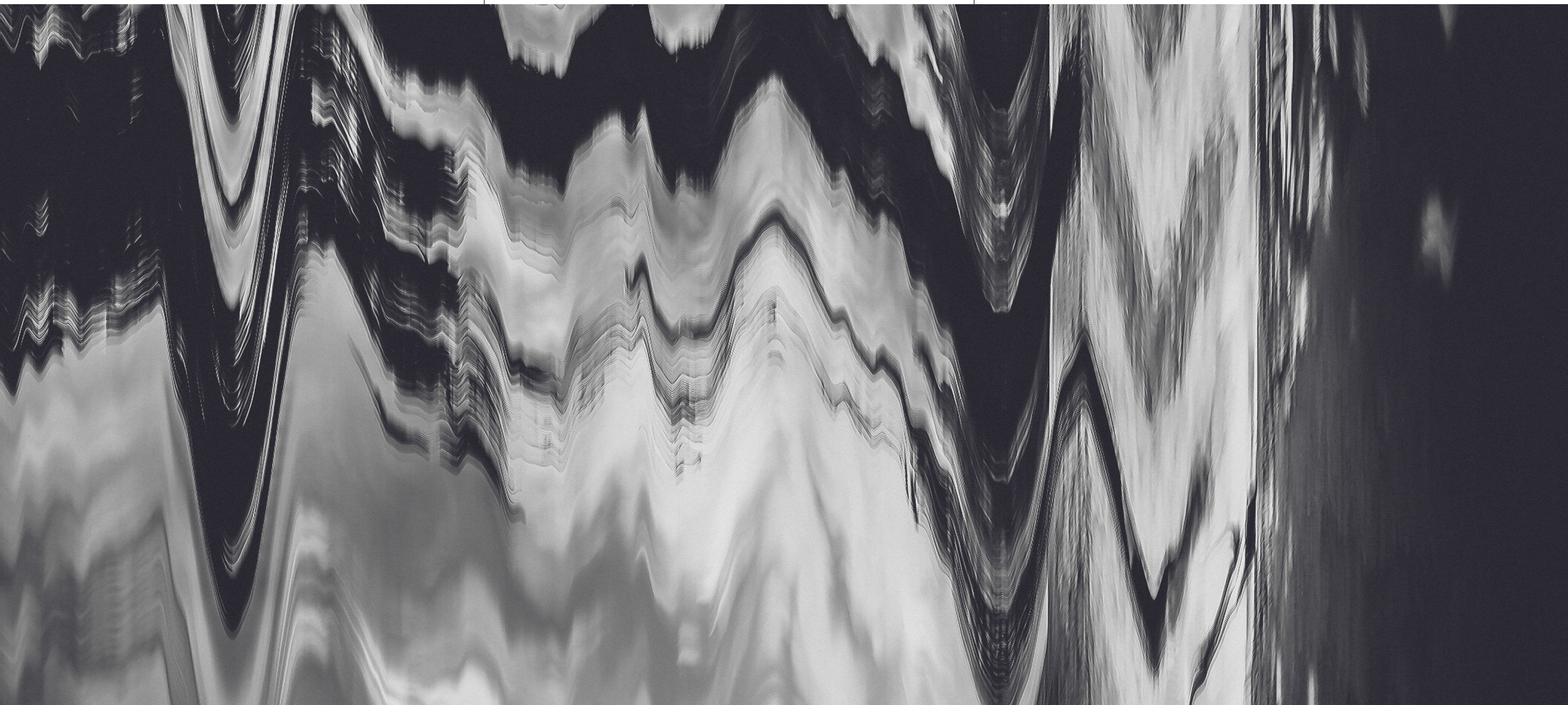
NO.2 | PORTFOLIO PROJECT

TIMESERIES ANOMALIES

TIMESERIES ANOMALY DETECTION
UTILIZING AN AUTOENCODER
DATA: NUMENTA ANOMALY BENCHMARK

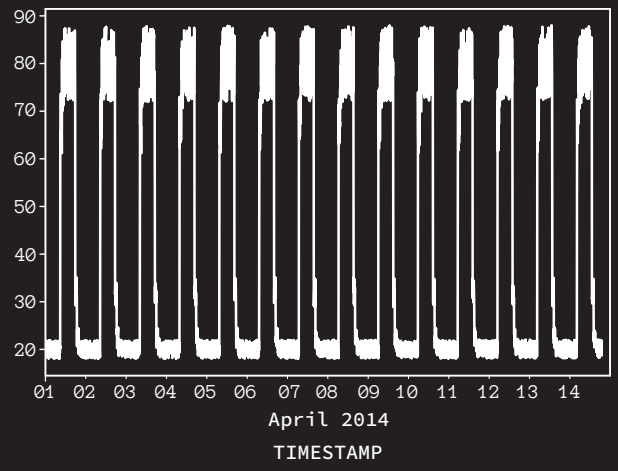
NAME: UMBERTO FASCI
EMAIL: UMBERTOFASCIDEV@GMAIL.COM

CREATED: 10.12.2022



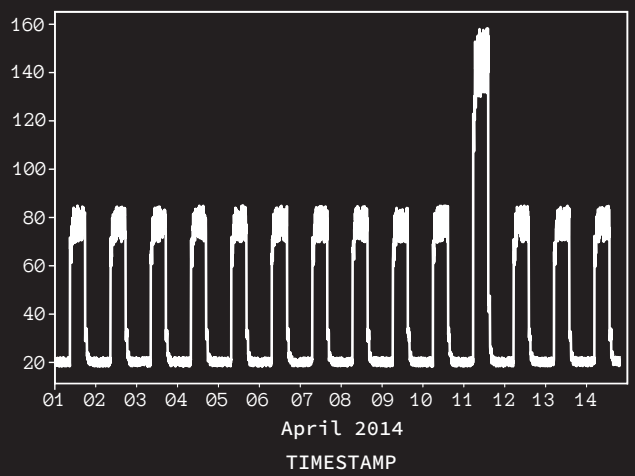
PULL TIMESTEP NOISE DATA

```
df_small_noise_url_suffix = "artificialNoAnomaly/art_daily_small_noise.csv"
df_small_noise_url = master_url_root + df_small_noise_url_suffix
df_small_noise = pd.read_csv(
    df_small_noise_url, parse_dates=True, index_col="timestamp")
```



PULL TIMESTEP NOISE DATA CONTAINING ANOMALY

```
df_daily_jumpsup_url_suffix = "artificialWithAnomaly/art_daily_jumpsup.csv"
df_daily_jumpsup_url = master_url_root + df_daily_jumpsup_url_suffix
df_daily_jumpsup = pd.read_csv(
    df_daily_jumpsup_url, parse_dates=True, index_col="timestamp")
```



DATA SOURCE: https://raw.githubusercontent.com/numenta/NAB/master/data/

TENSORFLOW MODEL

```
model = keras.Sequential(
    [
        layers.Input(shape=(x_train.shape[1],
            x_train.shape[2])),

        layers.Conv1D(
            filters=32, kernel_size=7, padding="same",
            strides=2, activation="relu"),

        layers.Dropout(rate=0.2),

        layers.Conv1D(
            filters=16, kernel_size=7, padding="same",
            strides=2, activation="relu"),

        layers.Conv1DTranspose(
            filters=16, kernel_size=7, padding="same",
            strides=2, activation="relu"),

        layers.Dropout(rate=0.2),

        layers.Conv1DTranspose(
            filters=32, kernel_size=7, padding="same",
            strides=2, activation="relu"),

        layers.Conv1DTranspose(filters=1, kernel_size=7,
            padding="same"),
    ]
)

model.compile(optimizer=keras.optimizers.Adam(
        learning_rate=0.001),
        loss="mse")
```

Continuing this, utilize the averages and standard deviation calculated from the pulled data, and compare it to the TIMESTEP NOISE DATA CONTAINING ANOMALY dataset. This will result in the expected test value of an accurate model.

Anomalies are detected by comparing the test MAE loss of the model run. Where this loss is shown to be above theshold (the maximum of the train_mae_loss), anomalies have been detected.

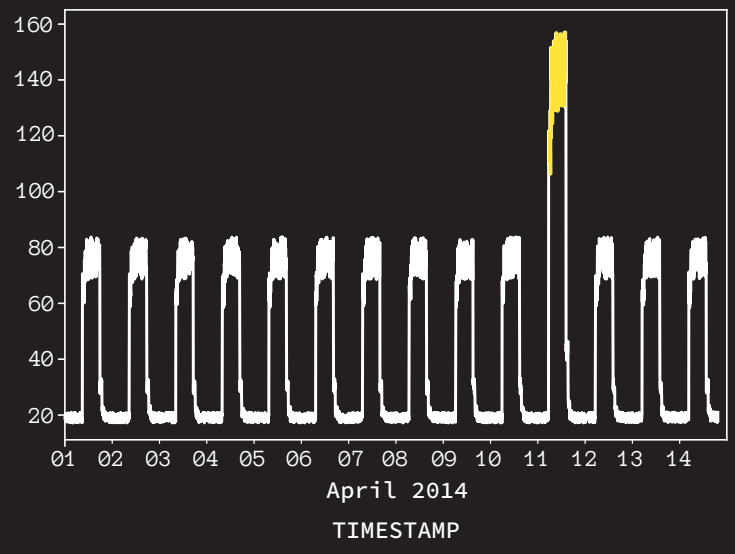
```
x_test_pred = model.predict(x_test)

test_mae_loss = np.mean(np.abs(x_test_pred - x_test),
     axis=1)

test_mae_loss = test_mae_loss.reshape((-1))


ANOMALIES = test_mae_loss > threshold
```

DETECTED ANOMALIES PLOTTED



DATA SOURCE: https://raw.githubusercontent.com/numenta/NAB/master/data/

FIND ALL THE CODE:
https://github.com/UmbertoFasci