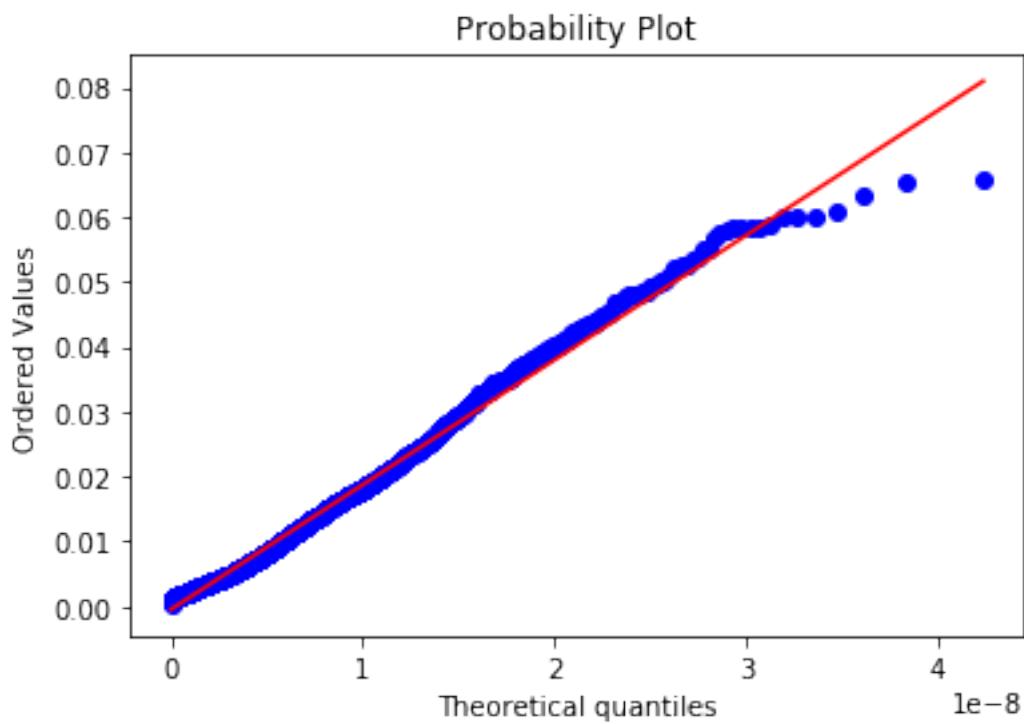
In [162]: model = Model(input_layer, output)
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

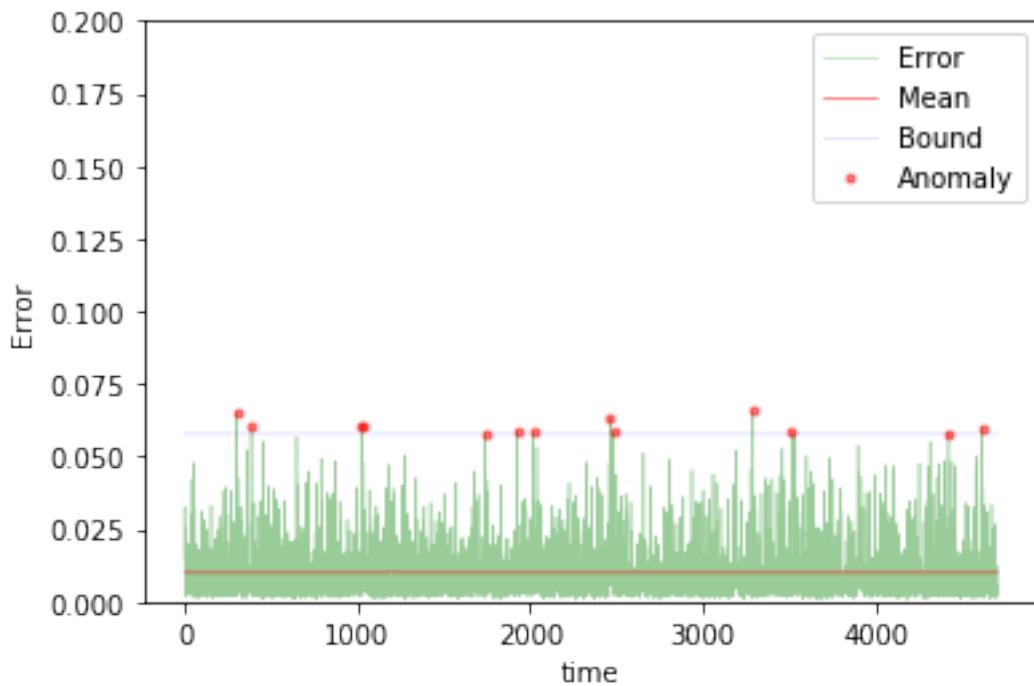
In [163]: train(model, tgen, vgen, name=name)
test(model, ravel=0, name=name, window=TIMESTEPS)
```



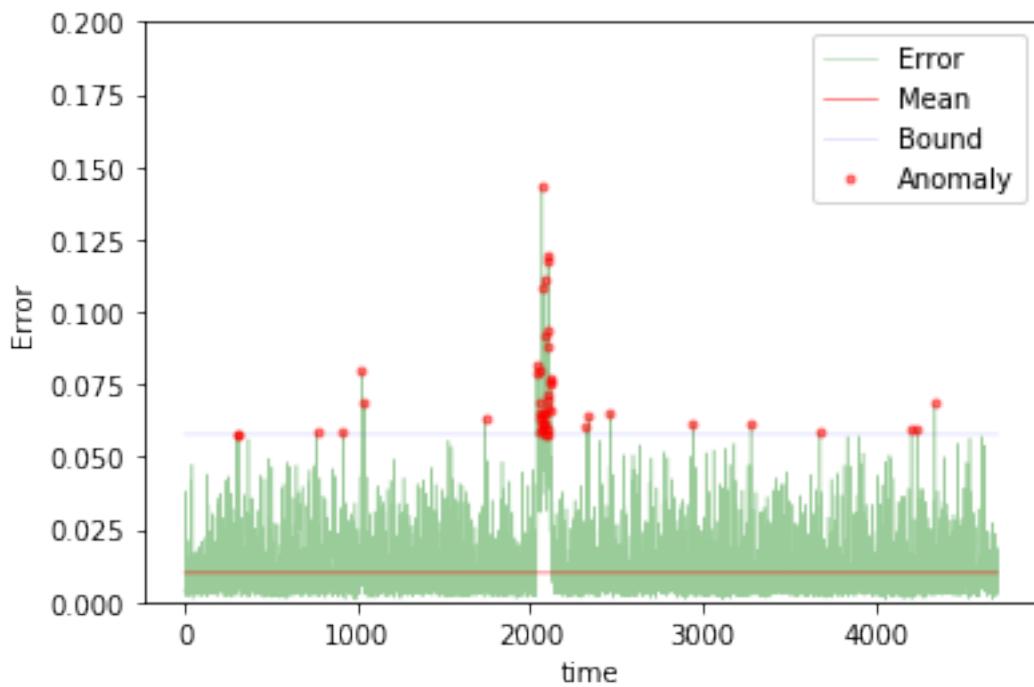
```
Training loss for final epoch is 0.010696047670207917
Validation loss for final epoch is 0.010592327292775735
----- Beginning tests for gru1_20 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
    warnings.warn(msg, RuntimeWarning)
```

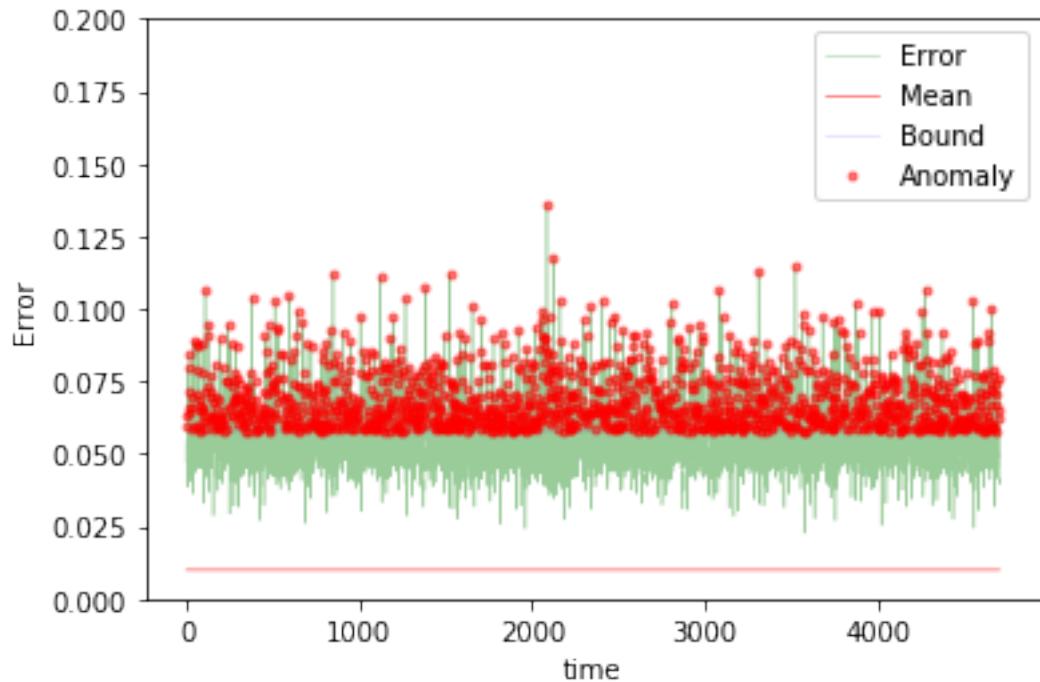




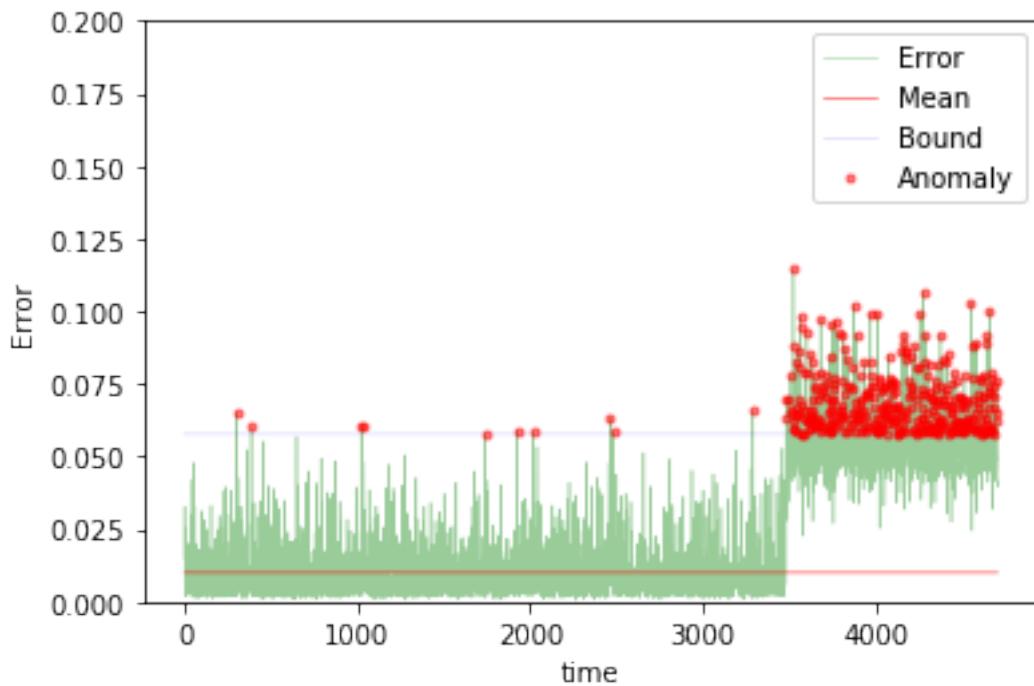
The mean error for gru1_20_normal_ is 0.010028442061040829 for length 4709
Testing on anomaly data.



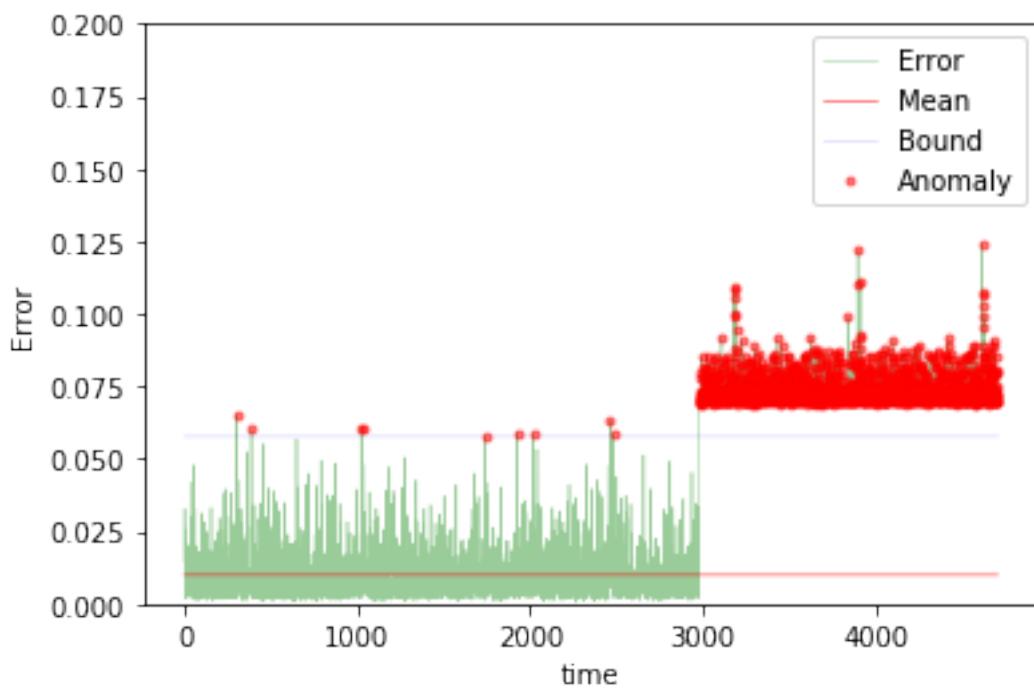
The mean error for gru1_20_anomaly_ is 0.011933373163449869 for length 4709
Testing on different app data.



The mean error for gru1_20_diff_app_ is 0.05515205139705226 for length 4709
Testing on App change synthetic data.



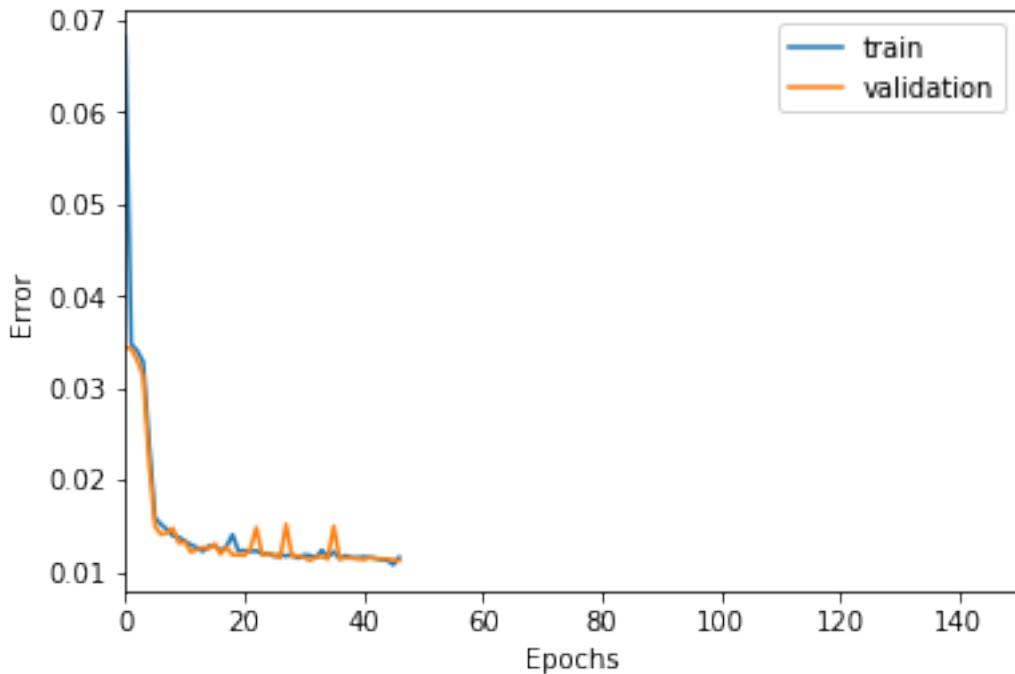
The mean error for gru1_20_app_change_ is 0.021686329200008358 for length 4709
Testing on Net flood synthetic data.



```
The mean error for gru1_20_net_flood_ is 0.03365387760949807 for length 4709  
=====
```

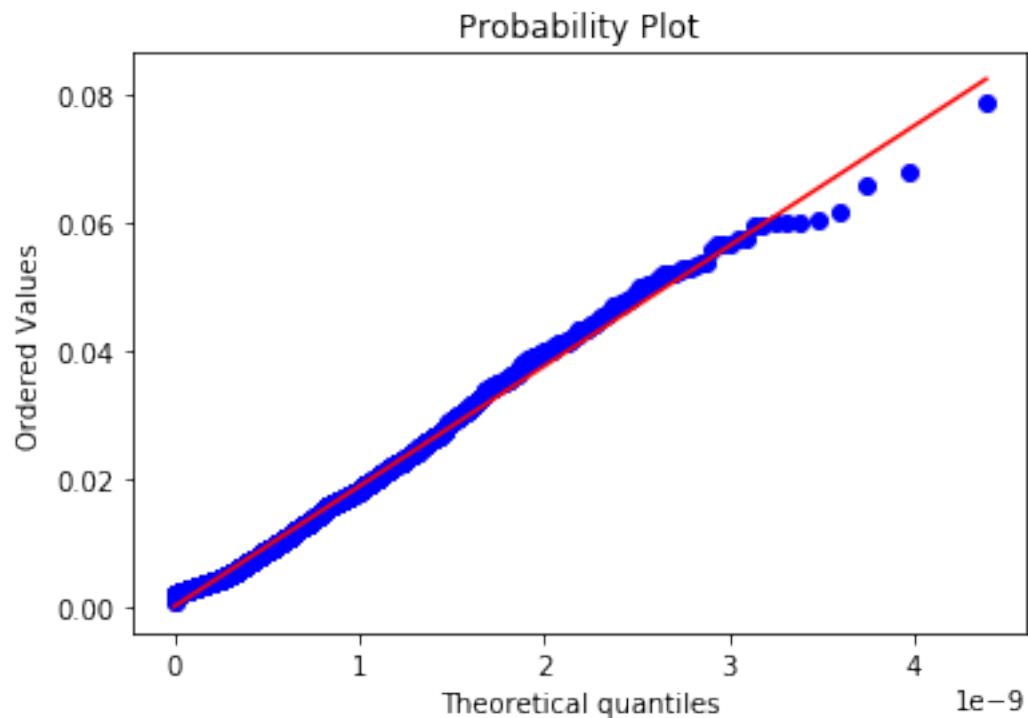
50 steps

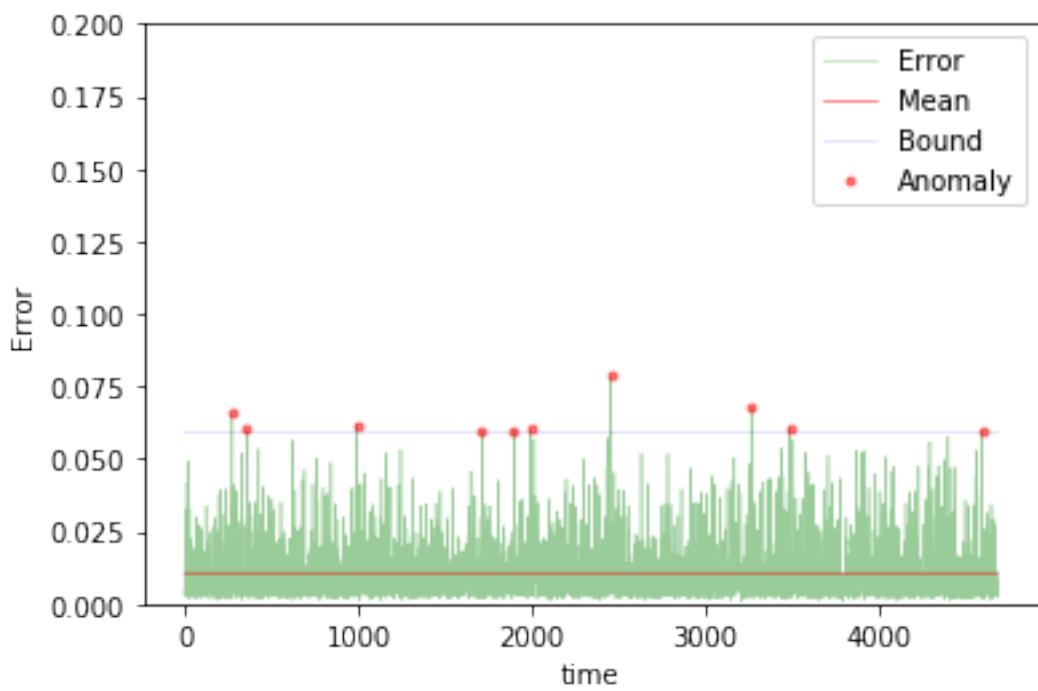
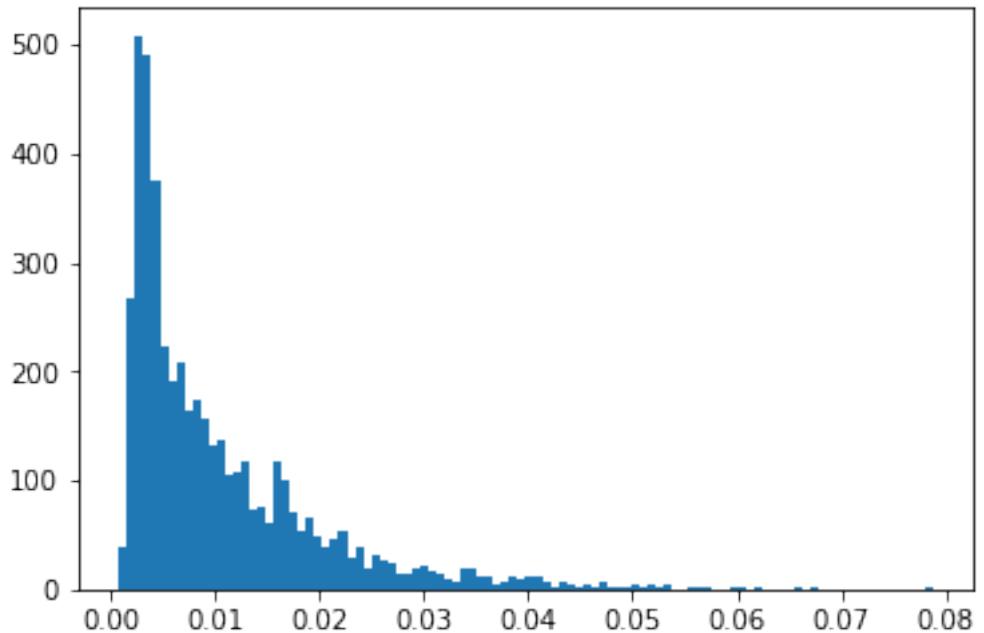
```
In [164]: TIMESTEPS = 50  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS,0)  
vgen = flat_generator(val_X, TIMESTEPS,0)  
name = "gru1_50"  
  
In [165]: input_layer = Input(shape=(TIMESTEPS,DIM))  
hidden = GRU(10, activation='relu')(input_layer)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [166]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [167]: train(model, tgen, vgen, name=name)  
test(model, ravel=0, name=name, window=TIMESTEPS)
```



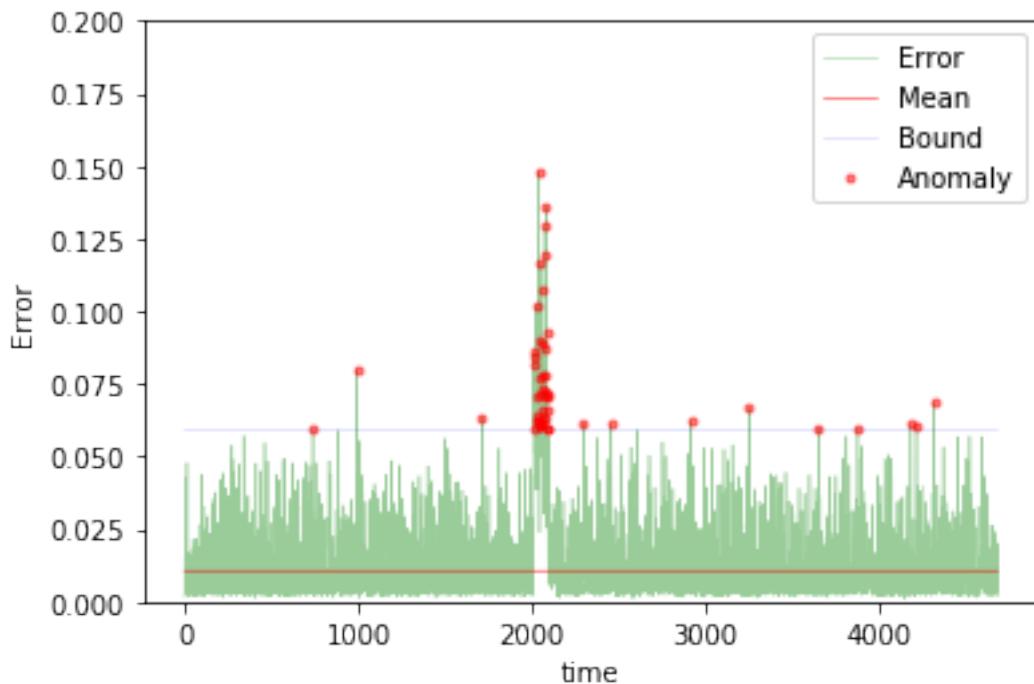
```
Training loss for final epoch is 0.011579646236496046
Validation loss for final epoch is 0.011202354178763927
----- Beginning tests for gru1_50 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

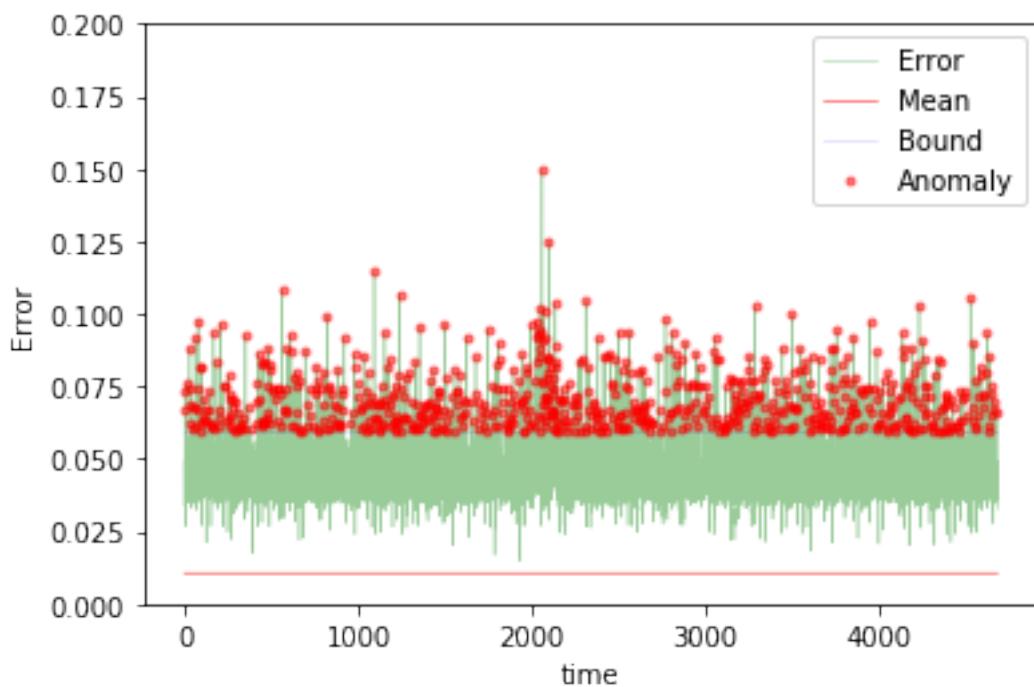




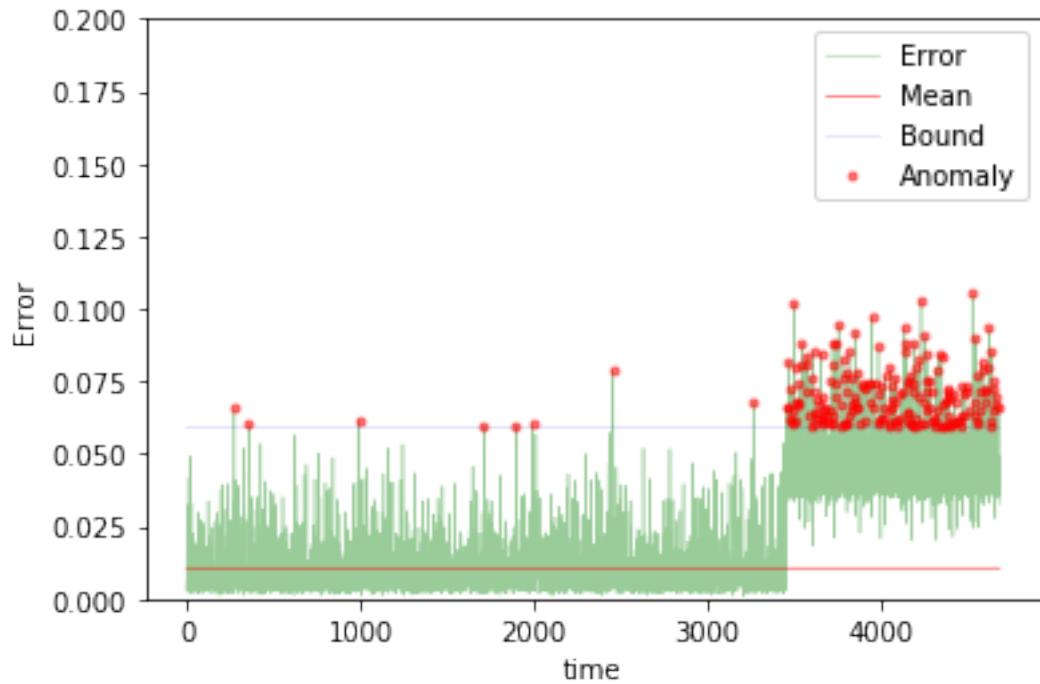
The mean error for gru1_50_normal_ is 0.010668821135111473 for length 4679
Testing on anomaly data.



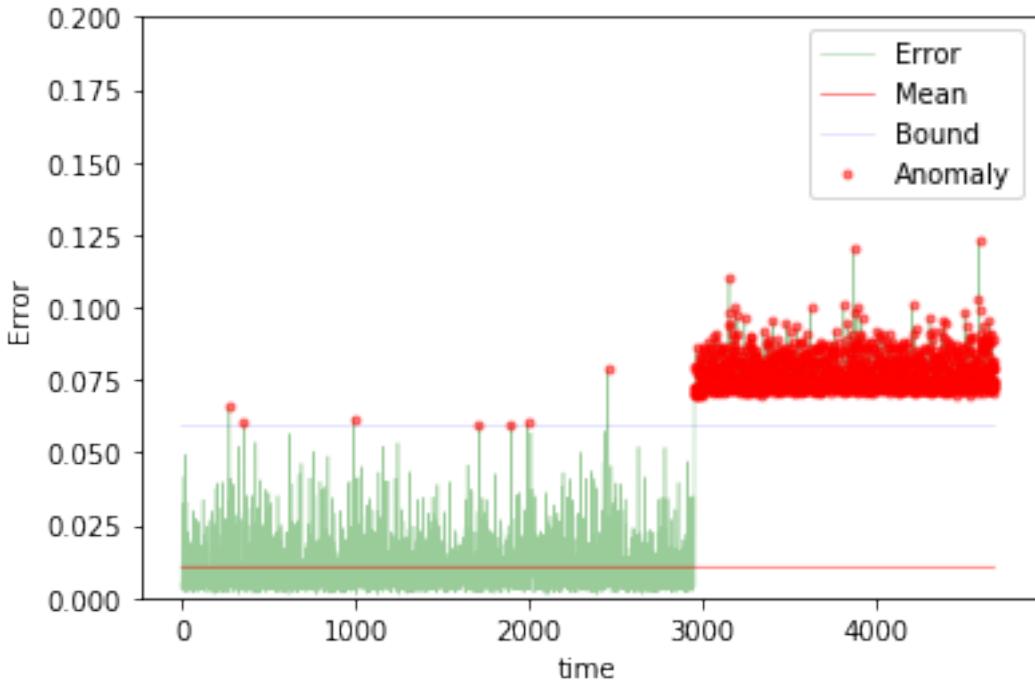
The mean error for gru1_50_anomaly_ is 0.012531055757954141 for length 4679
Testing on different app data.



The mean error for gru1_50_diff_app_ is 0.04683665767382045 for length 4679
Testing on App change synthetic data.



The mean error for gru1_50_app_change_ is 0.020053064202919125 for length 4679
Testing on Net flood synthetic data.



```
The mean error for gru1_50_net_flood_ is 0.034993480801099985 for length 4679
=====
```

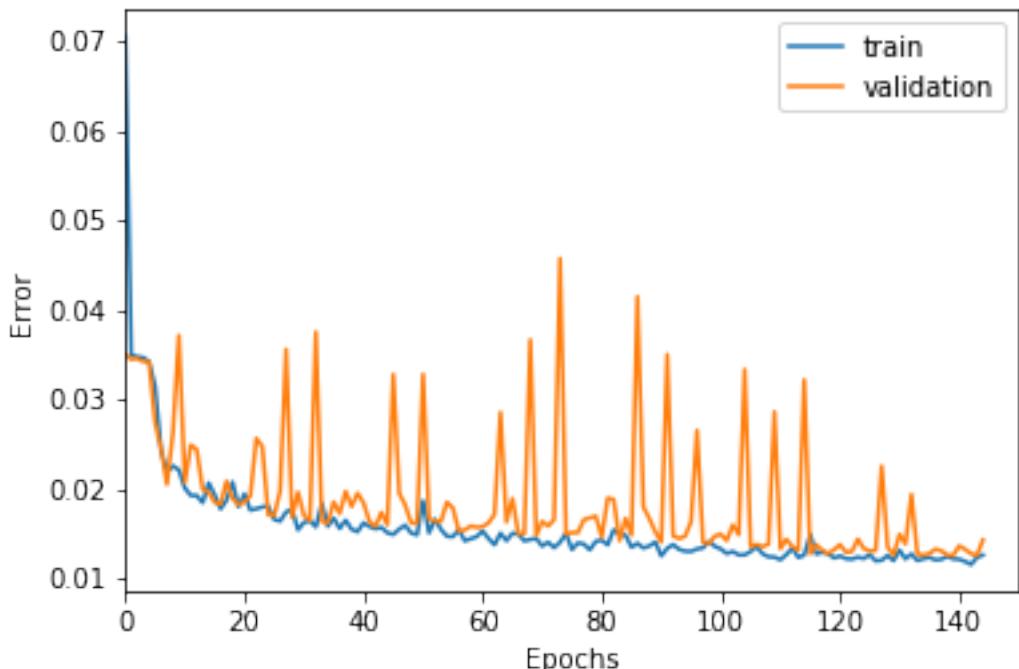
100 steps

```
In [168]: TIMESTEPS = 100
DIM = 29
tgen = flat_generator(X, TIMESTEPS,0)
vgen = flat_generator(val_X, TIMESTEPS,0)
name = "gru1_100"

In [169]: input_layer = Input(shape=(TIMESTEPS,DIM))
hidden = GRU(10, activation='relu')(input_layer)
output = Dense(DIM, activation='sigmoid')(hidden)

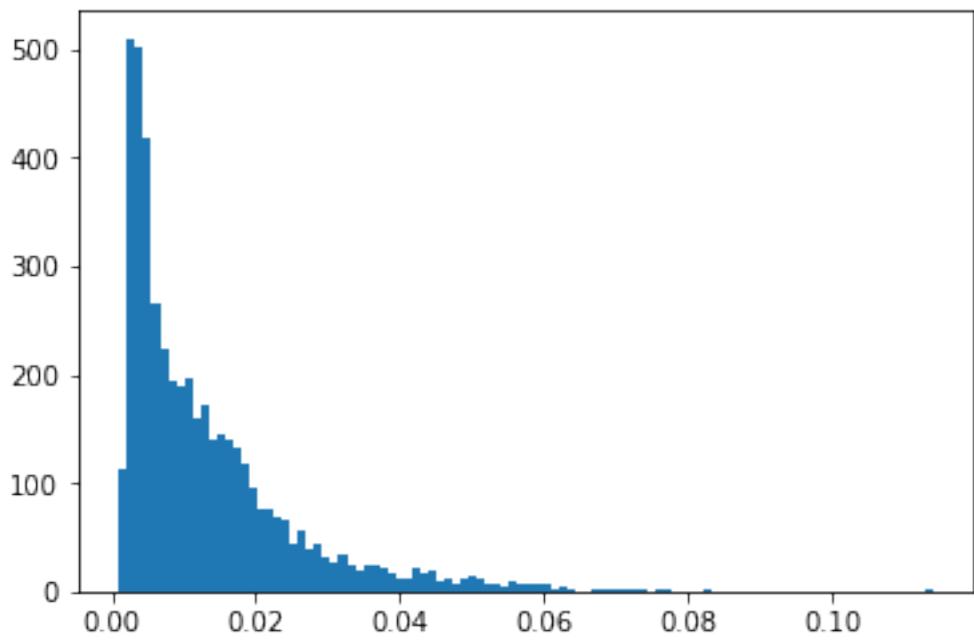
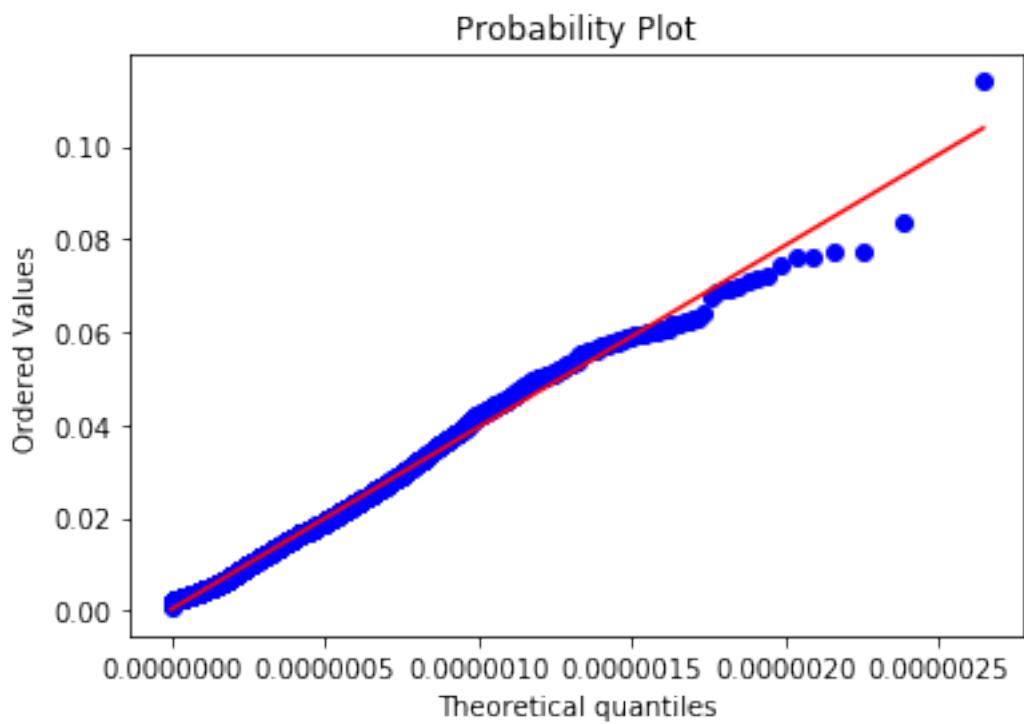
In [170]: model = Model(input_layer, output)
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

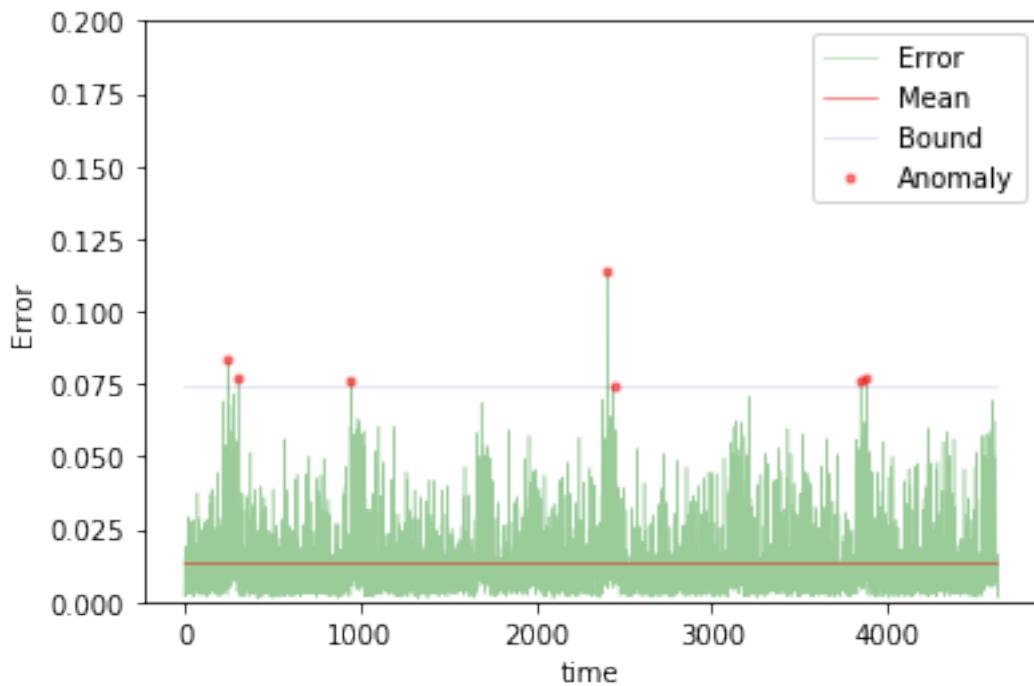
In [171]: train(model, tgen, vgen, name=name)
test(model, ravel=0, name=name, window=TIMESTEPS)
```



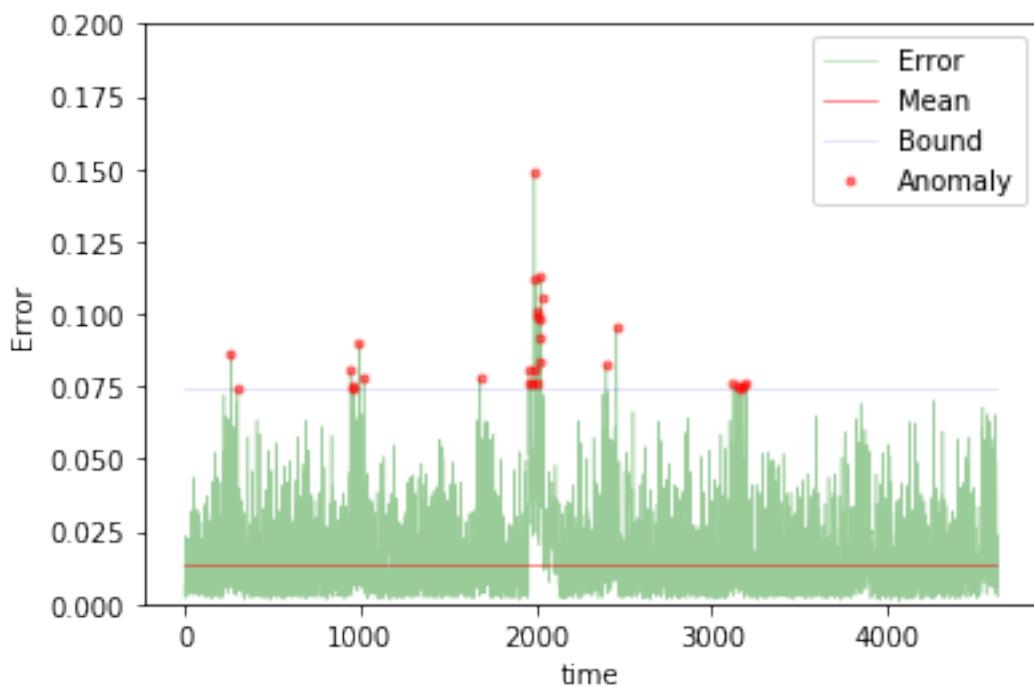
```
Training loss for final epoch is 0.0126045505075017
Validation loss for final epoch is 0.014285476626129822
----- Beginning tests for gru1_100 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

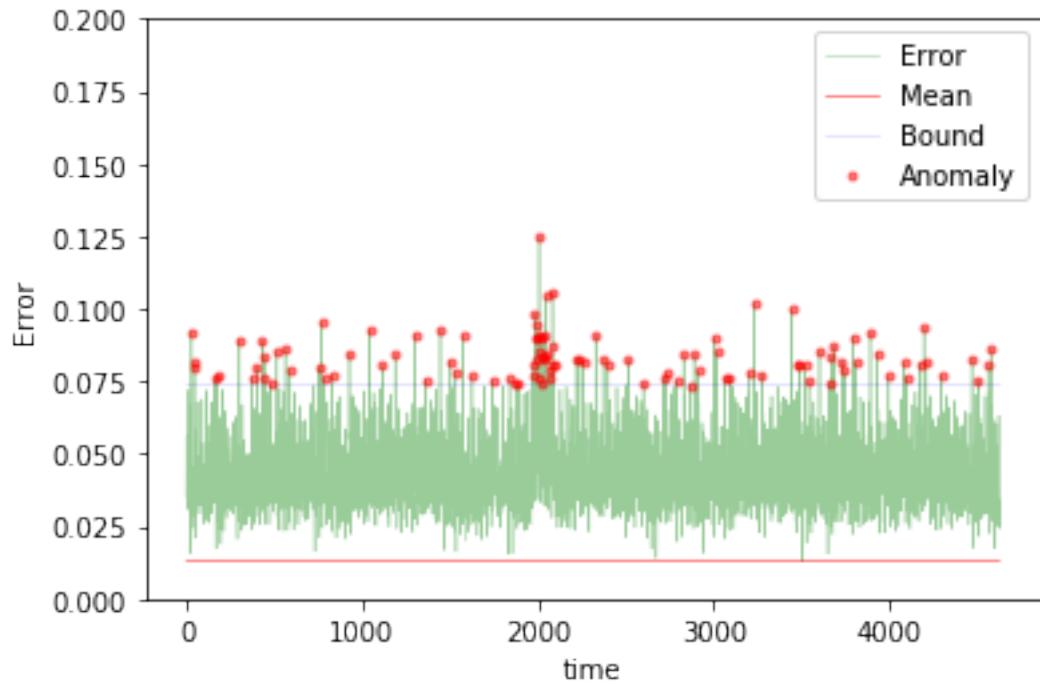




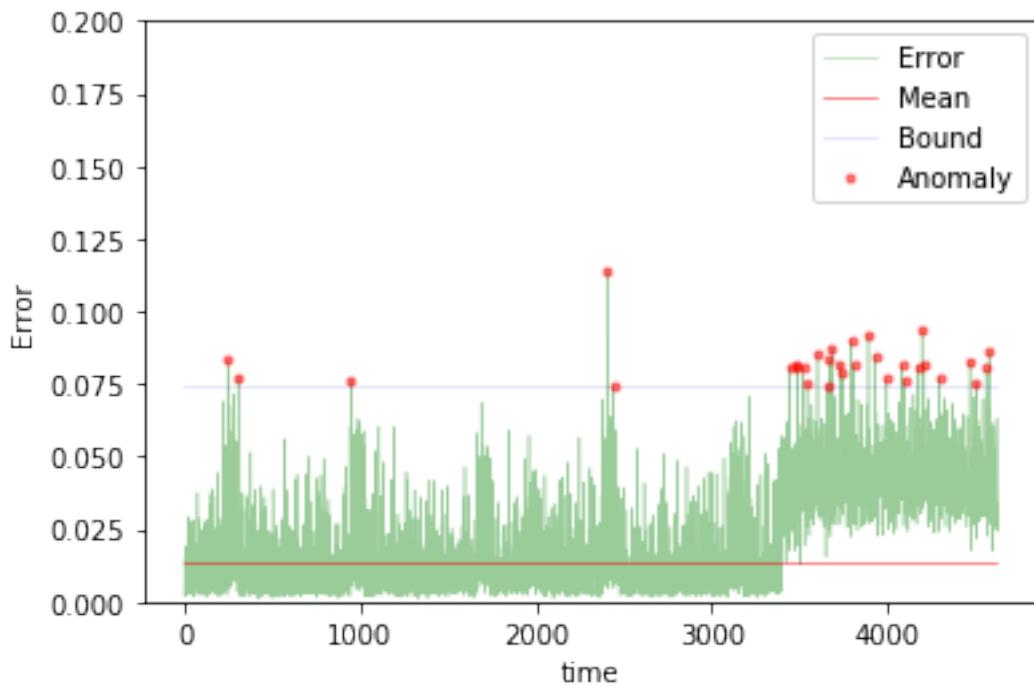
The mean error for gru1_100_normal_ is 0.013253470655520254 for length 4629
Testing on anomaly data.



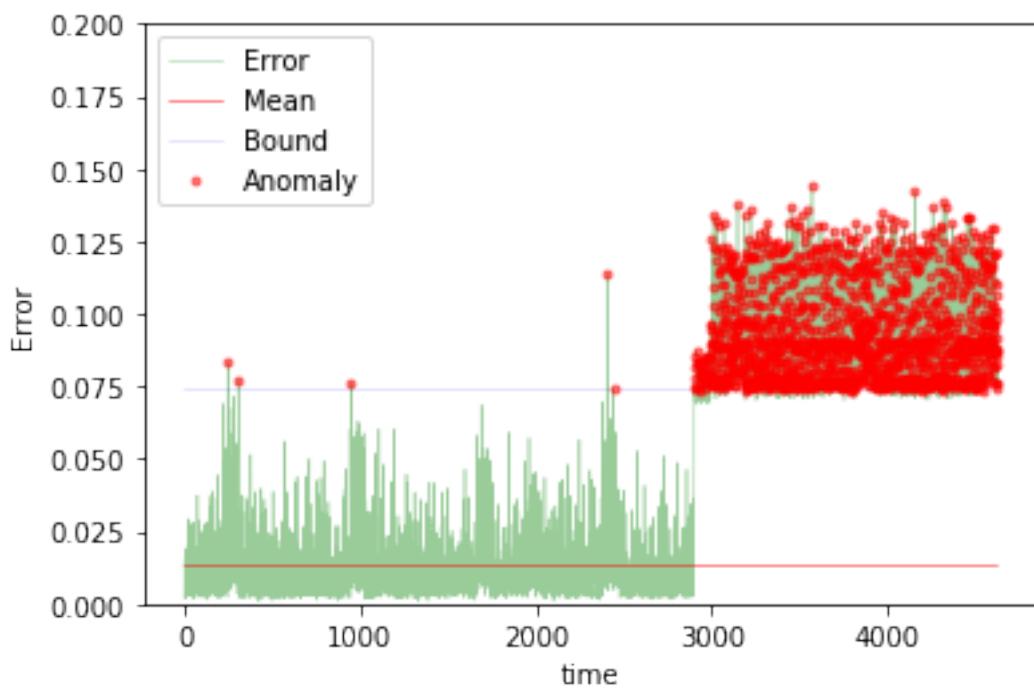
The mean error for gru1_100_anomaly_ is 0.01599120770544843 for length 4629
Testing on different app data.



The mean error for gru1_100_diff_app_ is 0.04141734817935168 for length 4629
Testing on App change synthetic data.



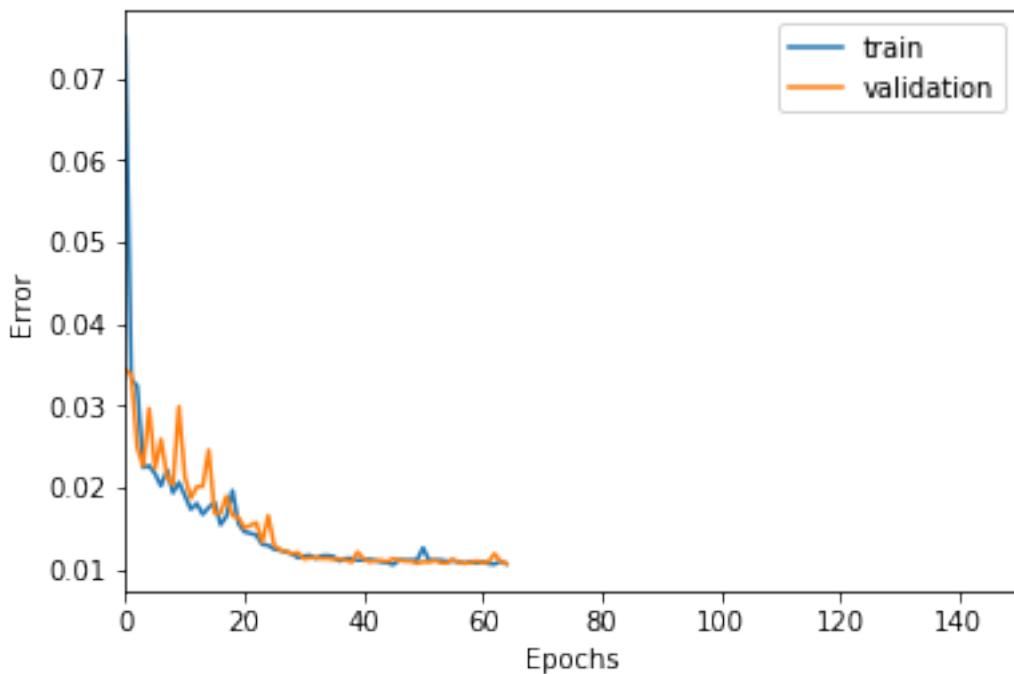
The mean error for gru1_100_app_change_ is 0.020495575056833814 for length 4629
Testing on Net flood synthetic data.



```
The mean error for gru1_100_net_flood_ is 0.042186197630955935 for length 4629  
=====
```

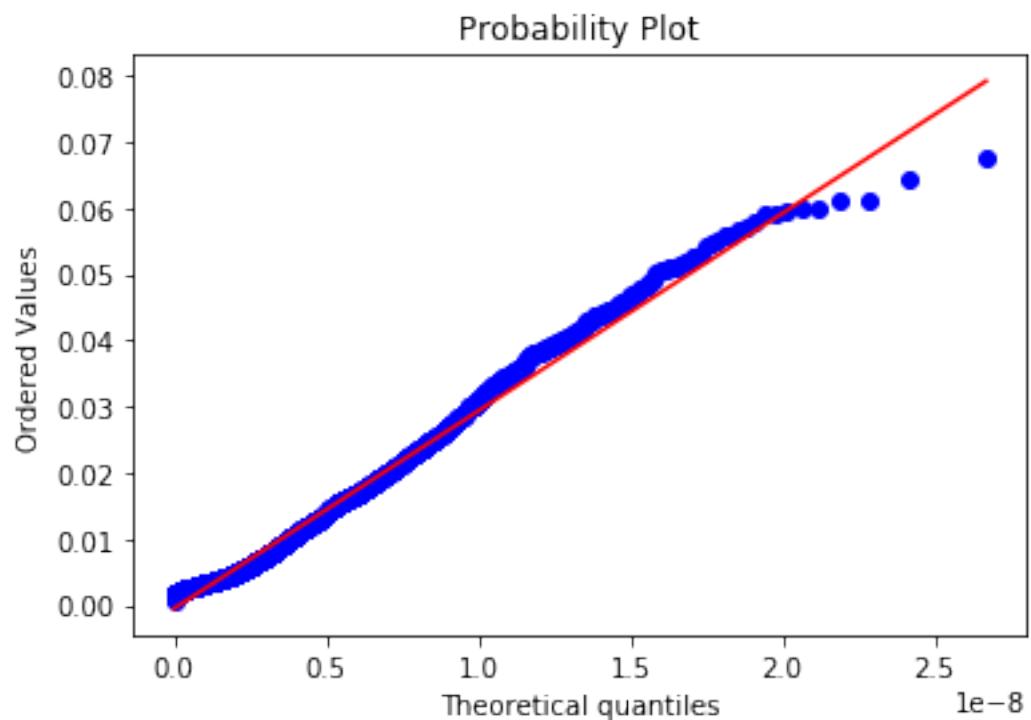
200 steps

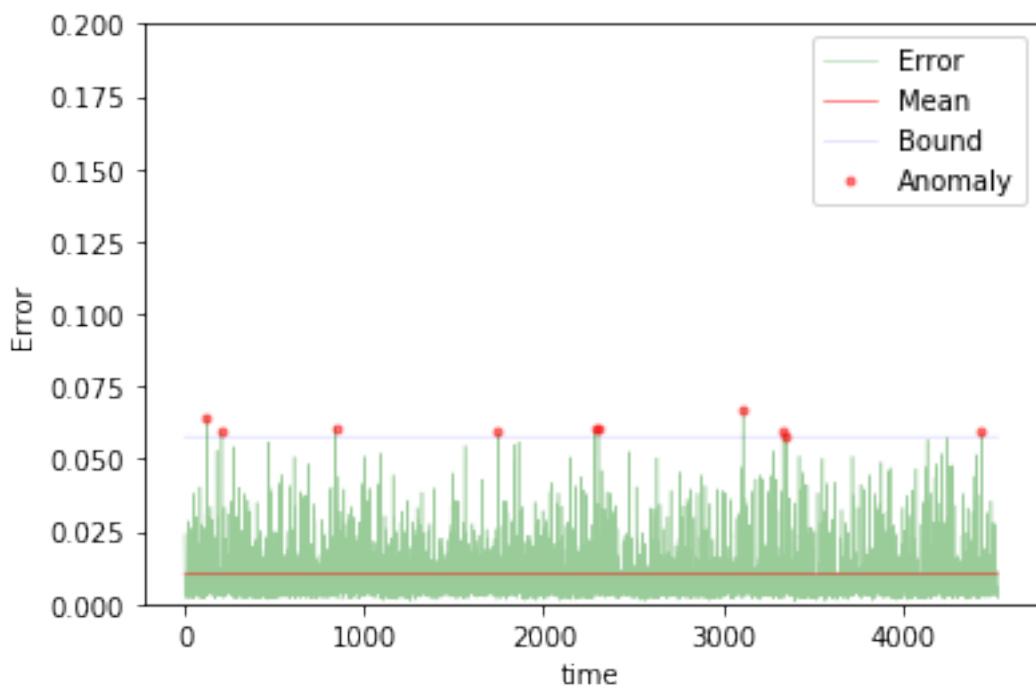
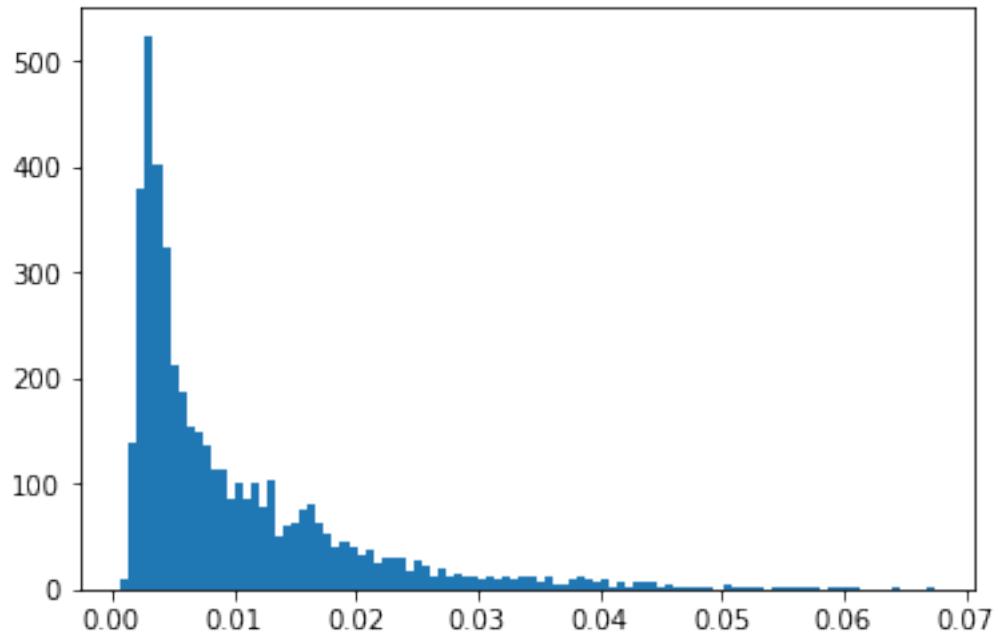
```
In [172]: TIMESTEPS = 200  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS,0)  
vgen = flat_generator(val_X, TIMESTEPS,0)  
name = "gru1_200"  
  
In [173]: input_layer = Input(shape=(TIMESTEPS,DIM))  
hidden = GRU(10, activation='relu')(input_layer)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [174]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [175]: train(model, tgen, vgen, name=name)  
test(model, ravel=0, name=name, window=TIMESTEPS)
```



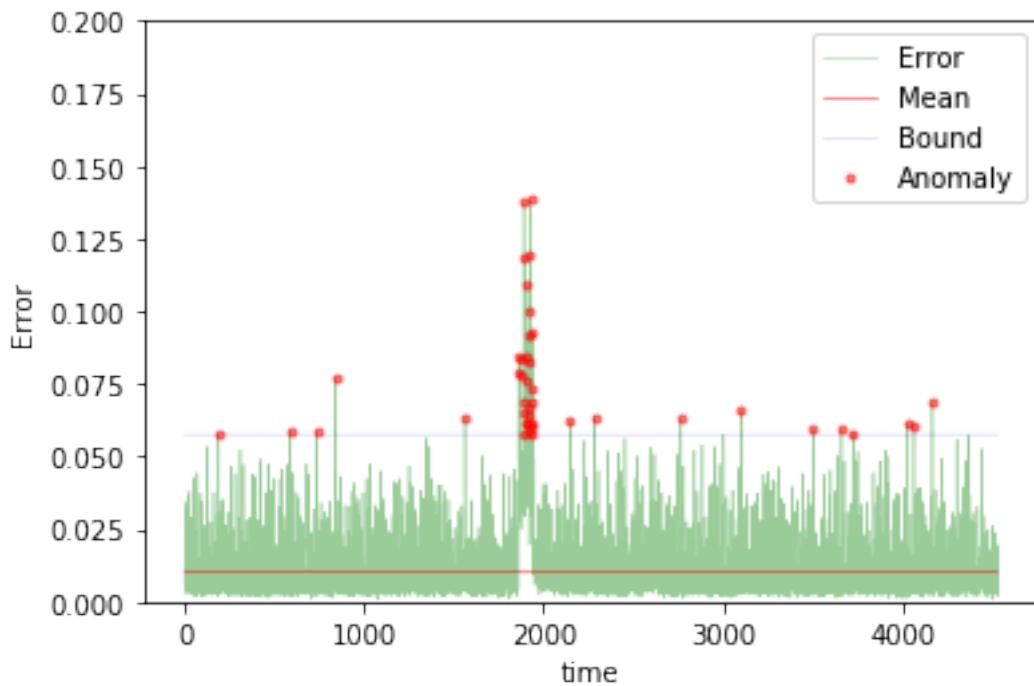
```
Training loss for final epoch is 0.010691348514636047
Validation loss for final epoch is 0.010734328928403557
----- Beginning tests for gru1_200 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

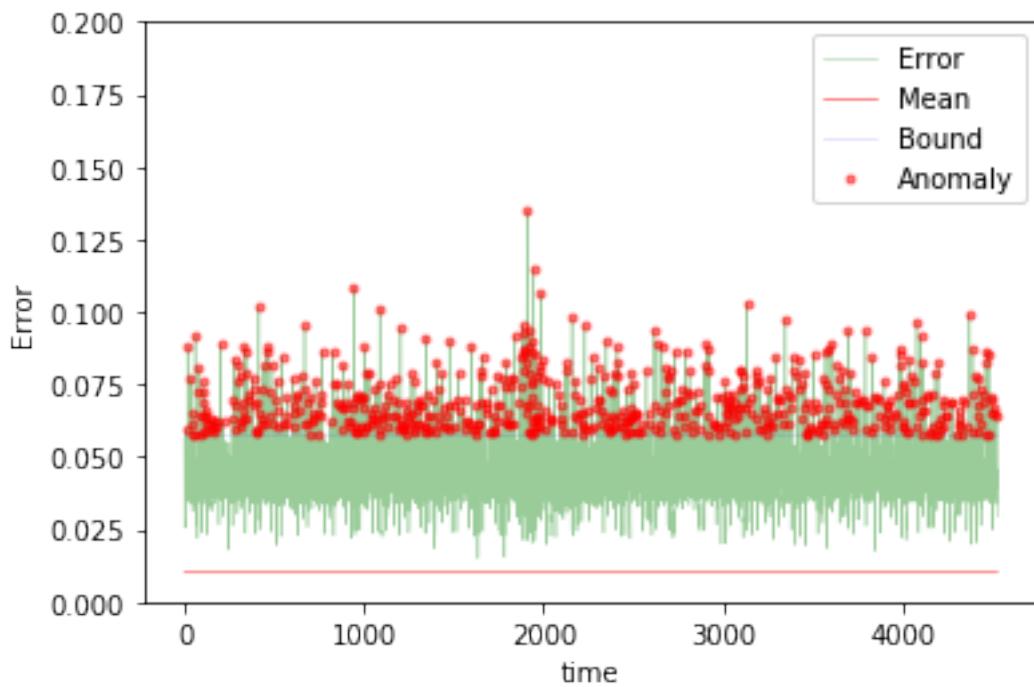




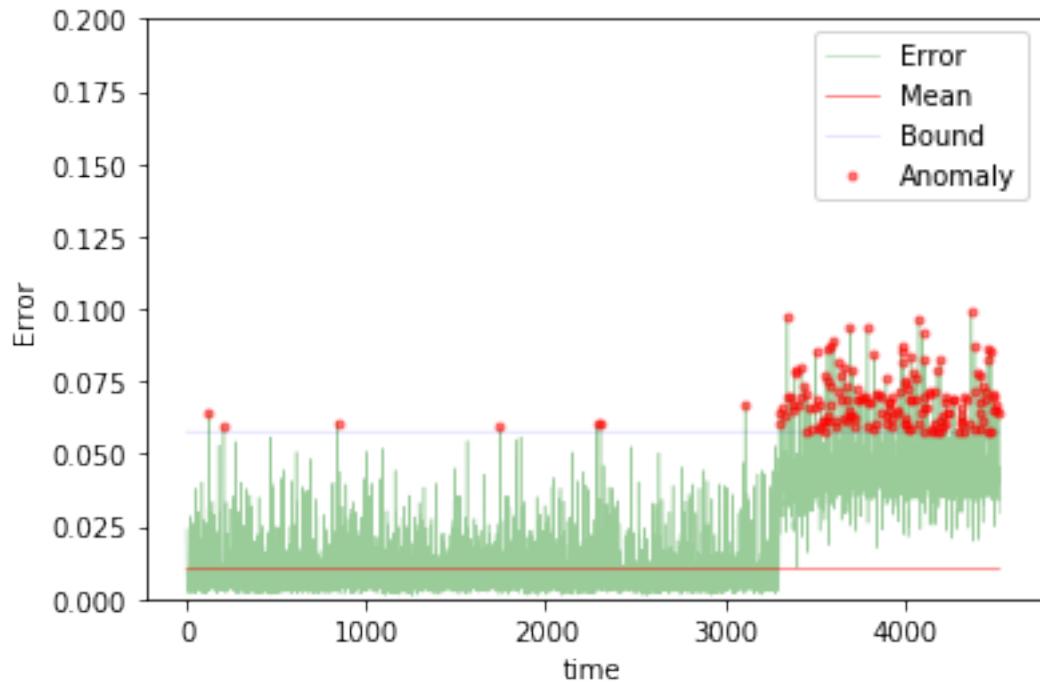
The mean error for gru1_200_normal_ is 0.010139247029719346 for length 4529
Testing on anomaly data.



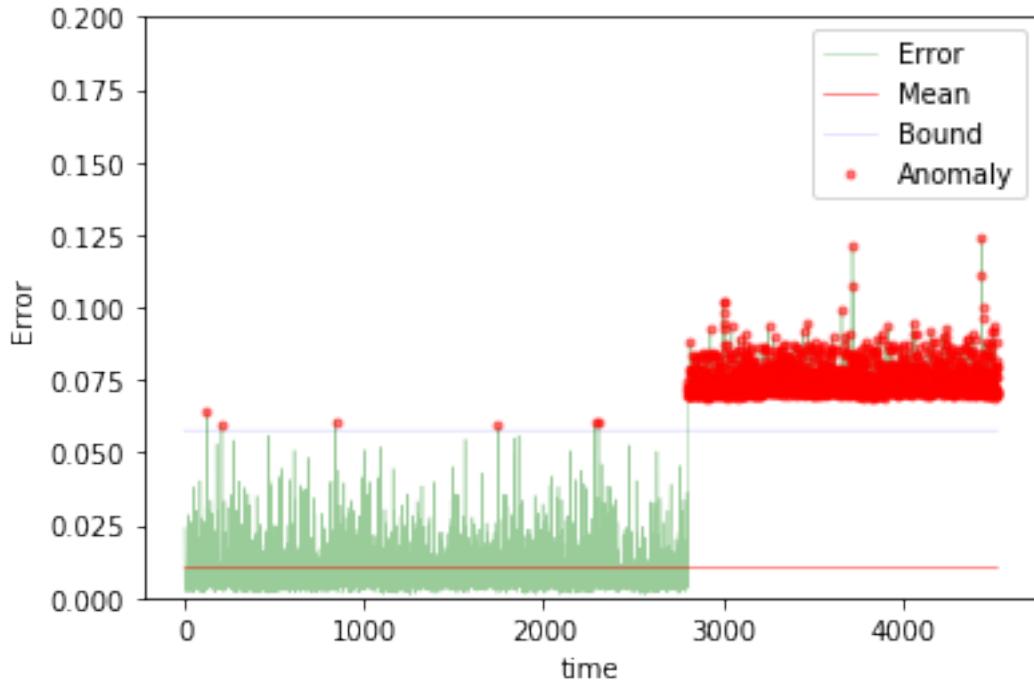
The mean error for gru1_200_anomaly_ is 0.0120621744914722 for length 4529
Testing on different app data.



The mean error for gru1_200_diff_app_ is 0.04503510962768084 for length 4529
Testing on App change synthetic data.



The mean error for gru1_200_app_change_ is 0.019360047609335466 for length 4529
Testing on Net flood synthetic data.



```
The mean error for gru1_200_net_flood_ is 0.03491526313086969 for length 4529
=====
```

2.1.6 RNN with 2 GRU layers

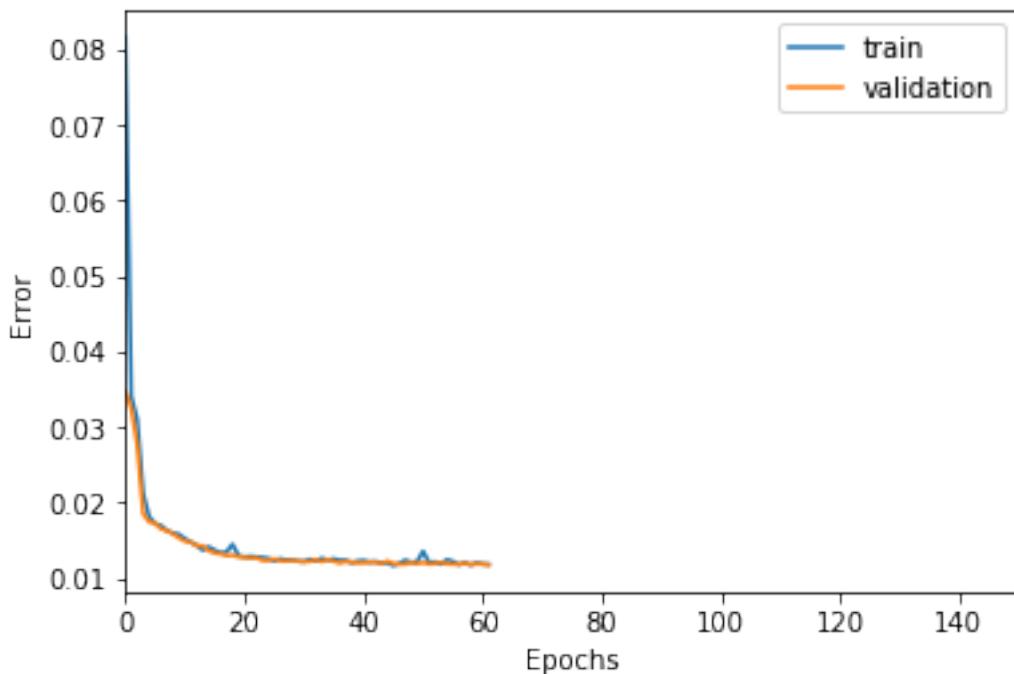
2 steps

```
In [176]: TIMESTEPS = 2
DIM = 29
tgen = flat_generator(X, TIMESTEPS,0)
vgen = flat_generator(val_X, TIMESTEPS,0)
name = "gru2_2"

In [177]: 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)

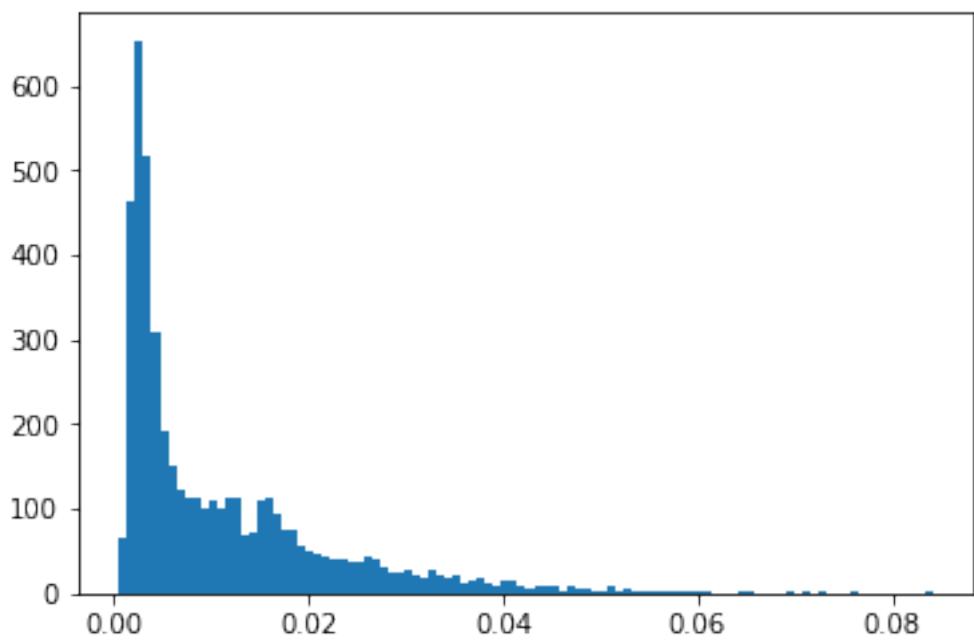
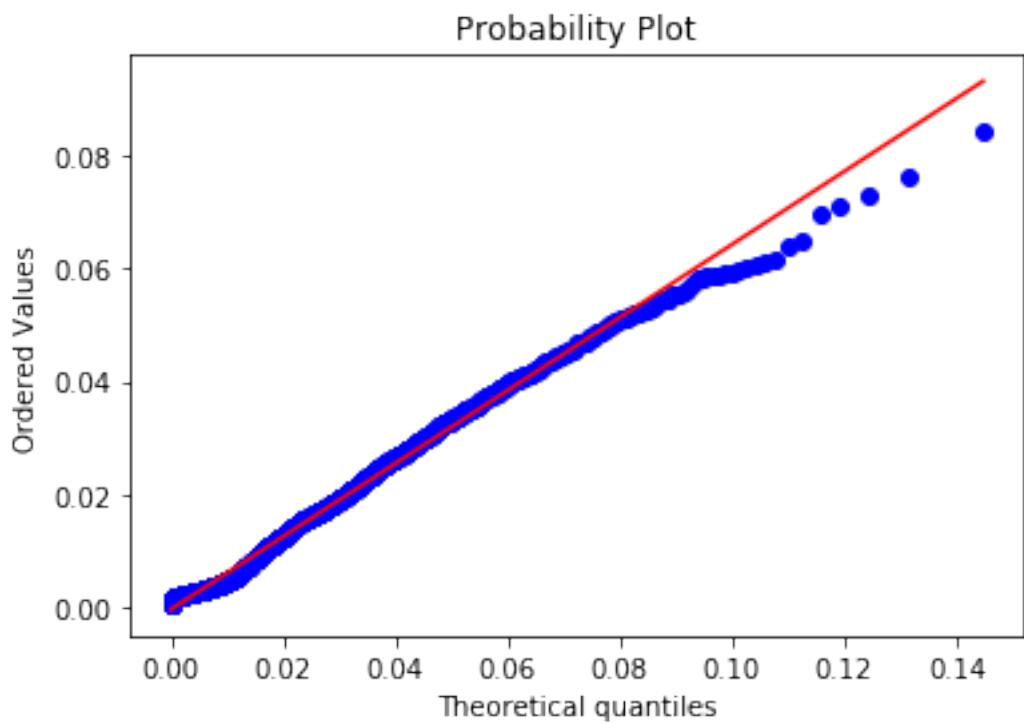
In [178]: model = Model(input_layer, output)
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])
```

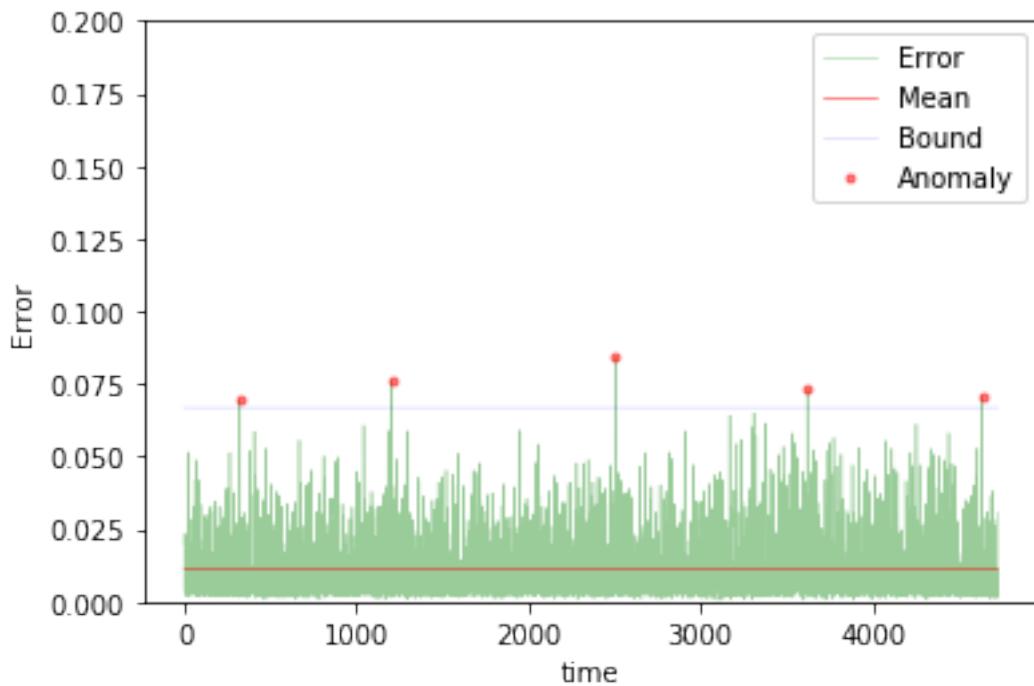
```
In [179]: train(model, tgen, vgen, name=name)
test(model, ravel=0, name=name, window=TIMESTEPS)
```



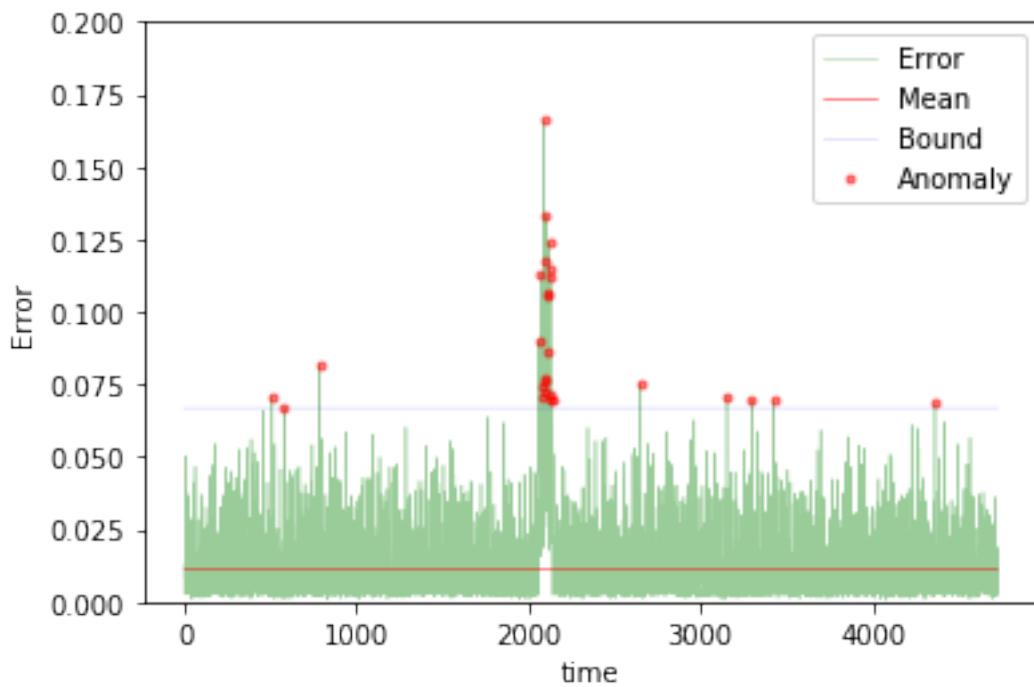
```
Training loss for final epoch is 0.011844495735131204
Validation loss for final epoch is 0.011777702249237337
----- Beginning tests for gru2_2 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWar
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

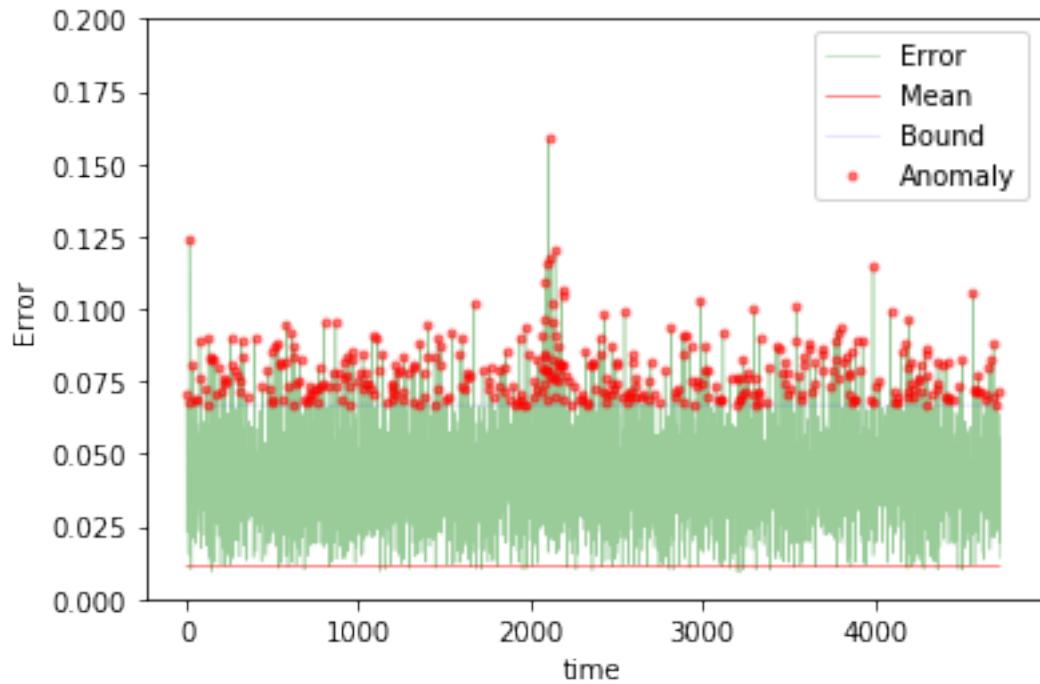




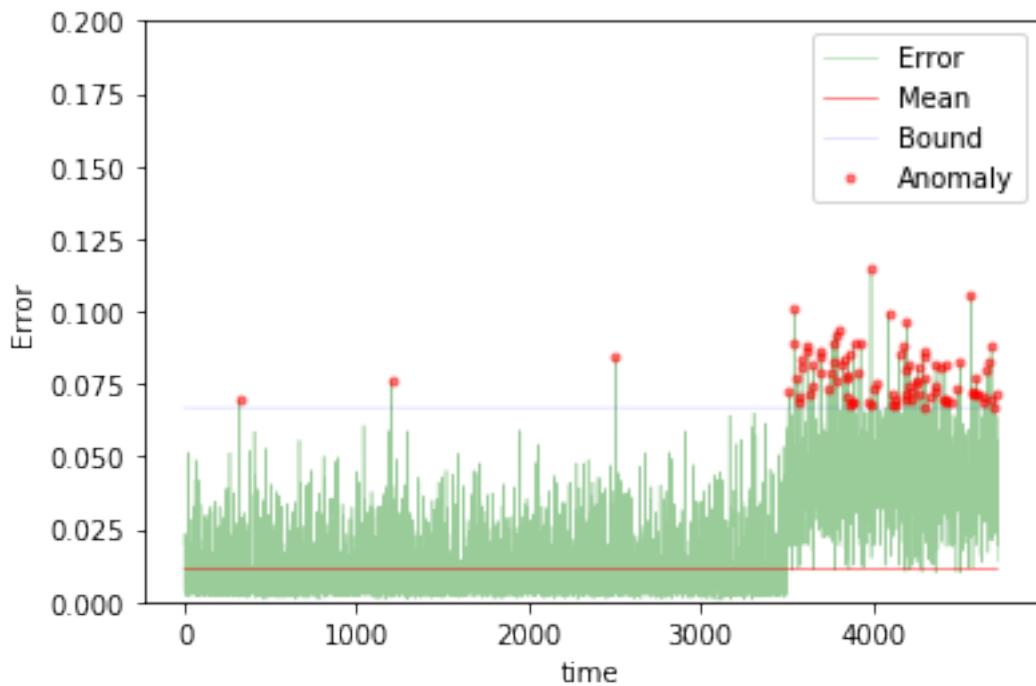
The mean error for gru2_2_normal_ is 0.01121554493767331 for length 4727
Testing on anomaly data.



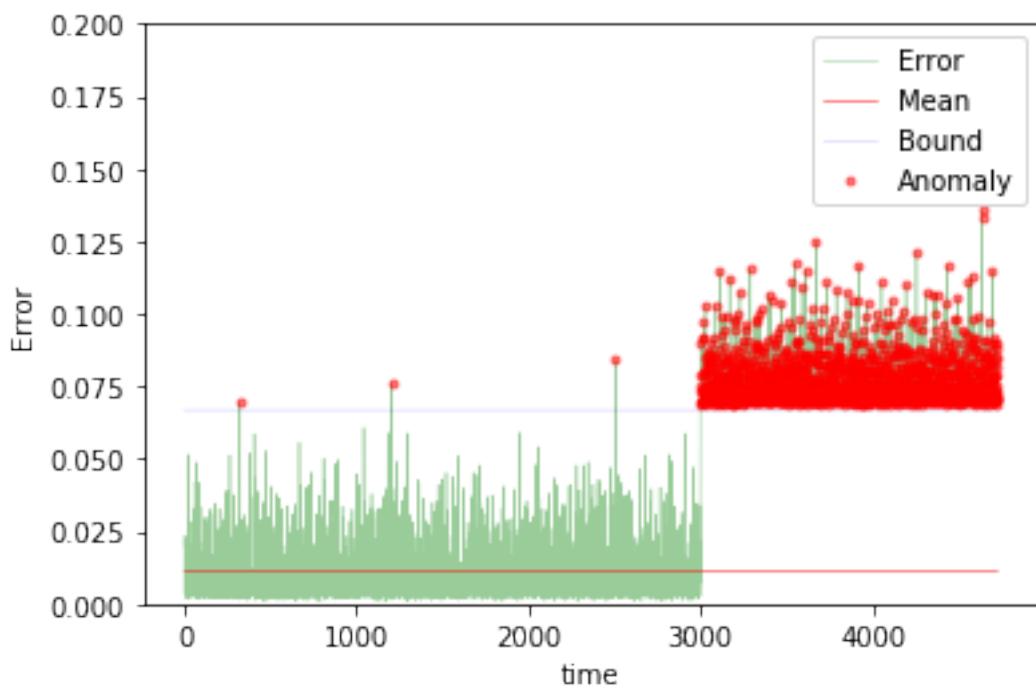
The mean error for gru2_2_anomaly_ is 0.01280580544778536 for length 4727
Testing on different app data.



The mean error for gru2_2_diff_app_ is 0.042563089234363166 for length 4727
Testing on App change synthetic data.



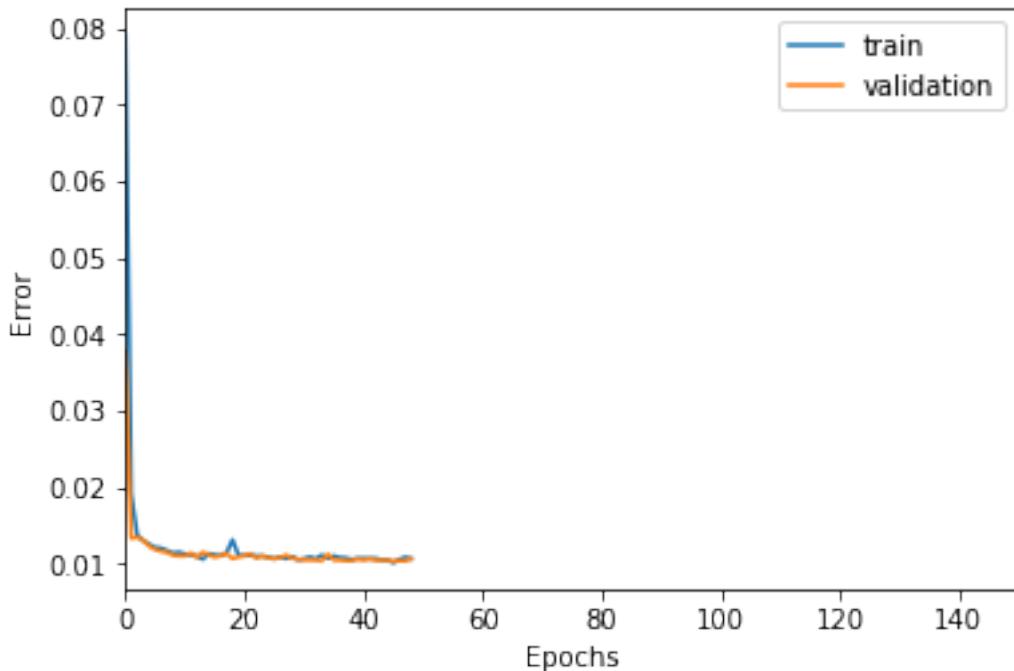
The mean error for gru2_2_app_change_ is 0.01927266389189374 for length 4727
Testing on Net flood synthetic data.



```
The mean error for gru2_2_net_flood_ is 0.035247514853181734 for length 4727  
=====
```

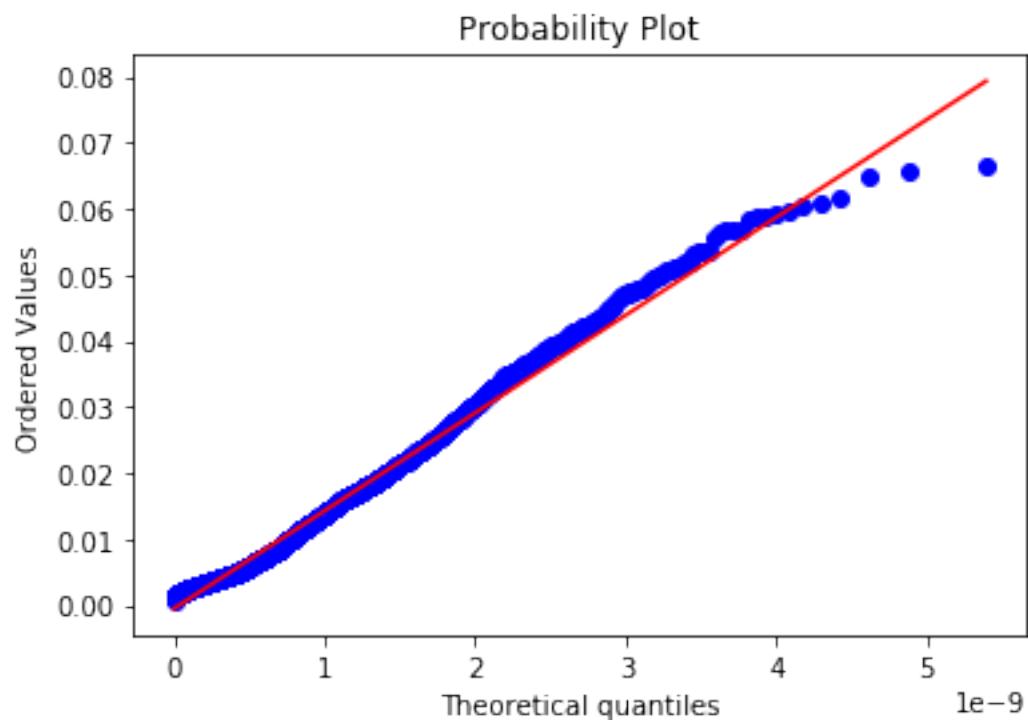
5 steps

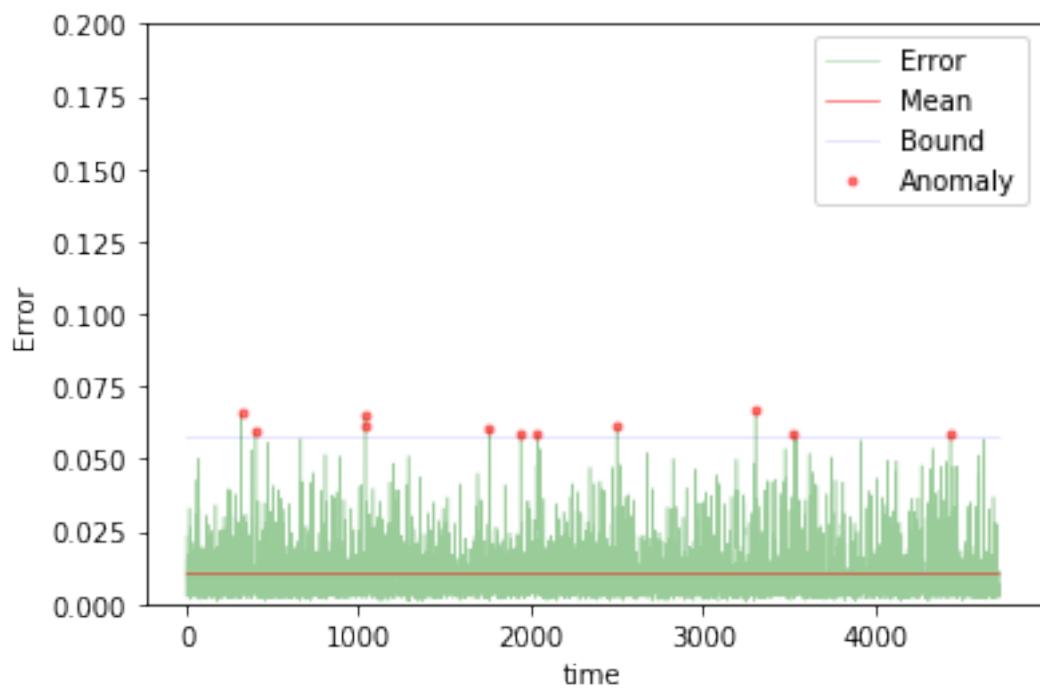
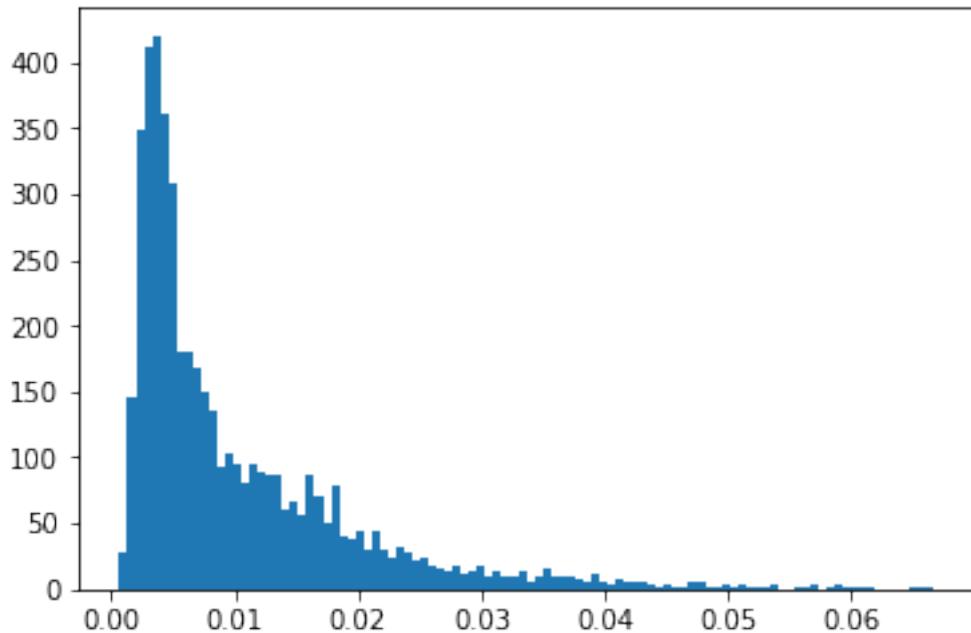
```
In [180]: TIMESTEPS = 5  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS,0)  
vgen = flat_generator(val_X, TIMESTEPS, 0)  
name = "gru2_5"  
  
In [181]: input_layer = Input(shape=(TIMESTEPS,DIM))  
hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)  
hidden = GRU(10, activation='relu')(hidden)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [182]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [183]: train(model, tgen, vgen, name=name)  
test(model, ravel=0, name=name, window=TIMESTEPS)
```



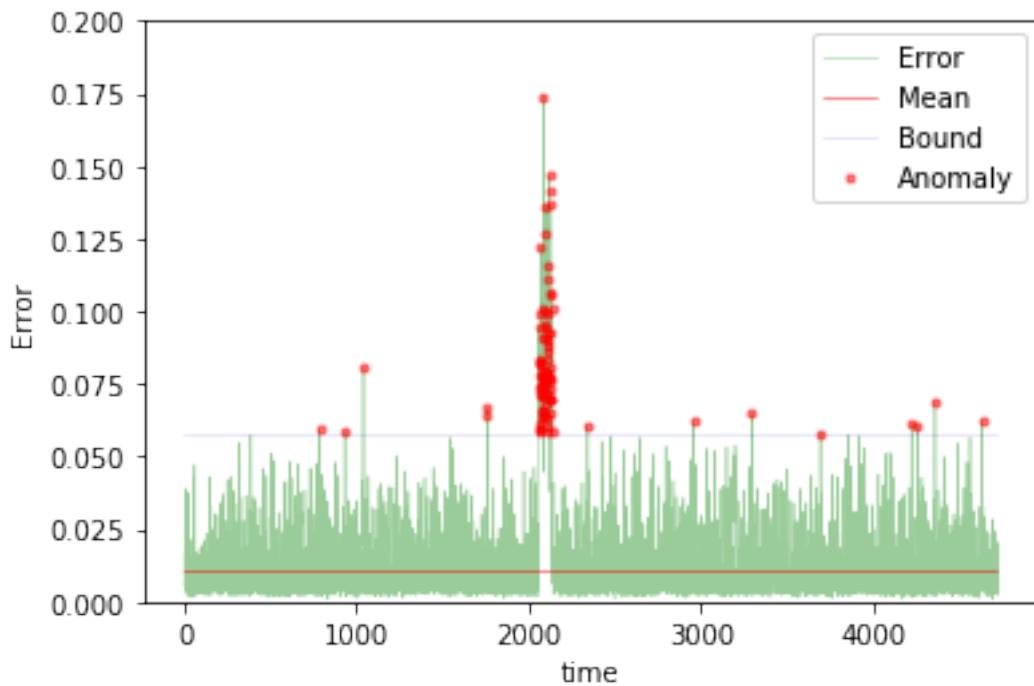
```
Training loss for final epoch is 0.010679421033710241
Validation loss for final epoch is 0.010646497472422198
----- Beginning tests for gru2_5 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

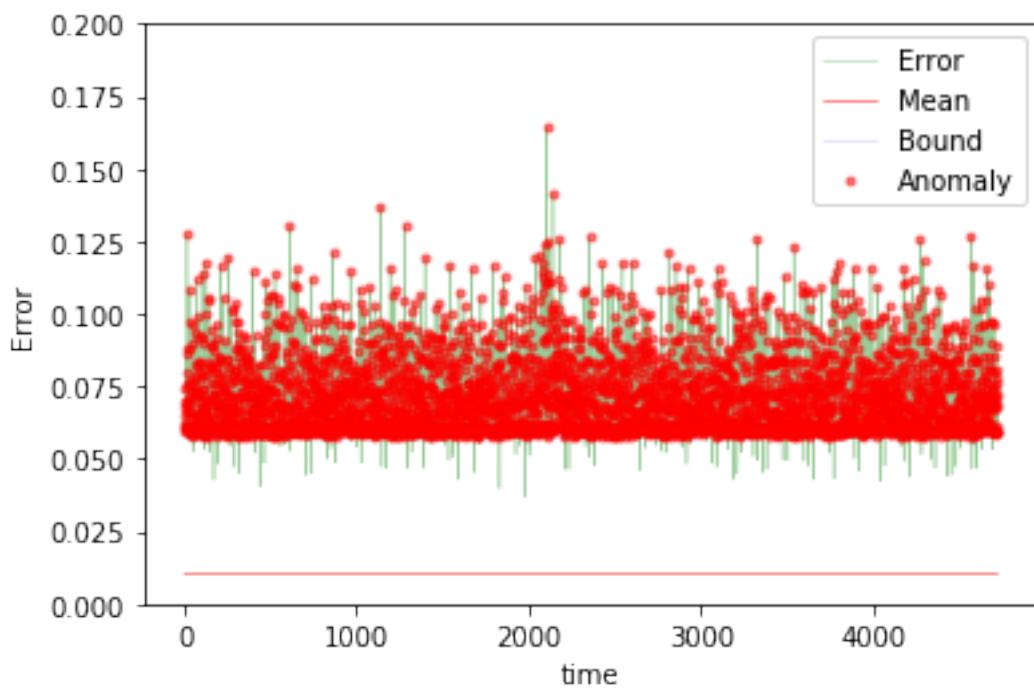




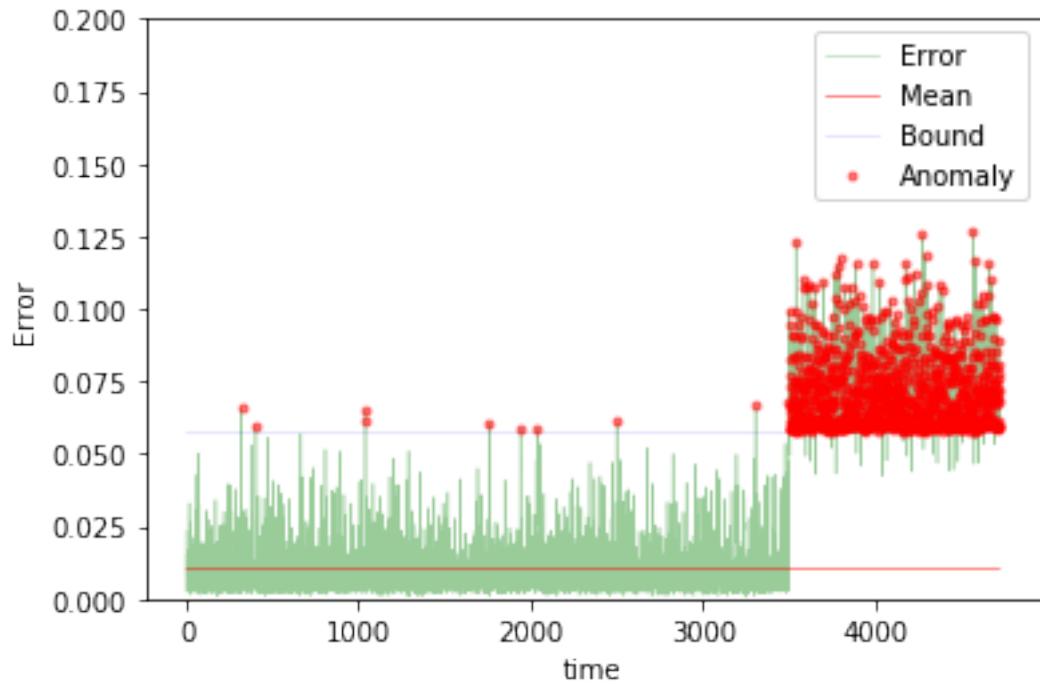
The mean error for gru2_5_normal_ is 0.010161049021653058 for length 4724
Testing on anomaly data.



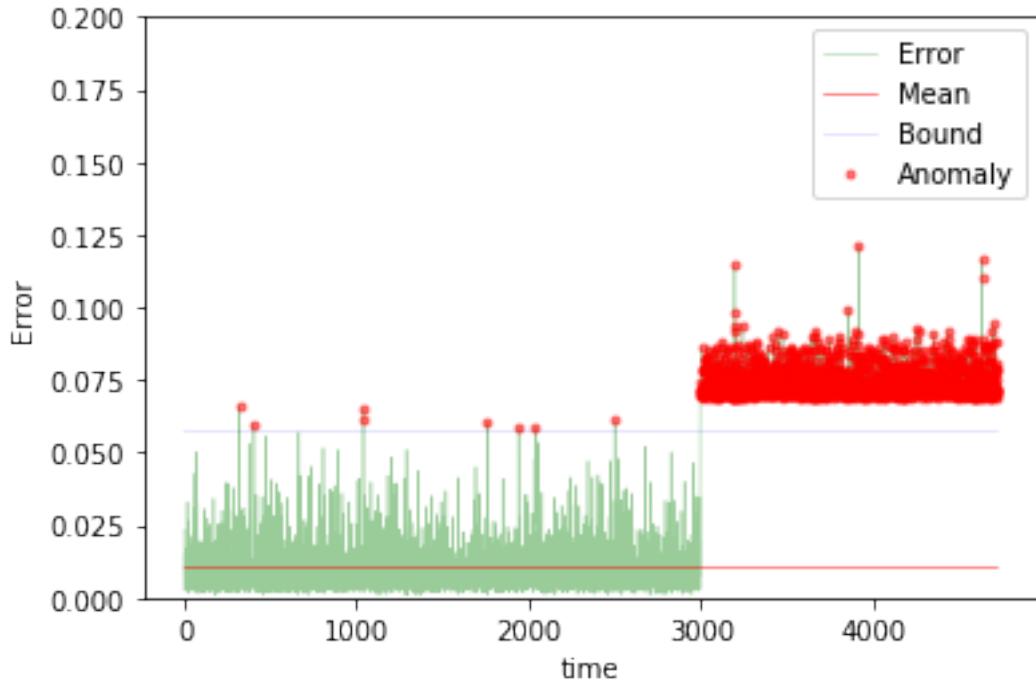
The mean error for gru2_5_anomaly_ is 0.012252551931311598 for length 4724
Testing on different app data.



The mean error for gru2_5_diff_app_ is 0.0703820354587448 for length 4724
Testing on App change synthetic data.



The mean error for gru2_5_app_change_ is 0.02568320434506144 for length 4724
Testing on Net flood synthetic data.



The mean error for gru2_5_net_flood_ is 0.03378434935245065 for length 4724
=====

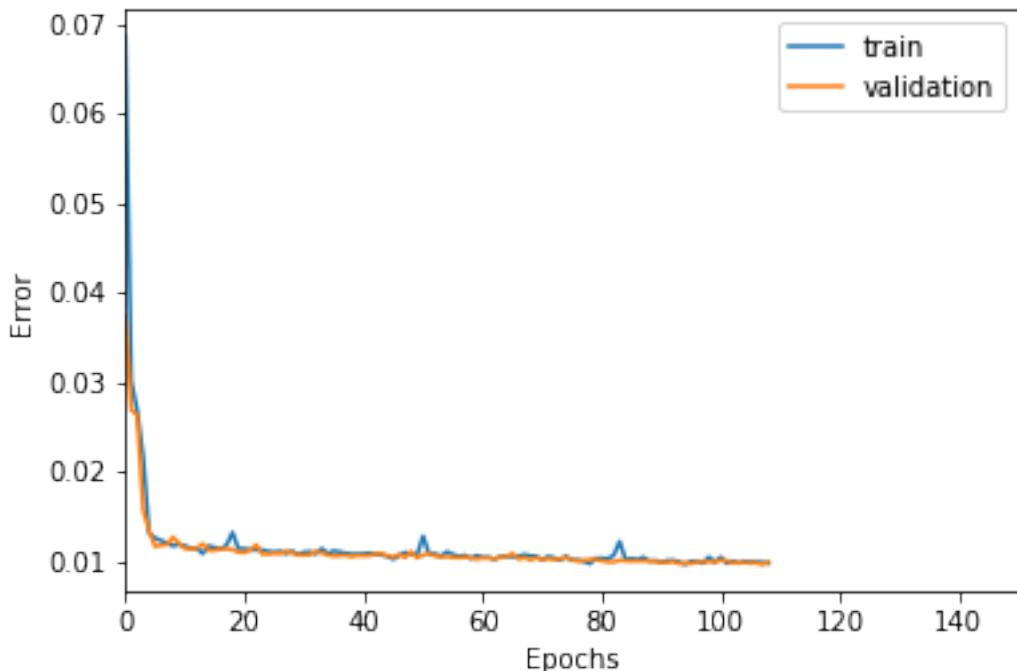
10 steps

```
In [184]: TIMESTEPS = 10
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS, 0)
          vgen = flat_generator(val_X, TIMESTEPS, 0)
          name = "gru2_10"

In [185]: 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)

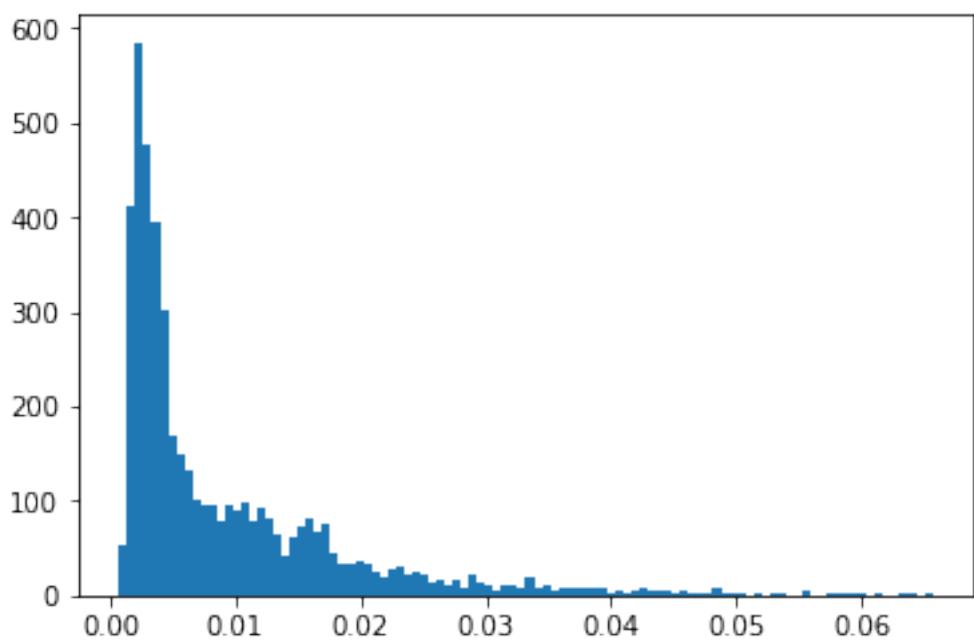
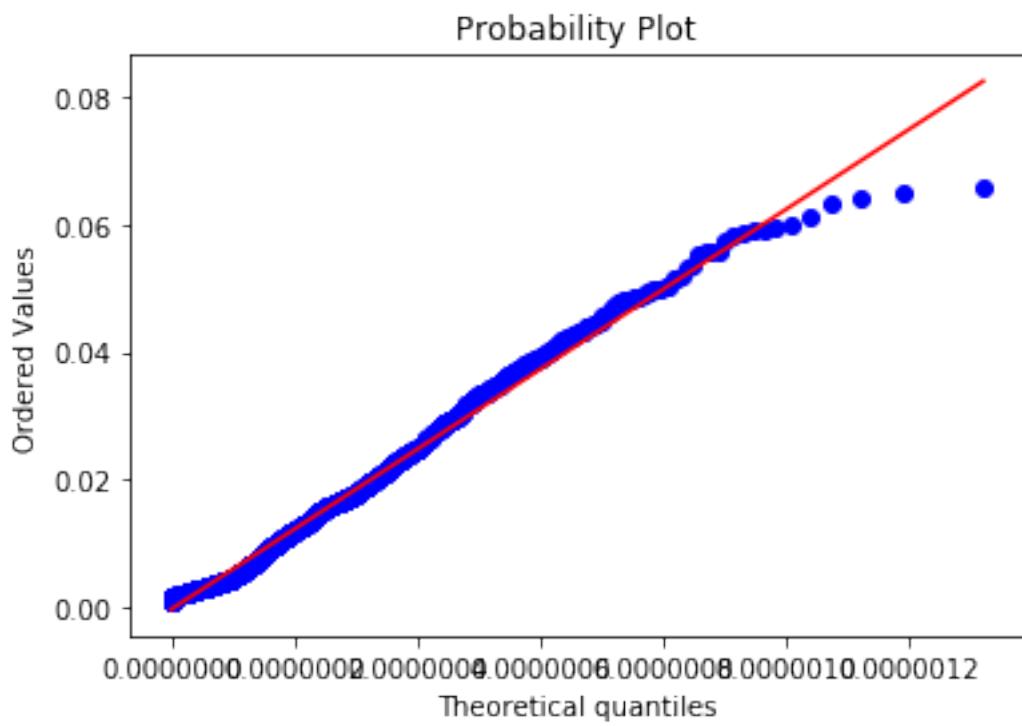
In [186]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

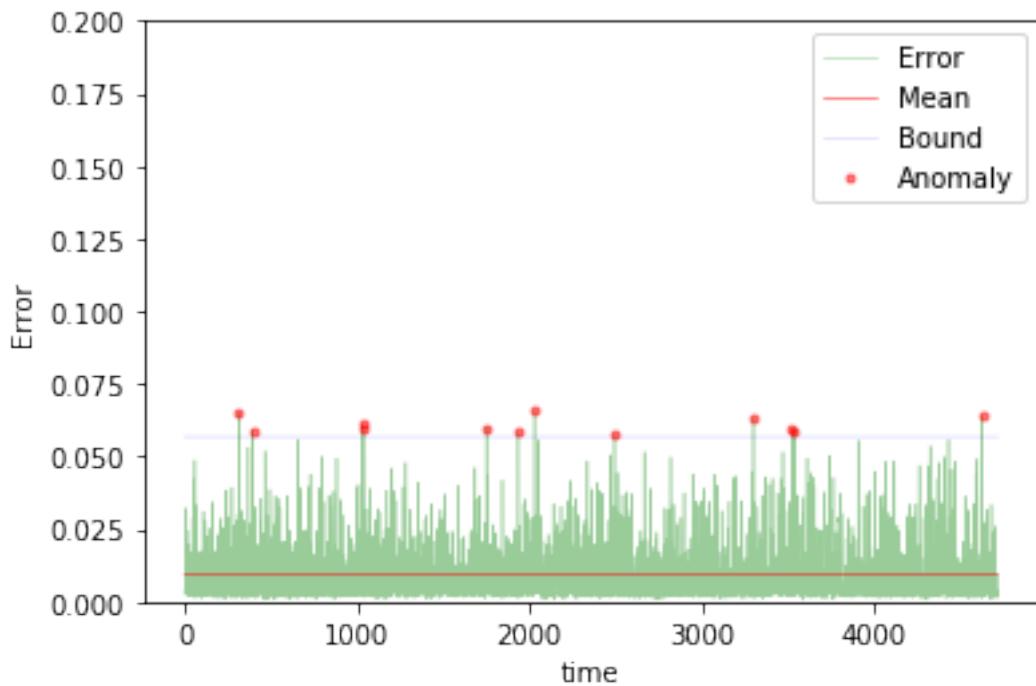
In [187]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```



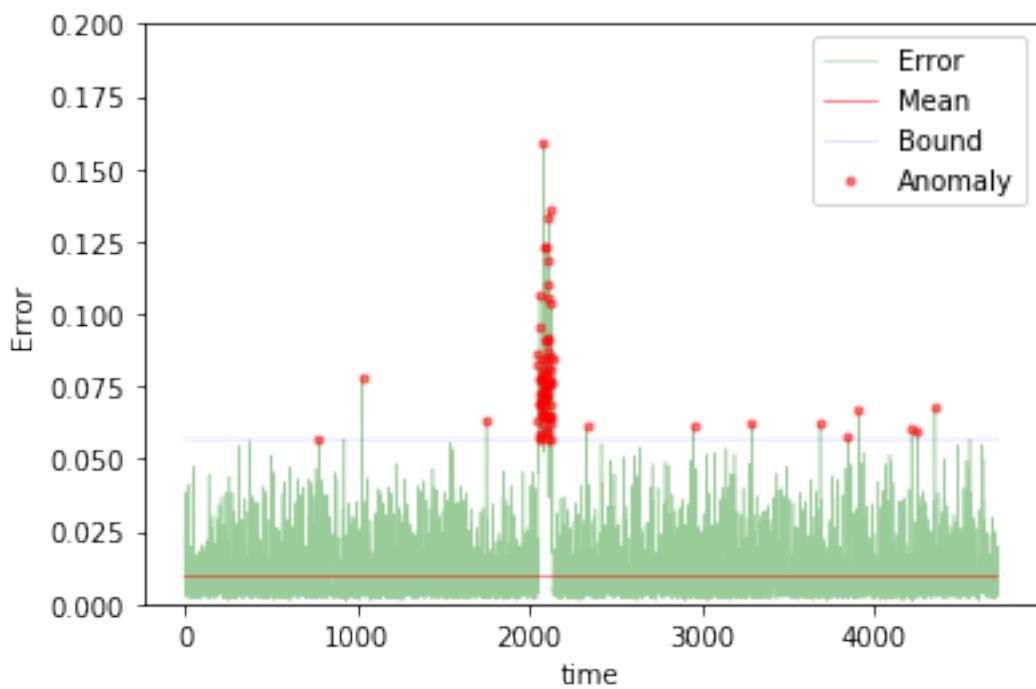
```
Training loss for final epoch is 0.009709420644911006
Validation loss for final epoch is 0.009861612751847134
----- Beginning tests for gru2_10 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWar
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

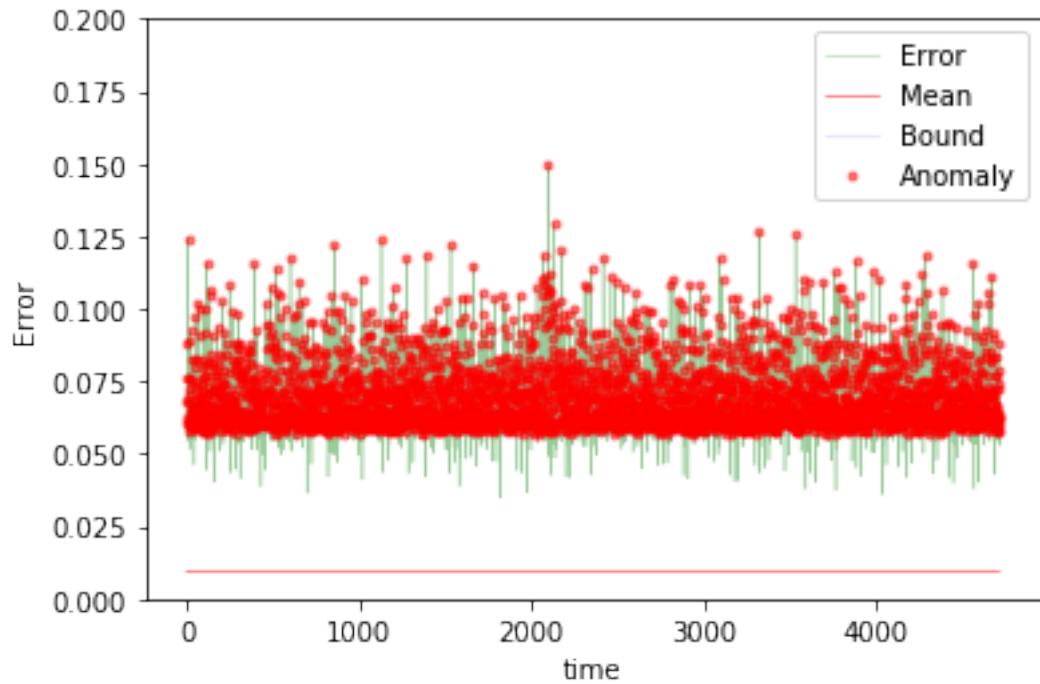




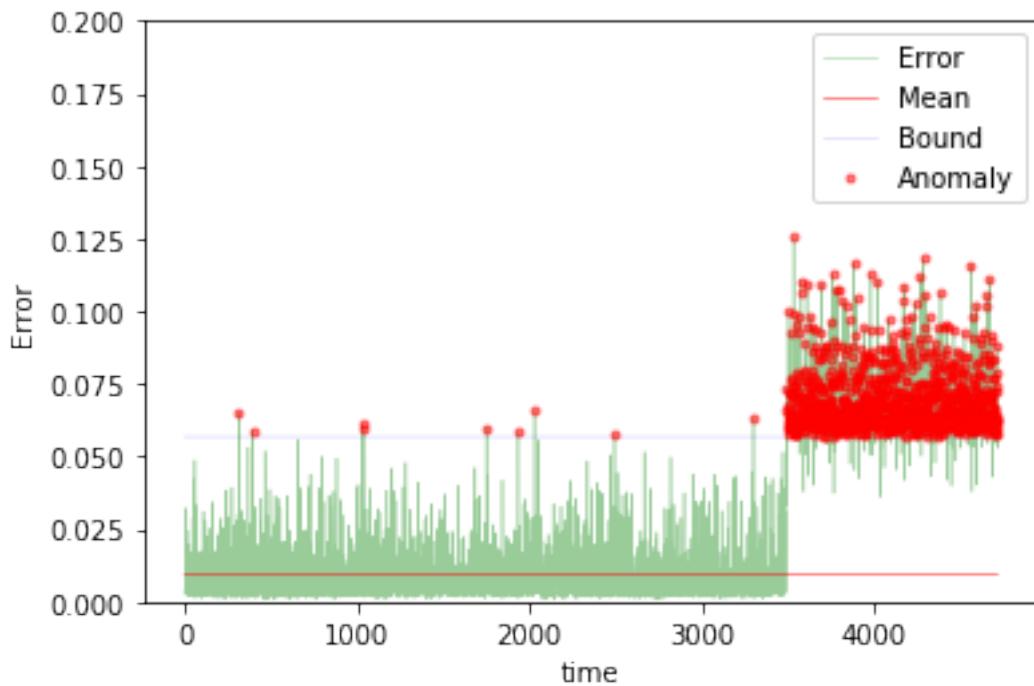
The mean error for gru2_10_normal_ is 0.009240892345496651 for length 4719
Testing on anomaly data.



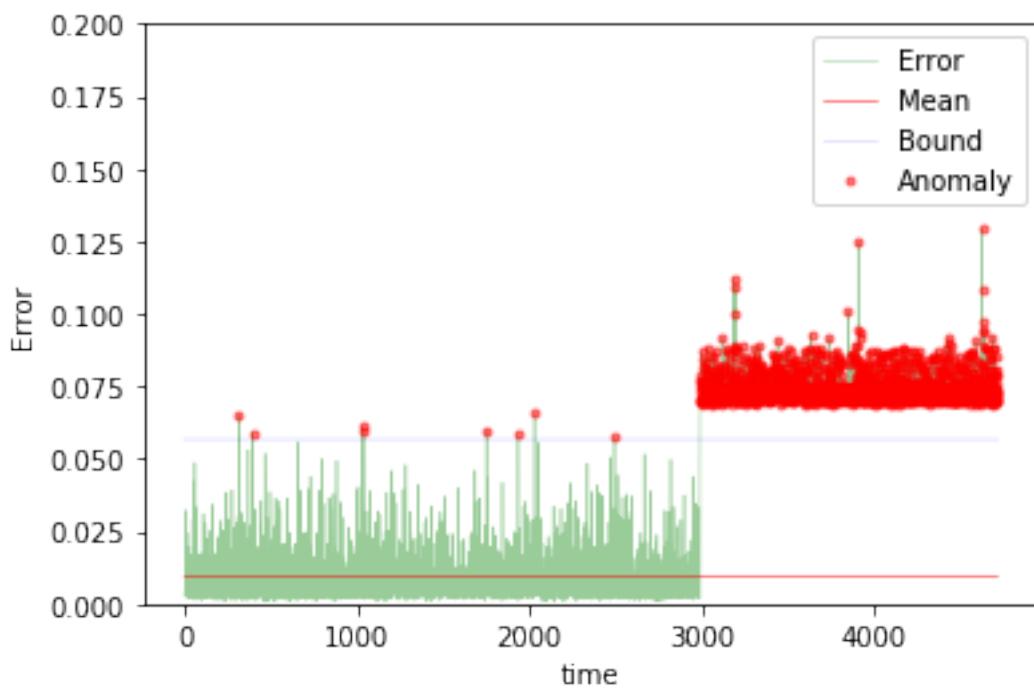
The mean error for gru2_10_anomaly_ is 0.011534300746764188 for length 4719
Testing on different app data.



The mean error for gru2_10_diff_app_ is 0.06802308220721404 for length 4719
Testing on App change synthetic data.



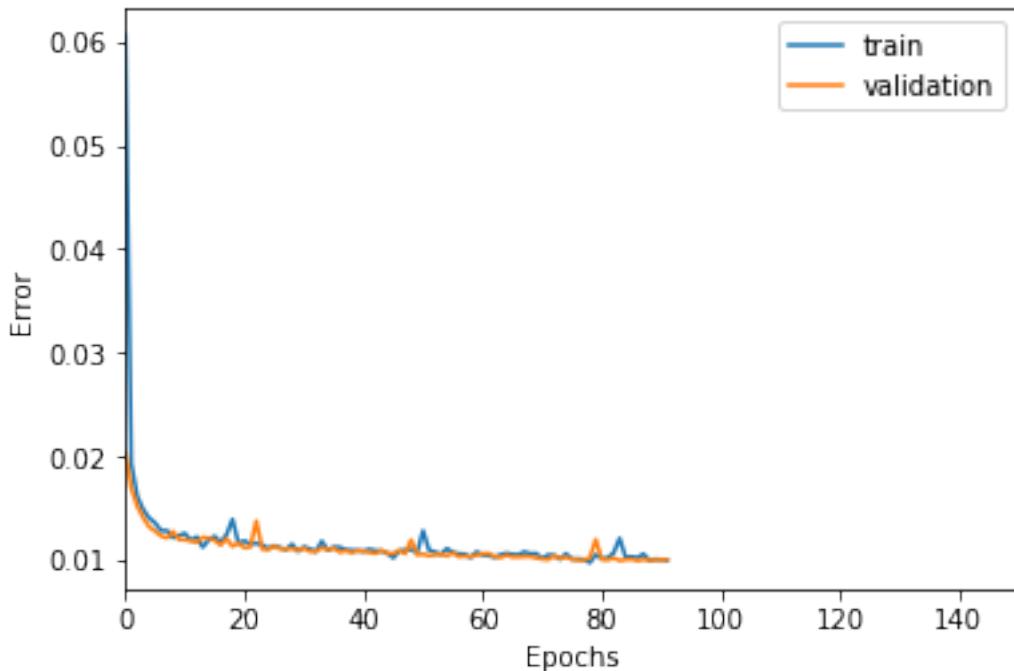
The mean error for gru2_10_app_change_ is 0.024400985301707773 for length 4719
Testing on Net flood synthetic data.



```
The mean error for gru2_10_net_flood_ is 0.03325063966955956 for length 4719  
=====
```

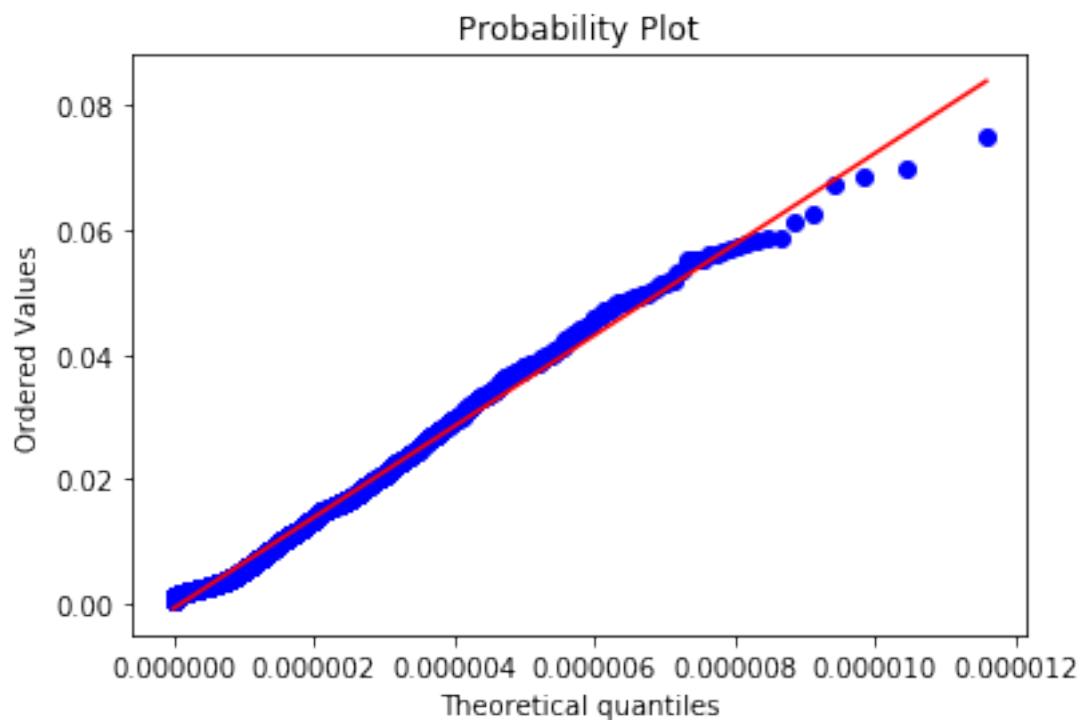
20 steps

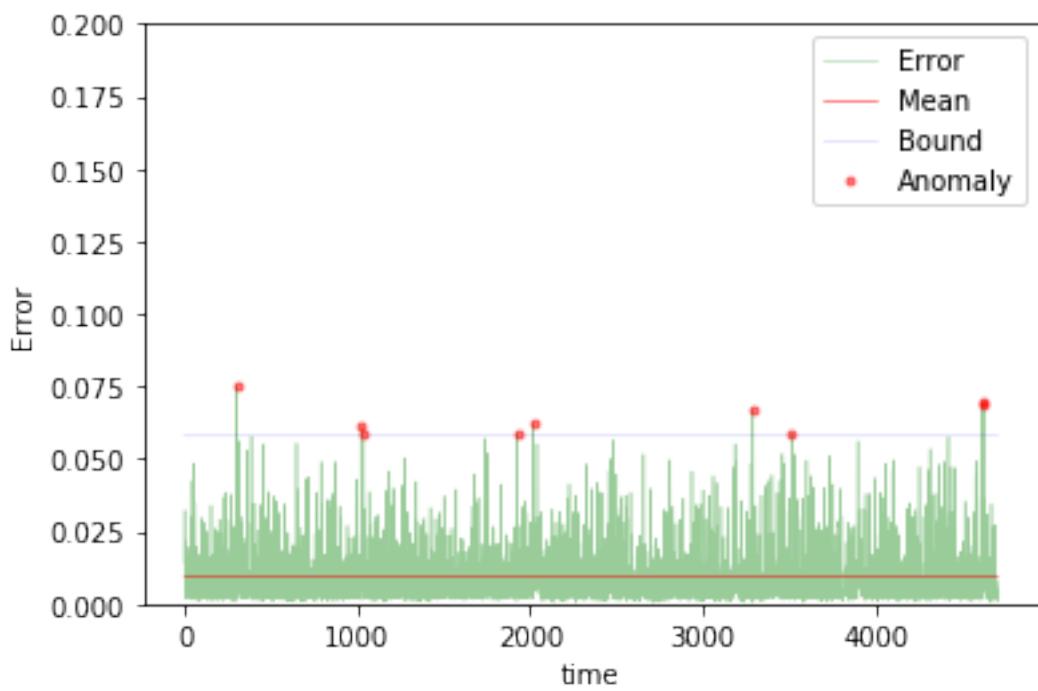
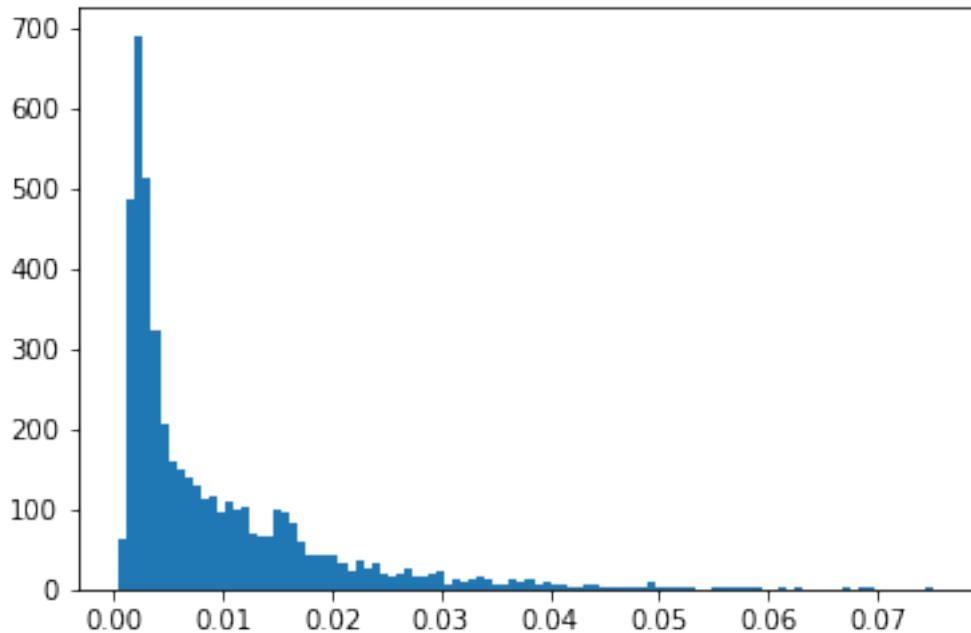
```
In [188]: TIMESTEPS = 20  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS,0)  
vgen = flat_generator(val_X, TIMESTEPS,0)  
name = "gru2_20"  
  
In [189]: input_layer = Input(shape=(TIMESTEPS,DIM))  
hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)  
hidden = GRU(10, activation='relu')(hidden)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [190]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [191]: train(model, tgen, vgen, name=name)  
test(model, ravel=0, name=name, window=TIMESTEPS)
```



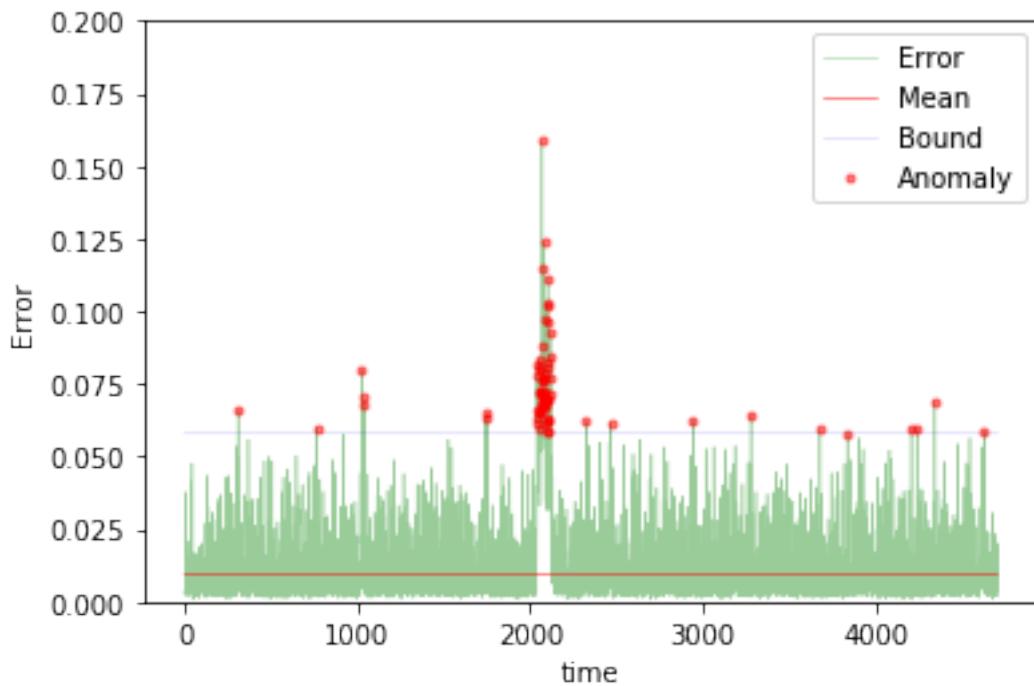
```
Training loss for final epoch is 0.010034221062785946
Validation loss for final epoch is 0.010070356840849854
----- Beginning tests for gru2_20 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

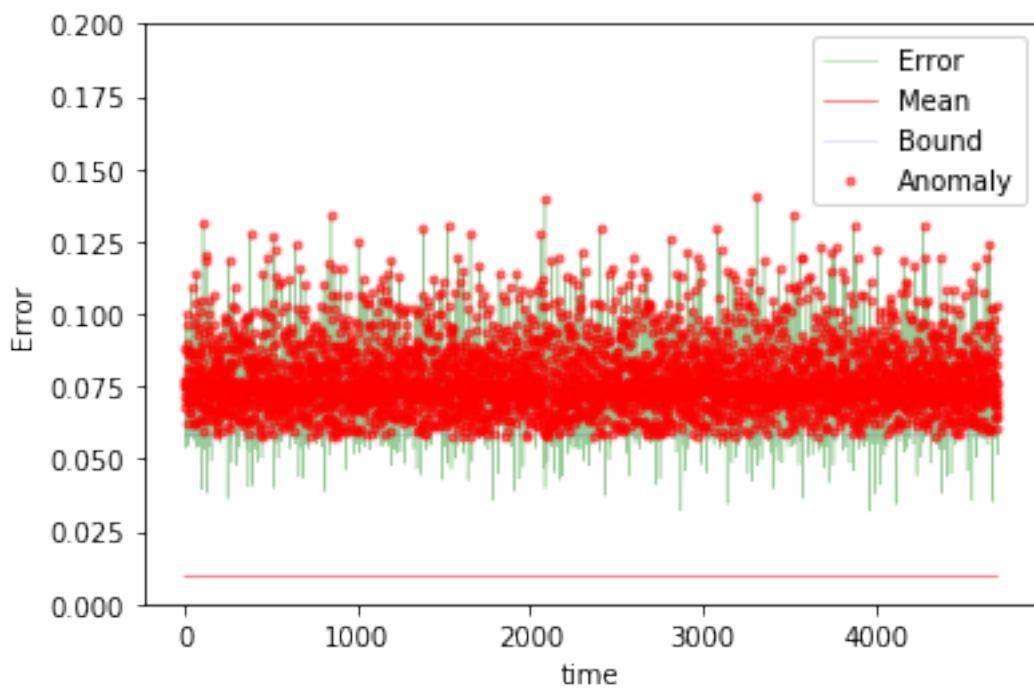




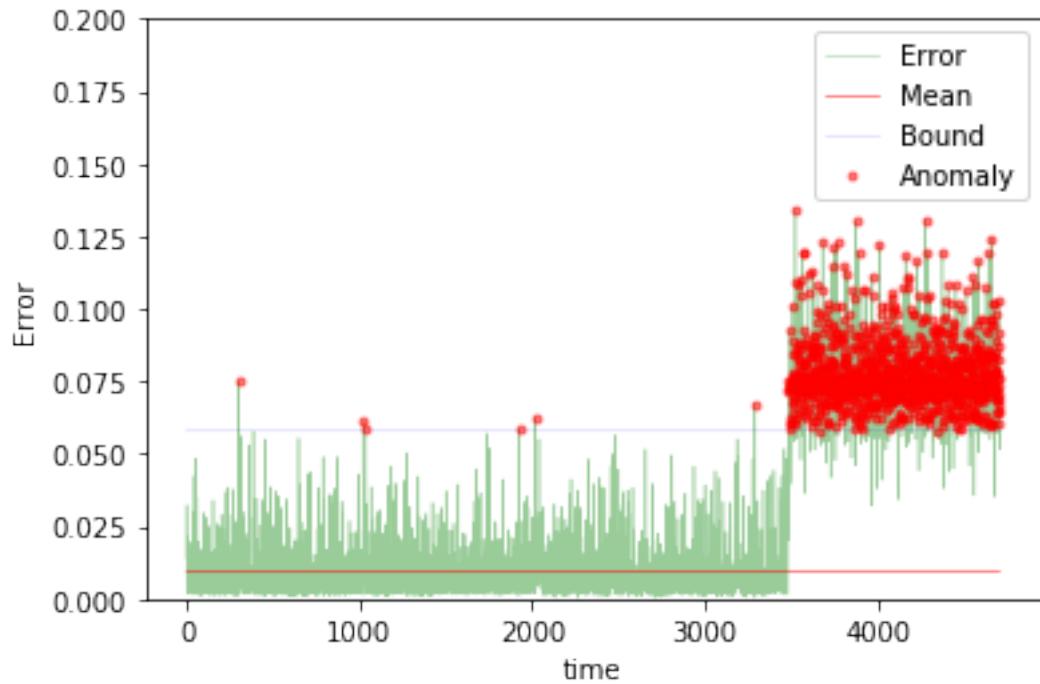
The mean error for gru2_20_normal_ is 0.009304604374833046 for length 4709
Testing on anomaly data.



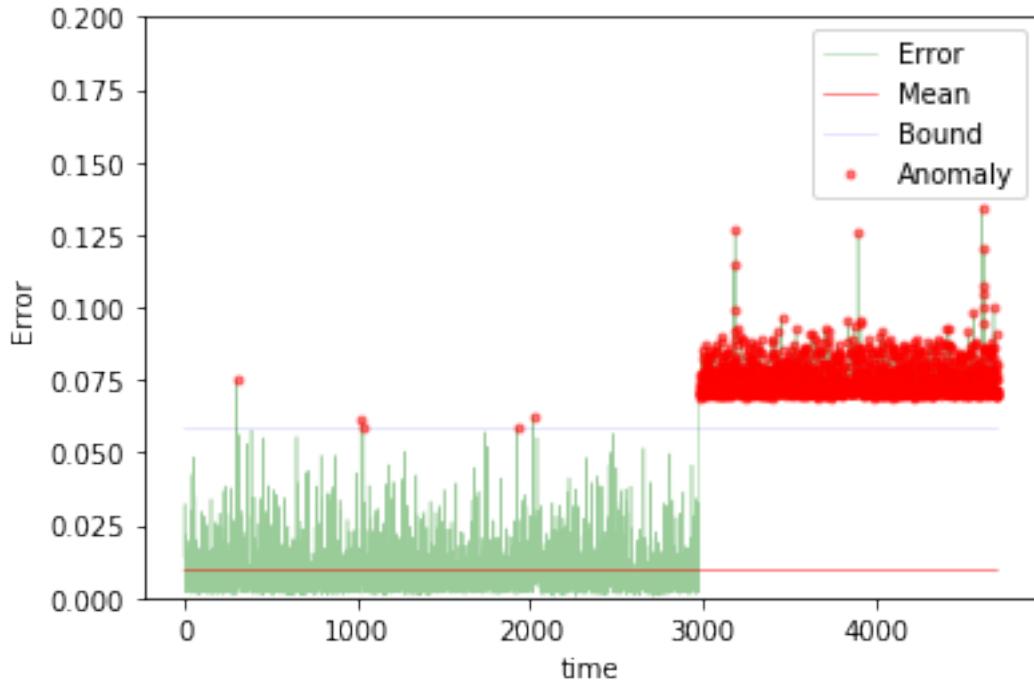
The mean error for gru2_20_anomaly_ is 0.01169213177449578 for length 4709
Testing on different app data.



The mean error for gru2_20_diff_app_ is 0.07610658294033276 for length 4709
Testing on App change synthetic data.



The mean error for gru2_20_app_change_ is 0.026614875030745777 for length 4709
Testing on Net flood synthetic data.



```
The mean error for gru2_20_net_flood_ is 0.03337780581912887 for length 4709
=====
```

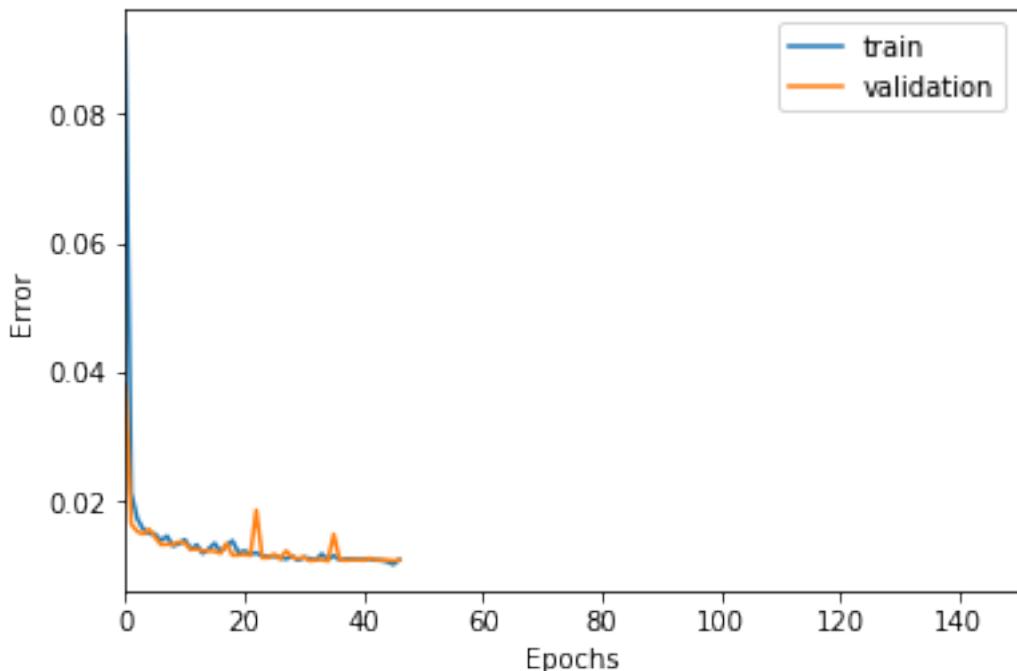
50 steps

```
In [192]: TIMESTEPS = 50
DIM = 29
tgen = flat_generator(X, TIMESTEPS,0)
vgen = flat_generator(val_X, TIMESTEPS,0)
name = "gru2_50"

In [193]: 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)

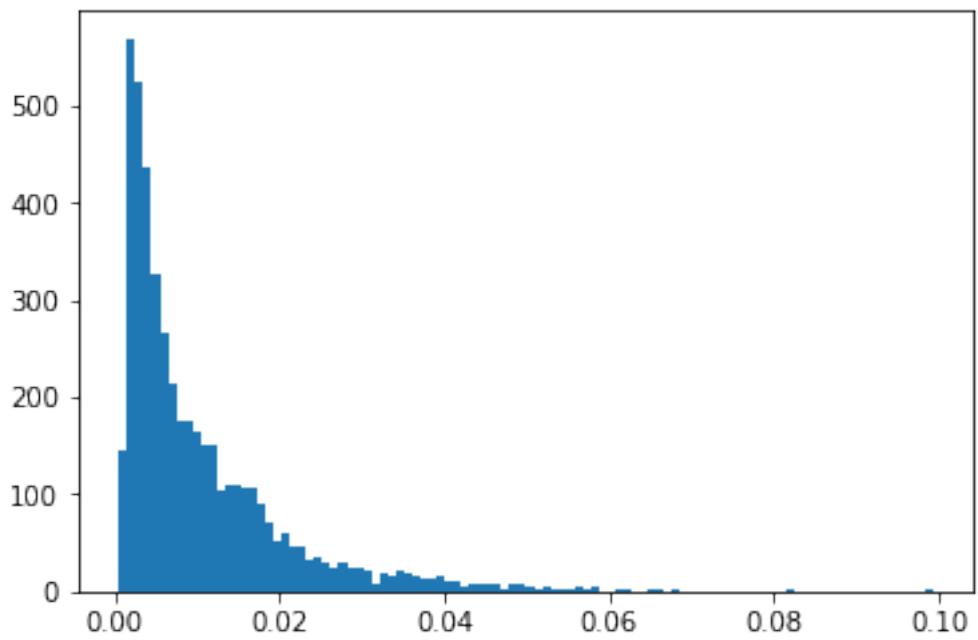
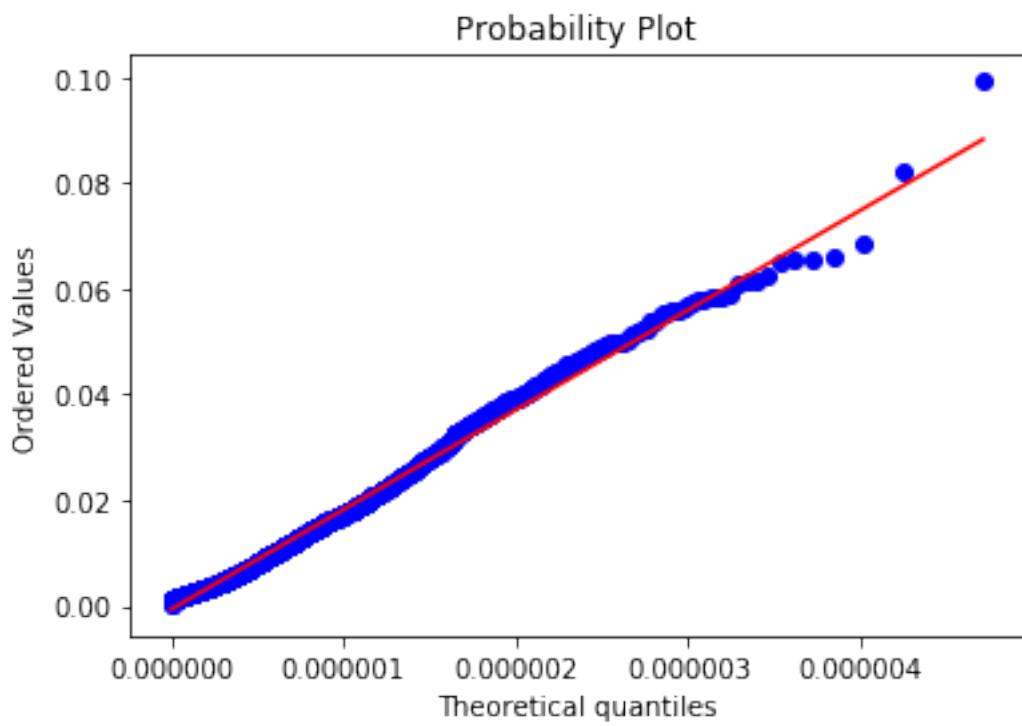
In [194]: model = Model(input_layer, output)
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

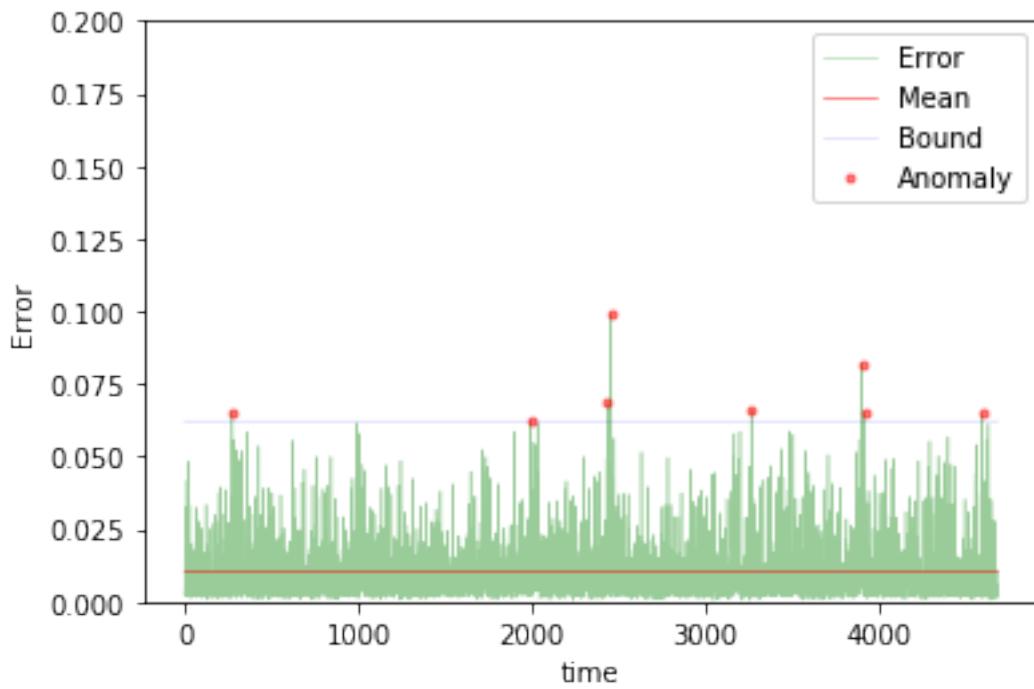
In [195]: train(model, tgen, vgen, name=name)
test(model, ravel=0, name=name, window=TIMESTEPS)
```



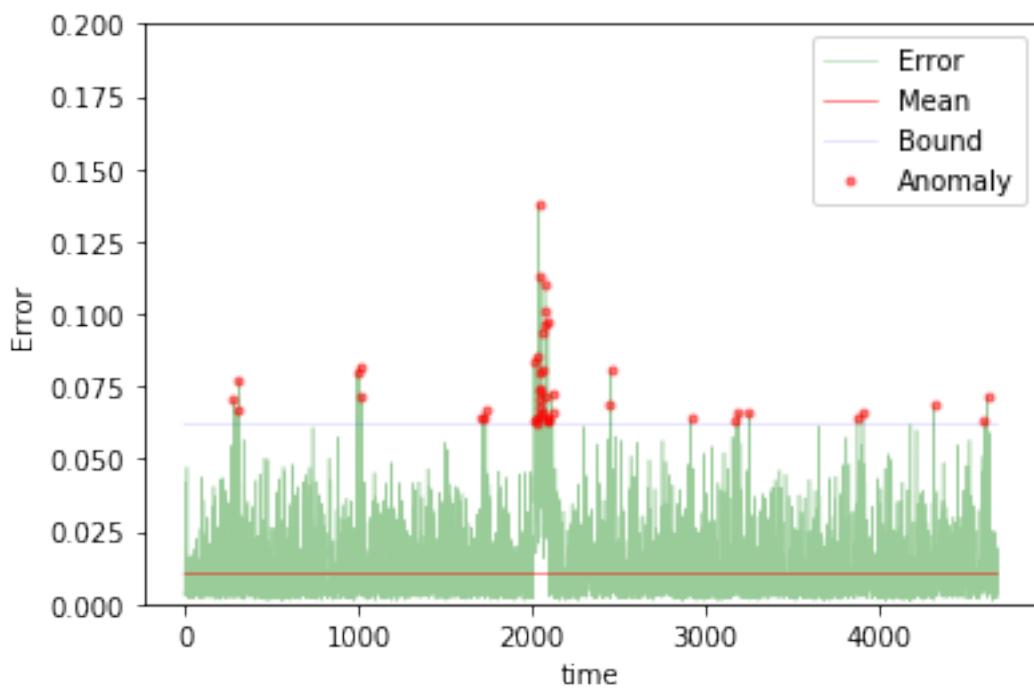
```
Training loss for final epoch is 0.010937342389719561
Validation loss for final epoch is 0.01086876728804782
----- Beginning tests for gru2_50 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

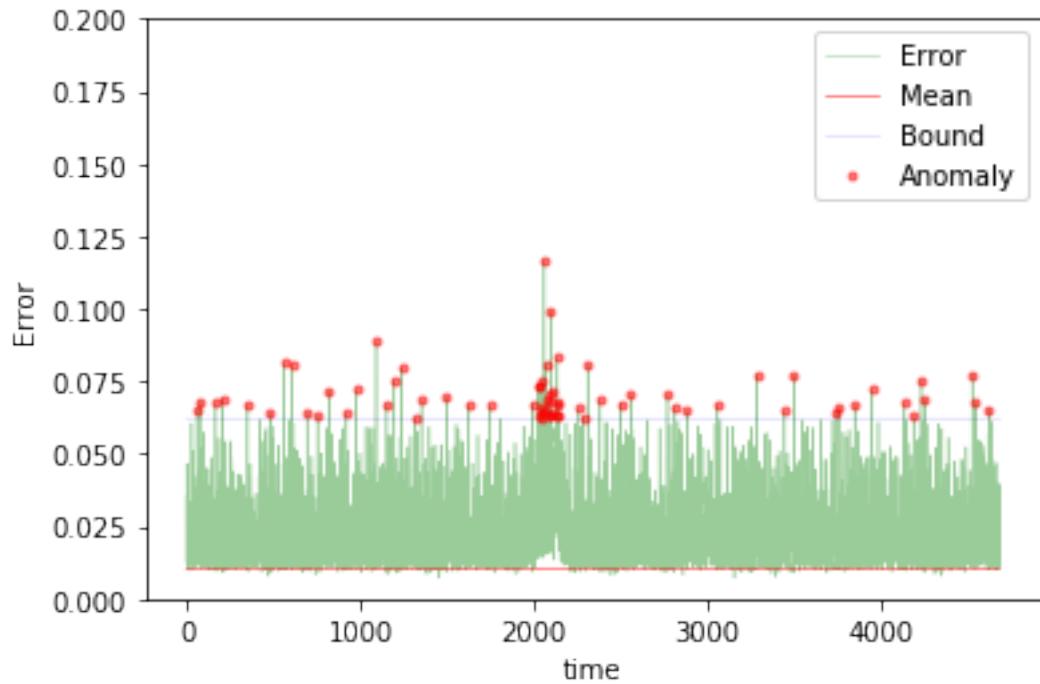




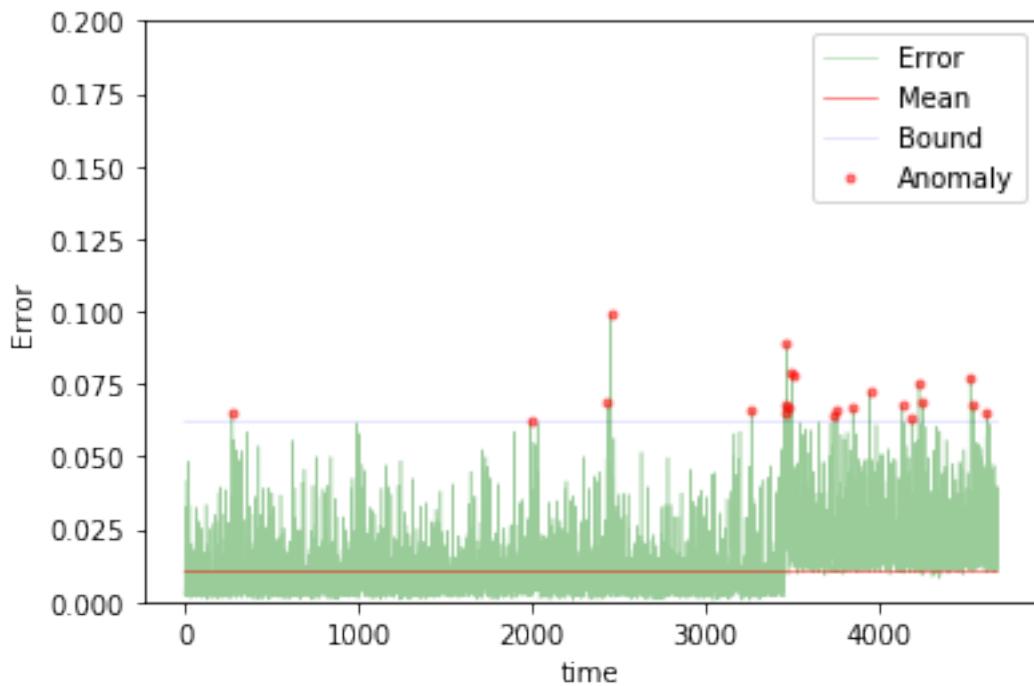
The mean error for gru2_50_normal_ is 0.010433076382721194 for length 4679
Testing on anomaly data.



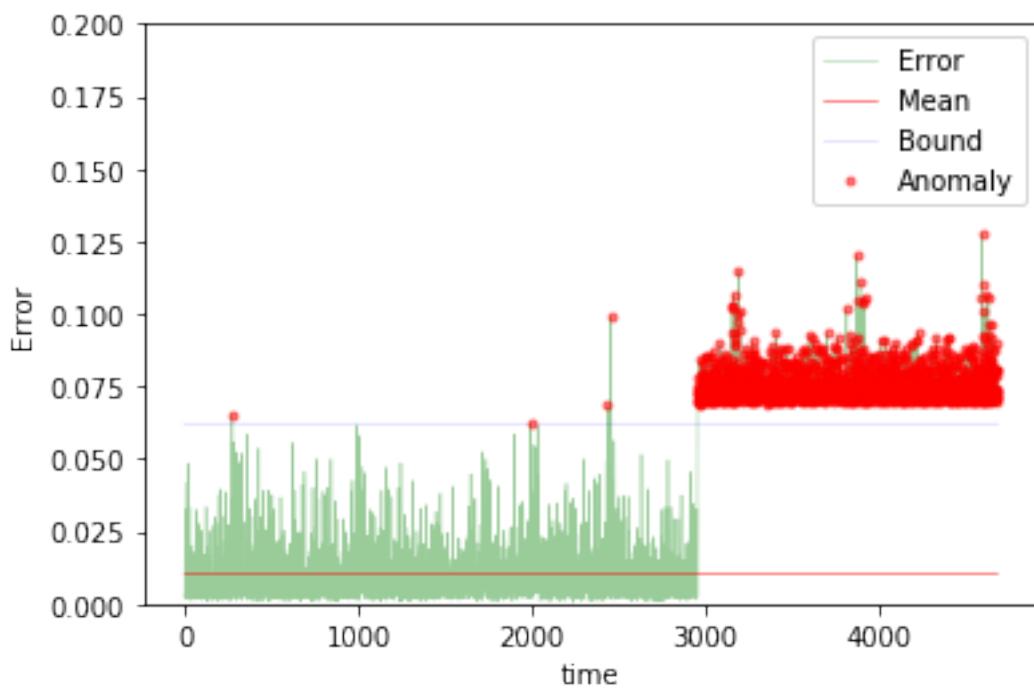
The mean error for gru2_50_anomaly_ is 0.012325716572733816 for length 4679
Testing on different app data.



The mean error for gru2_50_diff_app_ is 0.023388990814968 for length 4679
Testing on App change synthetic data.



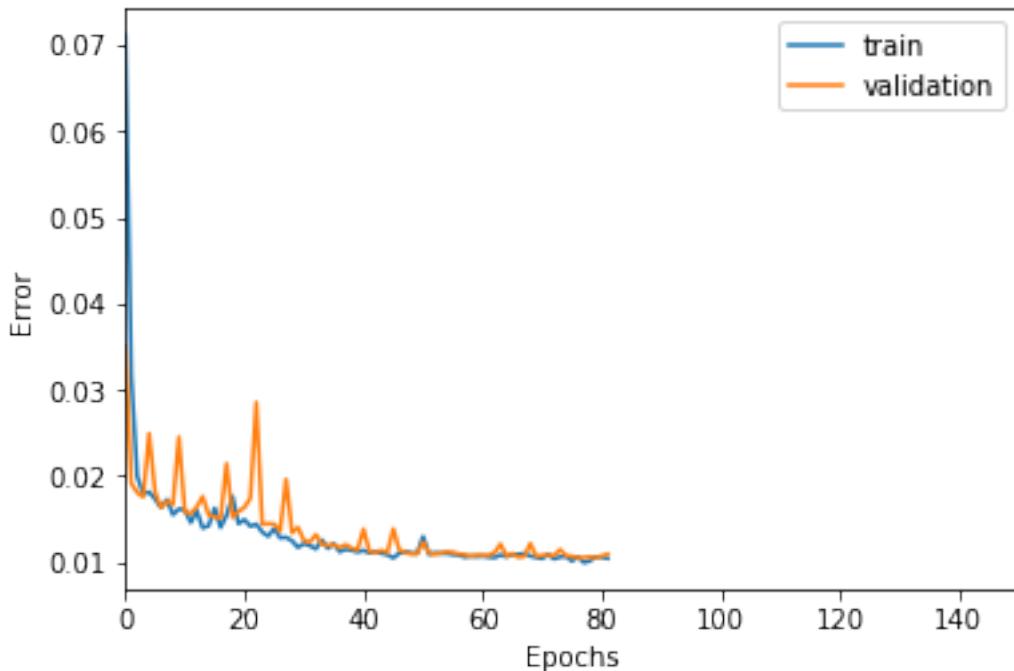
The mean error for gru2_50_app_change_ is 0.013871584419475481 for length 4679
Testing on Net flood synthetic data.



```
The mean error for gru2_50_net_flood_ is 0.03463766610878068 for length 4679  
=====
```

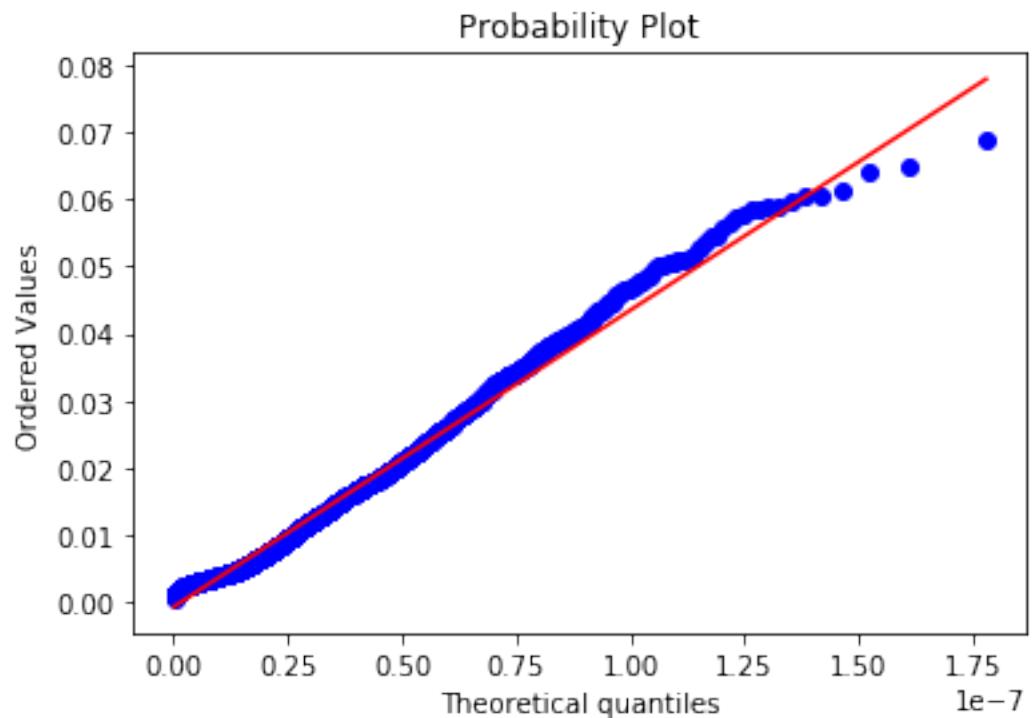
100 steps

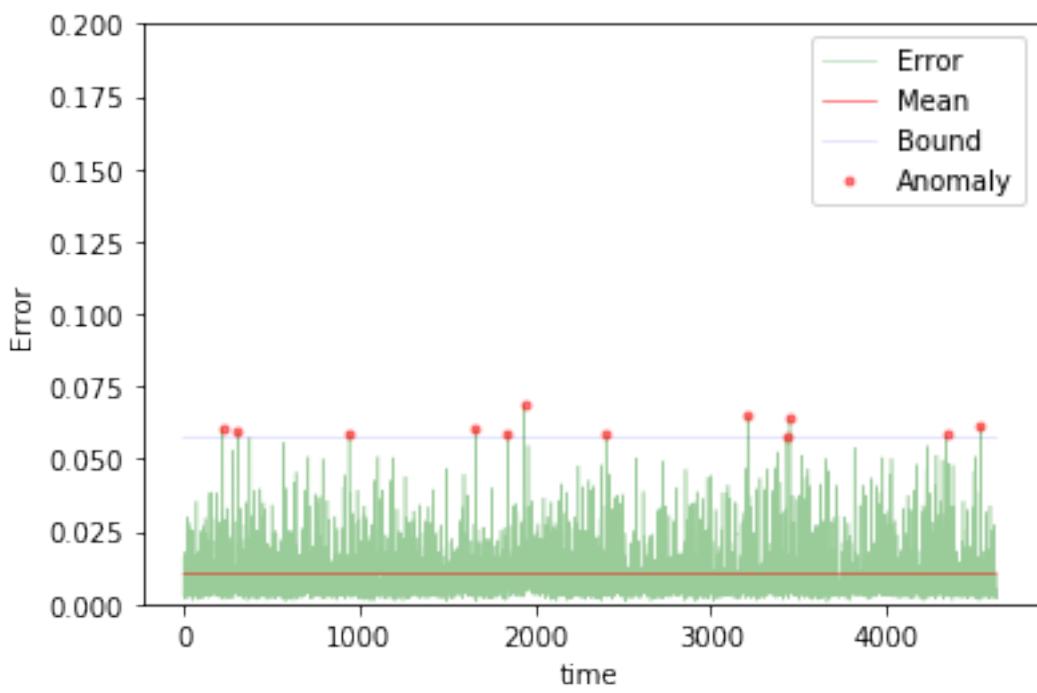
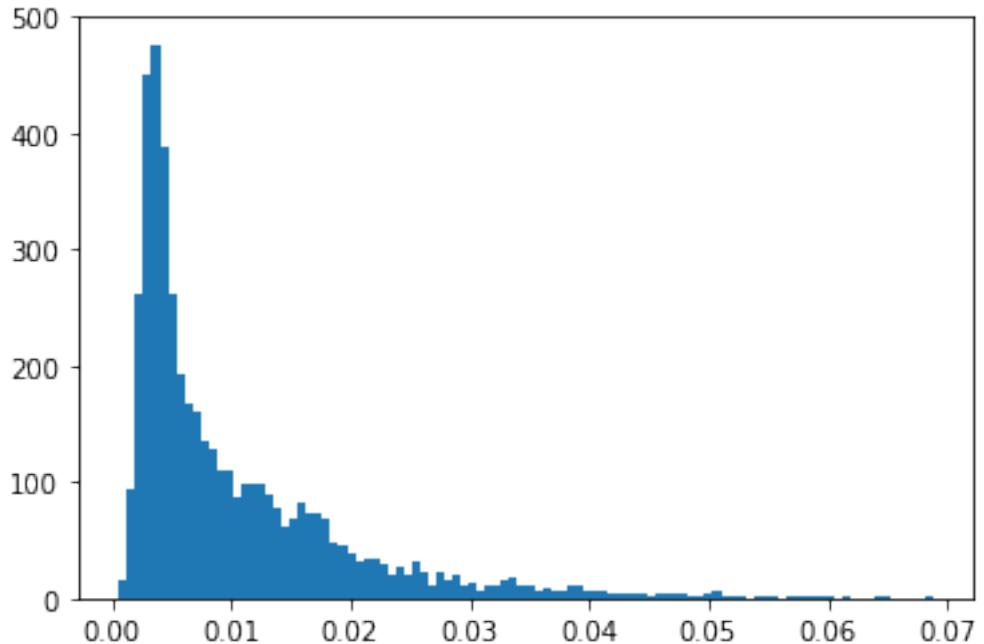
```
In [196]: TIMESTEPS = 100  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS,0)  
vgen = flat_generator(val_X, TIMESTEPS,0)  
name = "gru2_100"  
  
In [197]: input_layer = Input(shape=(TIMESTEPS,DIM))  
hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)  
hidden = GRU(10, activation='relu')(hidden)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [198]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [199]: train(model, tgen, vgen, name=name)  
test(model, ravel=0, name=name, window=TIMESTEPS)
```



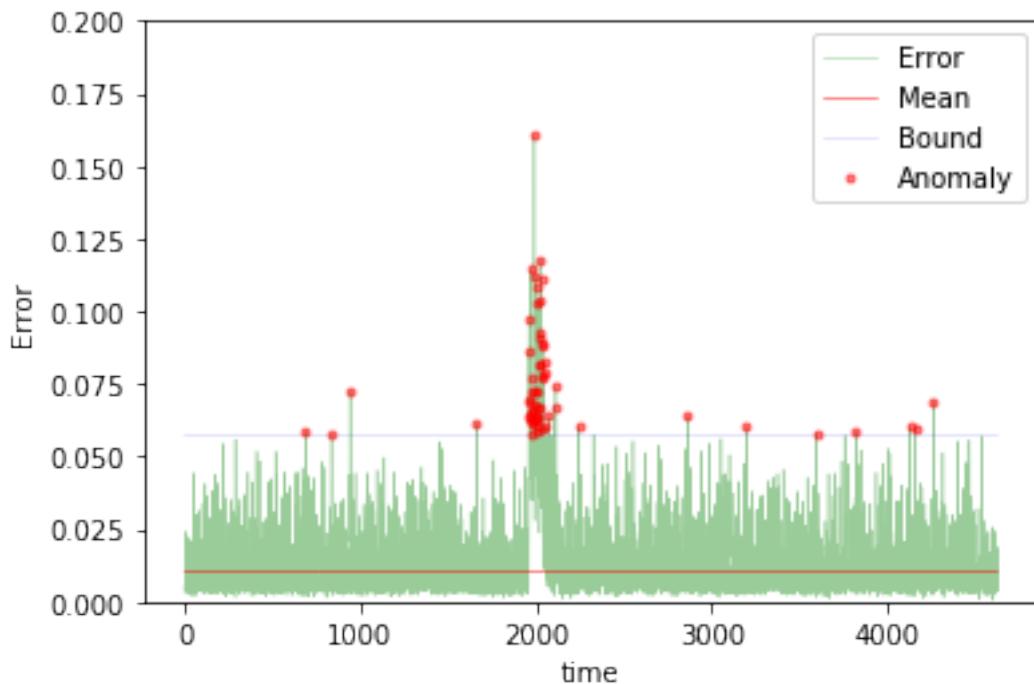
```
Training loss for final epoch is 0.010426716980990023
Validation loss for final epoch is 0.010911808256525546
----- Beginning tests for gru2_100 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

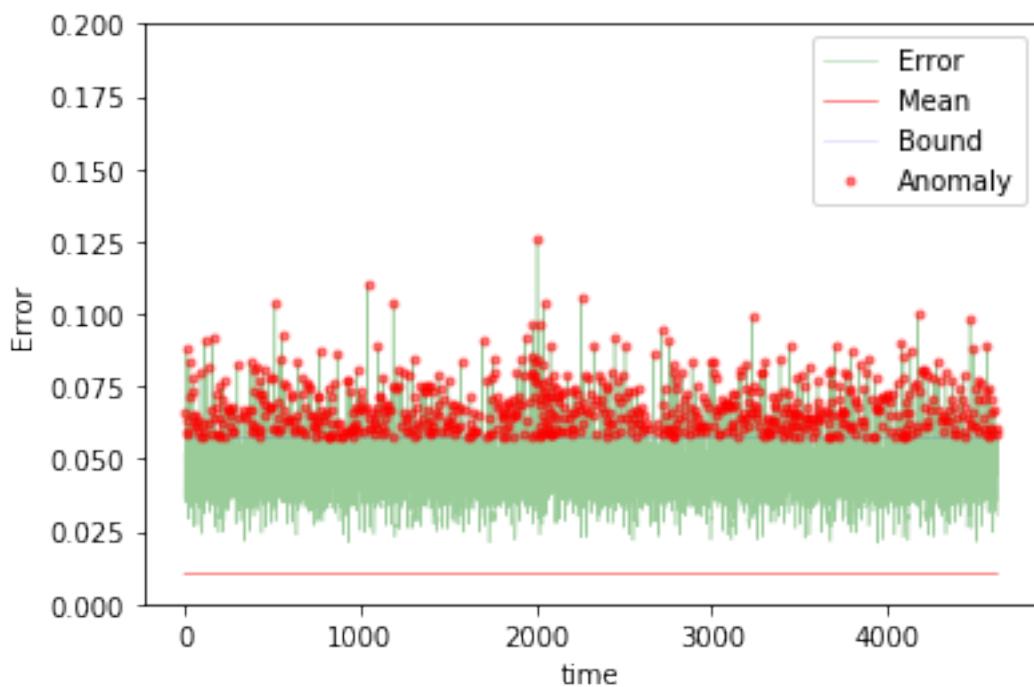




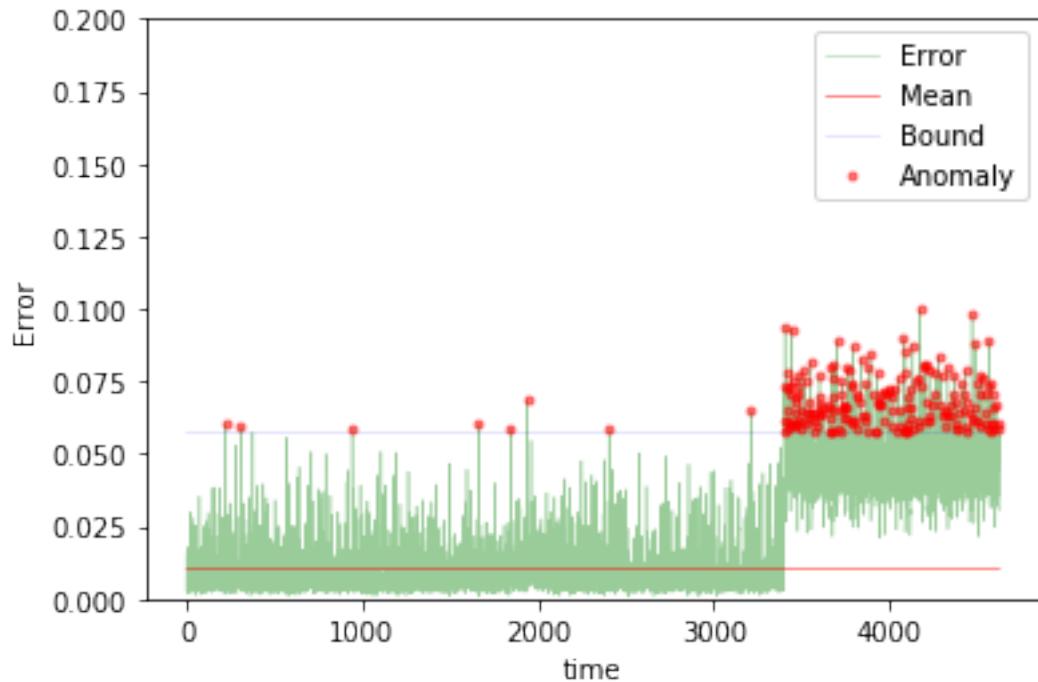
The mean error for gru2_100_normal_ is 0.010370510167541393 for length 4629
Testing on anomaly data.



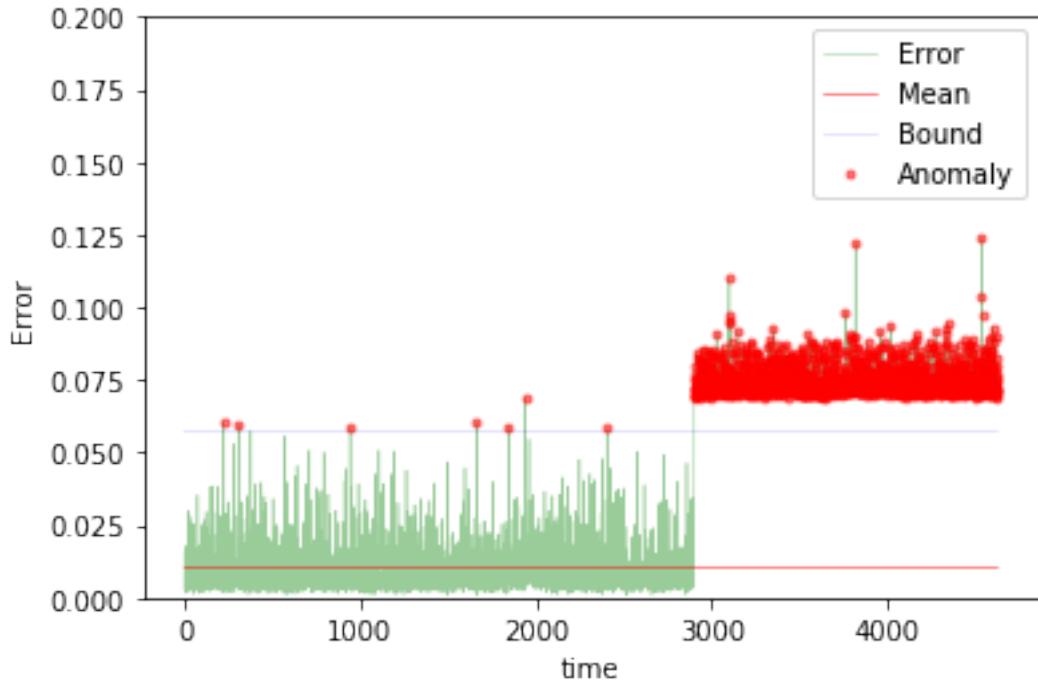
The mean error for gru2_100_anomaly_ is 0.012758181058038961 for length 4629
Testing on different app data.



The mean error for gru2_100_diff_app_ is 0.04495239016916485 for length 4629
Testing on App change synthetic data.



The mean error for gru2_100_app_change_ is 0.01951252136162146 for length 4629
Testing on Net flood synthetic data.



The mean error for gru2_100_net_flood_ is 0.034484630560082 for length 4629
=====

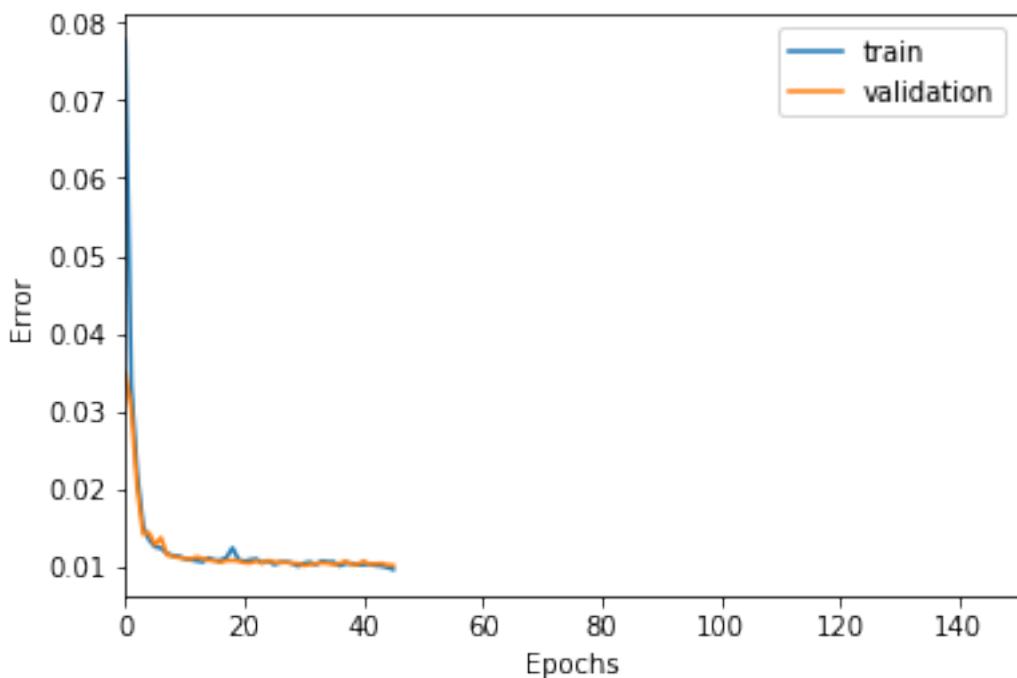
200 steps

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

In [201]: 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)

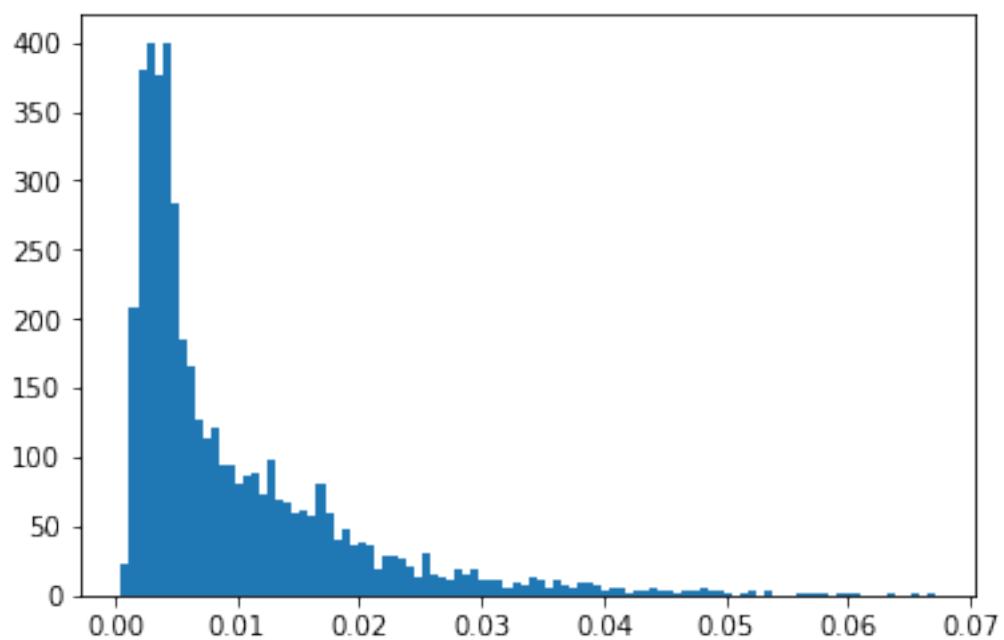
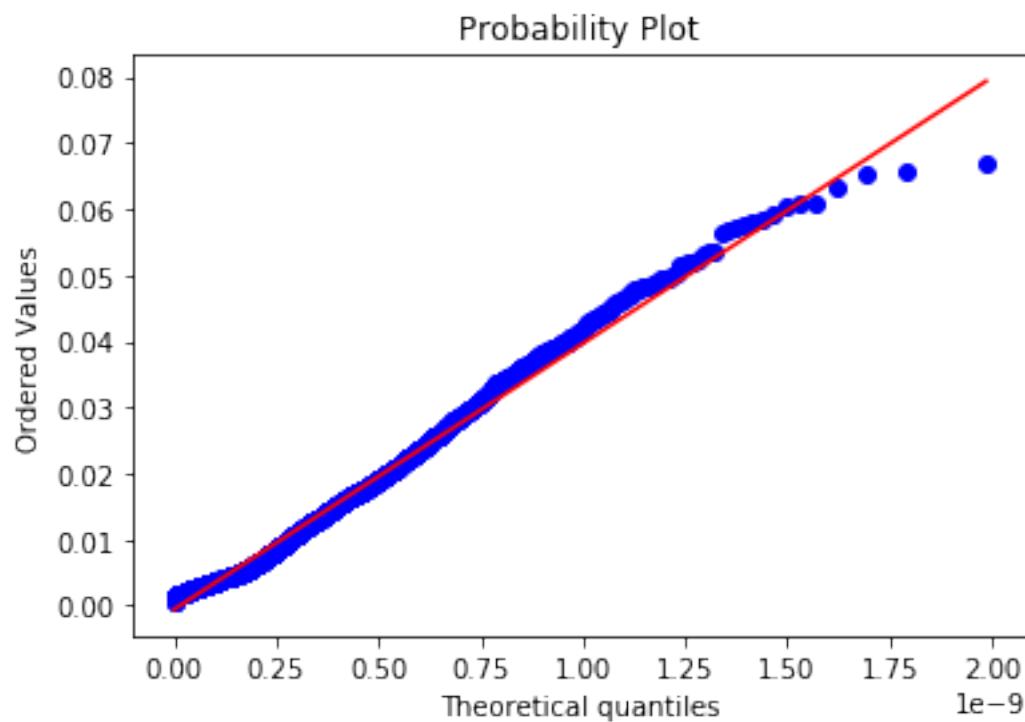
In [202]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

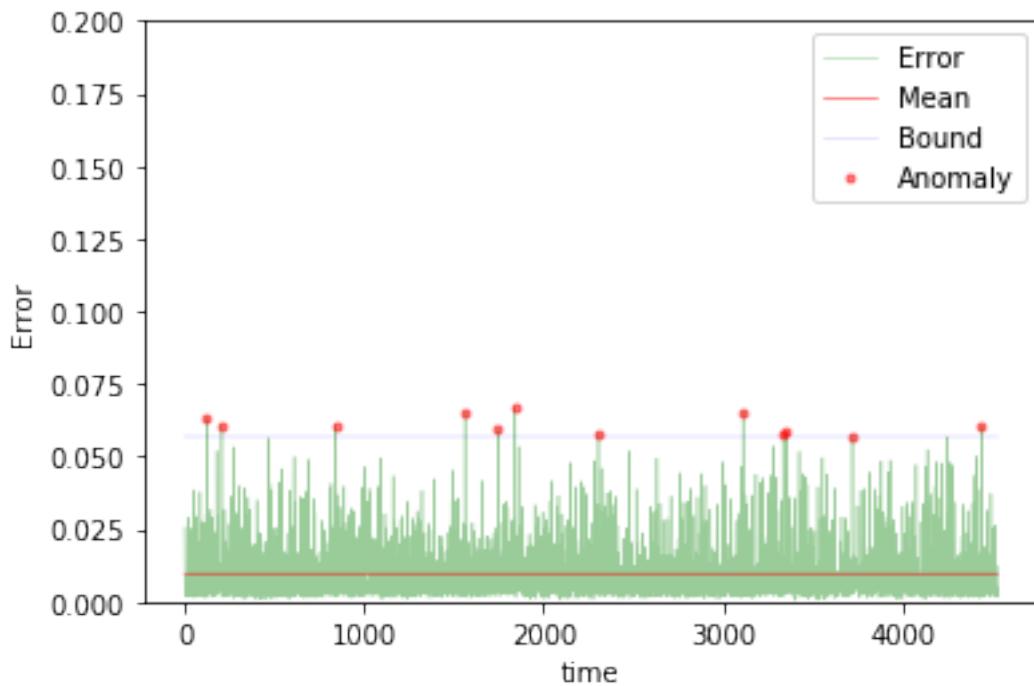
In [203]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```



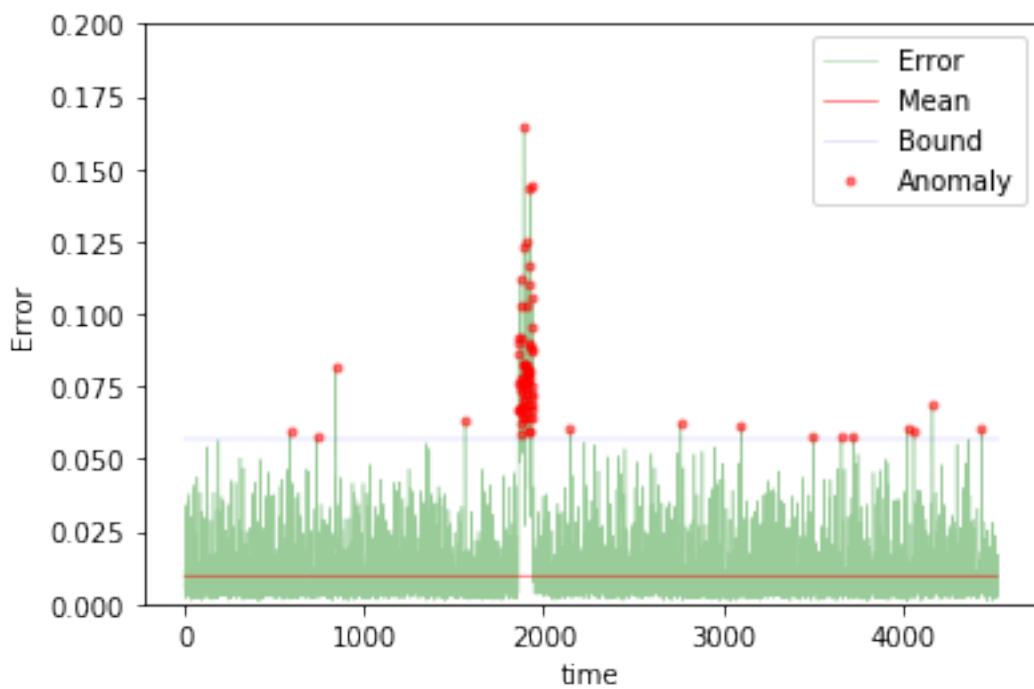
```
Training loss for final epoch is 0.009627584872650913
Validation loss for final epoch is 0.010159854871453718
----- Beginning tests for gru2_200 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
    warnings.warn(msg, RuntimeWarning)
```

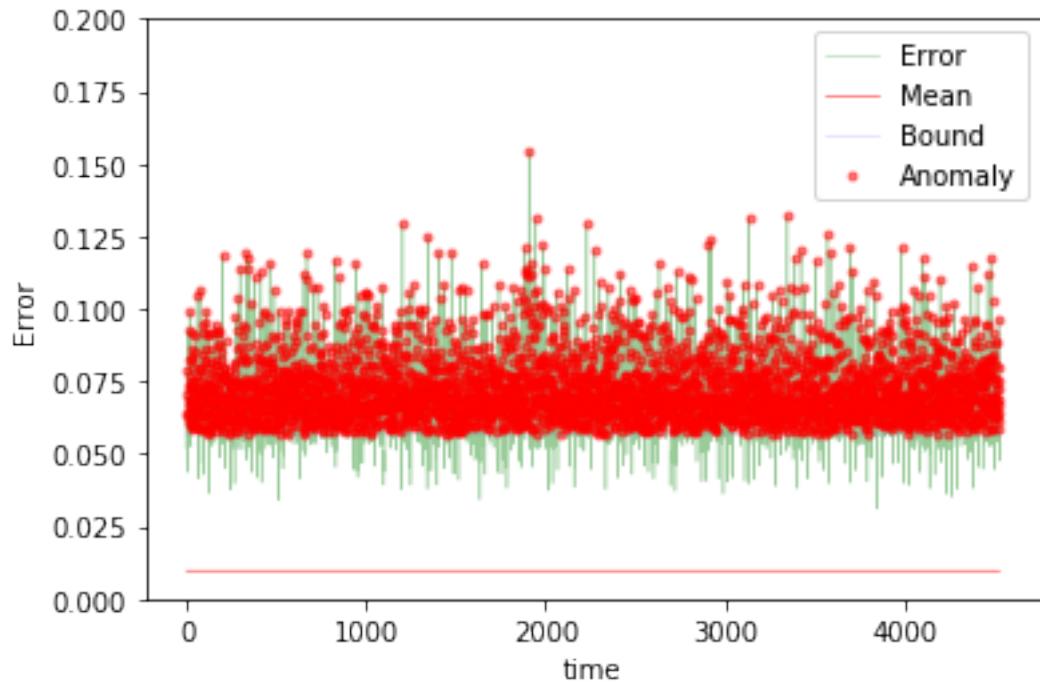




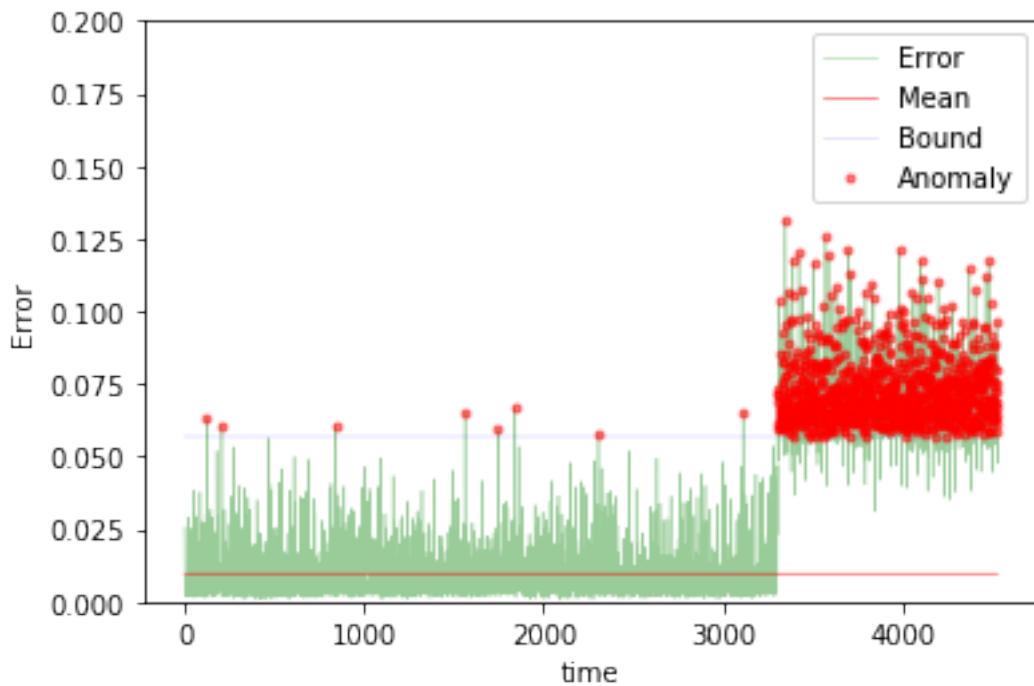
The mean error for gru2_200_normal_ is 0.00974393464783124 for length 4529
Testing on anomaly data.



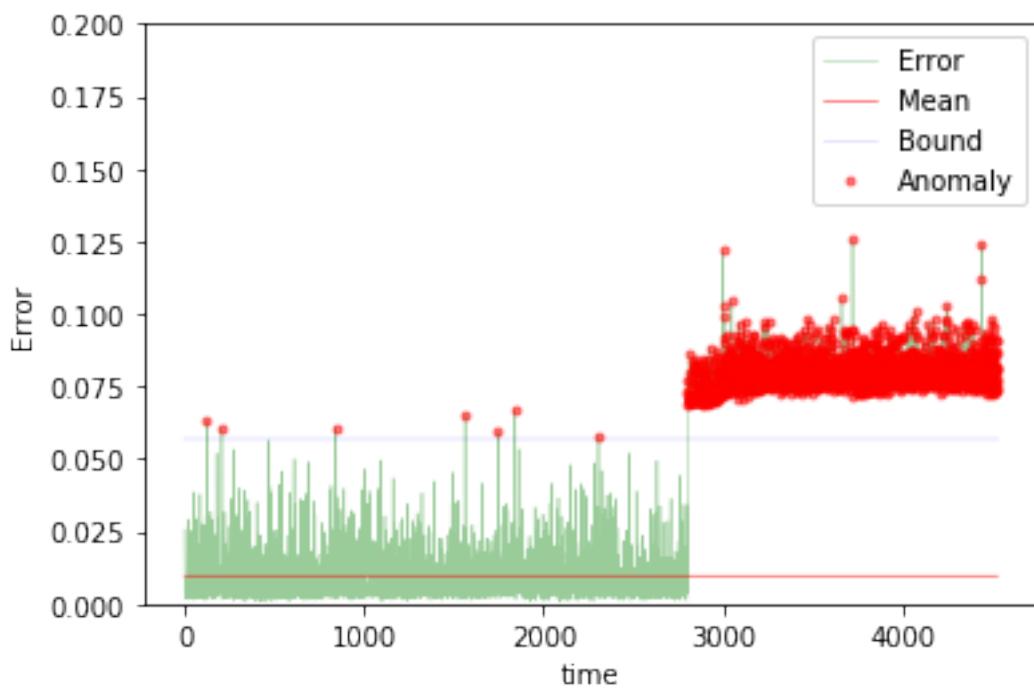
The mean error for gru2_200_anomaly_ is 0.011677434468942927 for length 4529
Testing on different app data.



The mean error for gru2_200_diff_app_ is 0.07034752420706437 for length 4529
Testing on App change synthetic data.



The mean error for gru2_200_app_change_ is 0.026010884701894117 for length 4529
Testing on Net flood synthetic data.

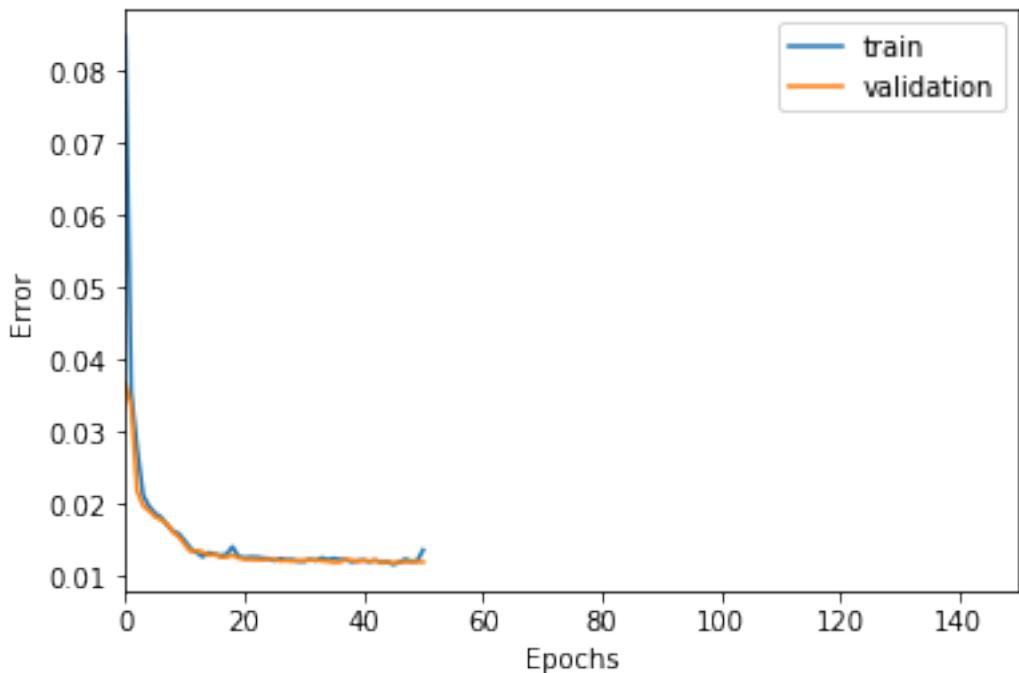


```
The mean error for gru2_200_net_flood_ is 0.036639525134781956 for length 4529  
=====
```

2.1.7 RNN with 3 GRU layers

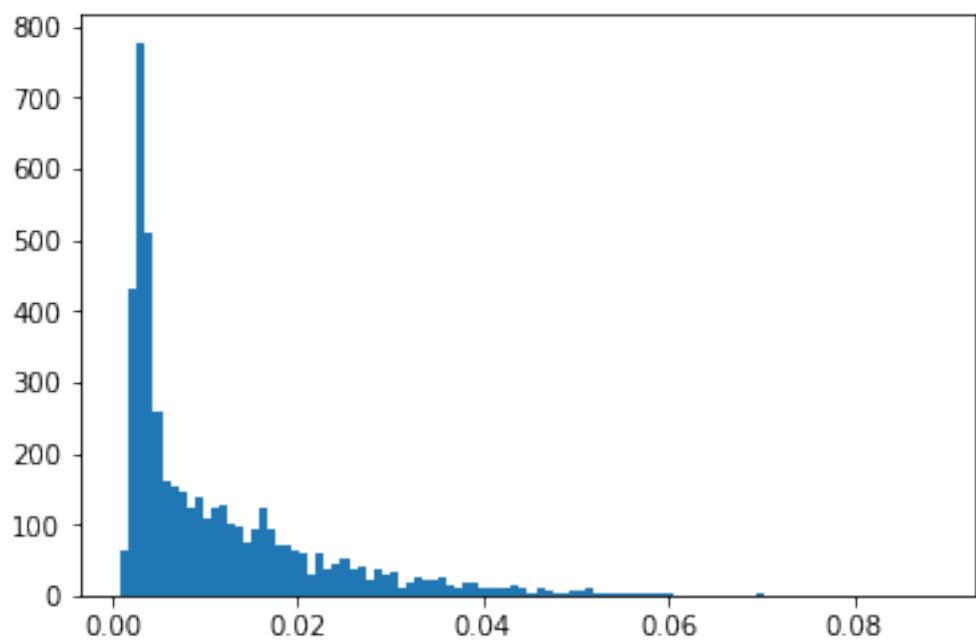
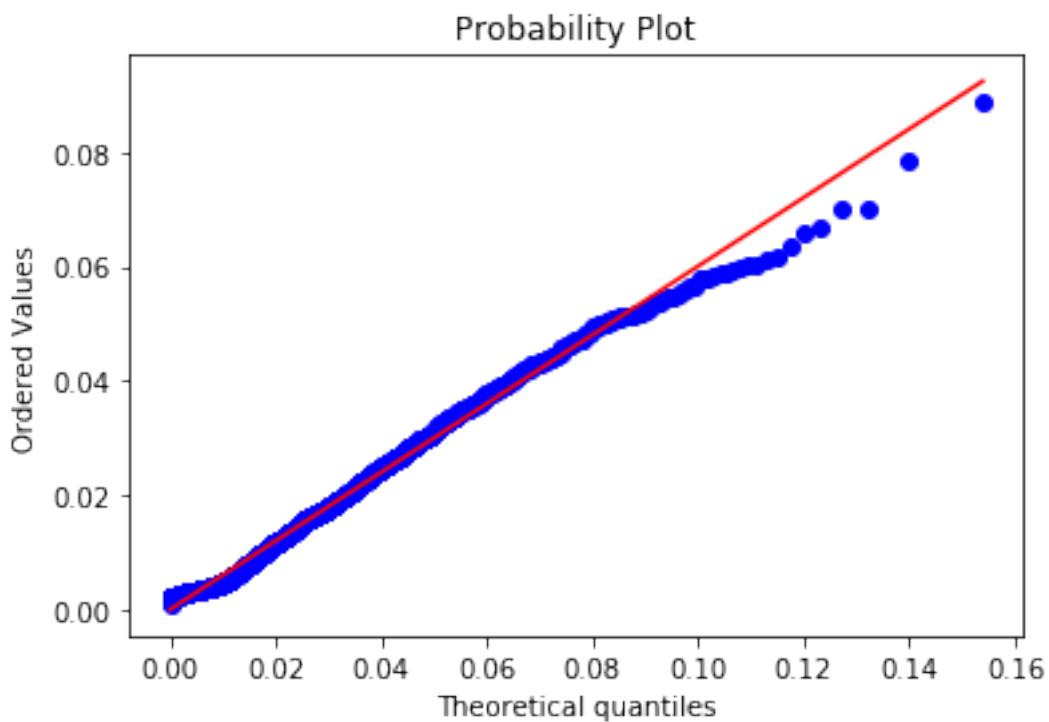
2 steps

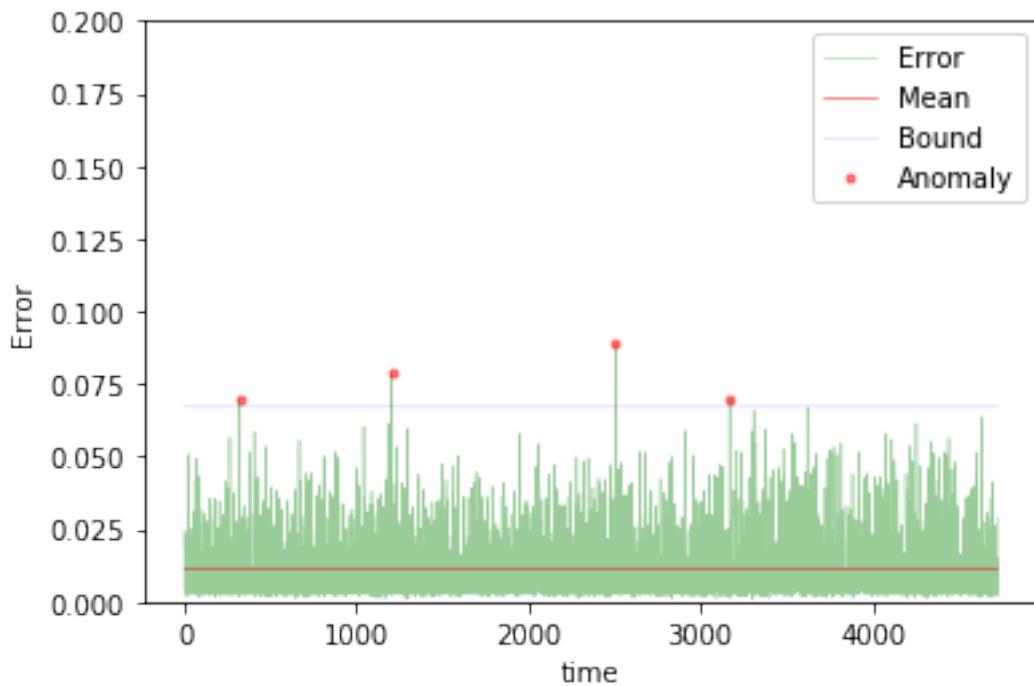
```
In [204]: TIMESTEPS = 2  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS,0)  
vgen = flat_generator(val_X, TIMESTEPS,0)  
name = "gru3_2"  
  
In [205]: input_layer = Input(shape=(TIMESTEPS,DIM))  
hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)  
hidden = GRU(10, activation='relu', return_sequences=True)(hidden)  
hidden = GRU(10, activation='relu')(hidden)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [206]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [207]: train(model, tgen, vgen, name=name)  
test(model, ravel=0, name=name, window=TIMESTEPS)
```



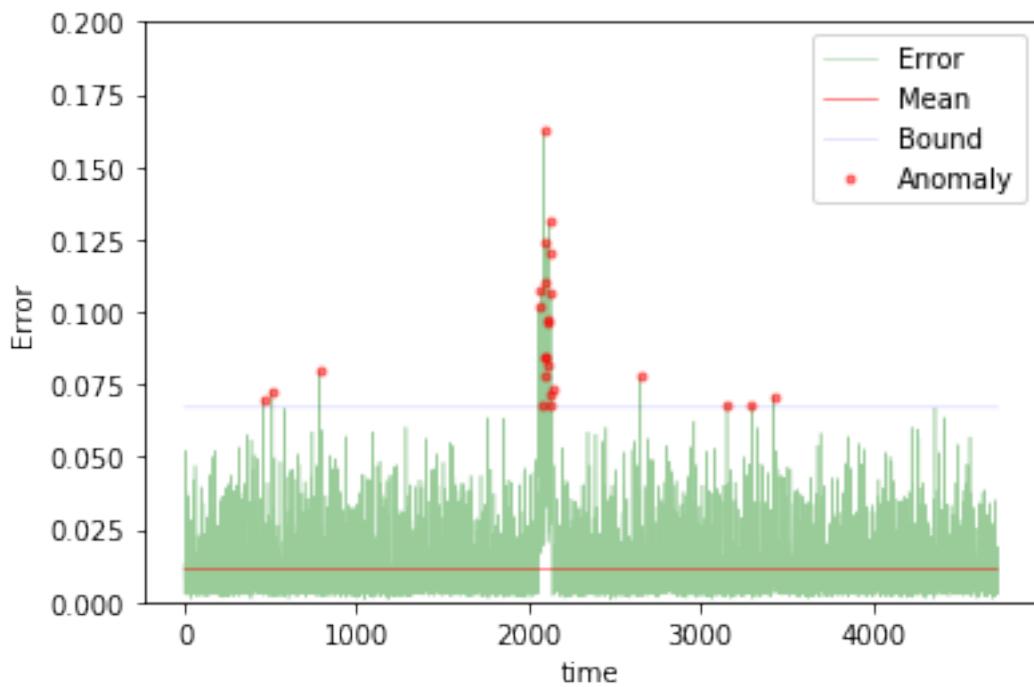
```
Training loss for final epoch is 0.013704184538917617
Validation loss for final epoch is 0.012020785969216376
----- Beginning tests for gru3_2 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
    warnings.warn(msg, RuntimeWarning)
```

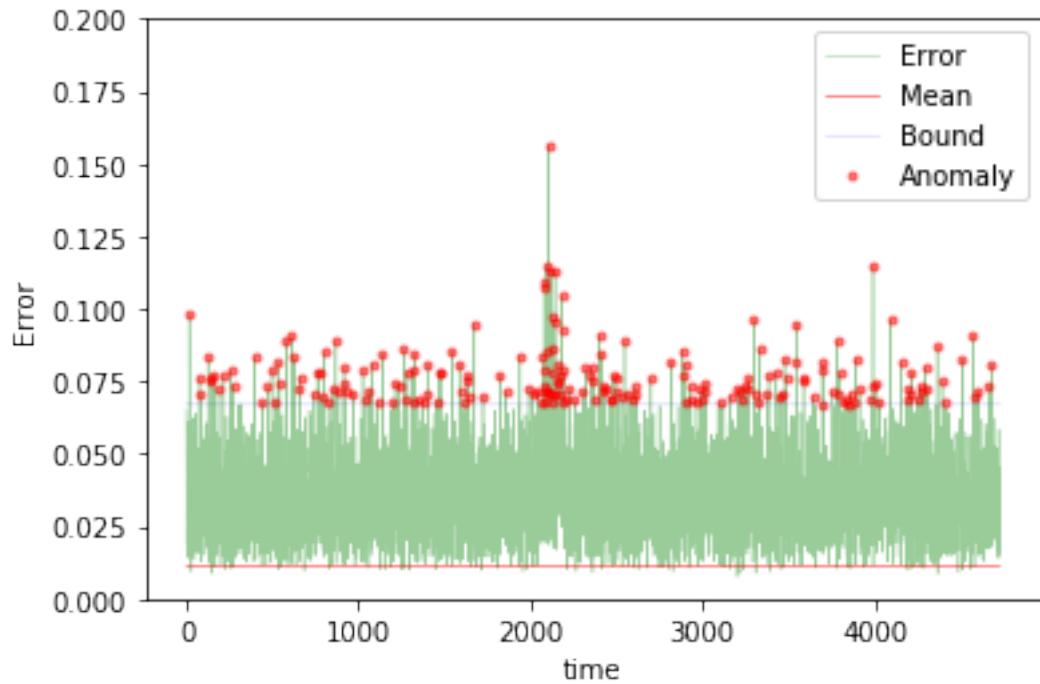




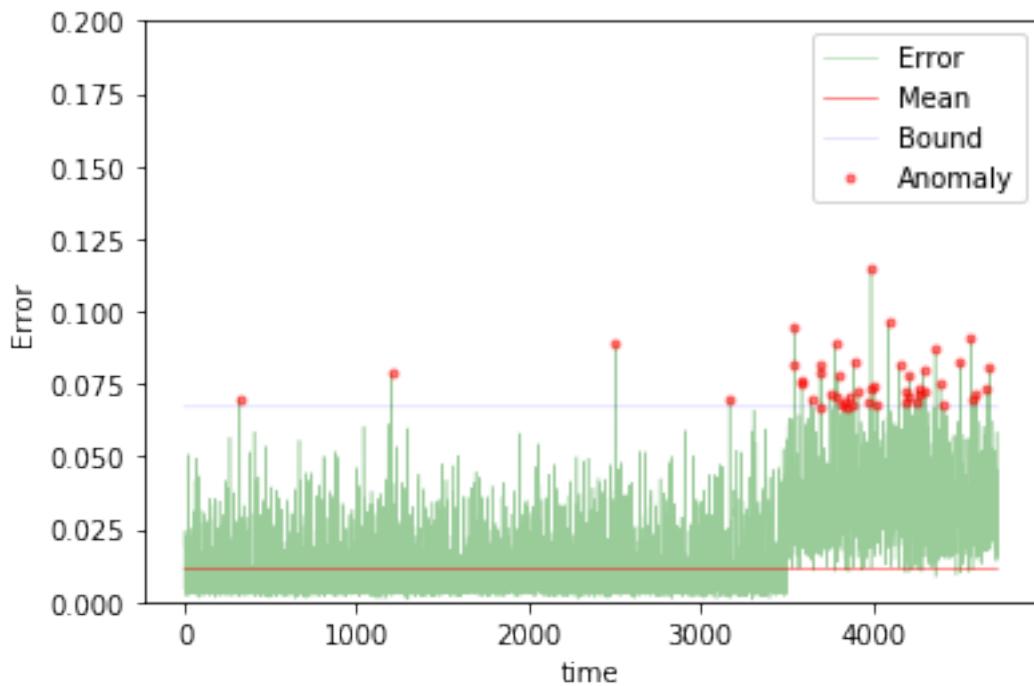
The mean error for gru3_2_normal_ is 0.011552955212682618 for length 4727
Testing on anomaly data.



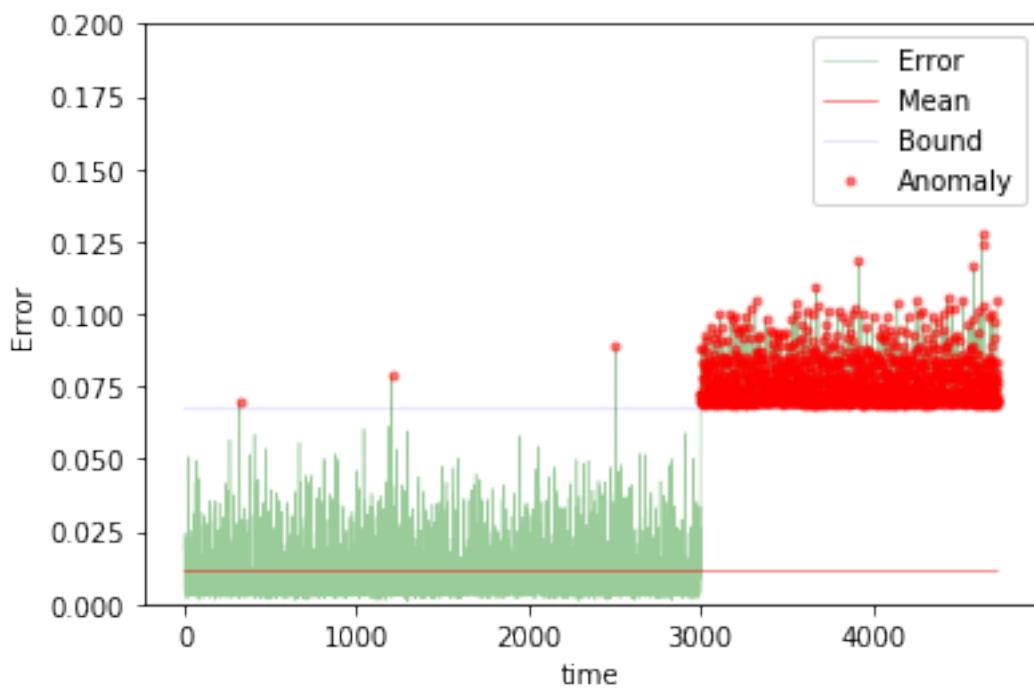
The mean error for gru3_2_anomaly_ is 0.01304741757342321 for length 4727
Testing on different app data.



The mean error for gru3_2_diff_app_ is 0.03514948885275833 for length 4727
Testing on App change synthetic data.



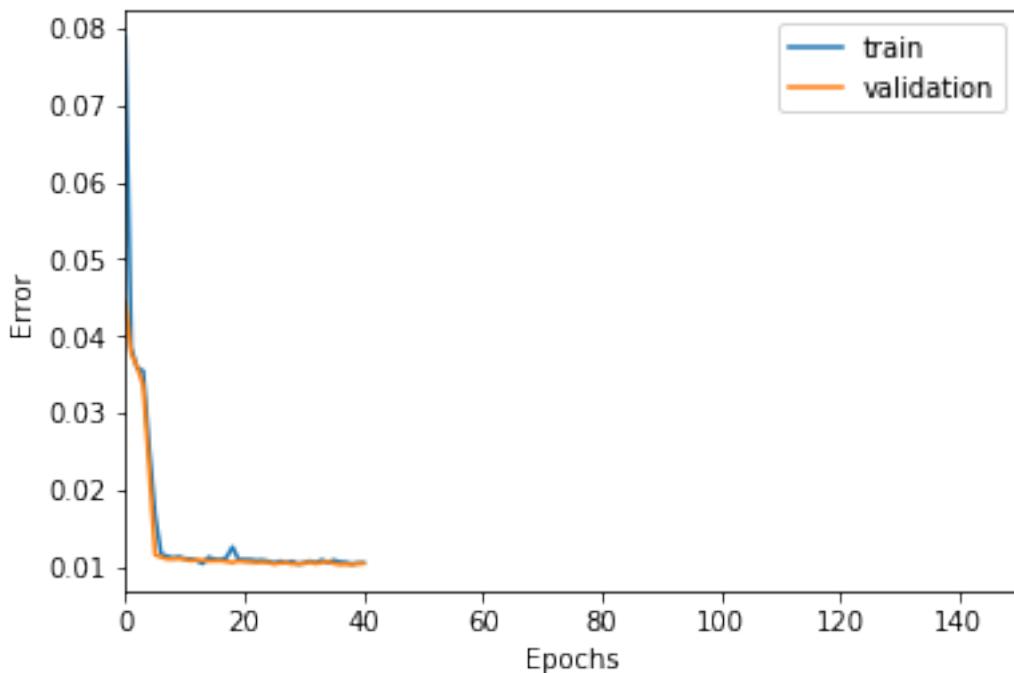
The mean error for gru3_2_app_change_ is 0.017571150048934635 for length 4727
Testing on Net flood synthetic data.



```
The mean error for gru3_2_net_flood_ is 0.03494976166506243 for length 4727  
=====
```

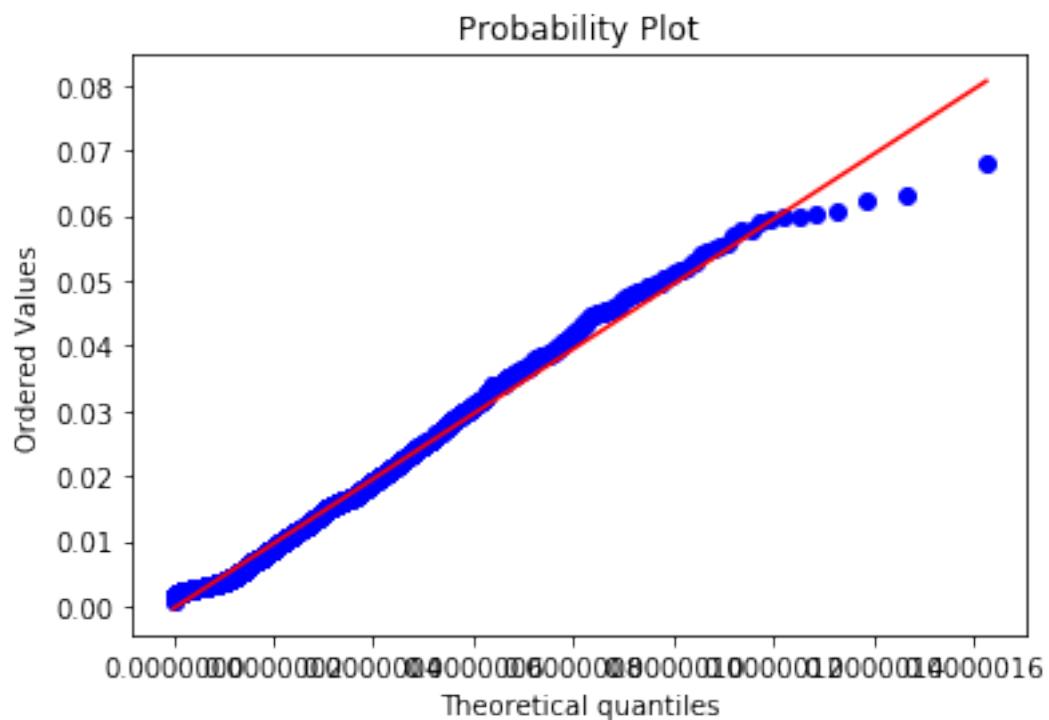
5 steps

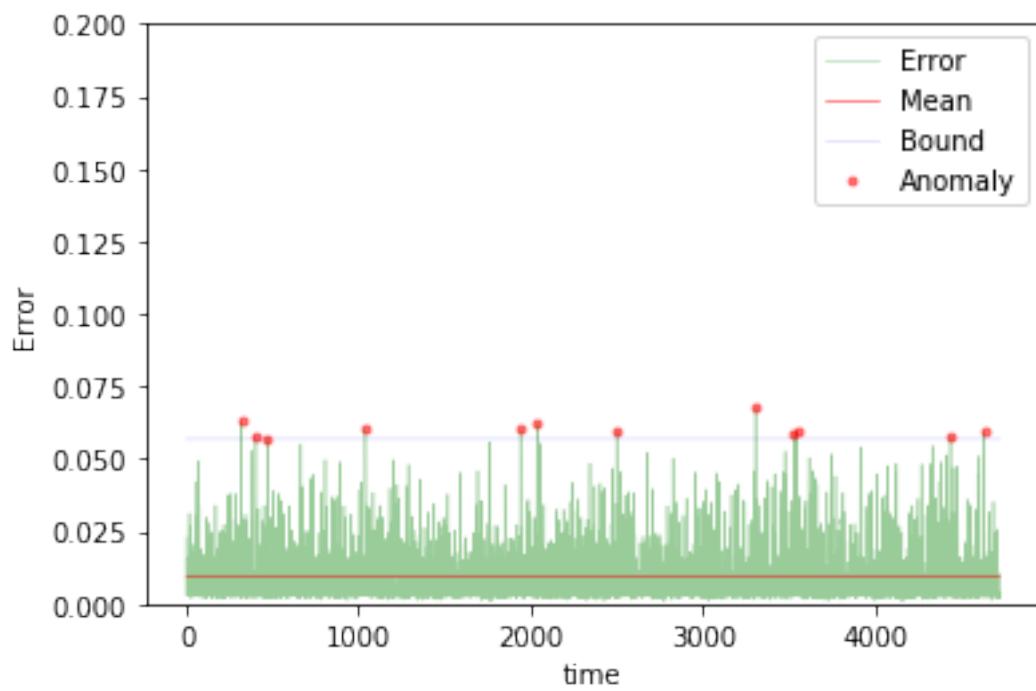
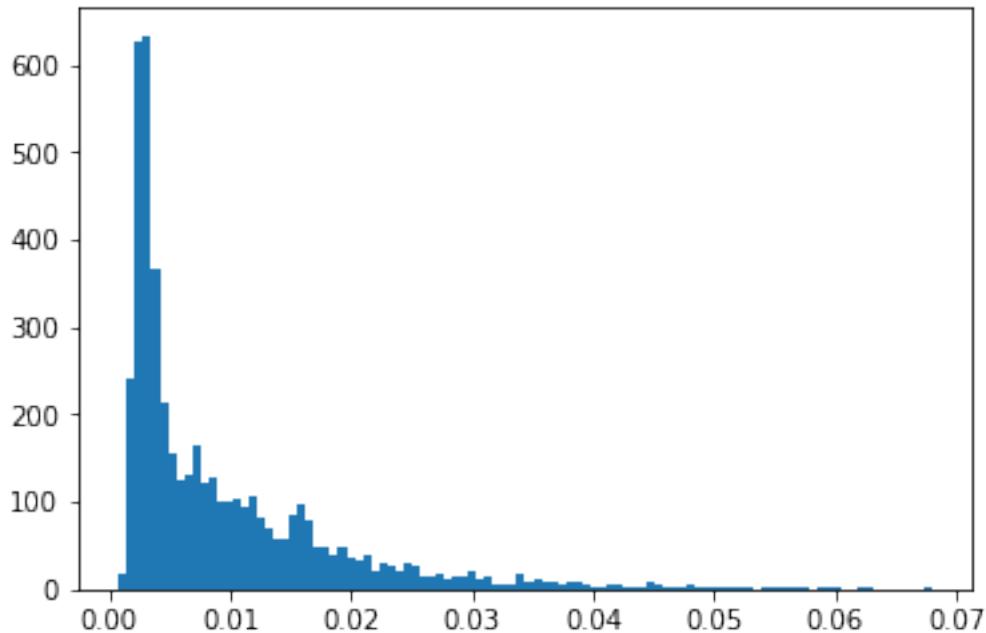
```
In [208]: TIMESTEPS = 5  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS, 0)  
vgen = flat_generator(val_X, TIMESTEPS, 0)  
name = "gru3_5"  
  
In [209]: input_layer = Input(shape=(TIMESTEPS,DIM))  
hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)  
hidden = GRU(10, activation='relu', return_sequences=True)(hidden)  
hidden = GRU(10, activation='relu')(hidden)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [210]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [211]: train(model, tgen, vgen, name=name)  
test(model, ravel=0, name=name, window=TIMESTEPS)
```



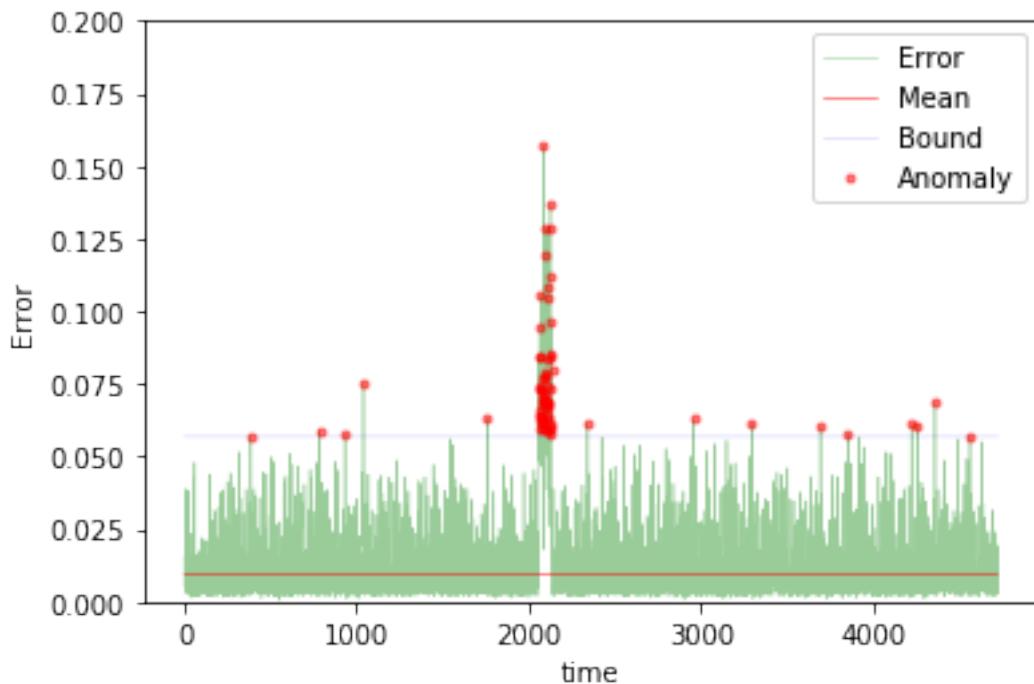
```
Training loss for final epoch is 0.010542703385232017
Validation loss for final epoch is 0.010487224749522284
----- Beginning tests for gru3_5 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

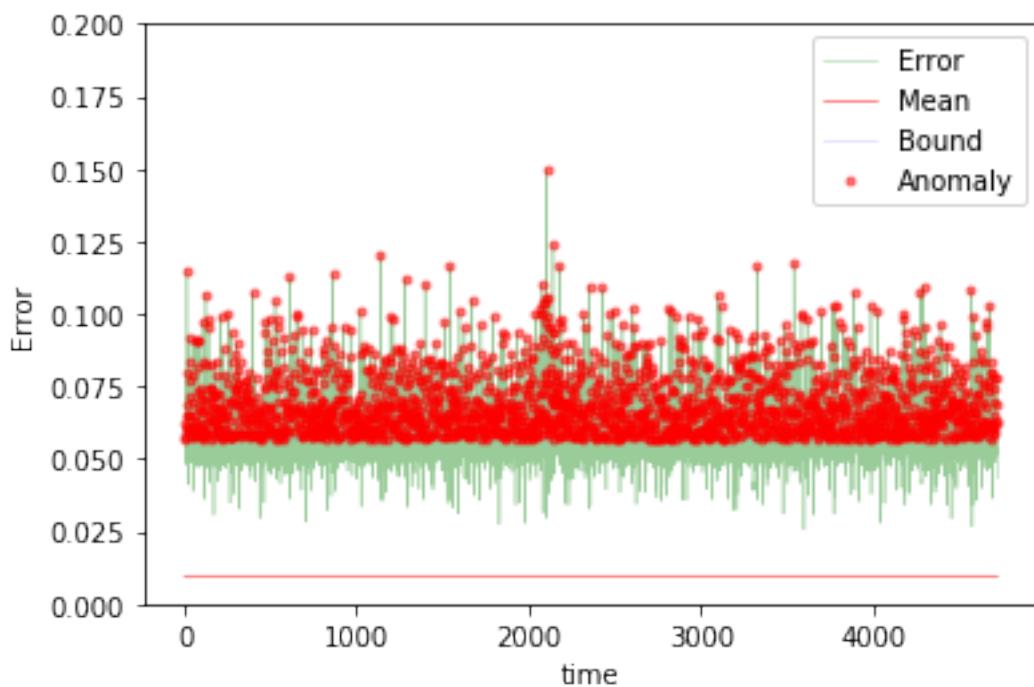




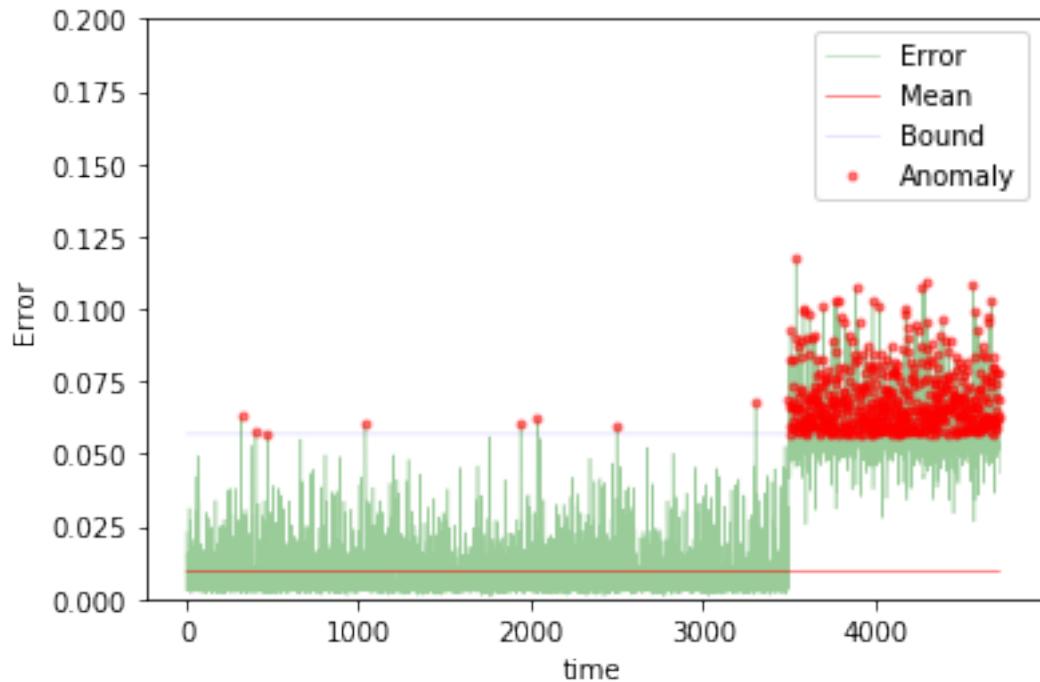
The mean error for gru3_5_normal_ is 0.009706190268213205 for length 4724
Testing on anomaly data.



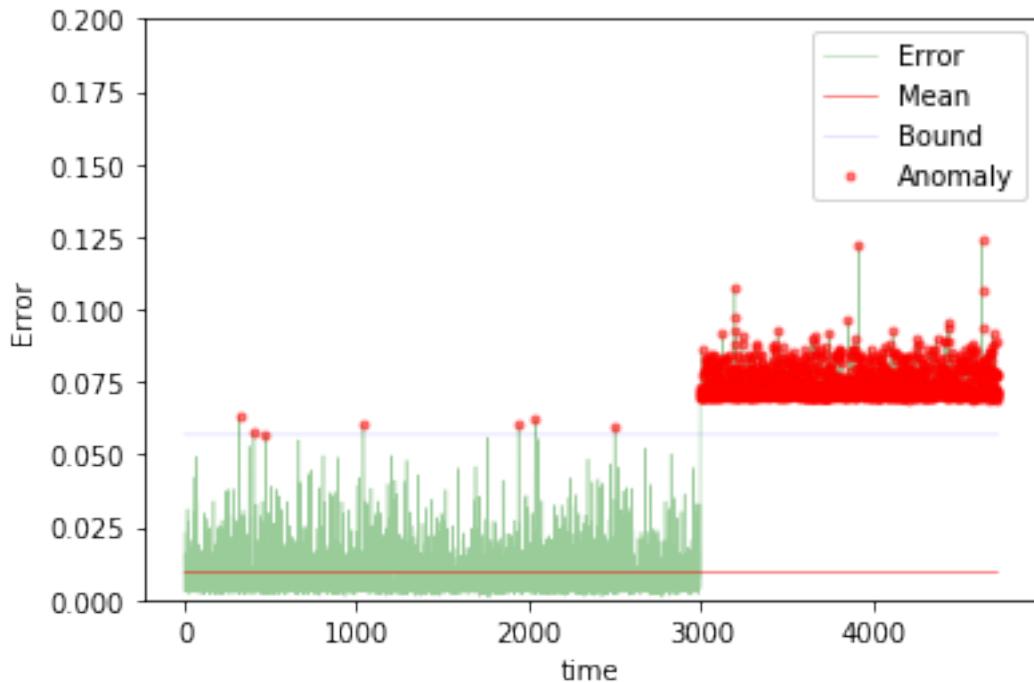
The mean error for gru3_5_anomaly_ is 0.011992832816883079 for length 4724
Testing on different app data.



The mean error for gru3_5_diff_app_ is 0.059177795884210294 for length 4724
Testing on App change synthetic data.



The mean error for gru3_5_app_change_ is 0.022447026826849206 for length 4724
Testing on Net flood synthetic data.



```
The mean error for gru3_5_net_flood_ is 0.033262392807005395 for length 4724
=====
```

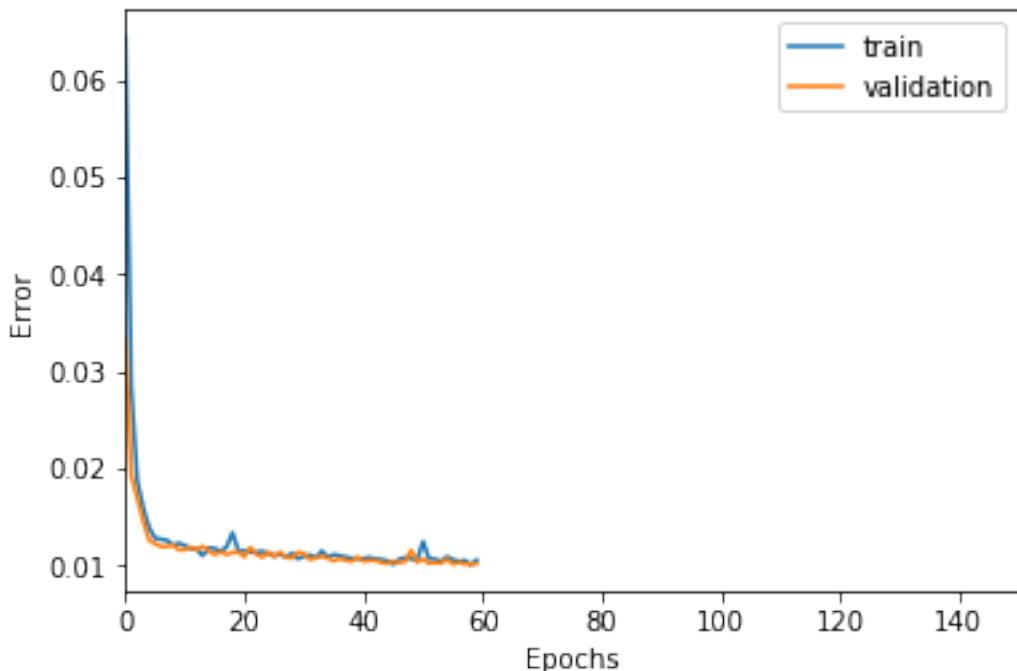
10 steps

```
In [212]: TIMESTEPS = 10
DIM = 29
tgen = flat_generator(X, TIMESTEPS, 0)
vgen = flat_generator(val_X, TIMESTEPS, 0)
name = "gru3_10"

In [213]: 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)

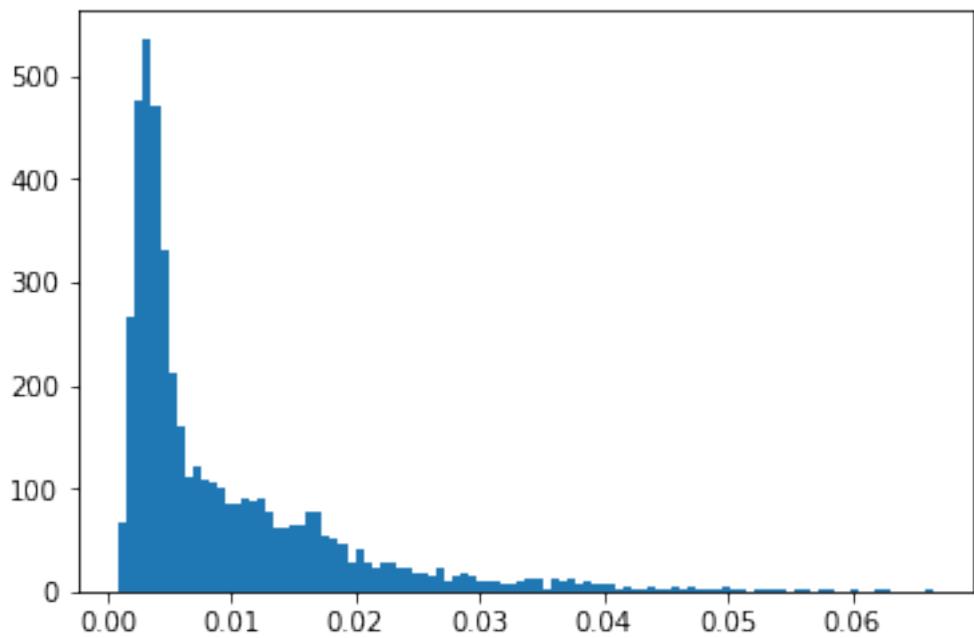
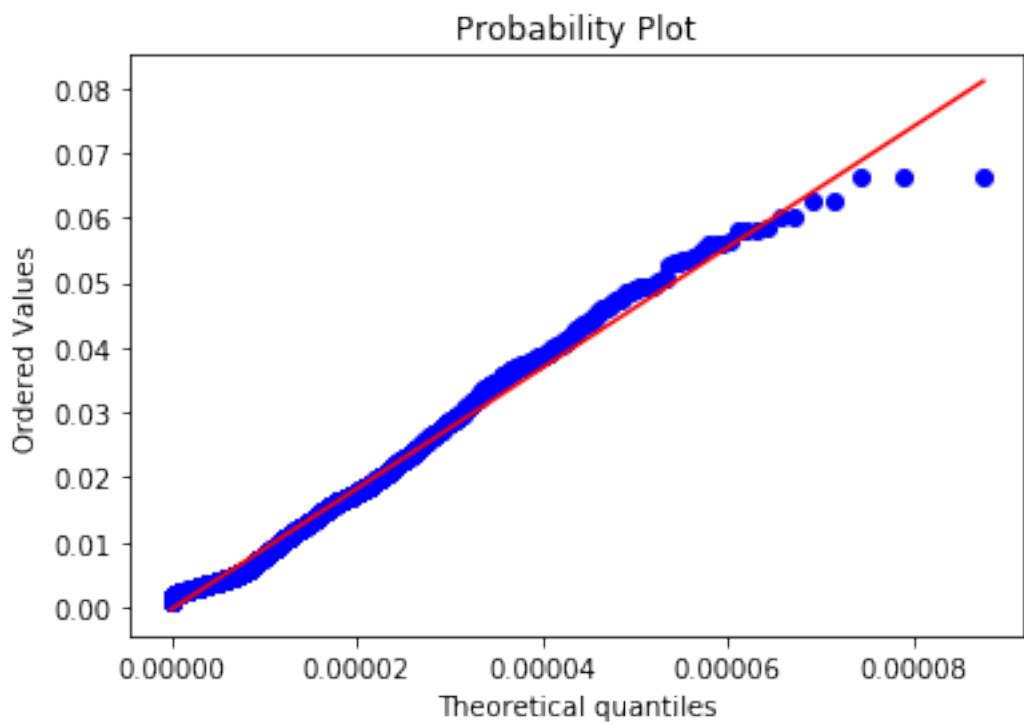
In [214]: model = Model(input_layer, output)
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

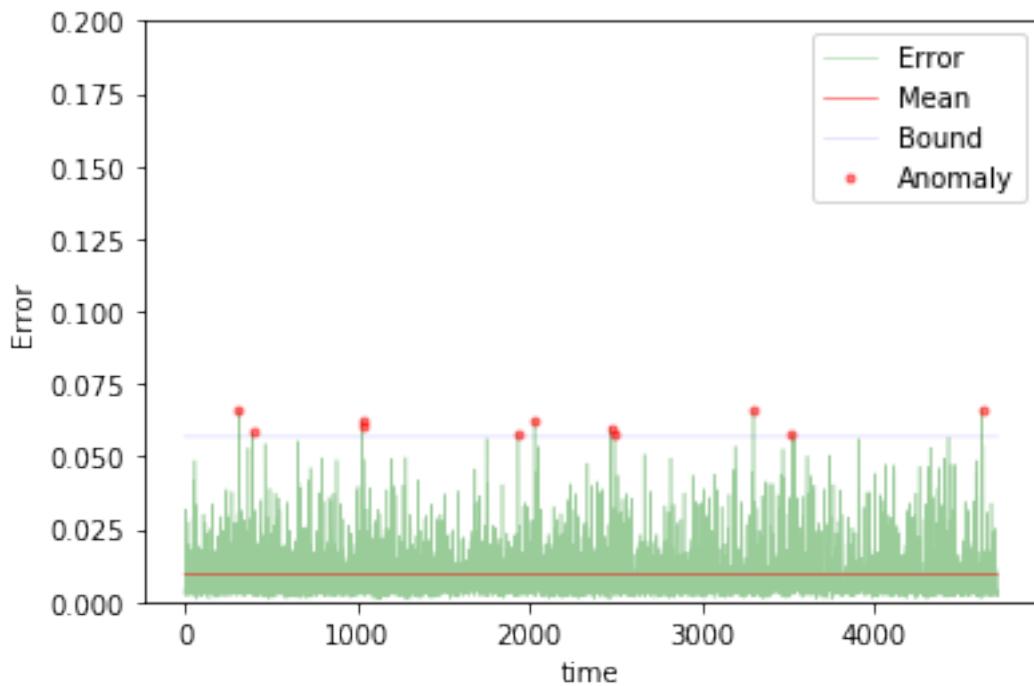
In [215]: train(model, tgen, vgen, name=name)
test(model, ravel=0, name=name, window=TIMESTEPS)
```



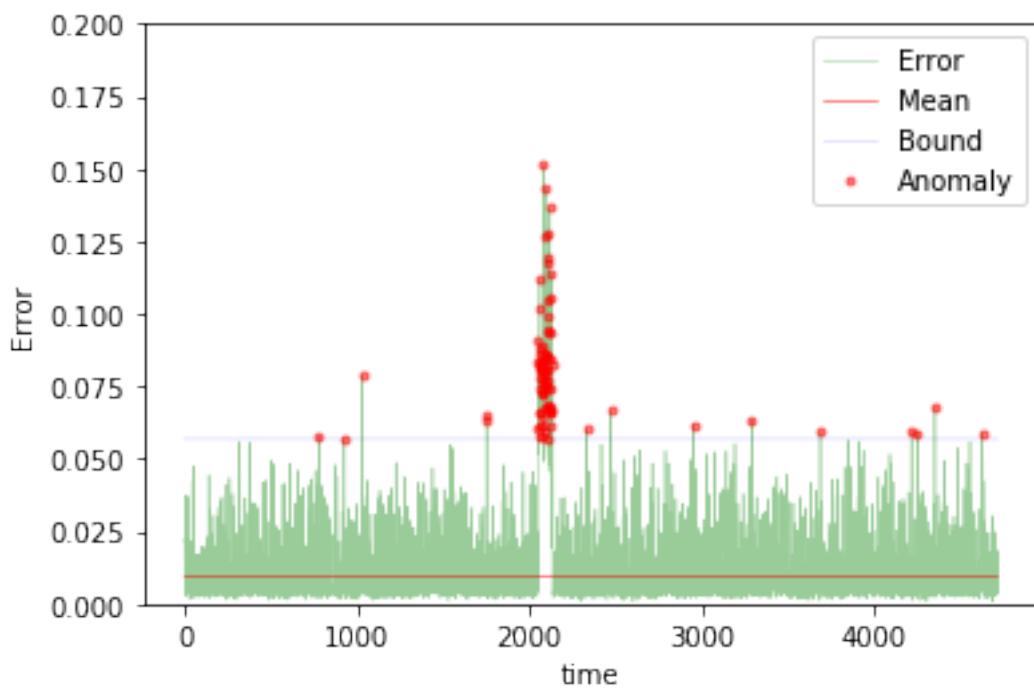
```
Training loss for final epoch is 0.010526178493862972
Validation loss for final epoch is 0.0101421512698289
----- Beginning tests for gru3_10 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

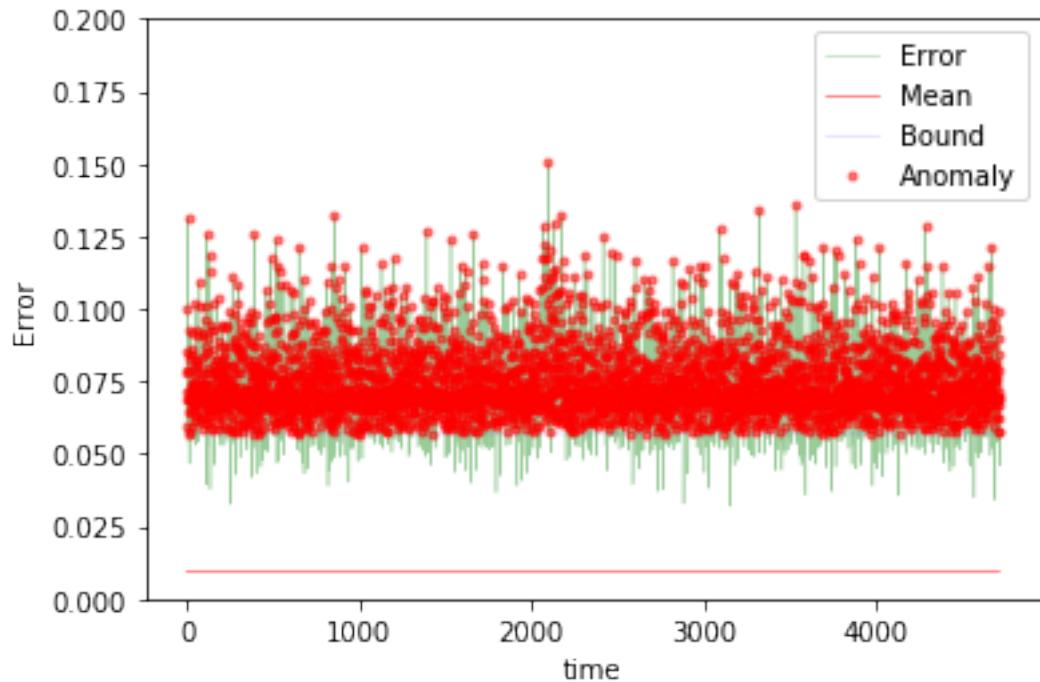




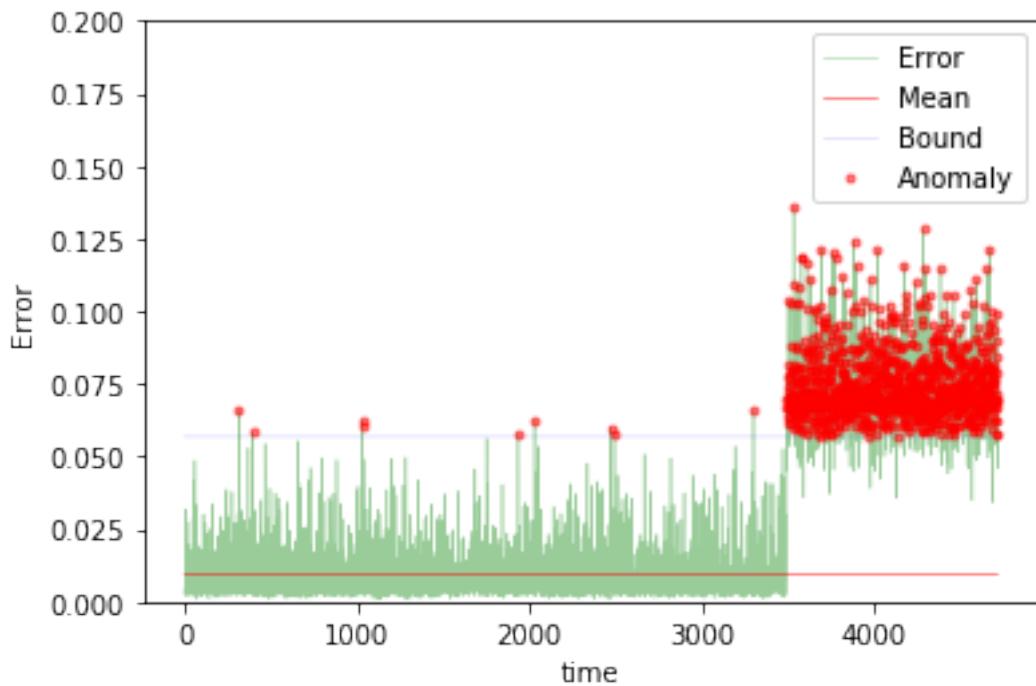
The mean error for gru3_10_normal_ is 0.009641125808913763 for length 4719
Testing on anomaly data.



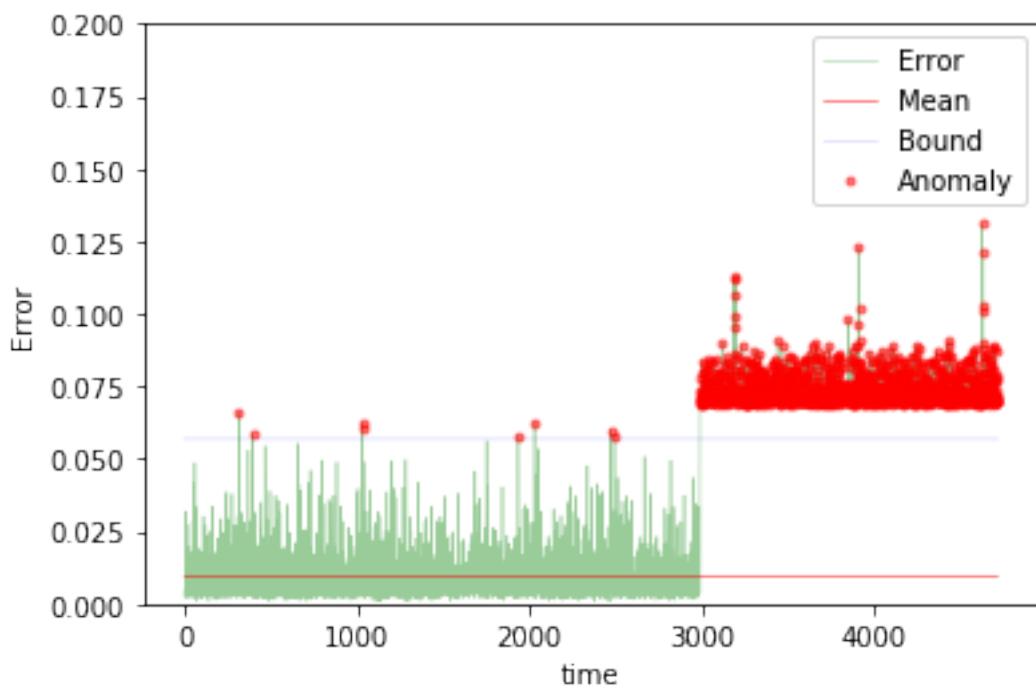
The mean error for gru3_10_anomaly_ is 0.012033171794253837 for length 4719
Testing on different app data.



The mean error for gru3_10_diff_app_ is 0.07307008603118245 for length 4719
Testing on App change synthetic data.



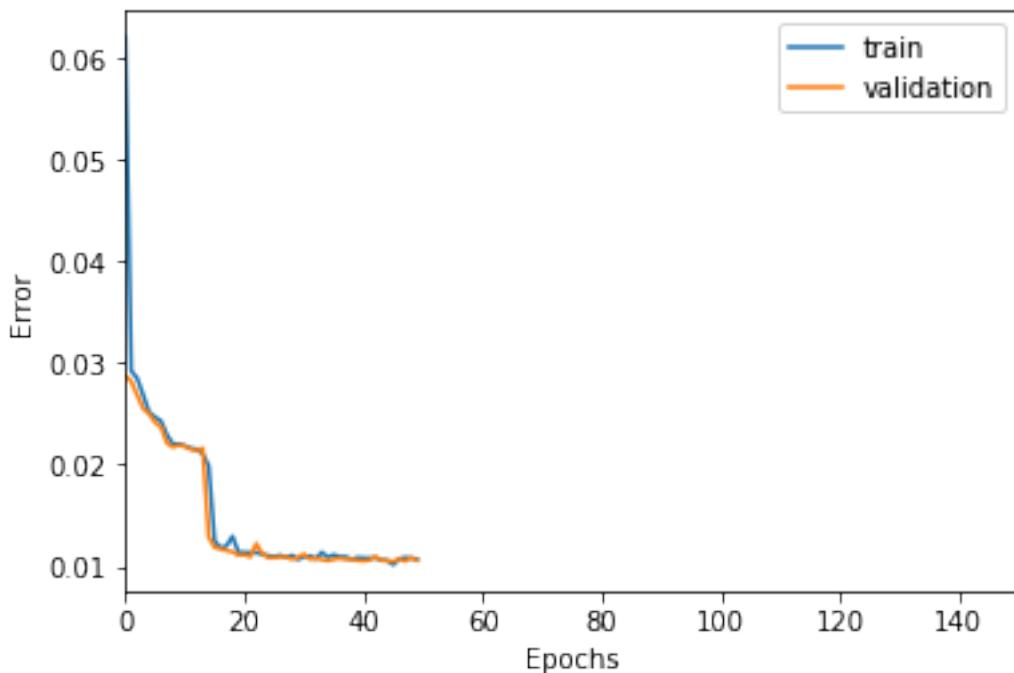
The mean error for gru3_10_app_change_ is 0.025987242215180218 for length 4719
Testing on Net flood synthetic data.



```
The mean error for gru3_10_net_flood_ is 0.03312433632328361 for length 4719  
=====
```

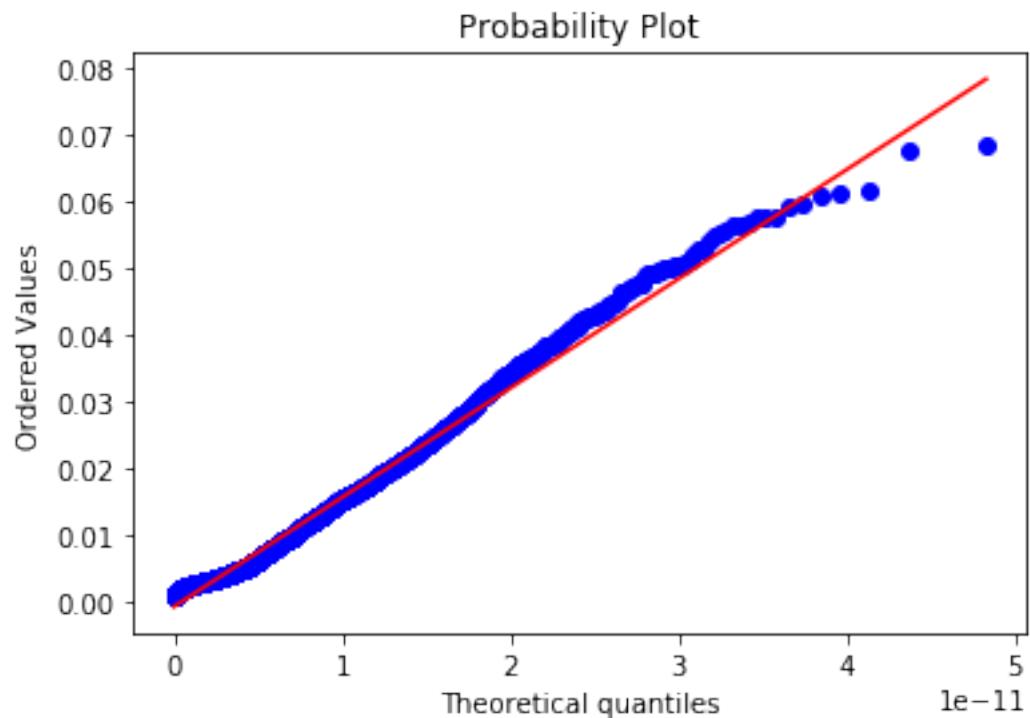
20 steps

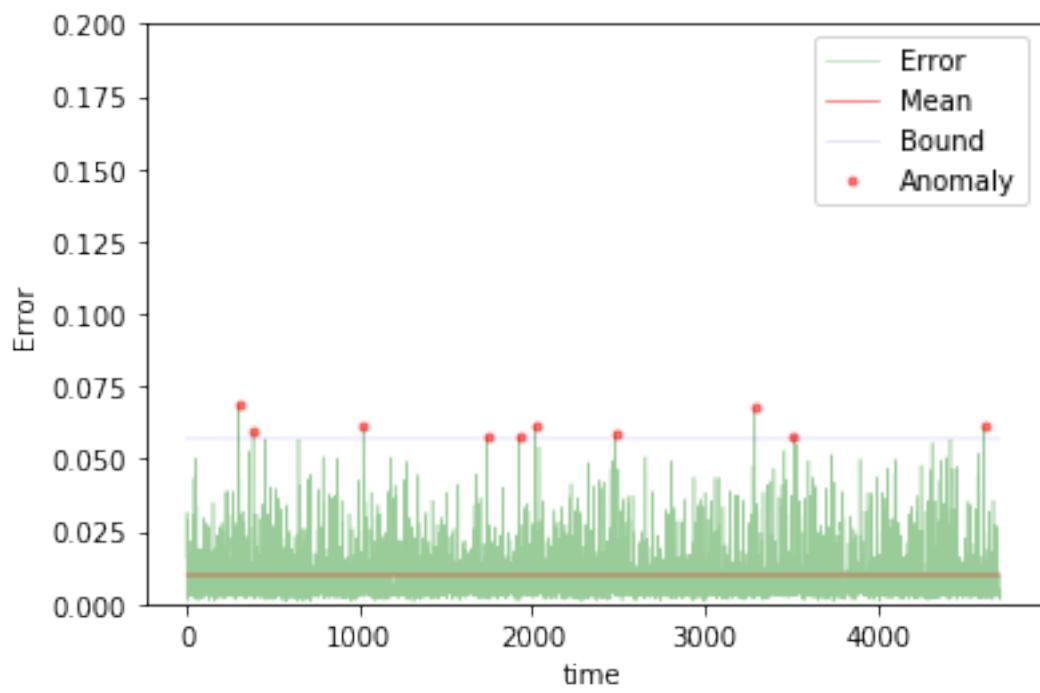
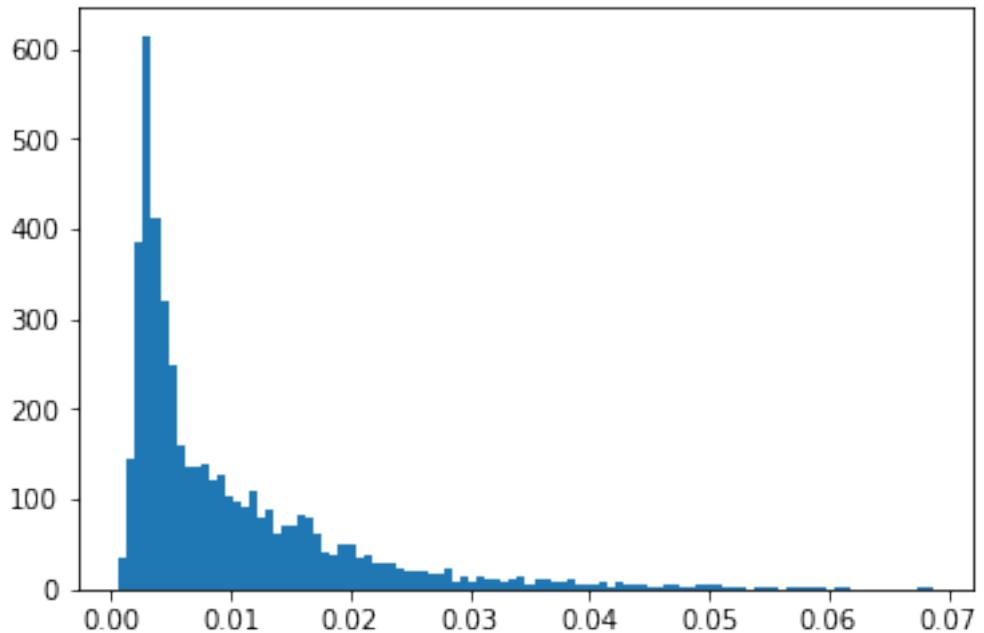
```
In [216]: TIMESTEPS = 20  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS,0)  
vgen = flat_generator(val_X, TIMESTEPS,0)  
name = "gru3_20"  
  
In [217]: input_layer = Input(shape=(TIMESTEPS,DIM))  
hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)  
hidden = GRU(10, activation='relu', return_sequences=True)(hidden)  
hidden = GRU(10, activation='relu')(hidden)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [218]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [219]: train(model, tgen, vgen, name=name)  
test(model, ravel=0, name=name, window=TIMESTEPS)
```



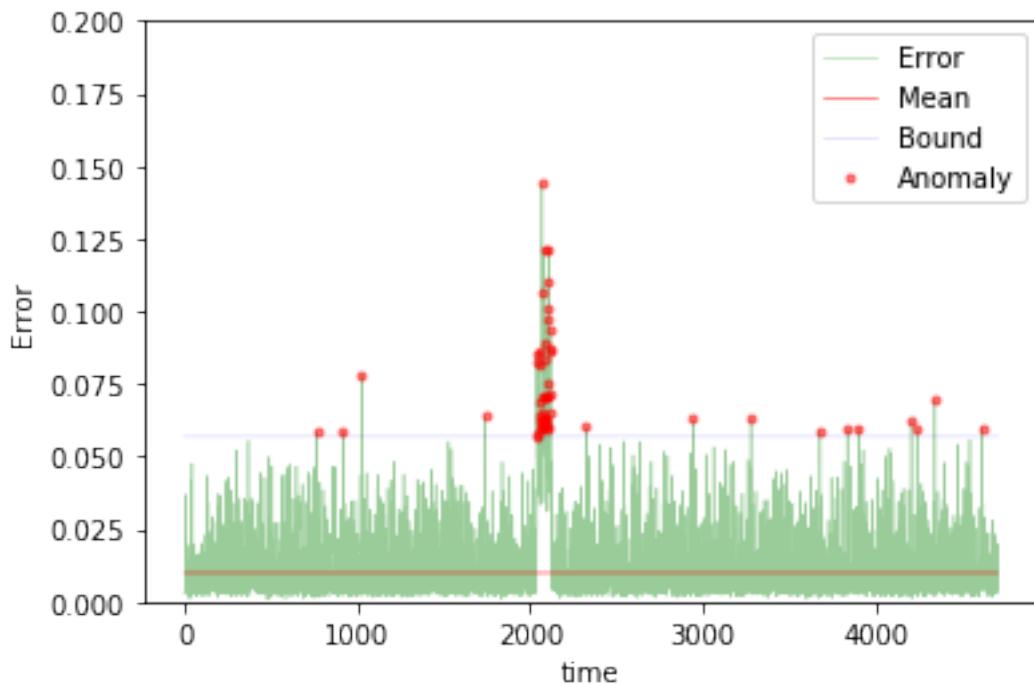
```
Training loss for final epoch is 0.010700519676320255
Validation loss for final epoch is 0.010593700208934024
----- Beginning tests for gru3_20 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

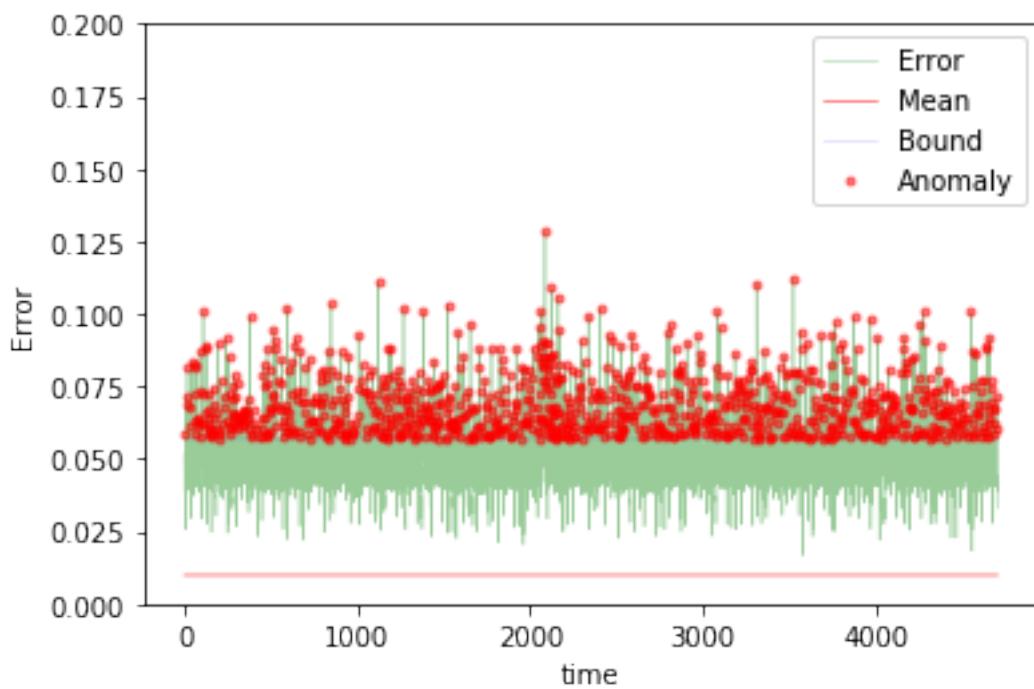




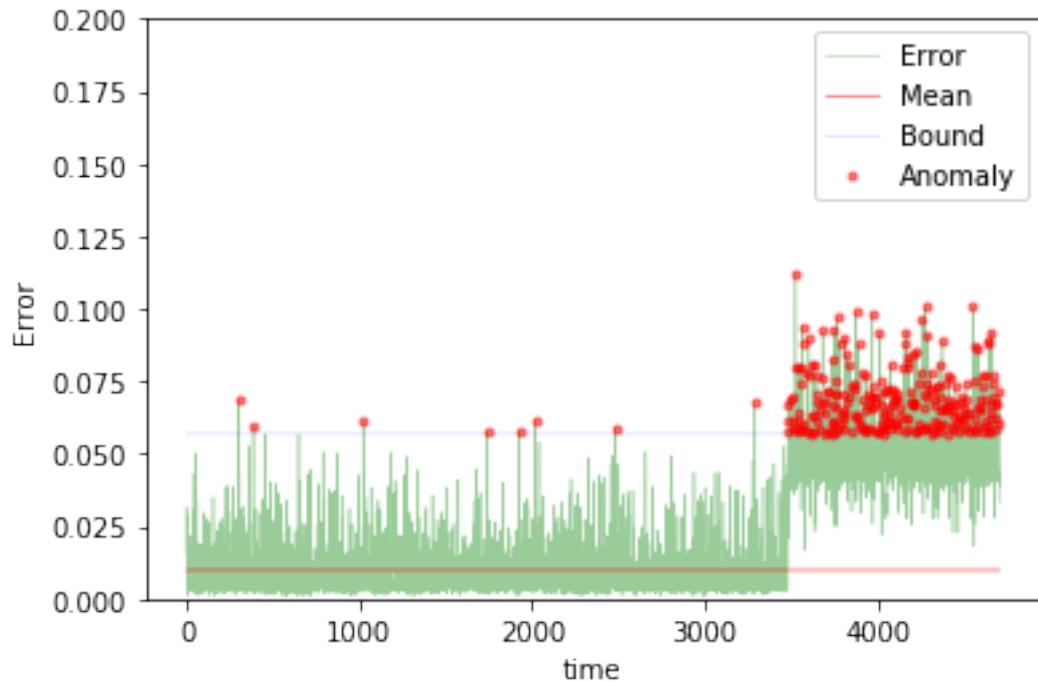
The mean error for gru3_20_normal_ is 0.009955298123860486 for length 4709
Testing on anomaly data.



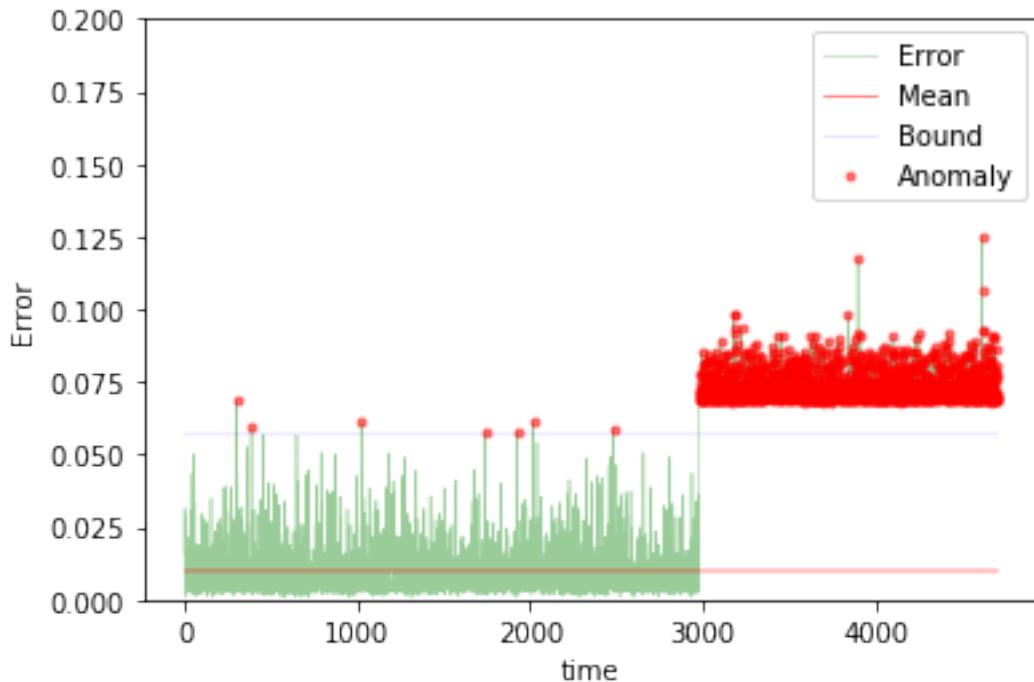
The mean error for gru3_20_anomaly_ is 0.01183788073493248 for length 4709
Testing on different app data.



The mean error for gru3_20_diff_app_ is 0.050084854560561005 for length 4709
Testing on App change synthetic data.



The mean error for gru3_20_app_change_ is 0.0202973115874488 for length 4709
Testing on Net flood synthetic data.



```
The mean error for gru3_20_net_flood_ is 0.03338211509038129 for length 4709
=====
```

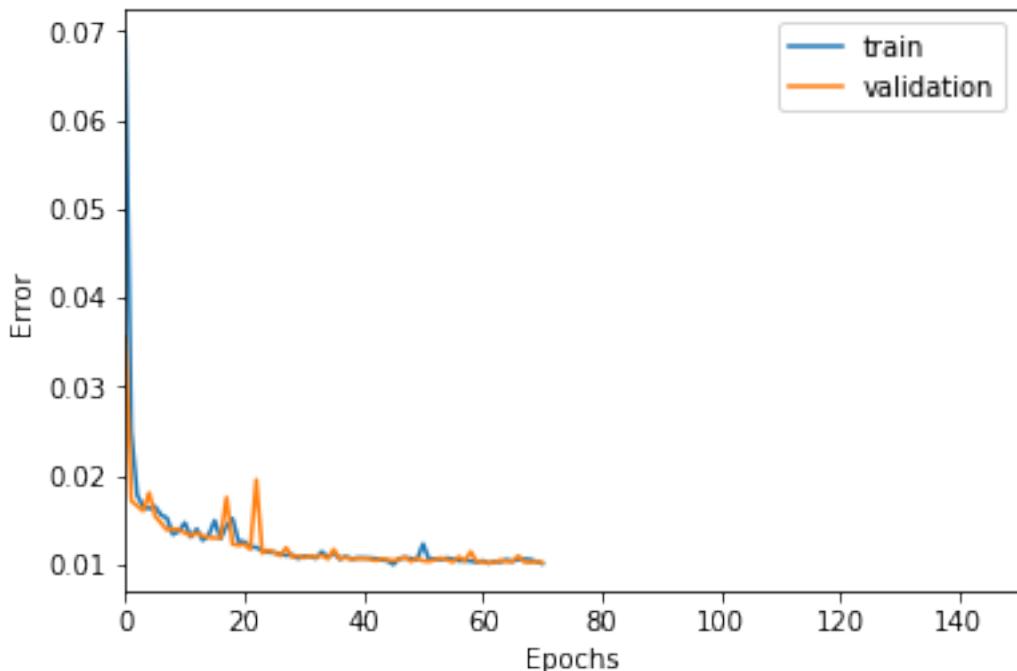
50 steps

```
In [220]: TIMESTEPS = 50
DIM = 29
tgen = flat_generator(X, TIMESTEPS,0)
vgen = flat_generator(val_X, TIMESTEPS,0)
name = "gru3_50"

In [221]: 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)

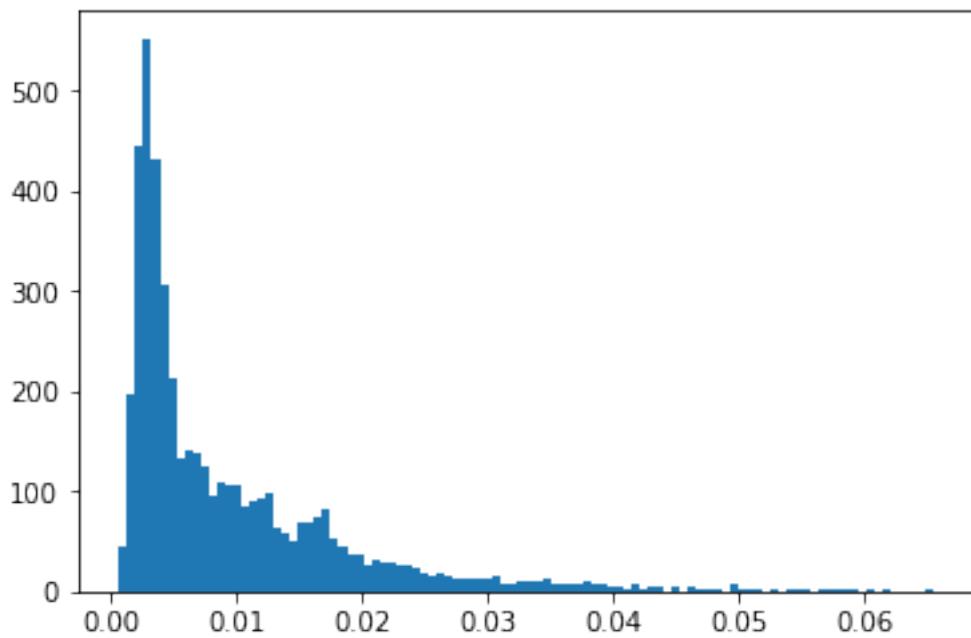
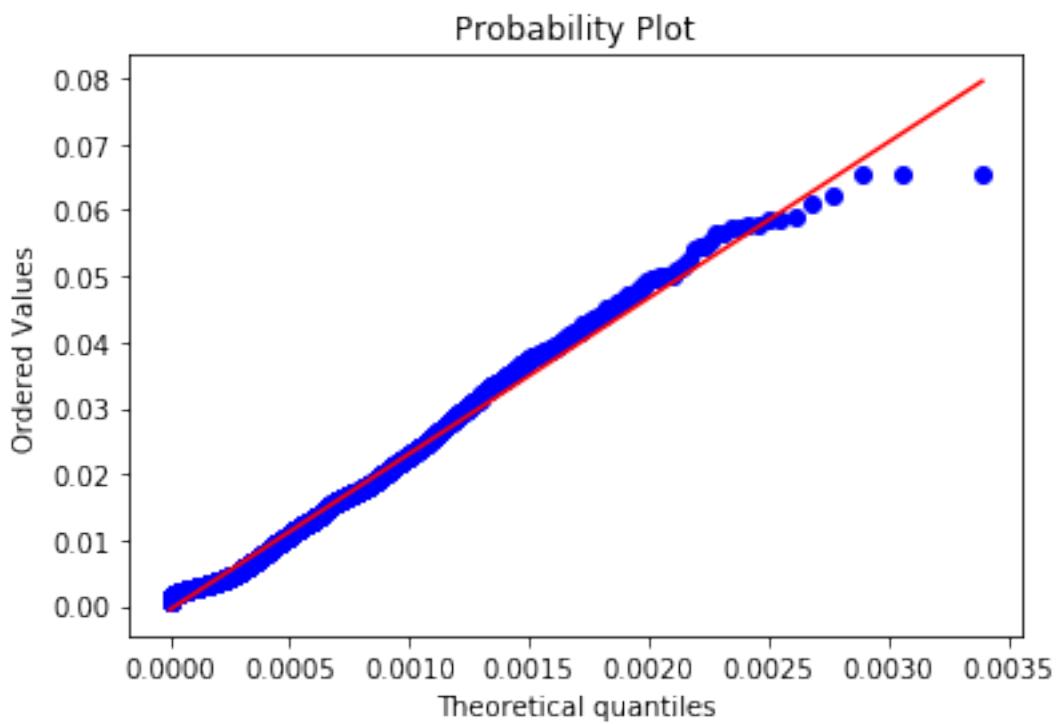
In [222]: model = Model(input_layer, output)
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

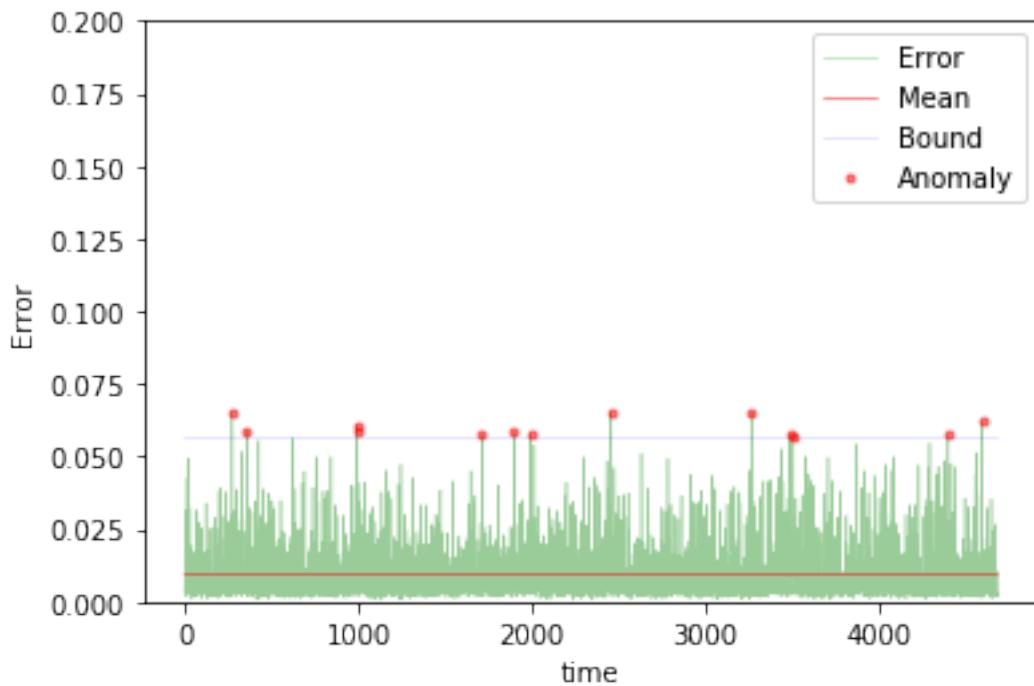
In [223]: train(model, tgen, vgen, name=name)
test(model, ravel=0, name=name, window=TIMESTEPS)
```



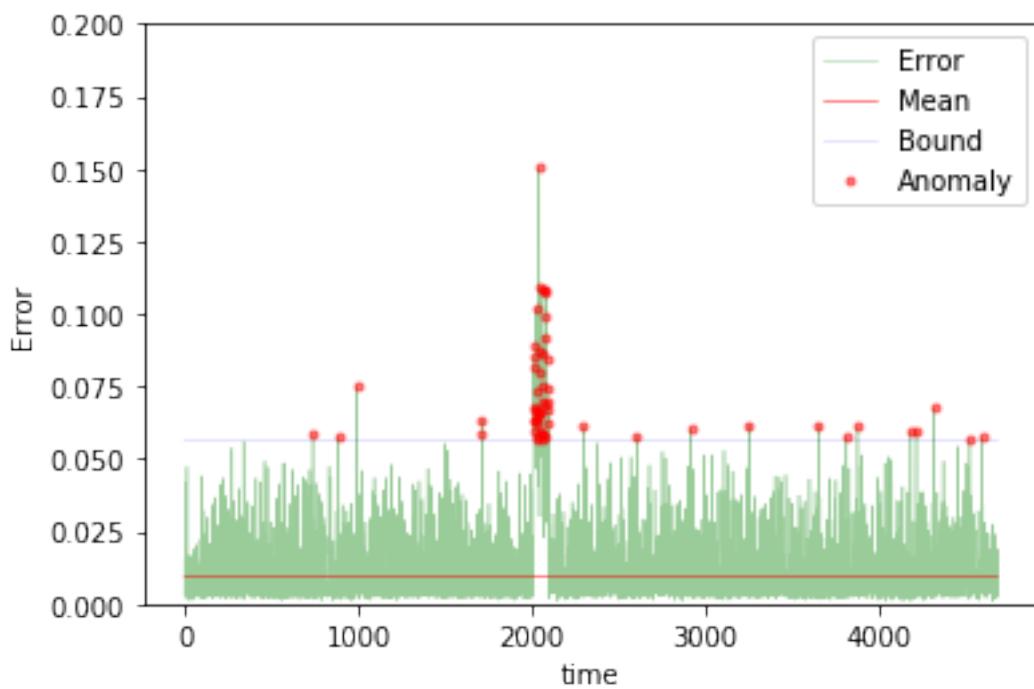
```
Training loss for final epoch is 0.010103780875797384
Validation loss for final epoch is 0.010183984480798245
----- Beginning tests for gru3_50 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

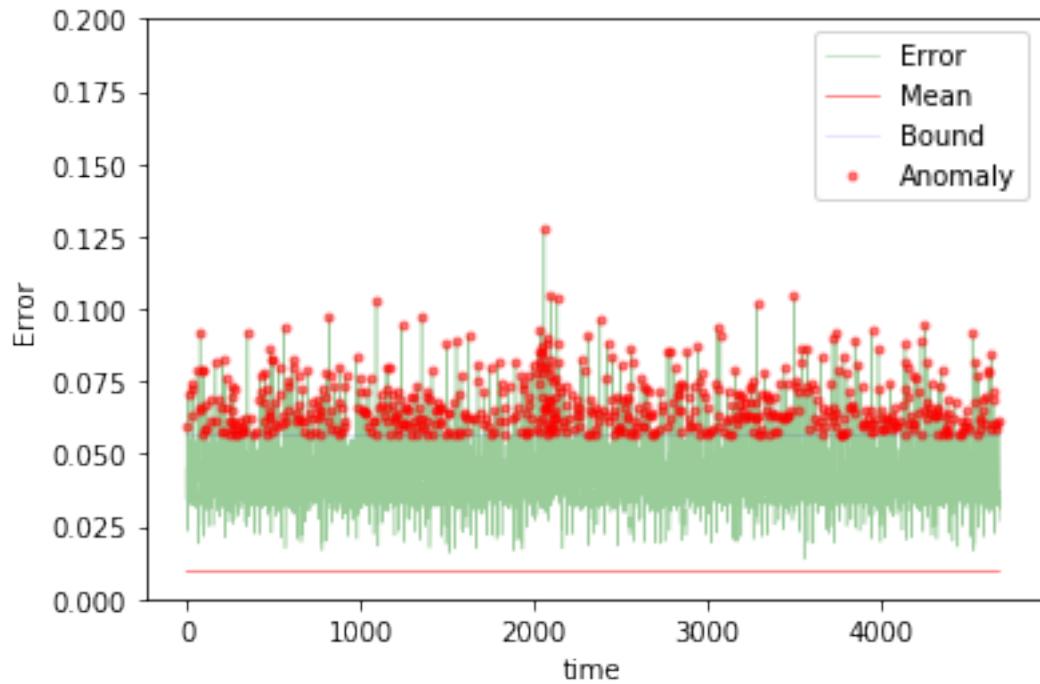




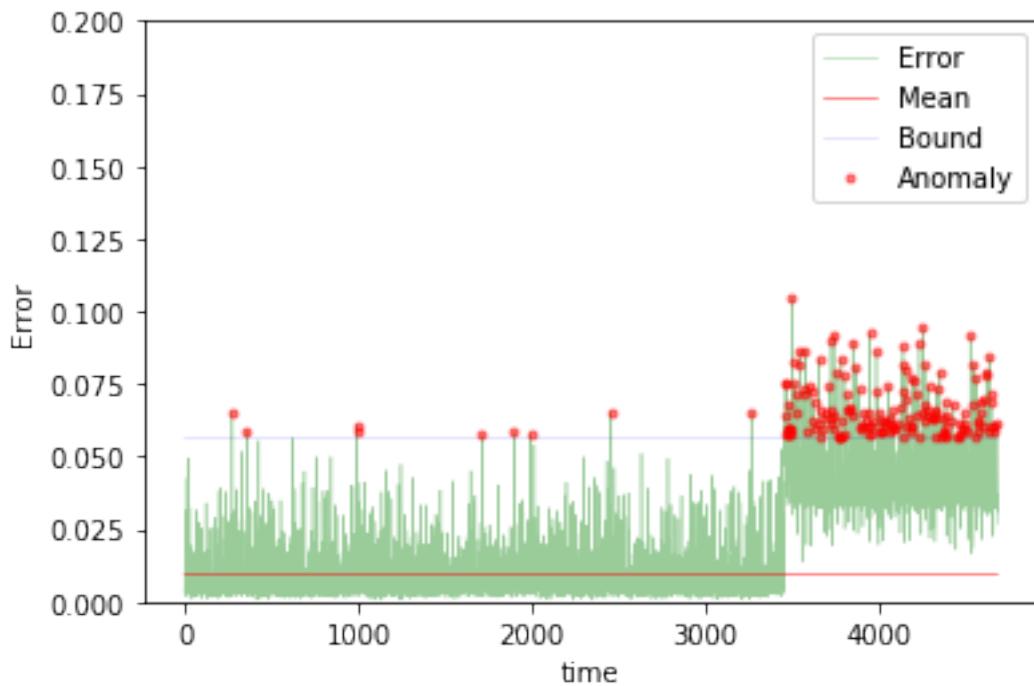
The mean error for gru3_50_normal_ is 0.009611114397125437 for length 4679
Testing on anomaly data.



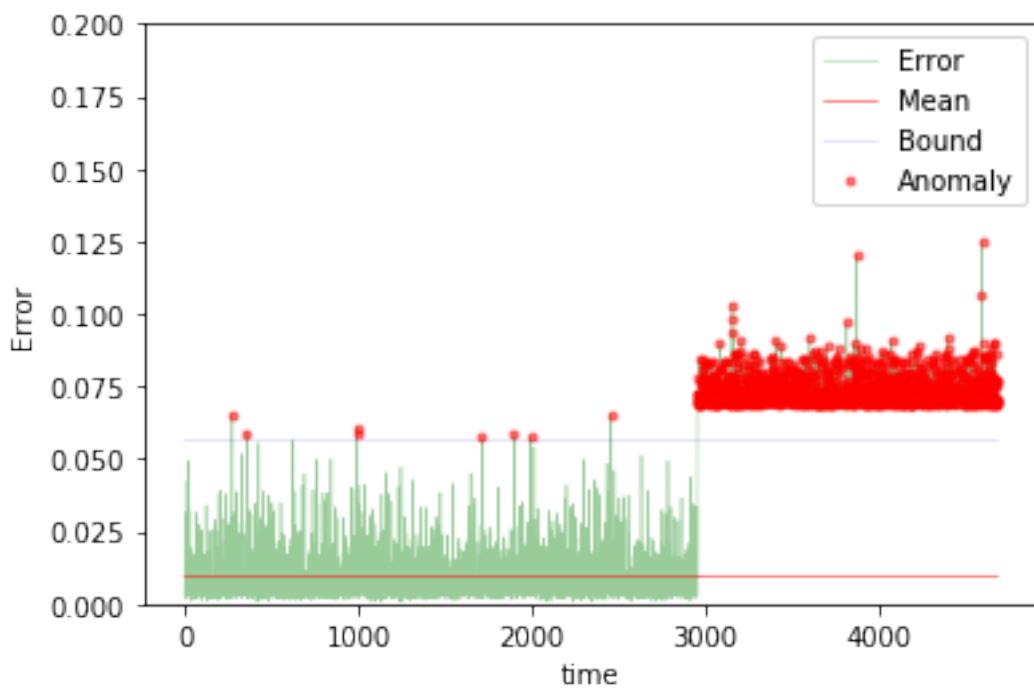
The mean error for gru3_50_anomaly_ is 0.011688573632440549 for length 4679
Testing on different app data.



The mean error for gru3_50_diff_app_ is 0.04259213061107416 for length 4679
Testing on App change synthetic data.



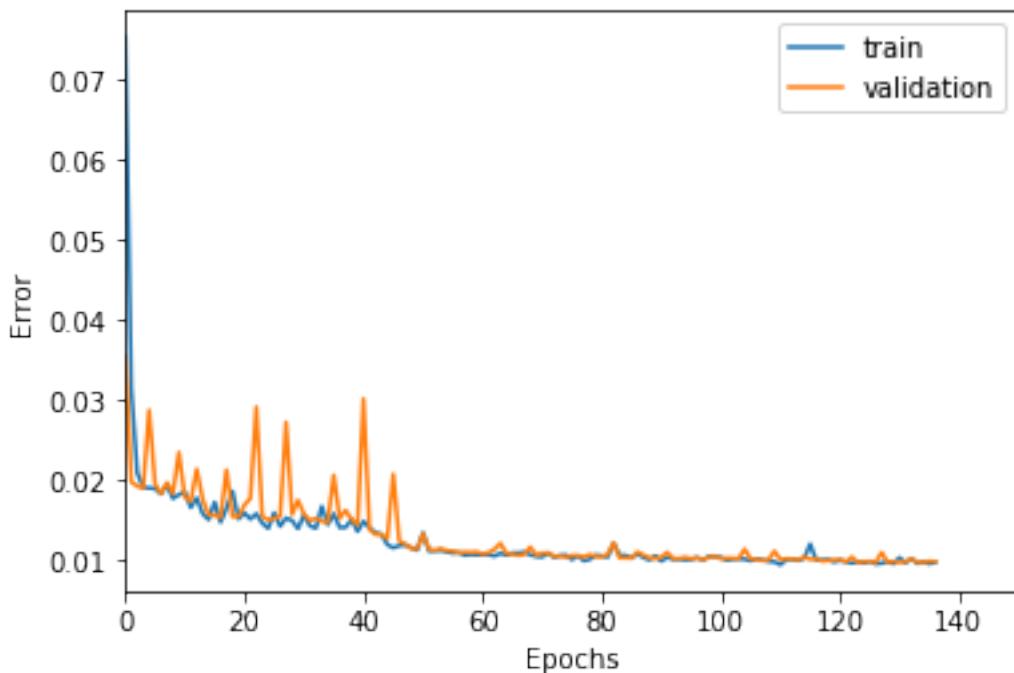
The mean error for gru3_50_app_change_ is 0.018244741073703677 for length 4679
Testing on Net flood synthetic data.



```
The mean error for gru3_50_net_flood_ is 0.03316723597591645 for length 4679  
=====
```

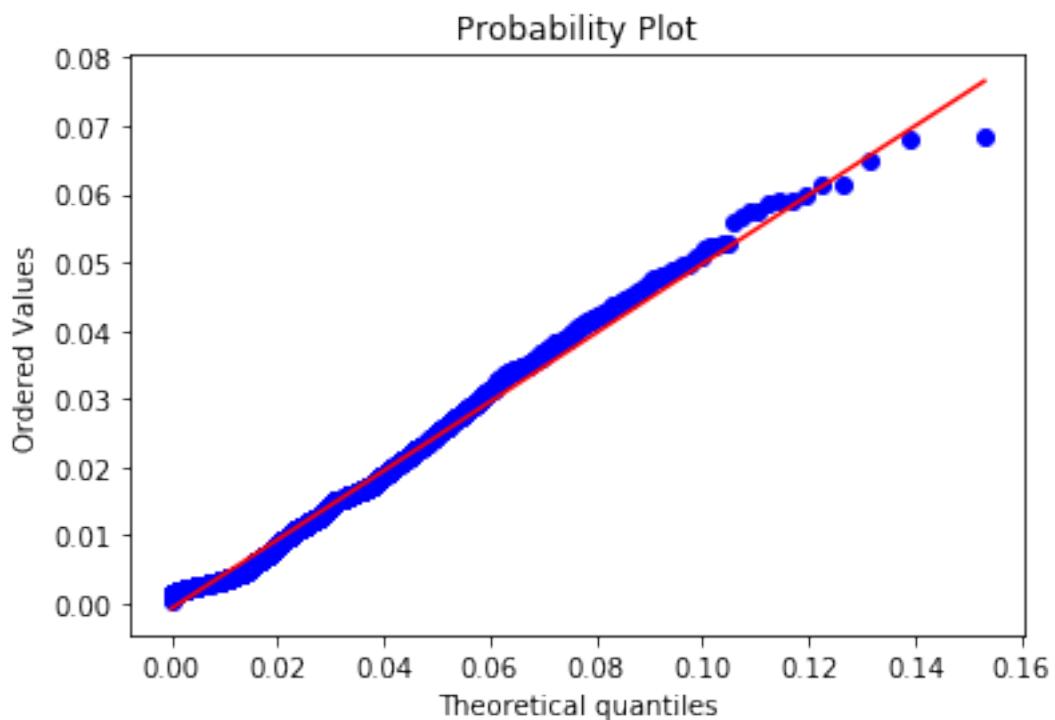
100 steps

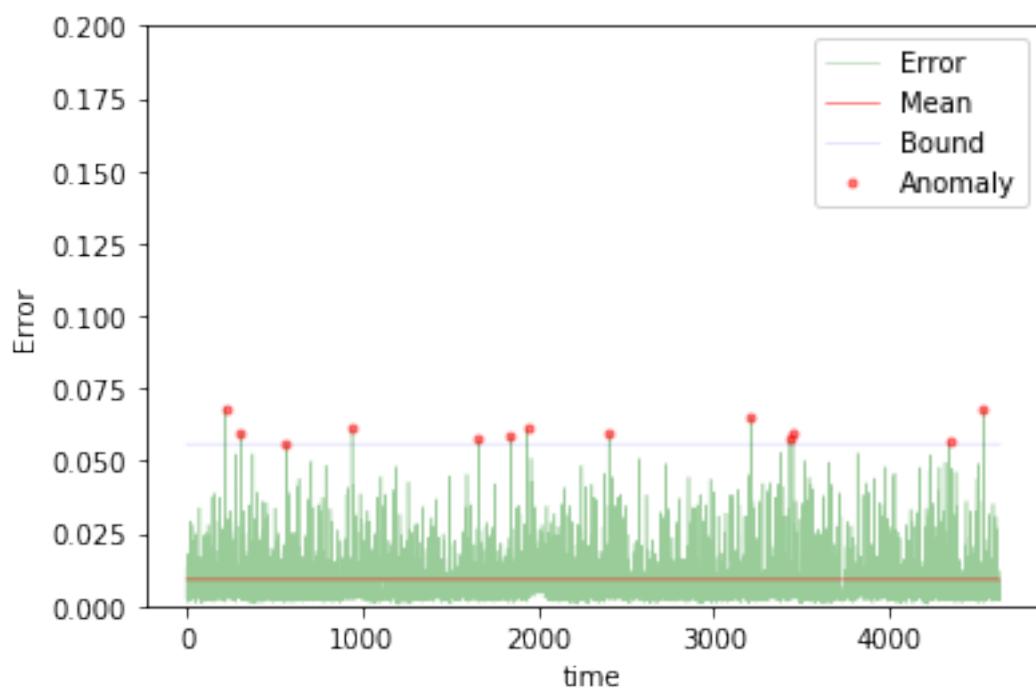
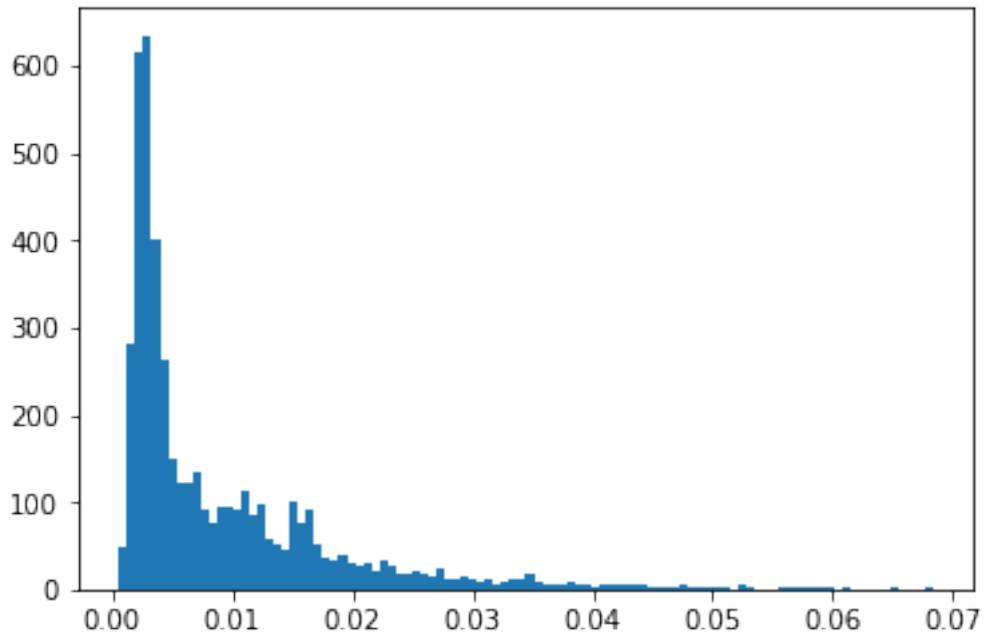
```
In [224]: TIMESTEPS = 100  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS,0)  
vgen = flat_generator(val_X, TIMESTEPS,0)  
name = "gru3_100"  
  
In [225]: input_layer = Input(shape=(TIMESTEPS,DIM))  
hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)  
hidden = GRU(10, activation='relu', return_sequences=True)(hidden)  
hidden = GRU(10, activation='relu')(hidden)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [226]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [227]: train(model, tgen, vgen, name=name)  
test(model, ravel=0, name=name, window=TIMESTEPS)
```



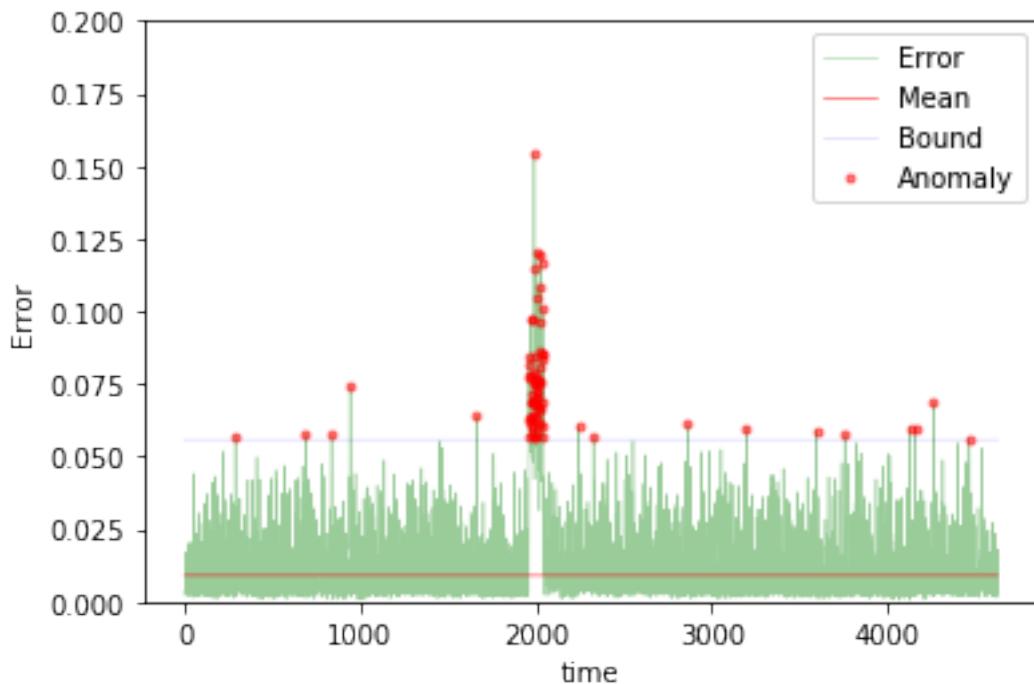
```
Training loss for final epoch is 0.009809592963079922
Validation loss for final epoch is 0.009876018312294036
----- Beginning tests for gru3_100 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
improvement from the last ten iterations.
warnings.warn(msg, RuntimeWarning)
```

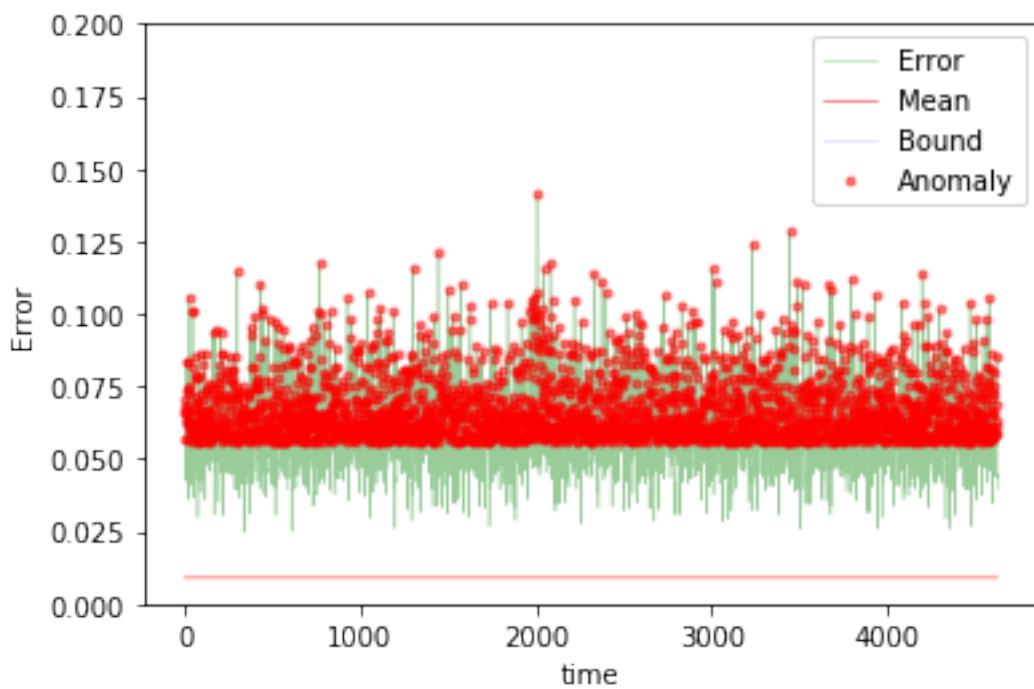




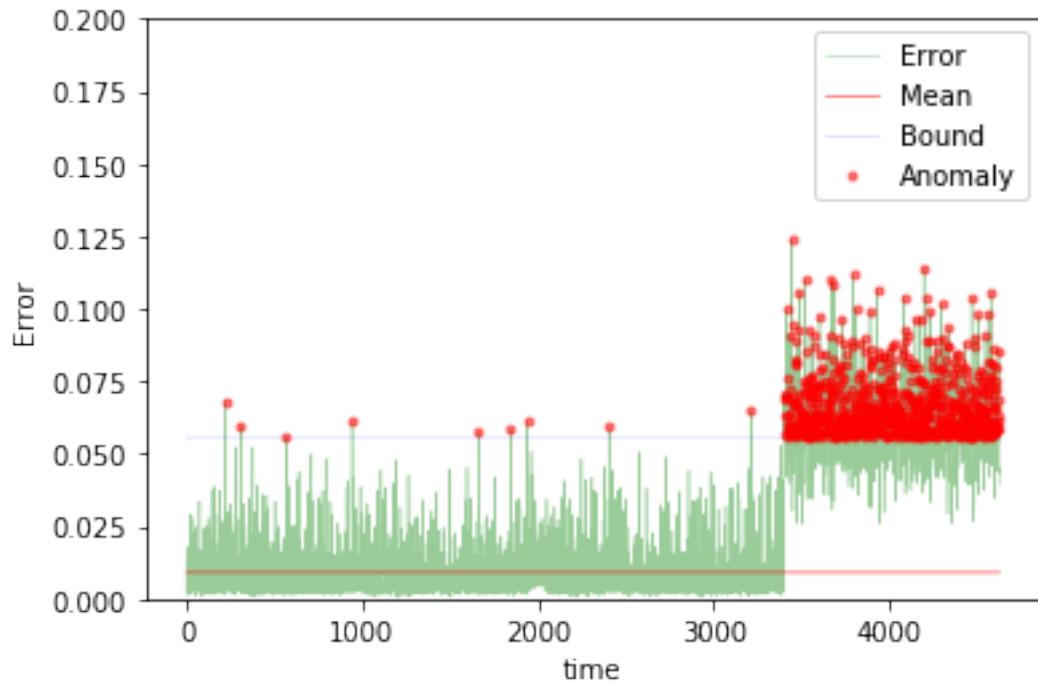
The mean error for gru3_100_normal_ is 0.009096355236680788 for length 4629
Testing on anomaly data.



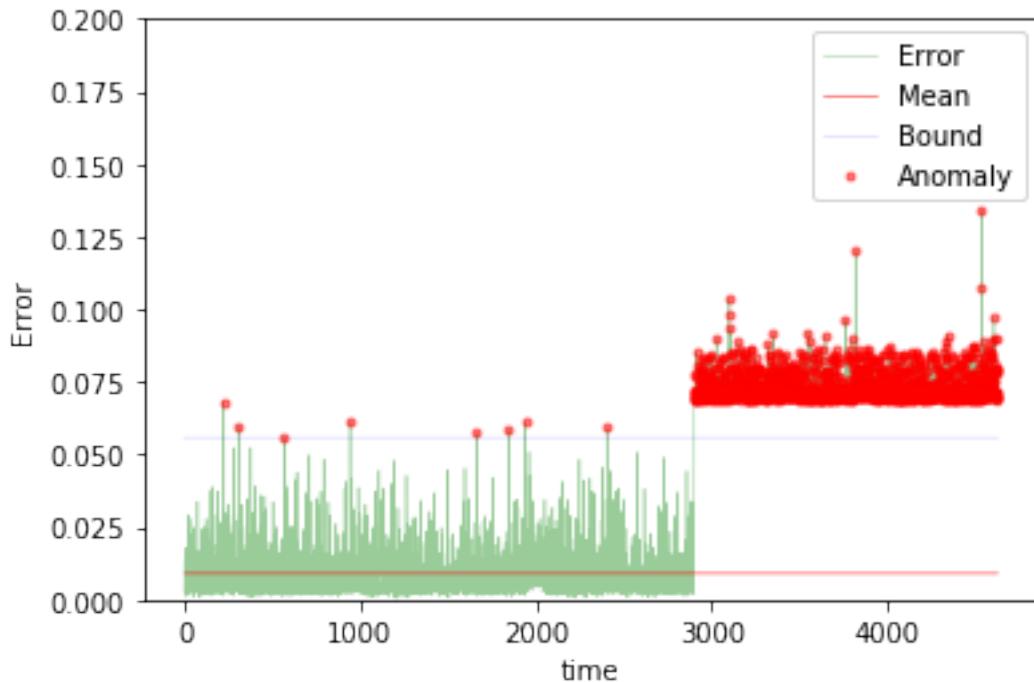
The mean error for gru3_100_anomaly_ is 0.011415723722420086 for length 4629
Testing on different app data.



The mean error for gru3_100_diff_app_ is 0.06050890346782051 for length 4629
Testing on App change synthetic data.



The mean error for gru3_100_app_change_ is 0.022596249972196468 for length 4629
Testing on Net flood synthetic data.



```
The mean error for gru3_100_net_flood_ is 0.033075999724027354 for length 4629
=====
```

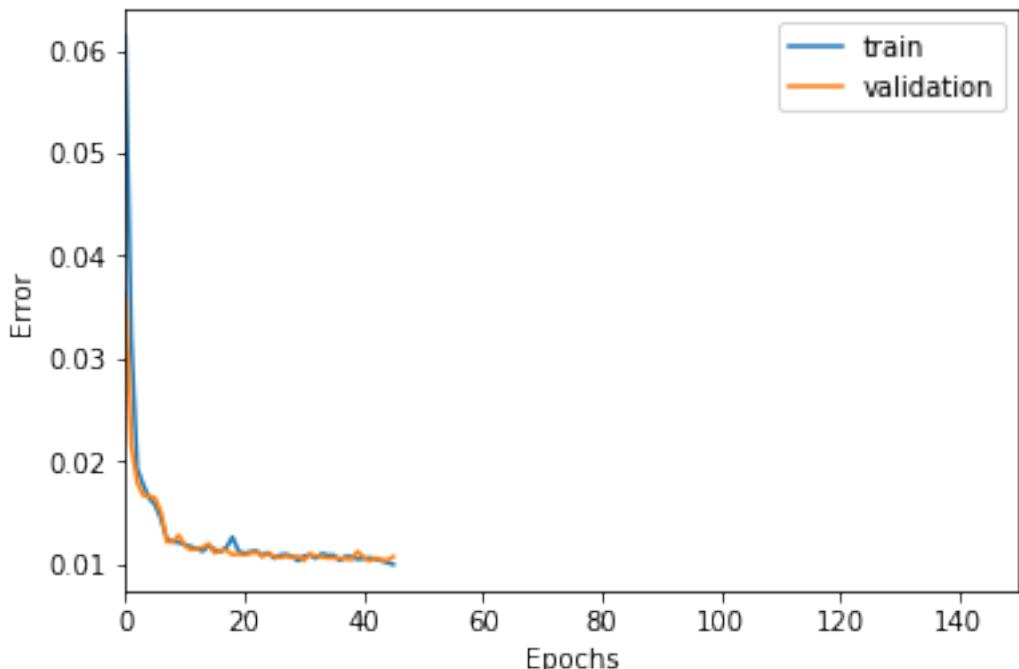
200 steps

```
In [228]: TIMESTEPS = 200
DIM = 29
tgen = flat_generator(X, TIMESTEPS,0)
vgen = flat_generator(val_X, TIMESTEPS,0)
name = "gru3_200"

In [229]: 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)

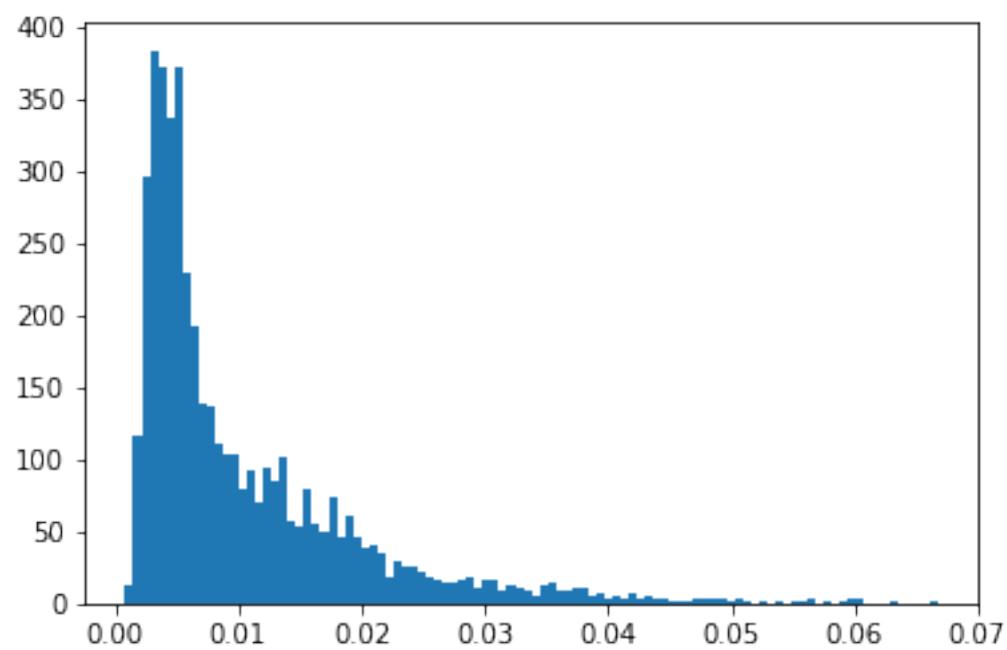
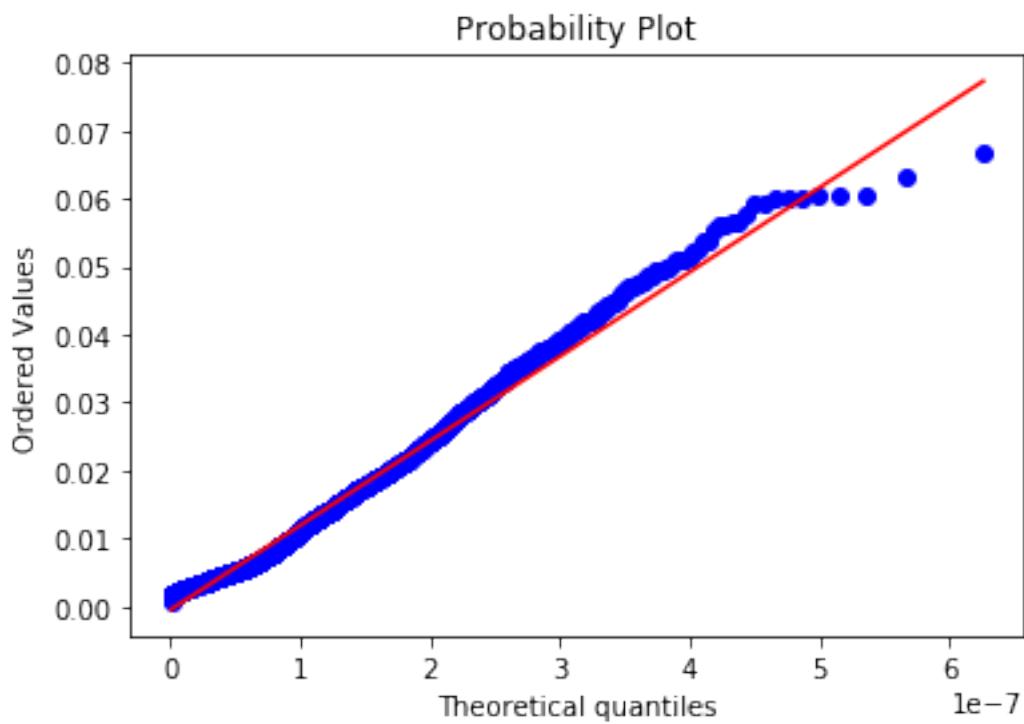
In [230]: model = Model(input_layer, output)
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

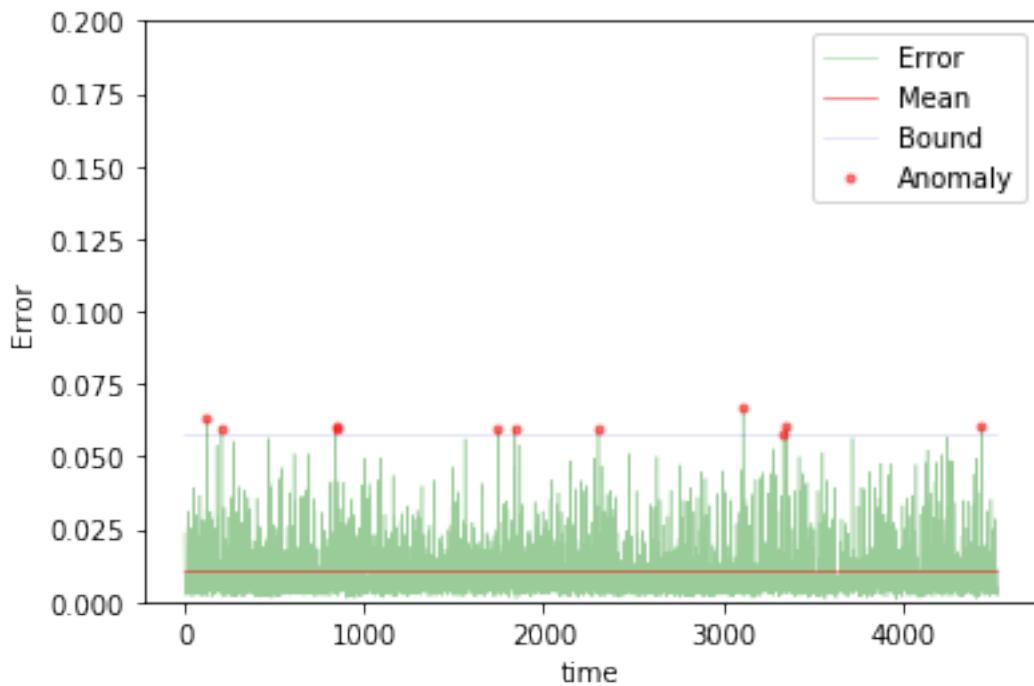
In [231]: train(model, tgen, vgen, name=name)
test(model, ravel=0, name=name, window=TIMESTEPS)
```



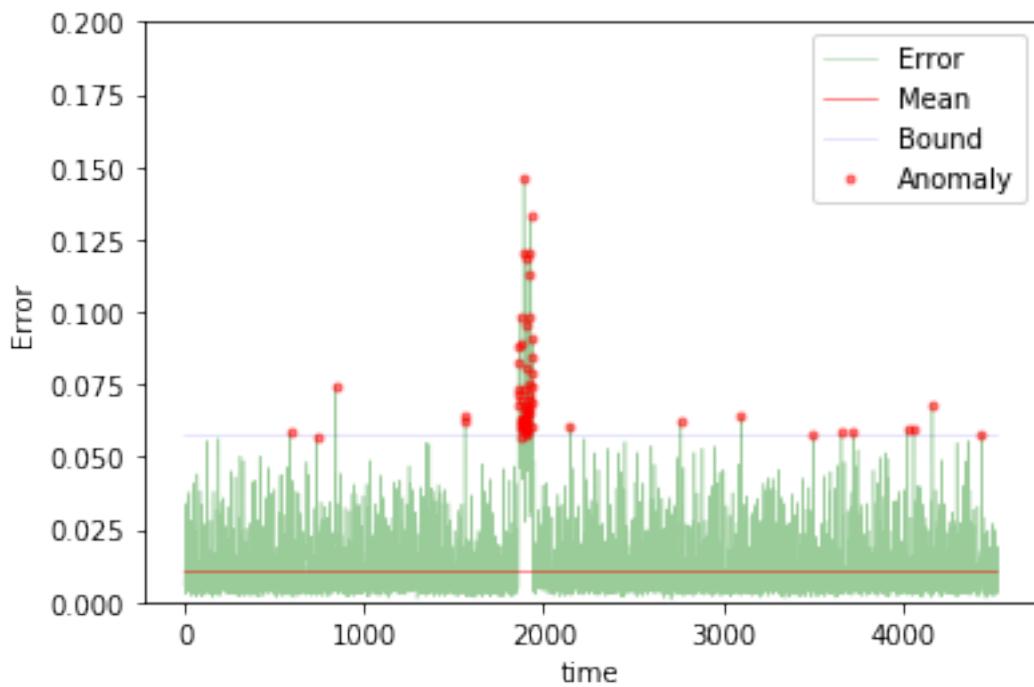
```
Training loss for final epoch is 0.009992767215473578
Validation loss for final epoch is 0.010746165899559856
----- Beginning tests for gru3_200 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

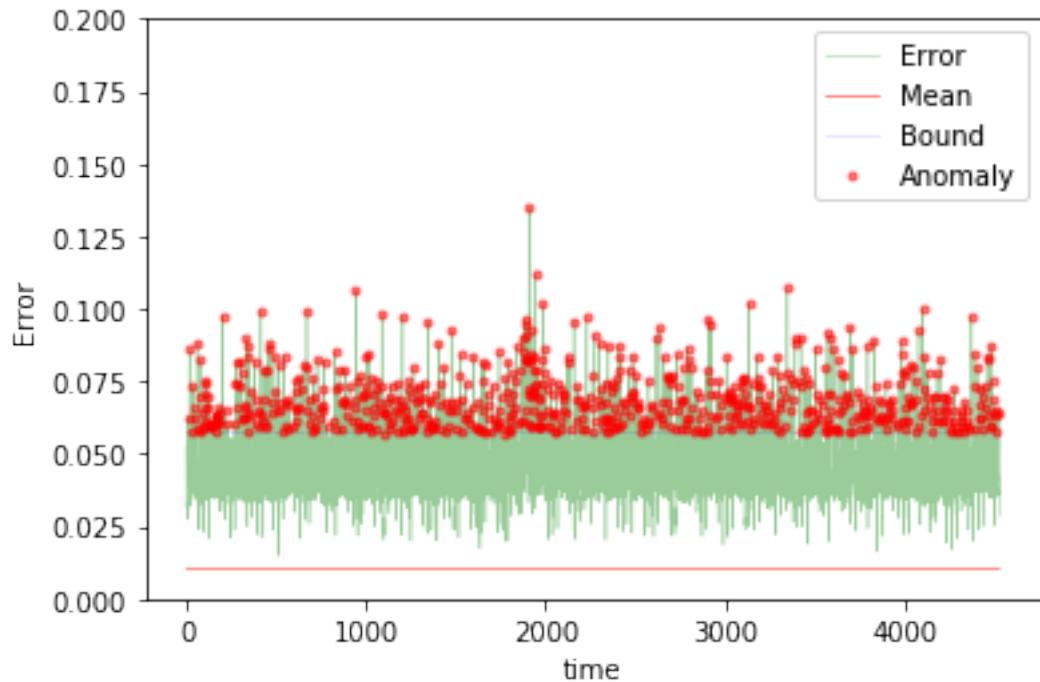




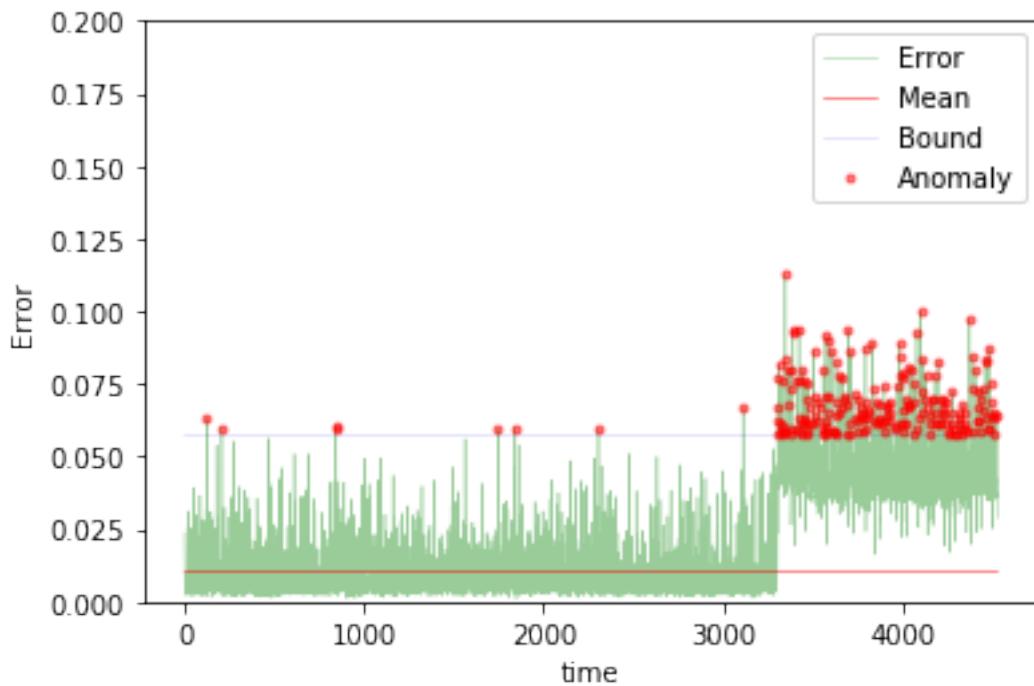
The mean error for gru3_200_normal_ is 0.010378692877892347 for length 4529
Testing on anomaly data.



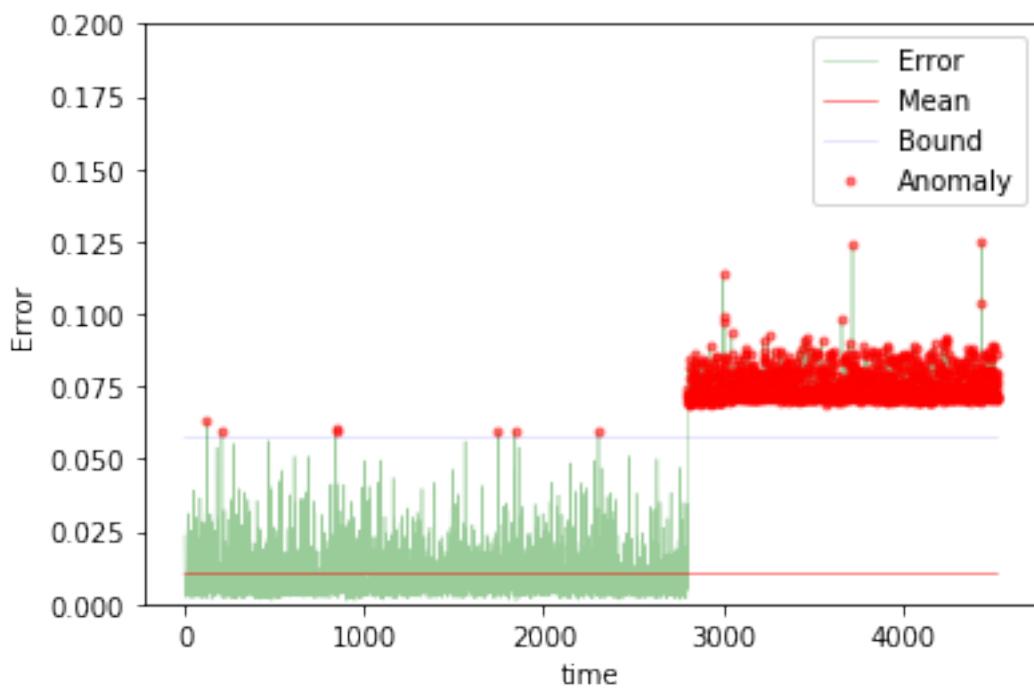
The mean error for gru3_200_anomaly_ is 0.012144407314042684 for length 4529
Testing on different app data.



The mean error for gru3_200_diff_app_ is 0.04589599983968481 for length 4529
Testing on App change synthetic data.



The mean error for gru3_200_app_change_ is 0.020069935096980095 for length 4529
Testing on Net flood synthetic data.



```
The mean error for gru3_200_net_flood_ is 0.034934996287726 for length 4529
=====
```

2.1.8 RNN with 4 GRU layers dim compression.

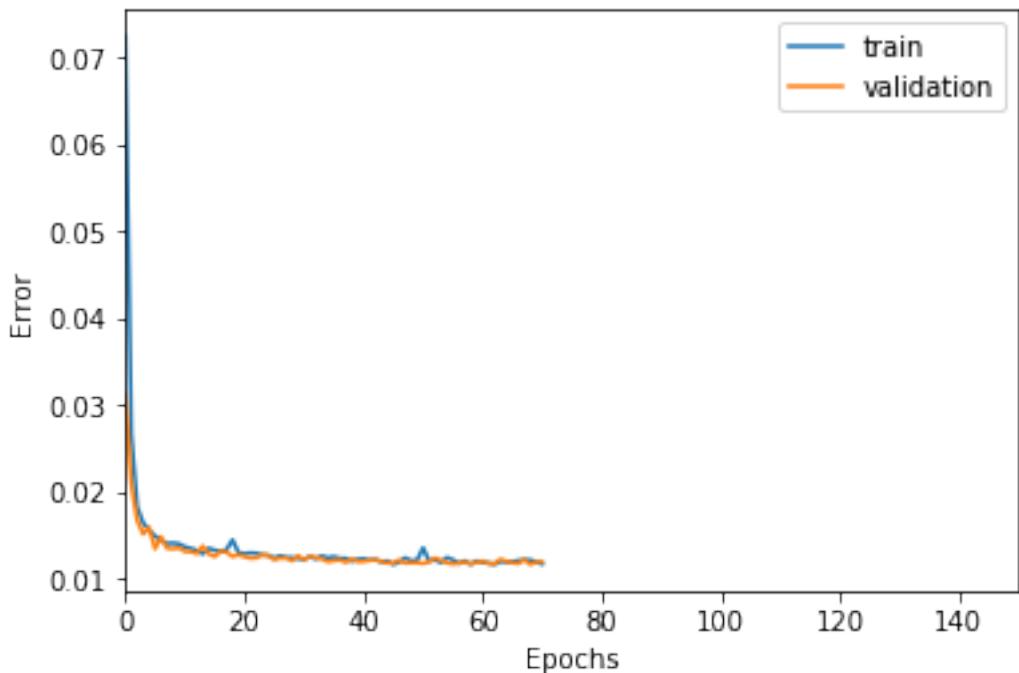
2 steps

```
In [232]: TIMESTEPS = 2
DIM = 29
tgen = flat_generator(X, TIMESTEPS,0)
vgen = flat_generator(val_X, TIMESTEPS,0)
name = "gru4_2"

In [233]: input_layer = Input(shape=(TIMESTEPS,DIM))
hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
hidden = GRU(7, activation='relu', return_sequences=True)(hidden)
hidden = GRU(5, activation='relu', return_sequences=True)(hidden)
hidden = GRU(DIM, activation='relu')(hidden)
output = Dense(DIM, activation='sigmoid')(hidden)

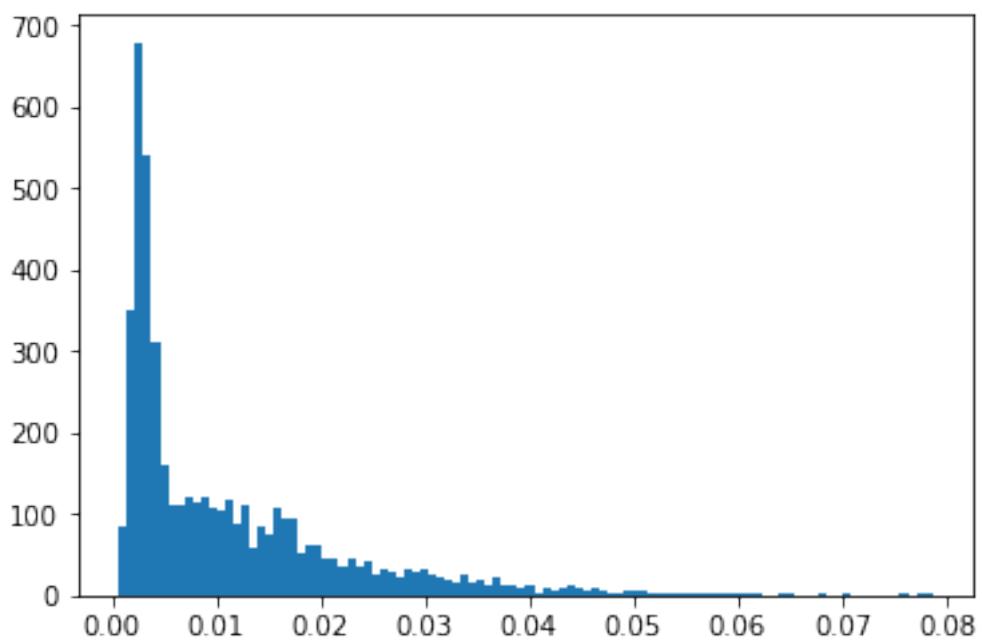
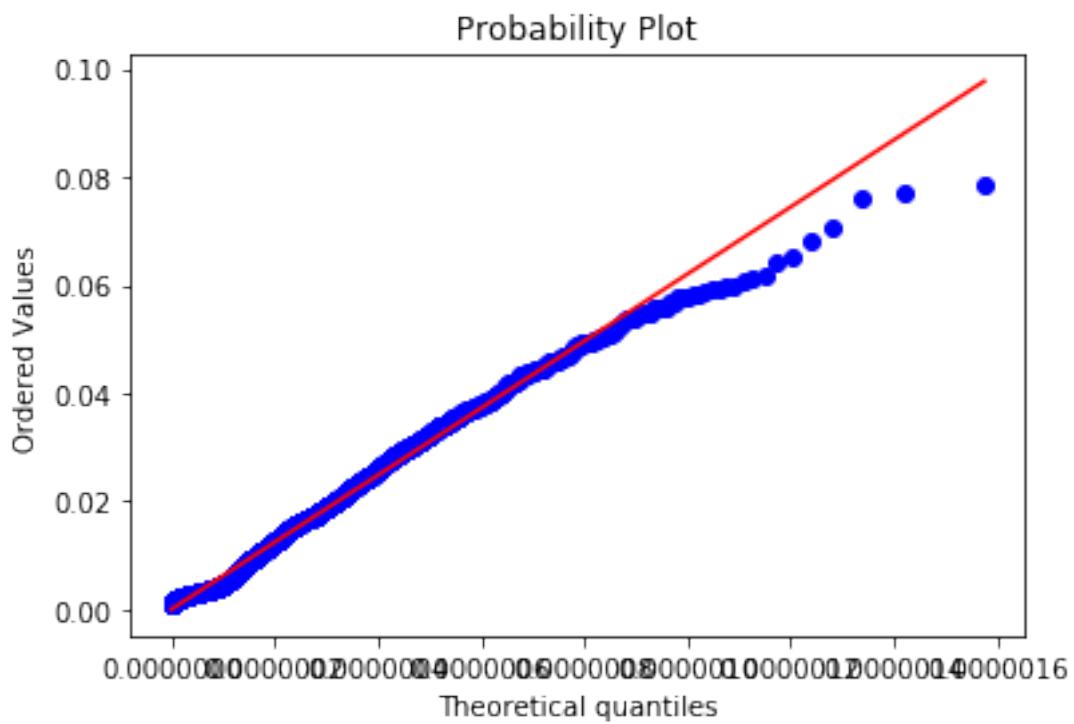
In [234]: model = Model(input_layer, output)
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

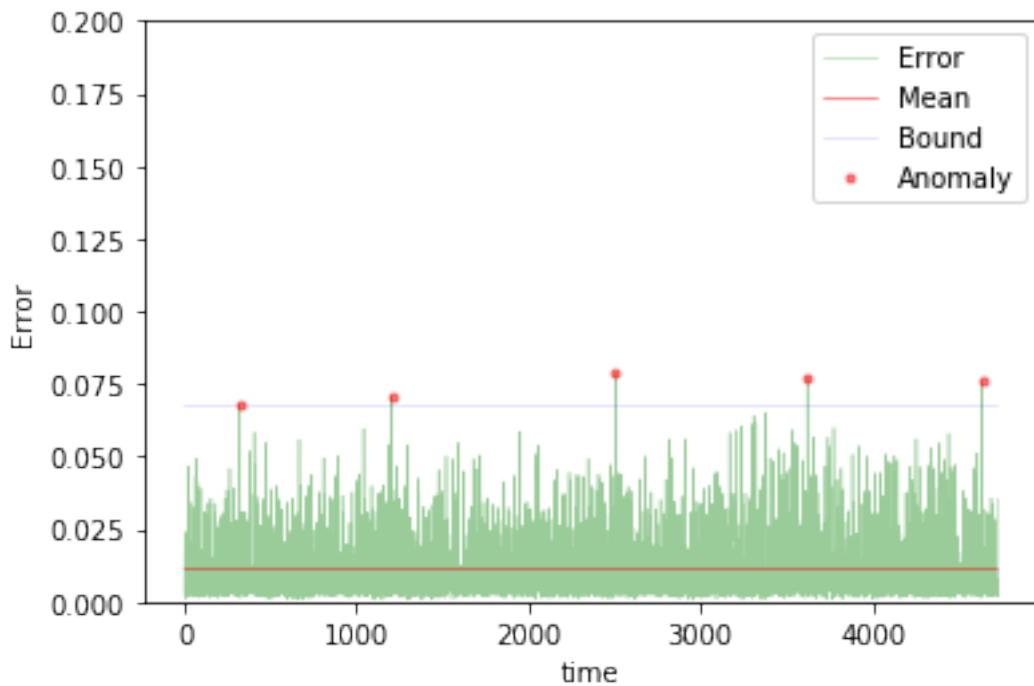
In [235]: train(model, tgen, vgen, name=name)
test(model, ravel=0, name=name, window=TIMESTEPS)
```



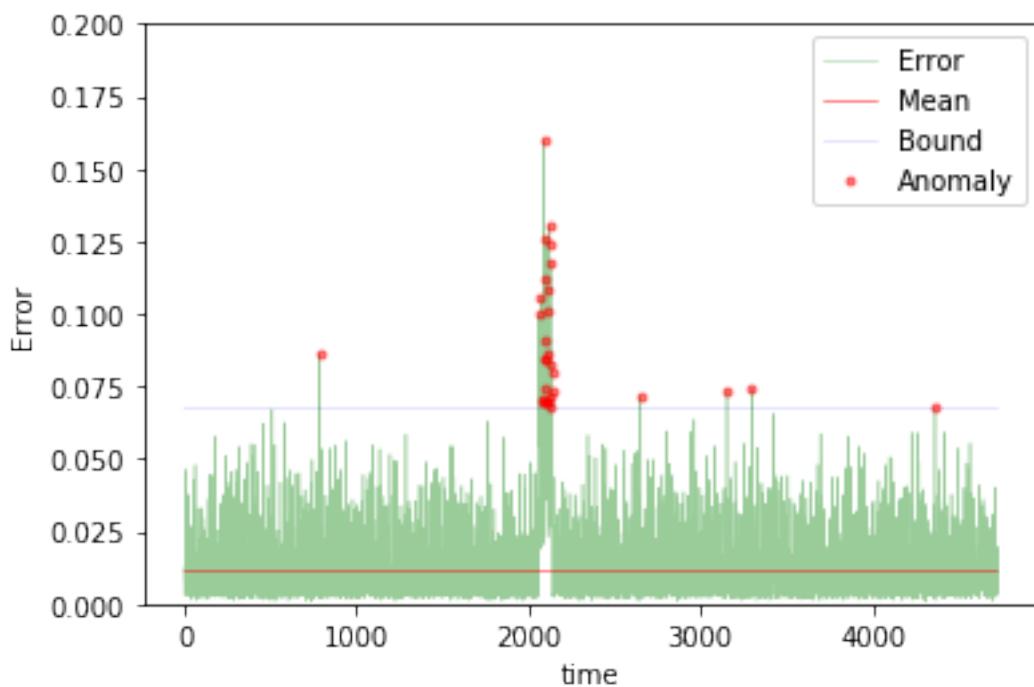
```
Training loss for final epoch is 0.011701393411494792
Validation loss for final epoch is 0.011948348972364329
----- Beginning tests for gru4_2 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

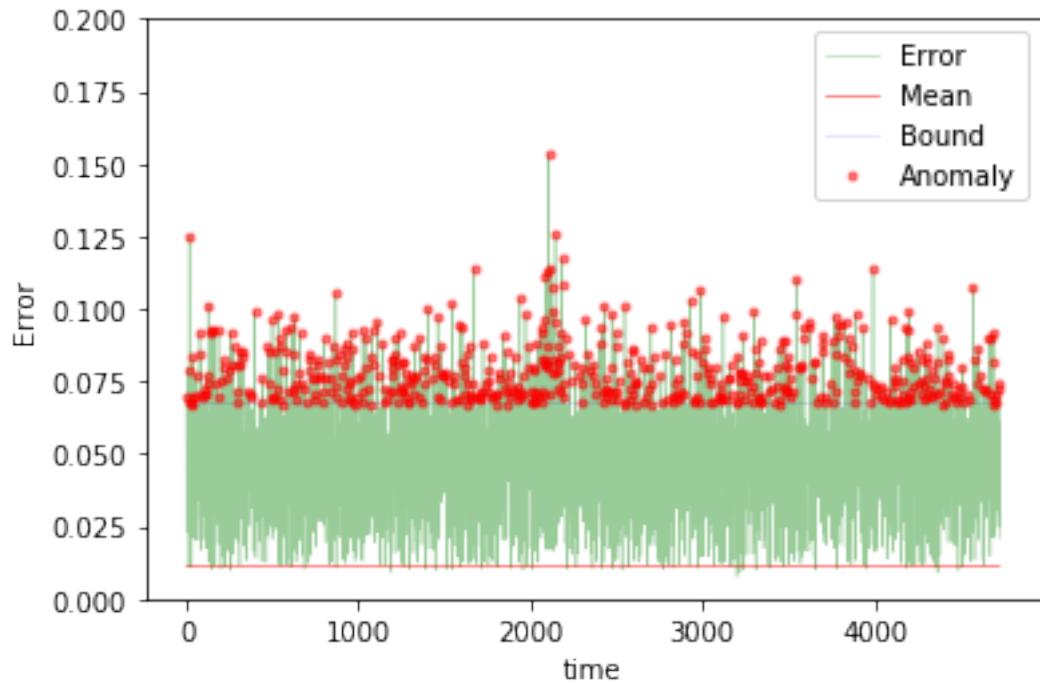




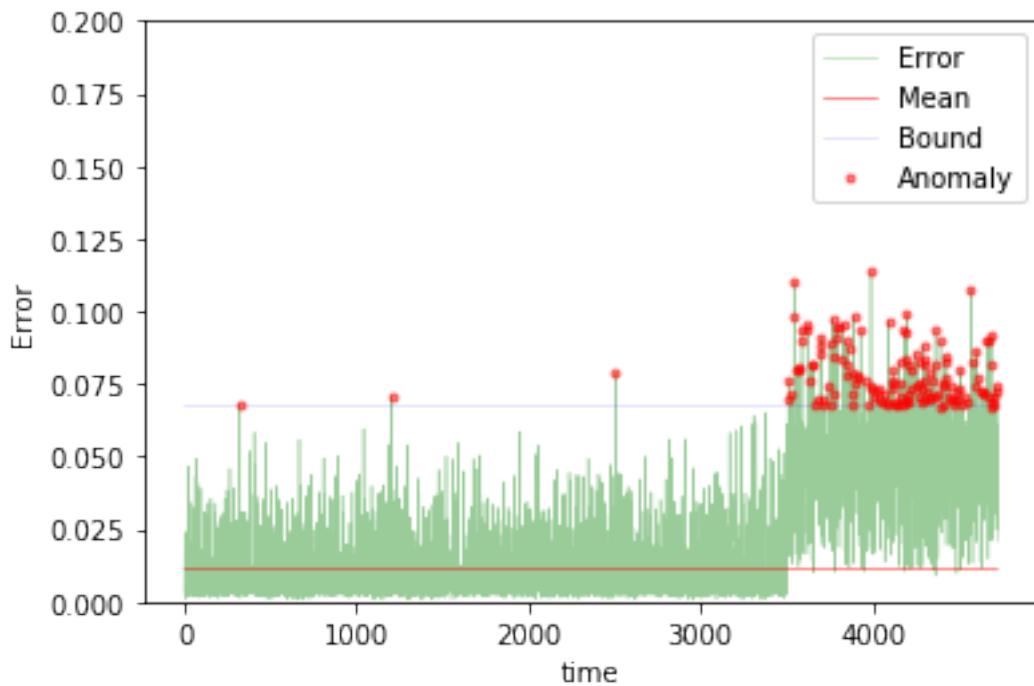
The mean error for gru4_2_normal_ is 0.011359818726701647 for length 4727
Testing on anomaly data.



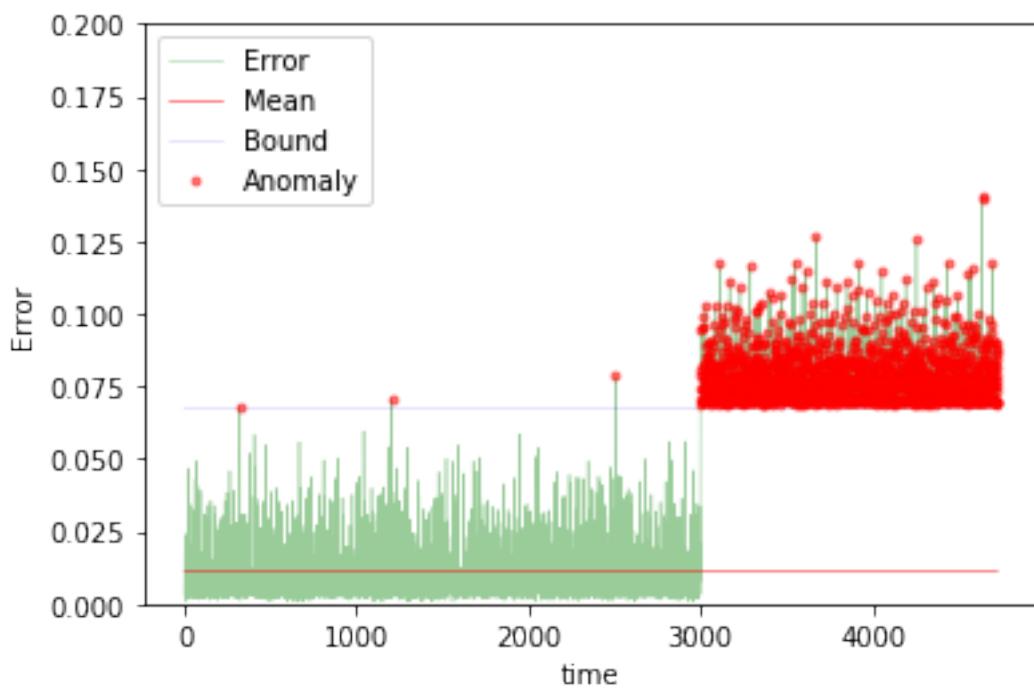
The mean error for gru4_2_anomaly_ is 0.013093132778607589 for length 4727
Testing on different app data.



The mean error for gru4_2_diff_app_ is 0.04844227379618291 for length 4727
Testing on App change synthetic data.



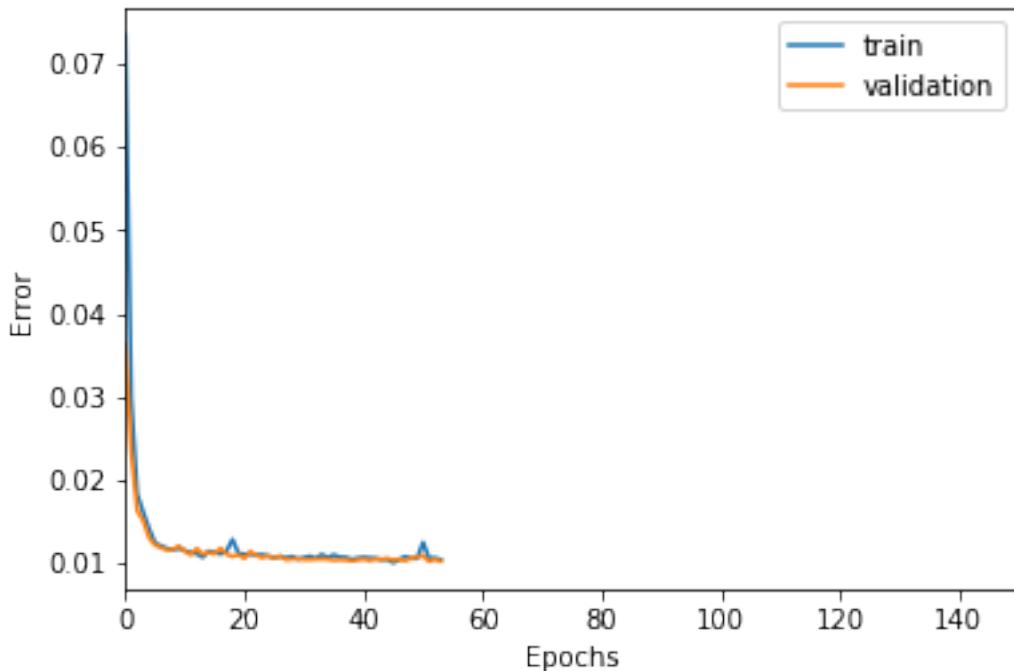
The mean error for gru4_2_app_change_ is 0.020928863634592867 for length 4727
Testing on Net flood synthetic data.



```
The mean error for gru4_2_net_flood_ is 0.035843095499551995 for length 4727  
=====
```

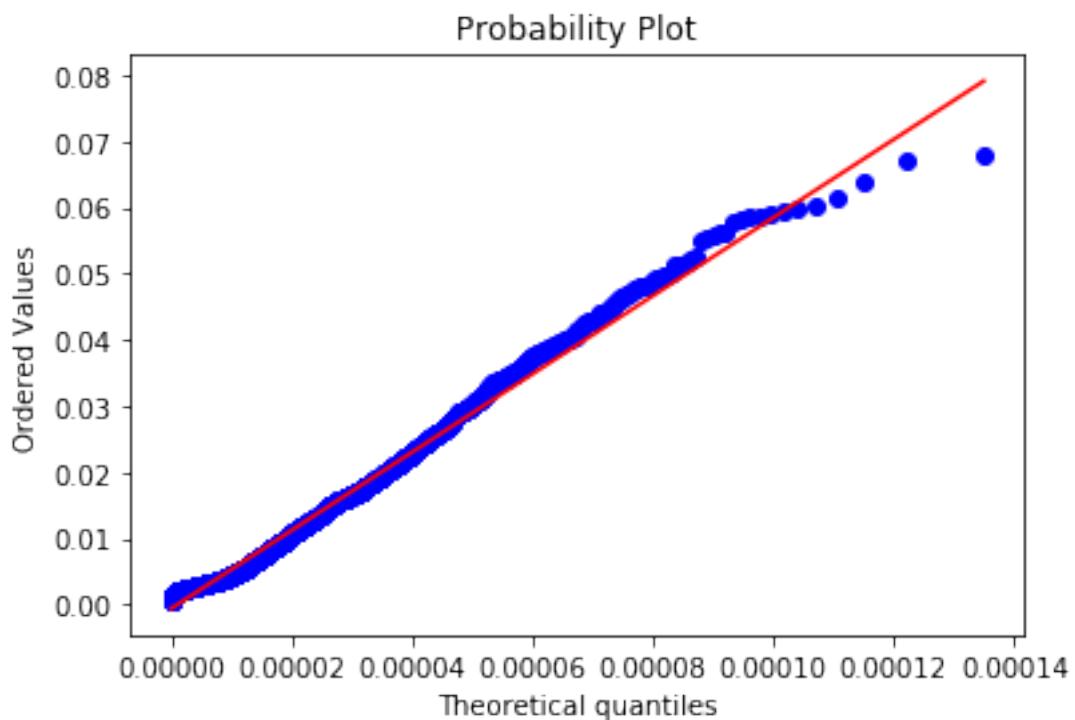
5 steps

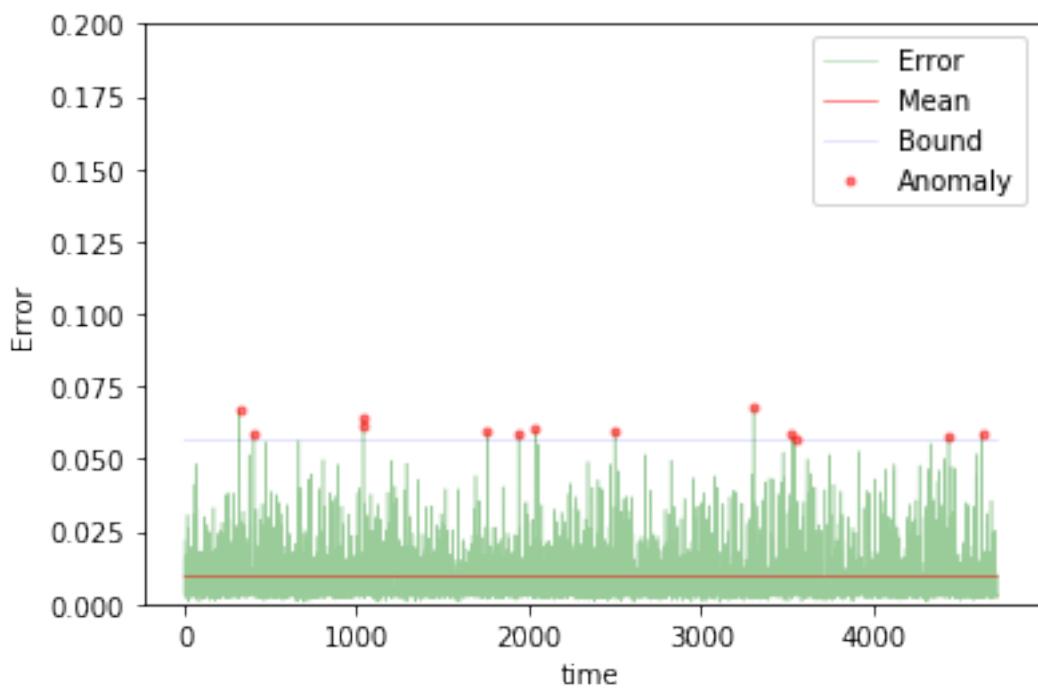
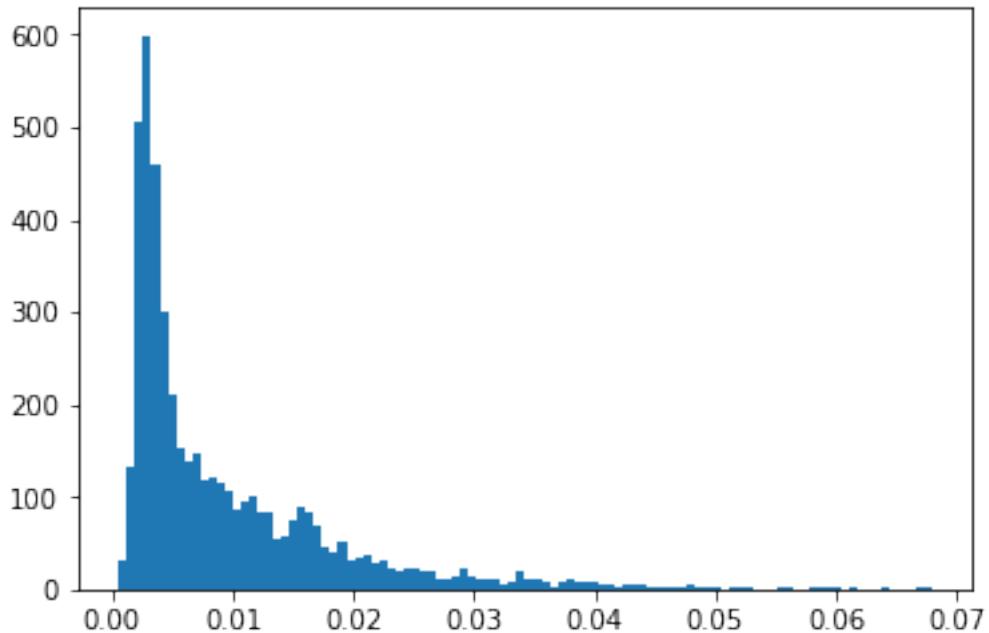
```
In [236]: TIMESTEPS = 5  
DIM = 29  
tgen = flat_generator(X, TIMESTEPS, 0)  
vgen = flat_generator(val_X, TIMESTEPS, 0)  
name = "gru4_5"  
  
In [237]: input_layer = Input(shape=(TIMESTEPS,DIM))  
hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)  
hidden = GRU(7, activation='relu', return_sequences=True)(hidden)  
hidden = GRU(5, activation='relu', return_sequences=True)(hidden)  
hidden = GRU(DIM, activation='relu')(hidden)  
output = Dense(DIM, activation='sigmoid')(hidden)  
  
In [238]: model = Model(input_layer, output)  
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])  
  
In [239]: train(model, tgen, vgen, name=name)  
test(model, ravel=0, name=name, window=TIMESTEPS)
```



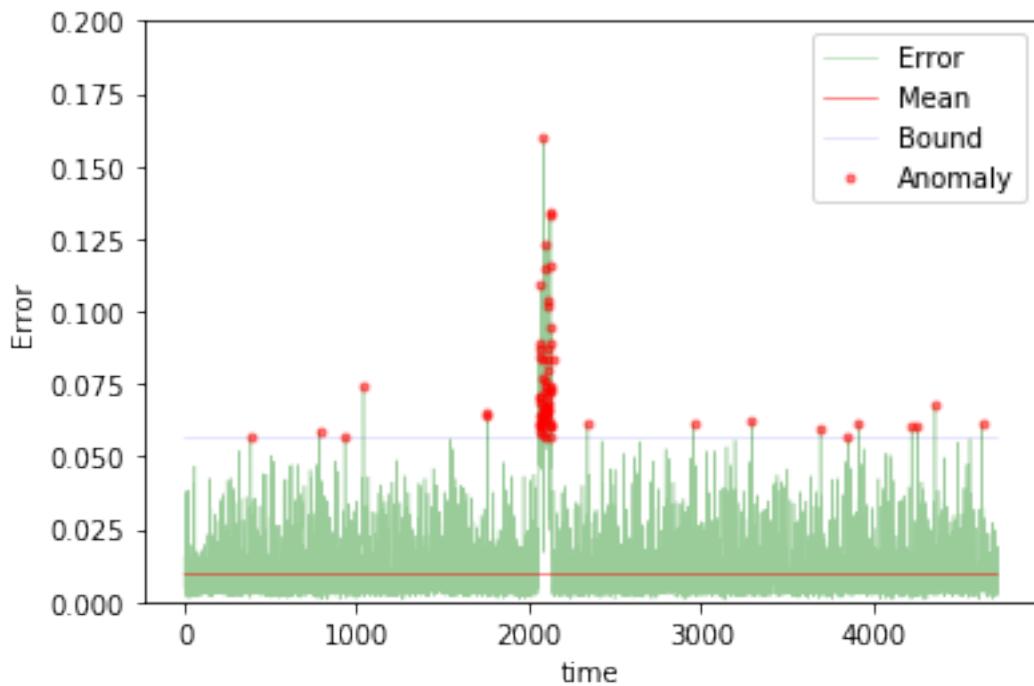
```
Training loss for final epoch is 0.010406905697658658
Validation loss for final epoch is 0.010243680570274591
----- Beginning tests for gru4_5 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

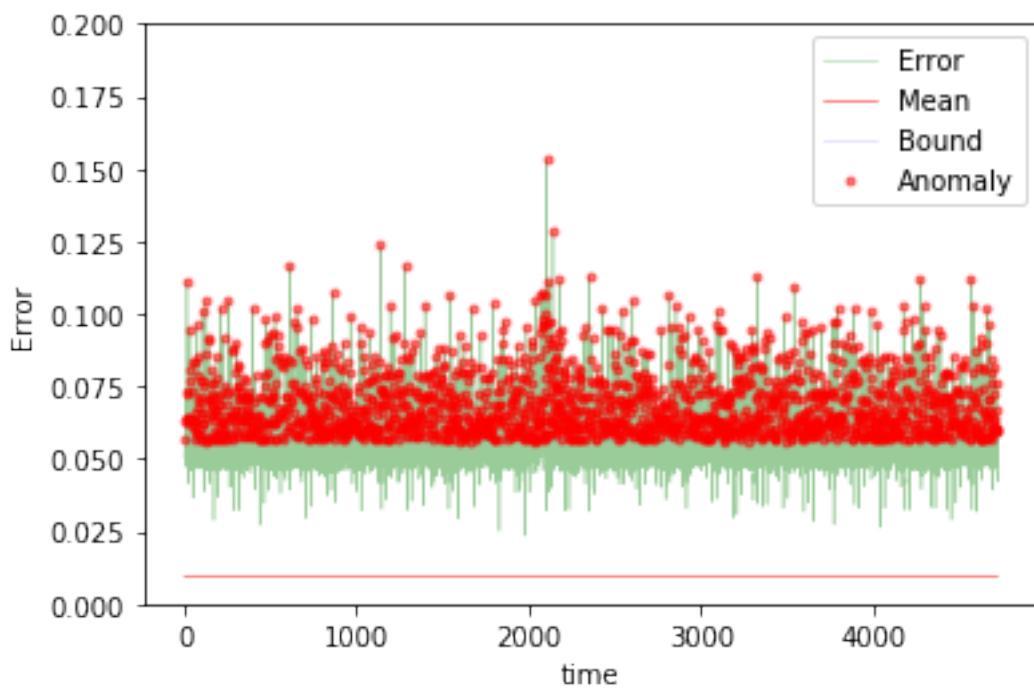




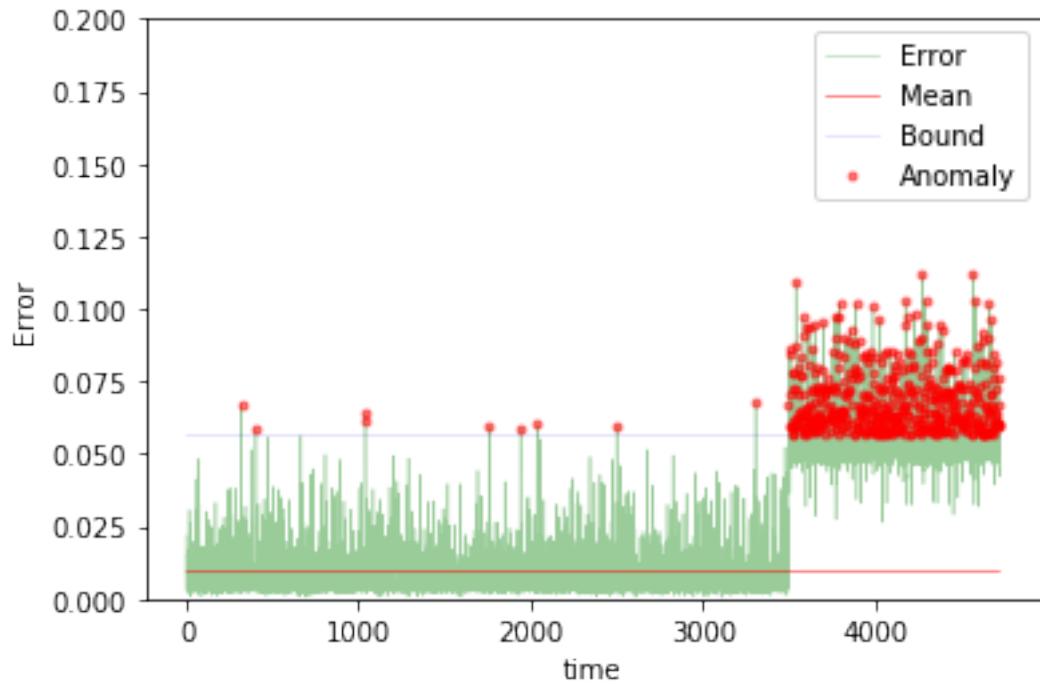
The mean error for gru4_5_normal_ is 0.009588797827326512 for length 4724
Testing on anomaly data.



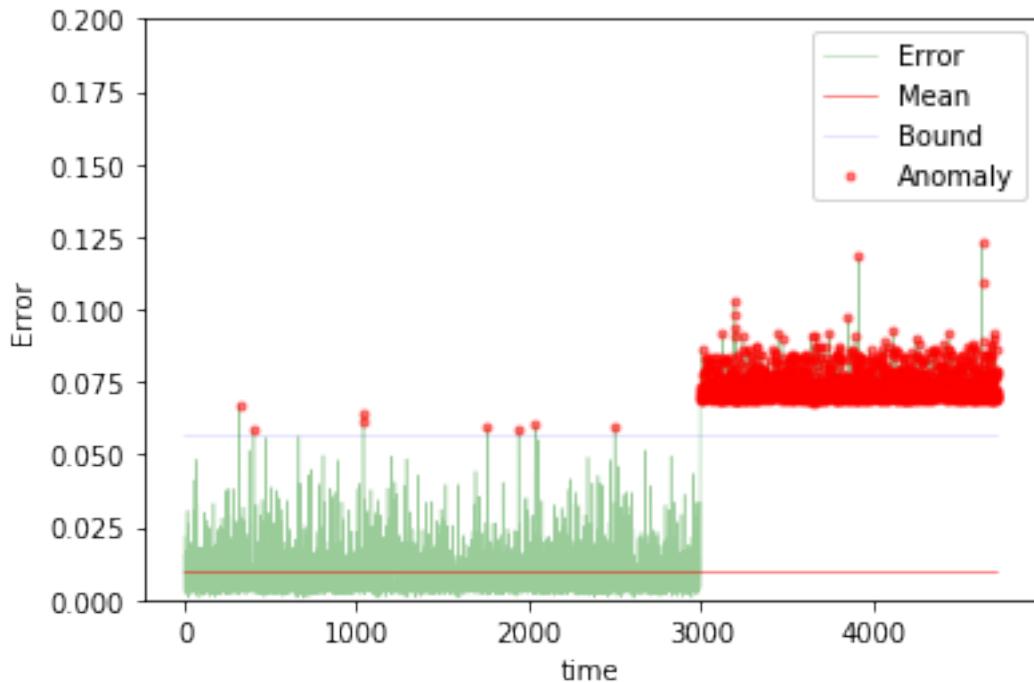
The mean error for gru4_5_anomaly_ is 0.01160979320251134 for length 4724
Testing on different app data.



The mean error for gru4_5_diff_app_ is 0.057270746116431986 for length 4724
Testing on App change synthetic data.



The mean error for gru4_5_app_change_ is 0.021850030855108064 for length 4724
Testing on Net flood synthetic data.



```
The mean error for gru4_5_net_flood_ is 0.033040620772288455 for length 4724
=====
```

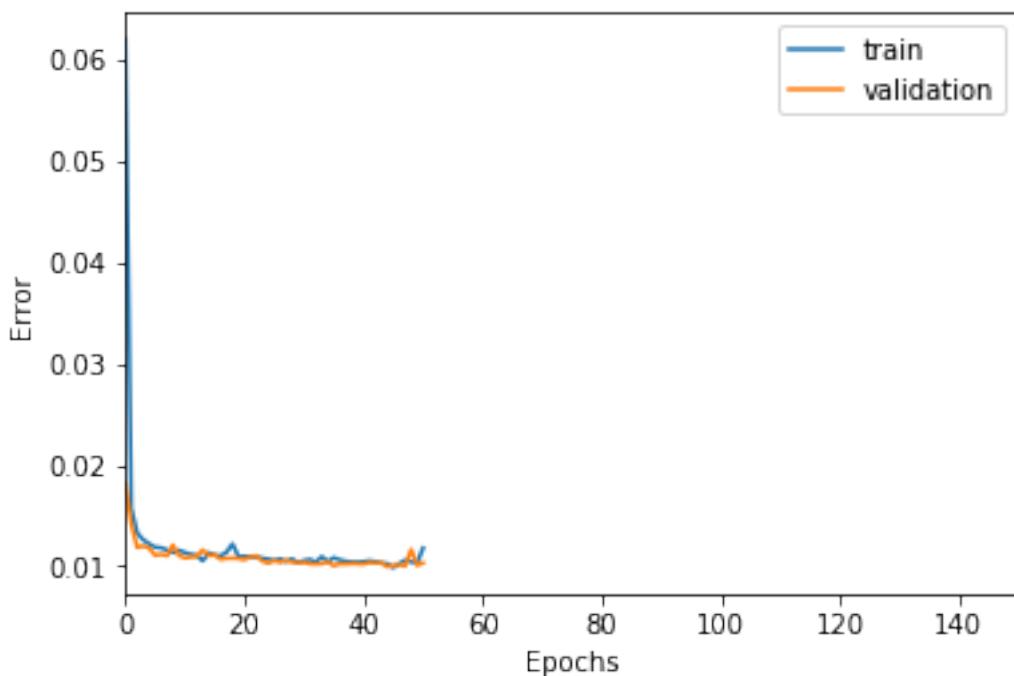
10 steps

```
In [240]: TIMESTEPS = 10
DIM = 29
tgen = flat_generator(X, TIMESTEPS, 0)
vgen = flat_generator(val_X, TIMESTEPS, 0)
name = "gru4_10"

In [241]: input_layer = Input(shape=(TIMESTEPS,DIM))
hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
hidden = GRU(7, activation='relu', return_sequences=True)(hidden)
hidden = GRU(5, activation='relu', return_sequences=True)(hidden)
hidden = GRU(DIM, activation='relu')(hidden)
output = Dense(DIM, activation='sigmoid')(hidden)

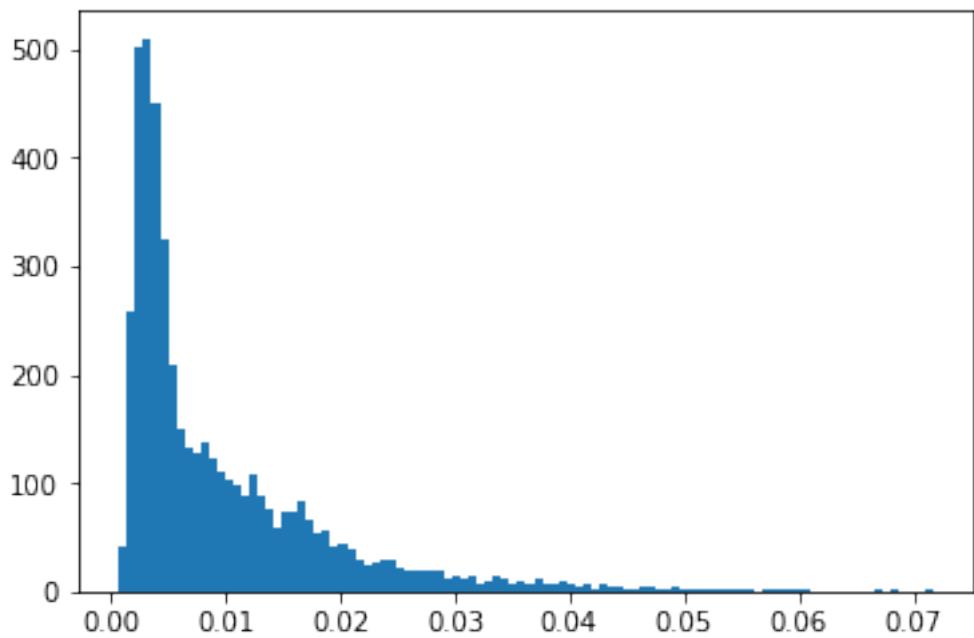
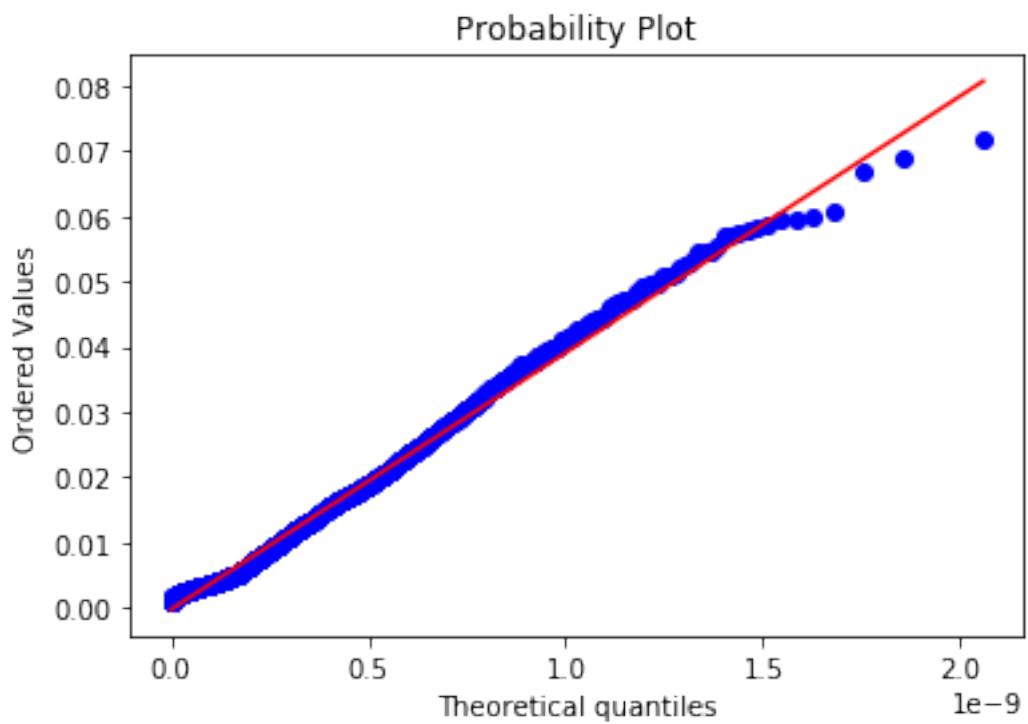
In [242]: model = Model(input_layer, output)
model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])
```

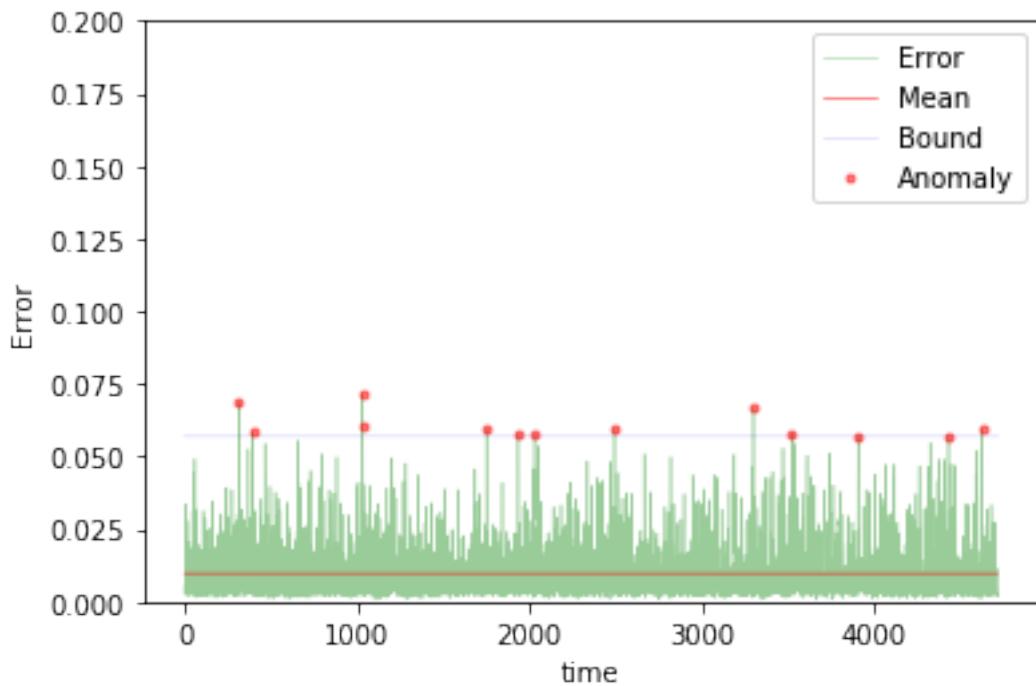
```
In [243]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```



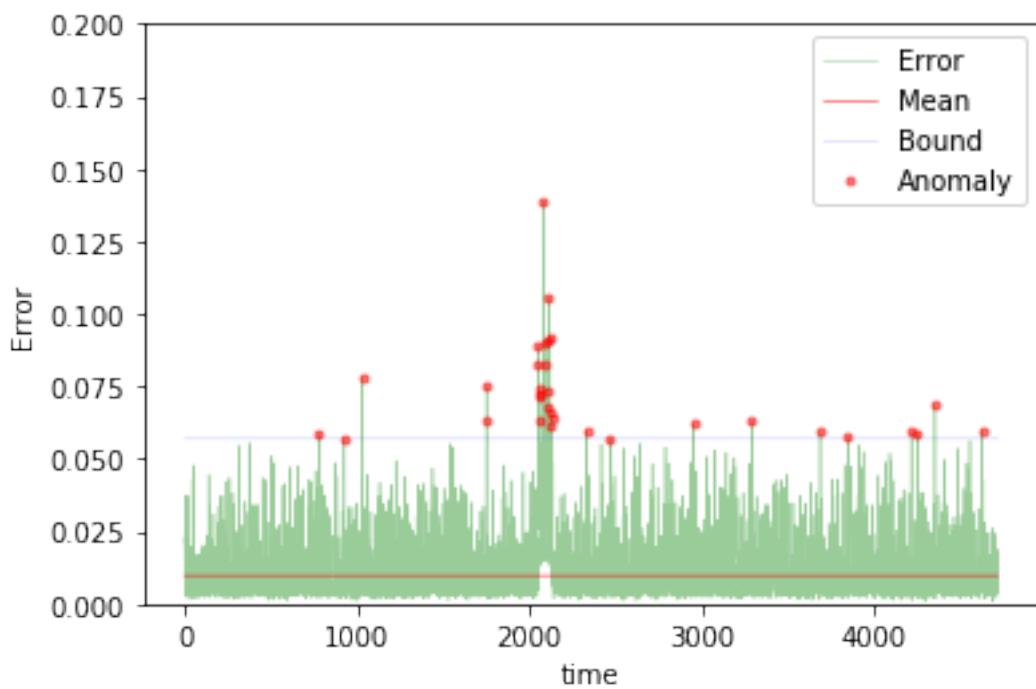
```
Training loss for final epoch is 0.011783752114628442
Validation loss for final epoch is 0.010329104818869382
----- Beginning tests for gru4_10 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
    improvement from the last ten iterations.
    warnings.warn(msg, RuntimeWarning)
```

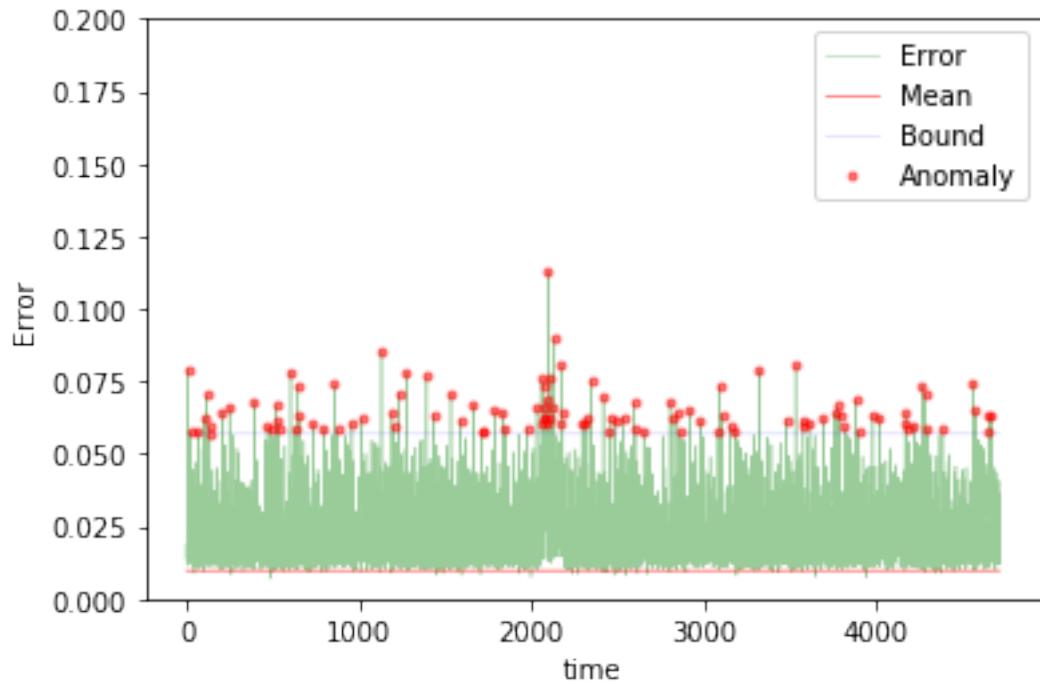




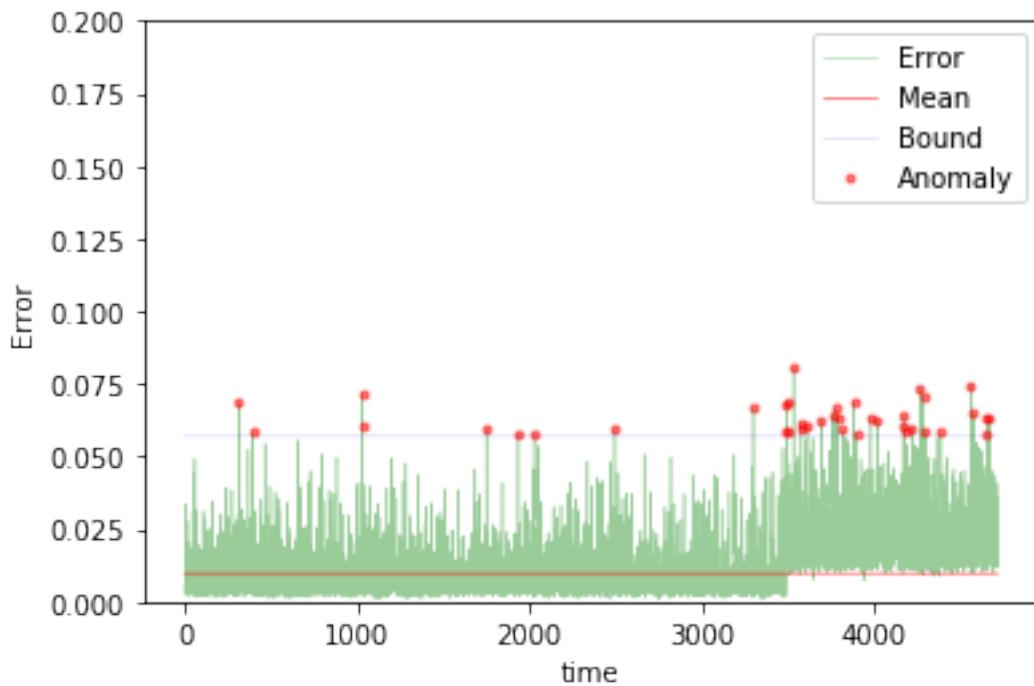
The mean error for gru4_10_normal_ is 0.009827070699616773 for length 4719
Testing on anomaly data.



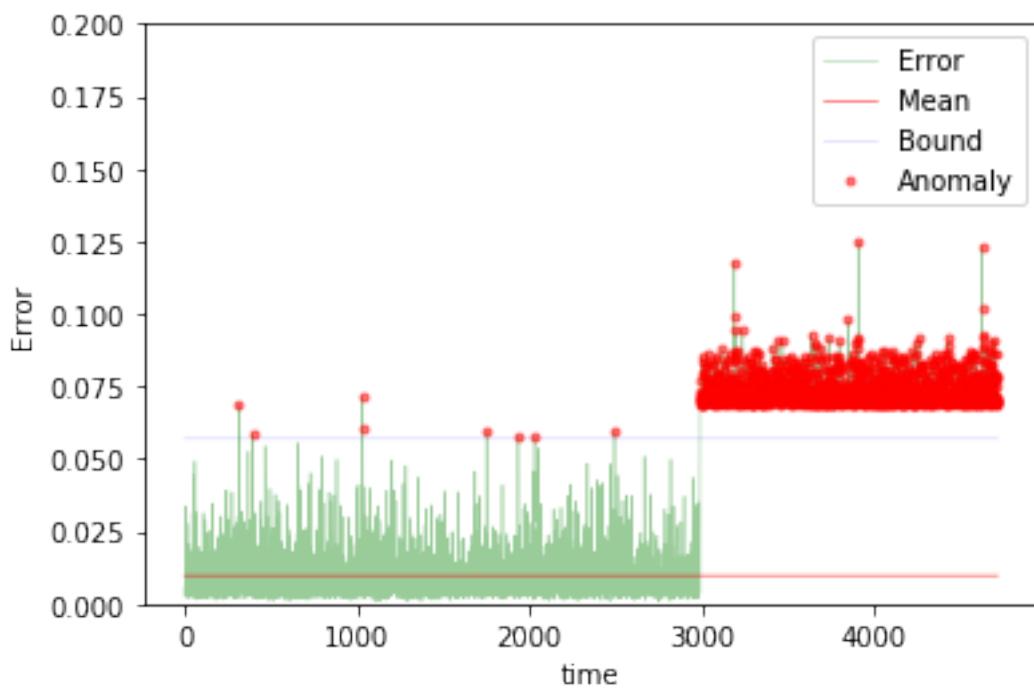
The mean error for gru4_10_anomaly_ is 0.011380686203668593 for length 4719
Testing on different app data.



The mean error for gru4_10_diff_app_ is 0.022768274298750165 for length 4719
Testing on App change synthetic data.



The mean error for gru4_10_app_change_ is 0.013109829834639588 for length 4719
Testing on Net flood synthetic data.



```
The mean error for gru4_10_net_flood_ is 0.03325717947350879 for length 4719
=====
```

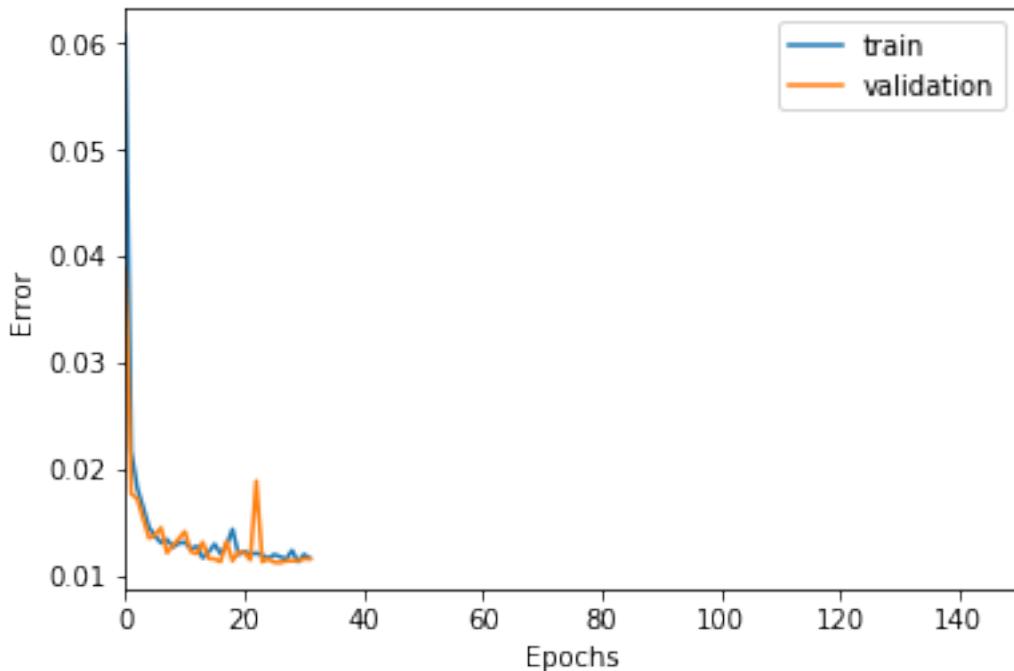
20 steps

```
In [244]: TIMESTEPS = 20
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru4_20"

In [245]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(7, activation='relu', return_sequences=True)(hidden)
          hidden = GRU(5, activation='relu', return_sequences=True)(hidden)
          hidden = GRU(DIM, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

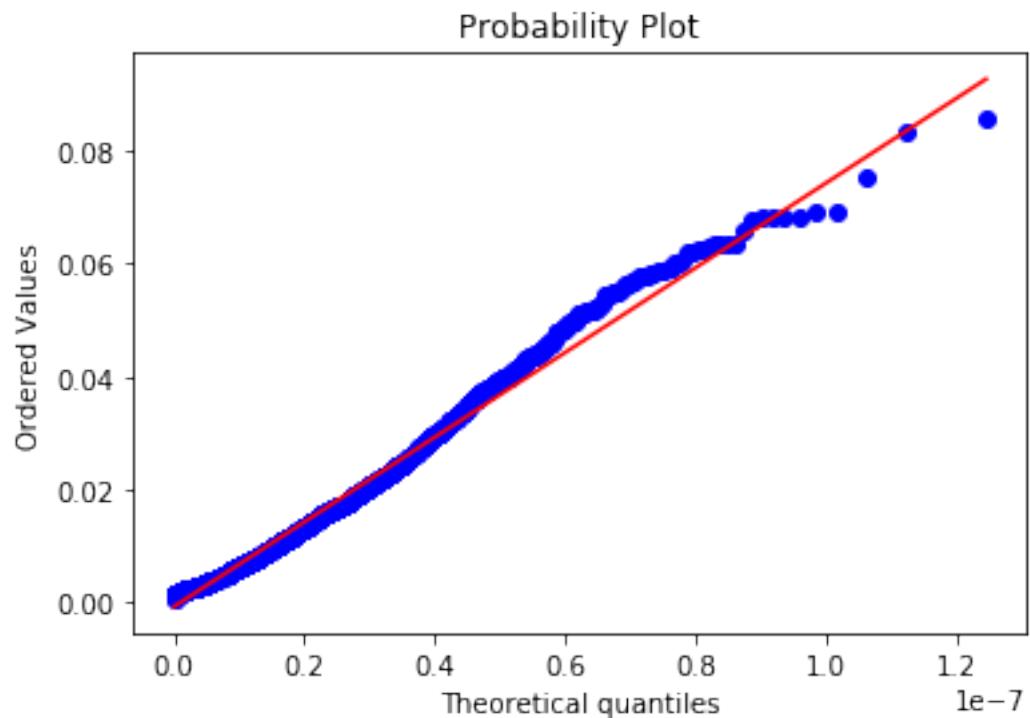
In [246]: model = Model(input_layer, output)
          model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mae'])

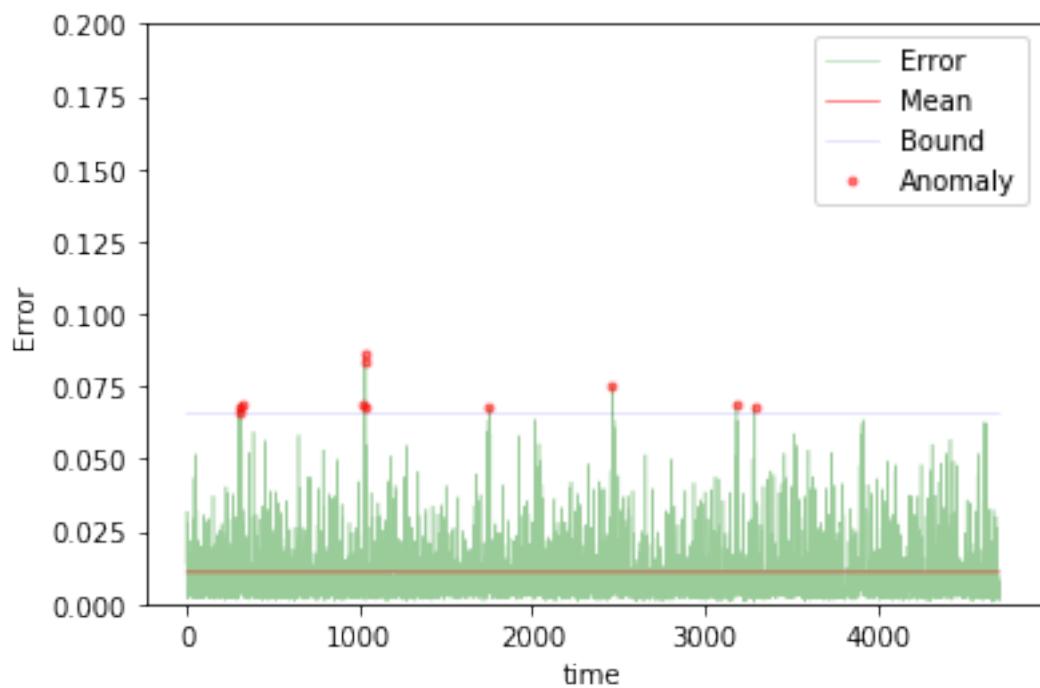
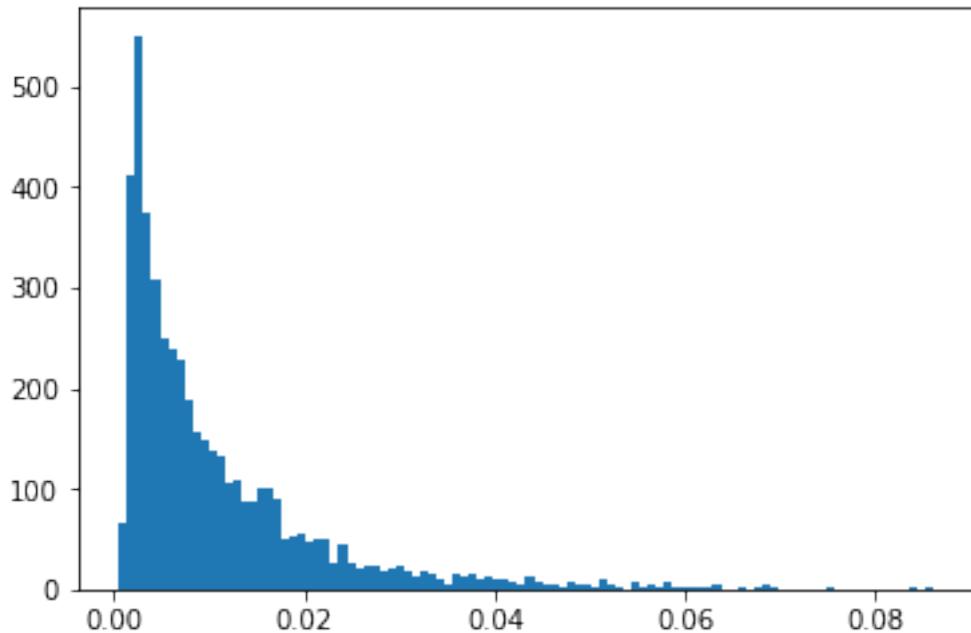
In [247]: train(model, tgen, vgen, name=name)
          test(model, ravel=0, name=name, window=TIMESTEPS)
```



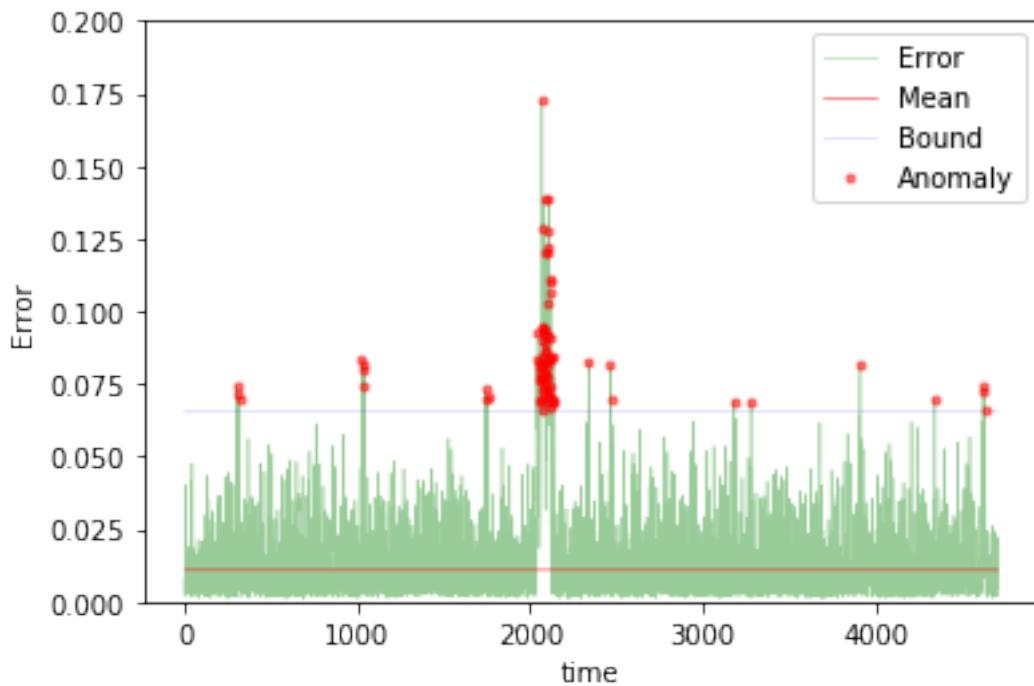
```
Training loss for final epoch is 0.011577315336093307
Validation loss for final epoch is 0.011534050096059218
----- Beginning tests for gru4_20 -----
Testing on normal data.
```

```
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/stats/_continuous_distns.py:412: Run
  sk = 2*(b-a)*np.sqrt(a + b + 1) / (a + b + 2) / np.sqrt(a*b)
/home/adityas/miniconda3/lib/python3.6/site-packages/scipy/optimize/minpack.py:161: RuntimeWarning:
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

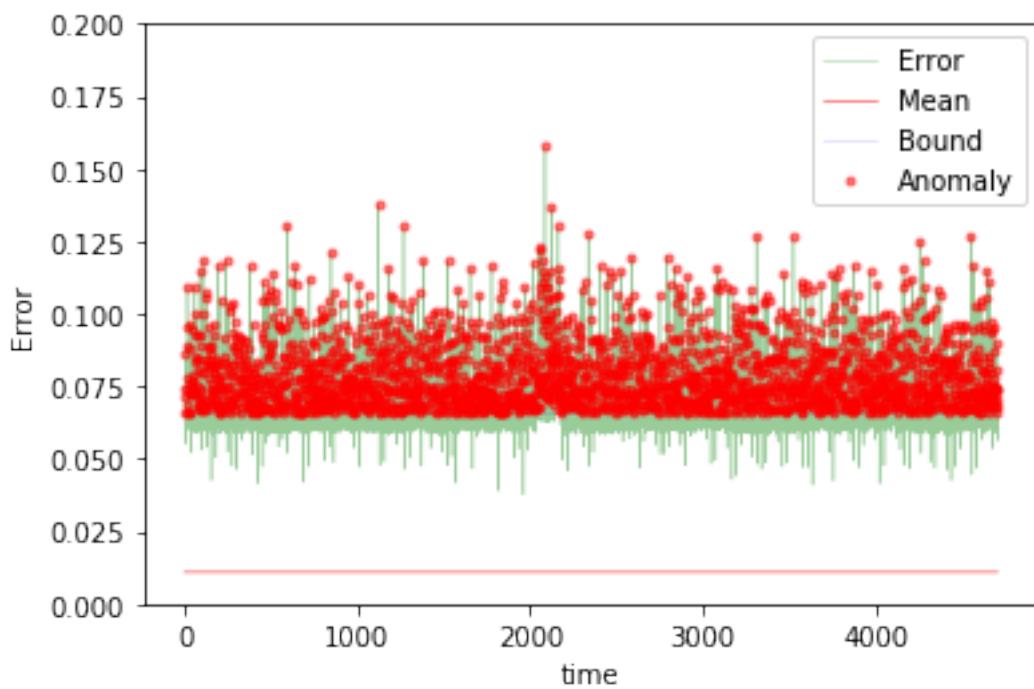




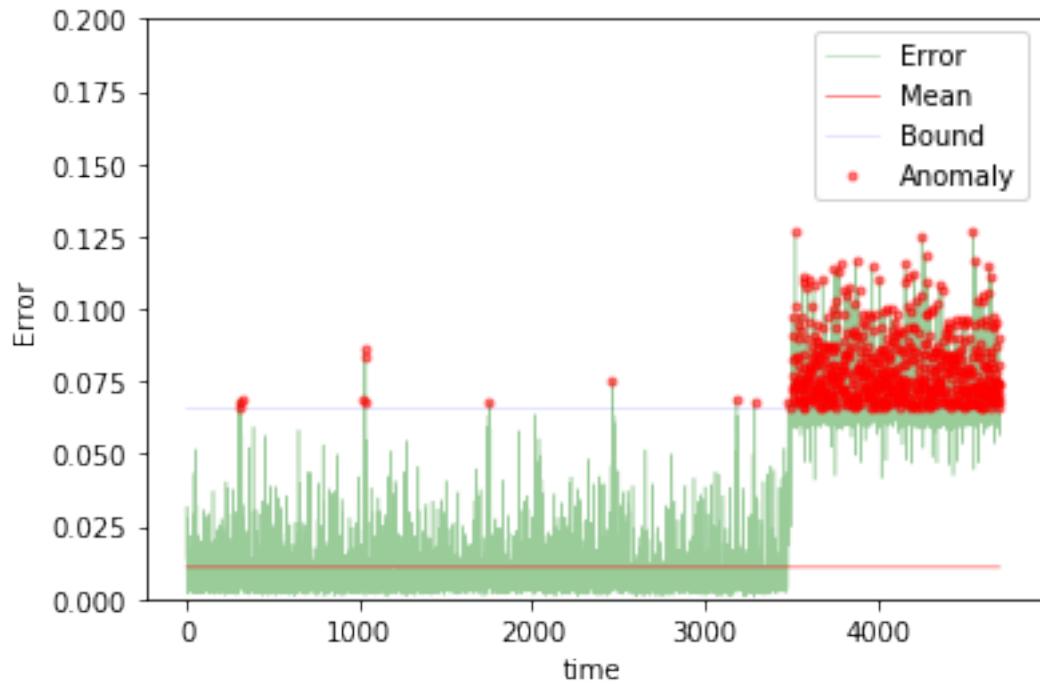
The mean error for gru4_20_normal_ is 0.010950932702517924 for length 4709
Testing on anomaly data.



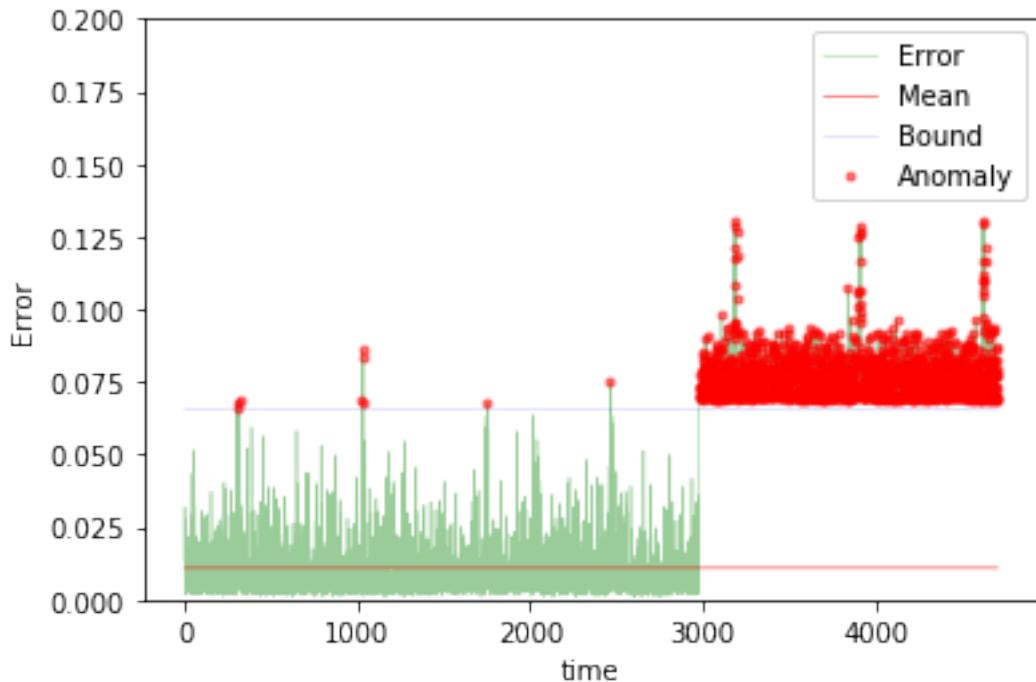
The mean error for gru4_20_anomaly_ is 0.013494113574516488 for length 4709
Testing on different app data.



The mean error for gru4_20_diff_app_ is 0.07094959276699372 for length 4709
Testing on App change synthetic data.



The mean error for gru4_20_app_change_ is 0.02636448788490206 for length 4709
Testing on Net flood synthetic data.



```
The mean error for gru4_20_net_flood_ is 0.03513780684818408 for length 4709
=====
```

50 steps

```
In [248]: TIMESTEPS = 50
          DIM = 29
          tgen = flat_generator(X, TIMESTEPS,0)
          vgen = flat_generator(val_X, TIMESTEPS,0)
          name = "gru4_50"

In [249]: input_layer = Input(shape=(TIMESTEPS,DIM))
          hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
          hidden = GRU(7, activation='relu', return_sequences=True)(hidden)
          hidden = GRU(5, activation='relu', return_sequences=True)(hidden)
          hidden = GRU(DIM, activation='relu')(hidden)
          output = Dense(DIM, activation='sigmoid')(hidden)

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

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

100 steps

```
In [ ]: TIMESTEPS = 100
        DIM = 29
        tgen = flat_generator(X, TIMESTEPS,0)
        vgen = flat_generator(val_X, TIMESTEPS,0)
        name = "gru4_100"

In [ ]: input_layer = Input(shape=(TIMESTEPS,DIM))
        hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
        hidden = GRU(7, activation='relu', return_sequences=True)(hidden)
        hidden = GRU(5, activation='relu', return_sequences=True)(hidden)
        hidden = GRU(DIM, activation='relu')(hidden)
        output = Dense(DIM, activation='sigmoid')(hidden)

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

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

200 steps

```
In [ ]: TIMESTEPS = 200
        DIM = 29
        tgen = flat_generator(X, TIMESTEPS,0)
        vgen = flat_generator(val_X, TIMESTEPS,0)
        name = "gru4_200"

In [ ]: input_layer = Input(shape=(TIMESTEPS,DIM))
        hidden = GRU(10, activation='relu', return_sequences=True)(input_layer)
        hidden = GRU(7, activation='relu', return_sequences=True)(hidden)
        hidden = GRU(5, activation='relu', return_sequences=True)(hidden)
        hidden = GRU(DIM, activation='relu')(hidden)
        output = Dense(DIM, activation='sigmoid')(hidden)

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

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