

Log Data Generation for GV12 Production

- Praktikum 1

Tobias Herbert

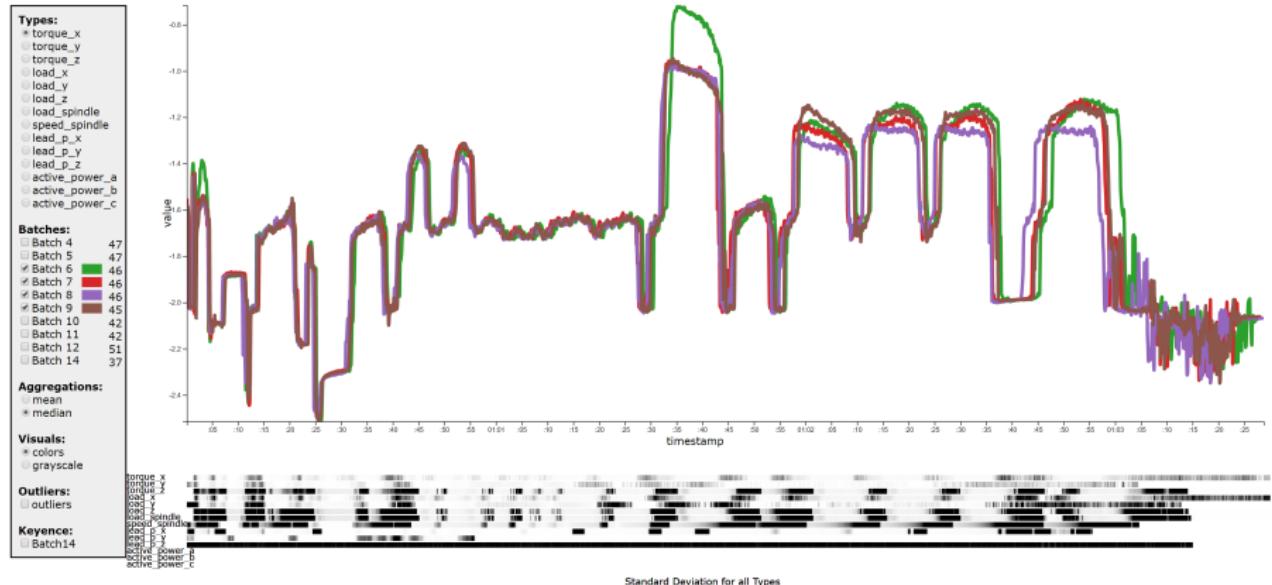
University of Vienna

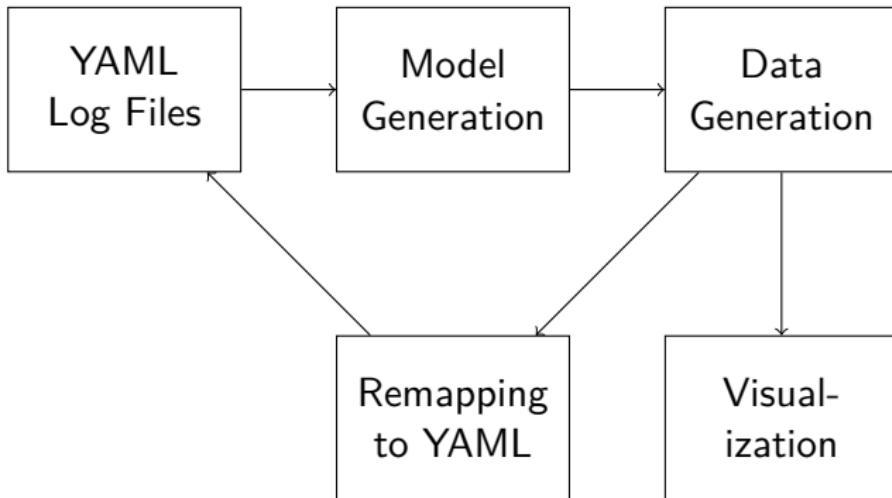
June 29, 2020



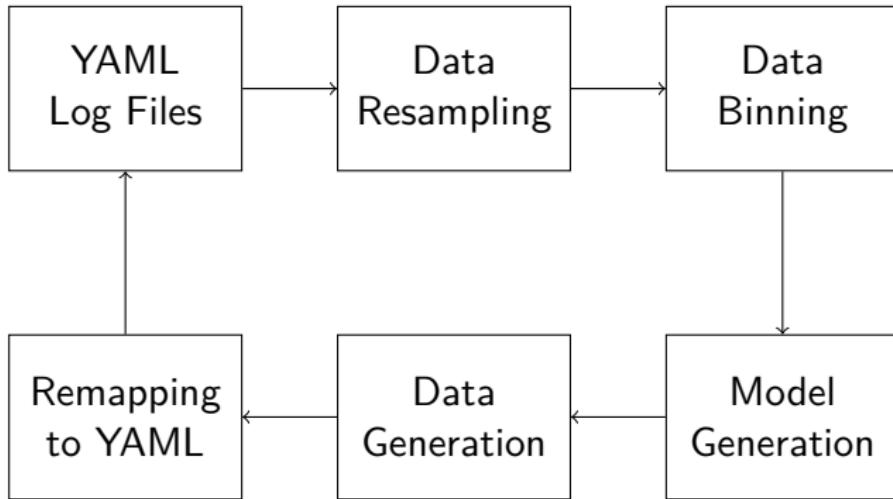
```
- ID: ns=2;s=/Channel/MachineAxis/aaTorque[u1,1]
  source: opcua
  name: Axis/X/aaTorque
  description: ''
  path: "/Object/Sinumerik/Channel/MachineAxis/aaTorque[u1,1]"
  value: "-2.06"
  timestamp: '2019-04-03T16:13:24.239'
  meta:
    StatusCode: Good
    ServerTimestamp: '2019-04-03 14:13:24.316427'
    VariantType: VariantType.Double
    ClientHandle: '225'
- ID: ns=2;s=/Bag/State/resetActive
  source: opcua
  name: State/resetActive
  description: Code whether all channels in mode group are in Reset
  path: "/Object/Sinumerik/Bag/State/resetActive"
  value: '0'
  timestamp: '2019-04-03T16:13:24.239'
  meta:
    StatusCode: Good
    ServerTimestamp: '2019-04-03 14:13:24.315427'
    VariantType: VariantType.UInt16
    ClientHandle: '205'
- ID: ns=2;s=/Channel/MachineAxis/aaLoad[u1,1]
  source: opcua
  name: Axis/X/aaLoad
  description: ''
  path: "/Object/Sinumerik/Channel/MachineAxis/aaLoad[u1,1]"
  value: '21.0205078125'
  timestamp: '2019-04-03T16:13:24.239'
  meta:
    StatusCode: Good
    ServerTimestamp: '2019-04-03 14:13:24.316427'
    VariantType: VariantType.Double
    ClientHandle: '234'
- ID: ns=2;s=/Channel/ProgramInfo/actLineNumber
  source: opcua
  name: Program/actLineNumber
  description: Line number of the current NC instruction (starting at 1)
  path: "/Object/Sinumerik/Channel/ProgramInfo/actLineNumber"
  value: "-2"
  timestamp: '2019-04-03T16:13:24.337'
```

Visual Analysis Tool





Actual Workflow



[

```
"Spindle/actSpeed",
"Spindle/driveLoad",
"Axis/X/aaLoad",
"Axis/Y/aaLoad",
"Axis/Z/aaLoad",
"Axis/X/aaTorque",
"Axis/Y/aaTorque",
"Axis/Z/aaTorque",
"Axis/X/aaLeadP",
"Axis/Y/aaLeadP",
"Axis/Z/aaLeadP",
"Axis/X/aaVactB",
"Axis/Y/aaVactB",
"Axis/Z/aaVactB"
```

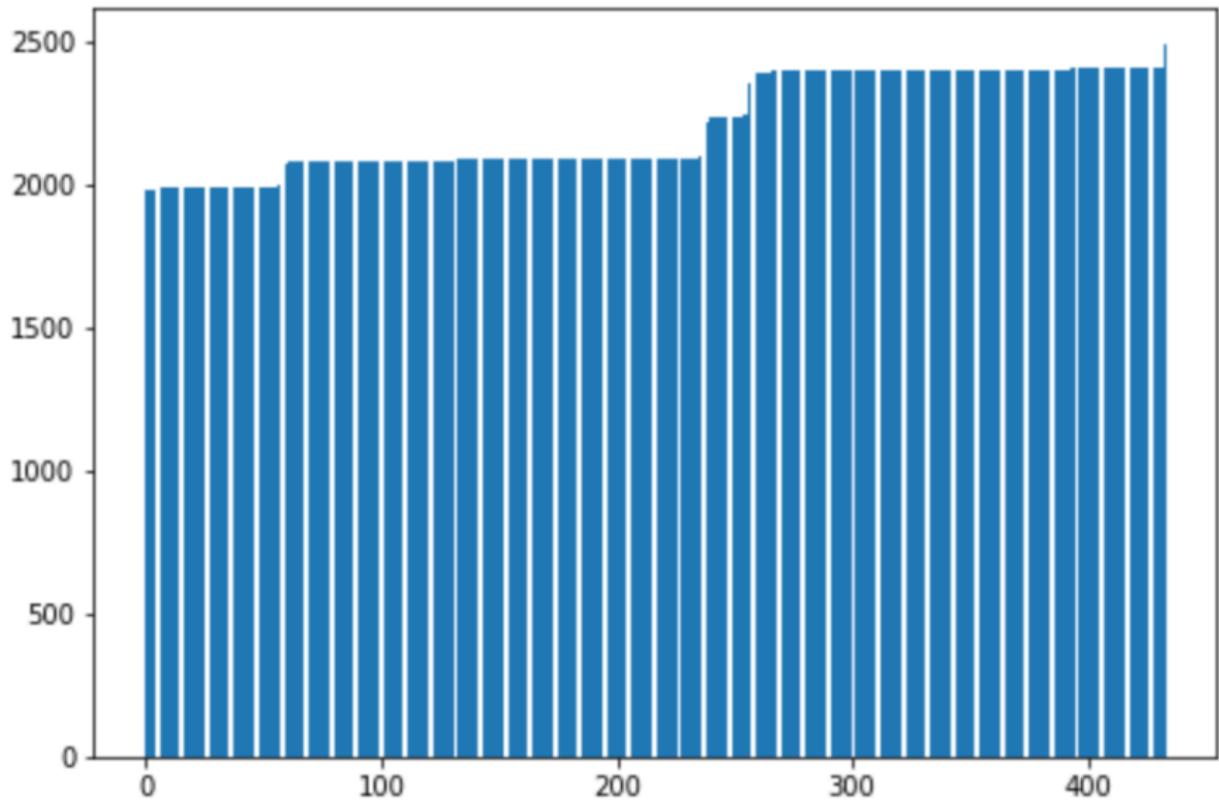
]



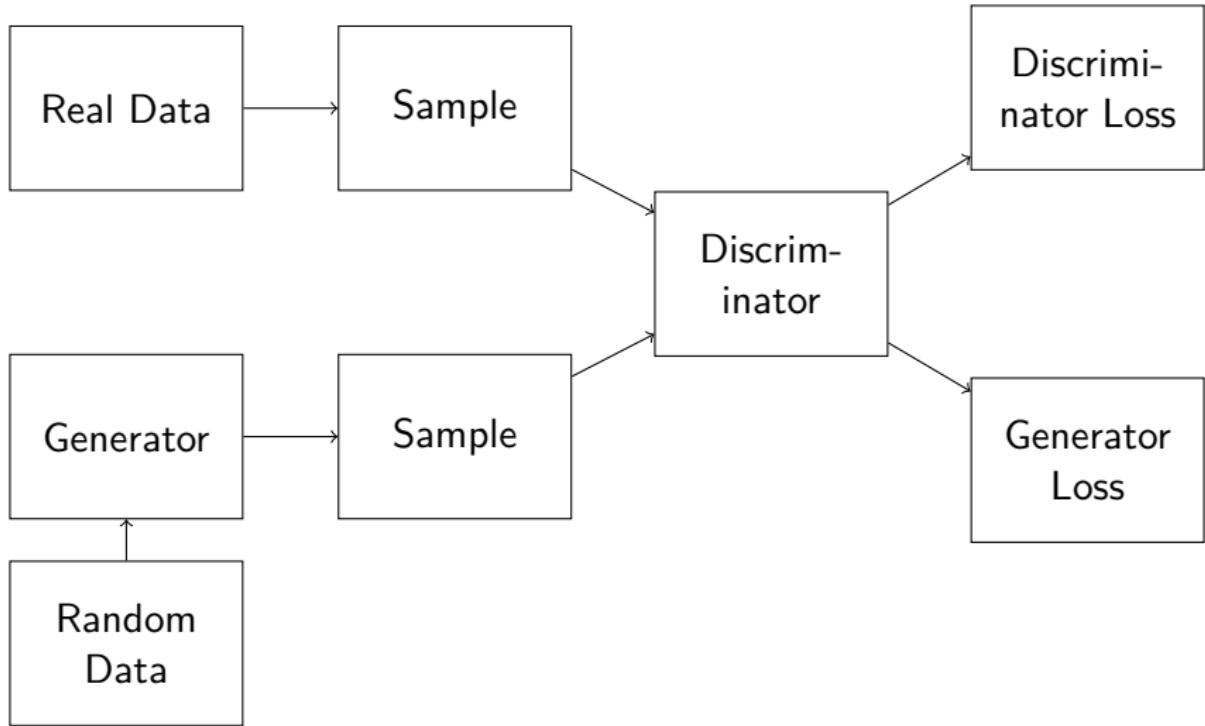
Why Data Resampling?

- Log Data is produced in varying frequency
- Neural Networks require fixed input shape
- Helps with recognizing unusually short or long runs

Data Binning



Model Generation - GAN



Generator:

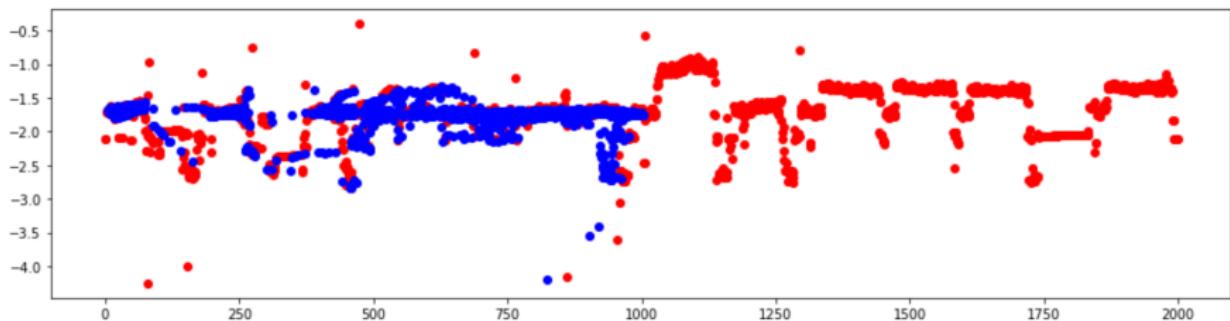
```
def define_generator(latent_dim , n_outputs=2):
    model = Sequential()
    model.add(Dense(1000, activation='relu' , kernel_initializer='he_uniform'
                   , input_dim=latent_dim))
    model.add(LeakyReLU(alpha=relu_alpha))
    model.add(Dense(500, kernel_initializer='he_uniform'))
    model.add(LeakyReLU(alpha=relu_alpha))
    model.add(Dense(250, kernel_initializer='he_uniform'))
    model.add(LeakyReLU(alpha=relu_alpha))
    model.add(Dense(100, kernel_initializer='he_uniform'))
    model.add(LeakyReLU(alpha=relu_alpha))
    model.add(Dense(50, kernel_initializer='he_uniform'))
    model.add(LeakyReLU(alpha=relu_alpha))
    model.add(Dense(n_outputs, activation='linear'))
    return model
```

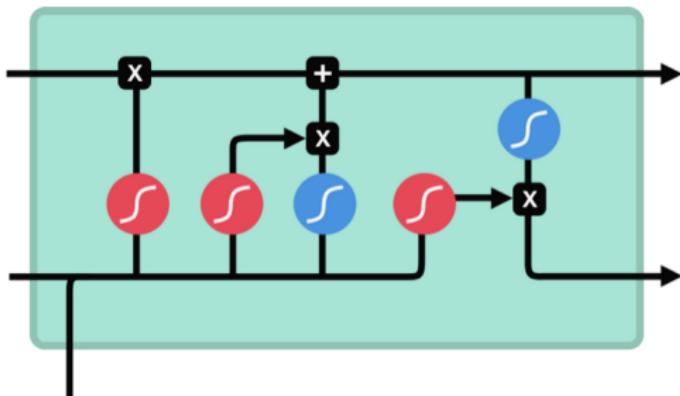
Discriminator:

```
def define_discriminator(n_inputs=2):
    model = Sequential()
    model.add(Dense(1000, kernel_initializer='he_uniform', input_dim=n_inputs))
    model.add(LeakyReLU(alpha=relu_alpha))
    model.add(Dense(500, kernel_initializer='he_uniform'))
    model.add(LeakyReLU(alpha=relu_alpha))
    model.add(Dense(250, kernel_initializer='he_uniform'))
    model.add(LeakyReLU(alpha=relu_alpha))
    model.add(Dense(100, kernel_initializer='he_uniform'))
    model.add(LeakyReLU(alpha=relu_alpha))
    model.add(Dense(50, kernel_initializer='he_uniform'))
    model.add(LeakyReLU(alpha=relu_alpha))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
```

Results:

776000 0.26649999618530273 0.8734999895095825





sigmoid



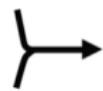
tanh



pointwise
multiplication



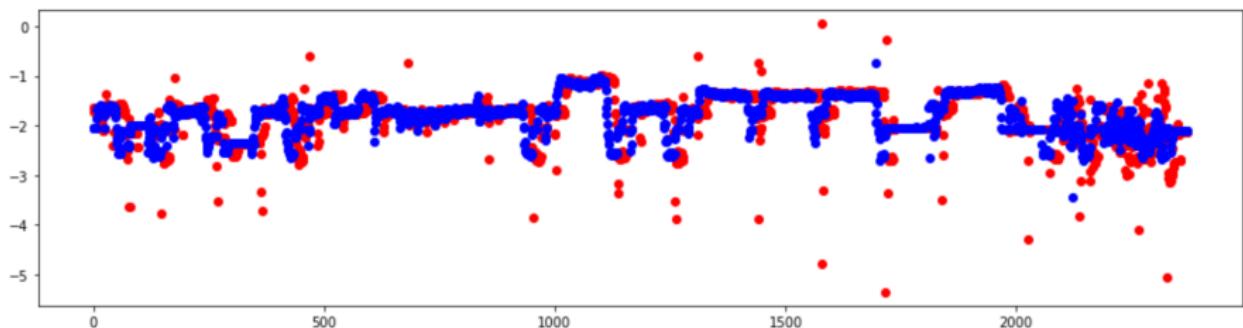
pointwise
addition



vector
concatenation

```
X, y = split_series(raw_seq, n_steps)
n_features = 1
X = X.reshape((X.shape[0], X.shape[1], n_features))
model = Sequential()
model.add(LSTM(50, activation='relu', input_shape=(n_steps, n_features)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
callbacks = [EarlyStopping(monitor='loss', patience=10)]
model.fit(X, y, epochs=100, verbose=0, callbacks=callbacks)
model.save('..../models/lstm/' + file_name)
```

Results:





Challenges:

- Varying number of log entries between YAML files
- Varying number of log entries between metrics within YAML file
- Varying intervals between log entries



Solution:

- Varying number of log entries between YAML files
 - **Make selection of YAML files as templates and match with closest size to generated data**
- Varying number of log entries between metrics within YAML file
- Varying intervals between log entries



Solution:

- Varying number of log entries between YAML files
 - Make selection of YAML files as templates and match with closest size to generated data
- Varying number of log entries between metrics within YAML file
 - **Calculate total duration of template and extract number of entries per metric**
 - **Calculate sample rate based and total duration and desired number of entries**
- Varying intervals between log entries

Solution:

- Varying number of log entries between YAML files
 - Make selection of yaml files as templates and match with closets size to generated data
- Varying number of log entries between metrics within YAML file
 - Calculate total duration of template and extract number of entries per metric
 - Calculate sample rate based and total duration and desired number of entries
- Varying intervals between log entries
 - **Replace both value and timestamp for each log entry**

User Interface

batch4	batch5	batch6	batch7	batch8	batch9	batch10	batch11	batch12	batch14	batch15
Spindle_actSpeed										
Spindle_driveLoad										
Axis_X_aaLoad										
Axis_Y_aaLoad										
Axis_Z_aaLoad										
Axis_X_aaTorque										
Axis_Y_aaTorque										
Axis_Z_aaTorque										
Axis_X_aaLeadP										
Axis_Y_aaLeadP										
Axis_Z_aaLeadP										
Axis_X_aaVactB										
Axis_Y_aaVactB										
Axis_Z_aaVactB										

Generate Batch

User Interface



batch4	batch5	batch6	batch7	batch8	batch9	batch10	batch11	batch12	batch14	batch15
Spindle_actSpeed										
Spindle_driveLoad										
Axis_X_aaLoad										
Axis_Y_aaLoad										
Axis_Z_aaLoad										
Axis_X_aaTorque										
Axis_Y_aaTorque										
Axis_Z_aaTorque										
Axis_X_aaLeadP										
Axis_Y_aaLeadP										
Axis_Z_aaLeadP										
Axis_X_aaVactB										
Axis_Y_aaVactB										
Axis_Z_aaVactB										

Generate Batch

User Interface

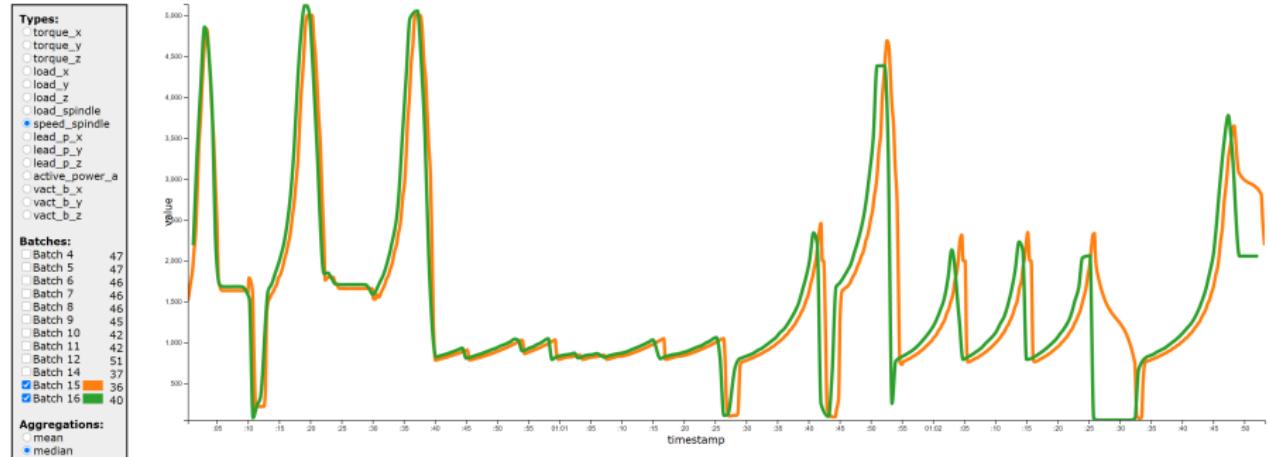
batch4	batch5	batch6	batch7	batch8	batch9	batch10	batch11	batch12	batch14	batch15
Spindle_actSpeed										
Spindle_driveLoad										
Axis_X_aaLoad										
Axis_Y_aaLoad										
Axis_Z_aaLoad										
Axis_X_aaTorque										
Axis_Y_aaTorque										
Axis_Z_aaTorque										
Axis_X_aaLeadP										
Axis_Y_aaLeadP										
Axis_Z_aaLeadP										
Axis_X_aaVectB										
Axis_Y_aaVectB										
Axis_Z_aaVectB										

User Interface

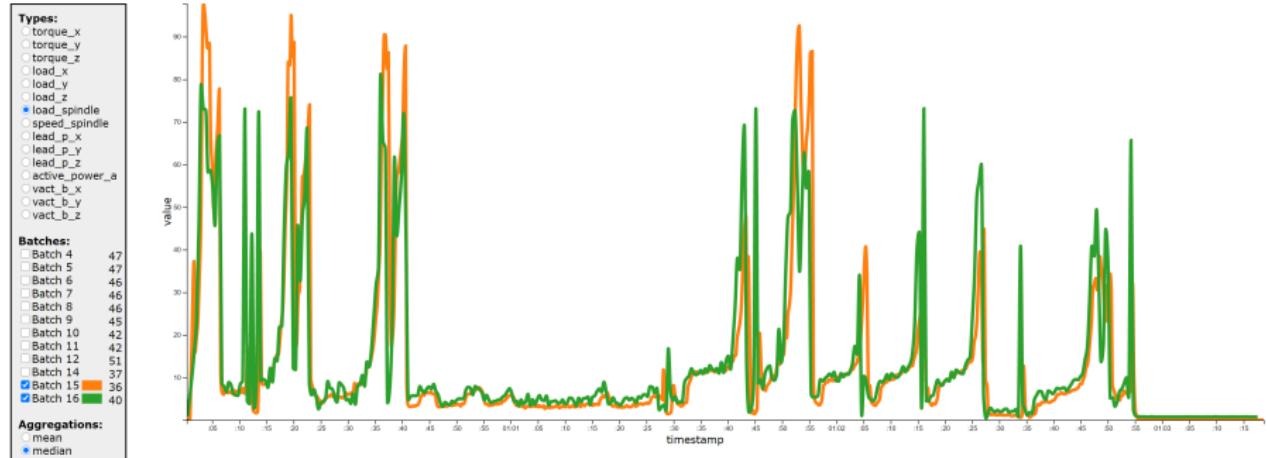
batch4	batch5	batch6	batch7	batch8	batch9	batch10	batch11	batch12	batch14	batch15
Spindle_actSpeed										
Spindle_driveLoad										
Axis_X_aaLoad										
Axis_Y_aaLoad										
Axis_Z_aaLoad										
Axis_X_aaTorque										
Axis_Y_aaTorque										
Axis_Z_aaTorque										
Axis_X_aaLeadP										
Axis_Y_aaLeadP										
Axis_Z_aaLeadP										
Axis_X_aaVactB										
Axis_Y_aaVactB										
Axis_Z_aaVactB										

40/40

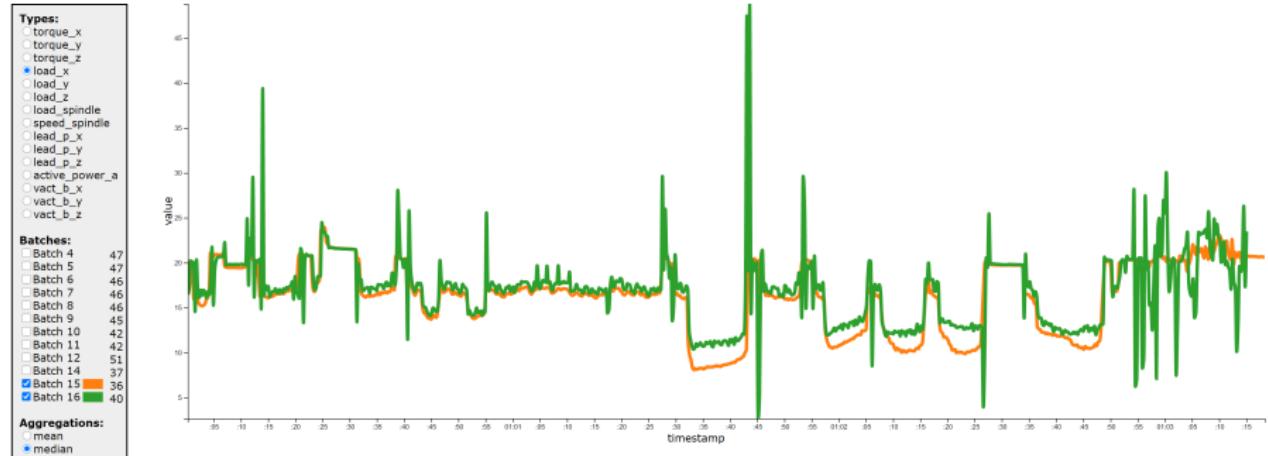
Results



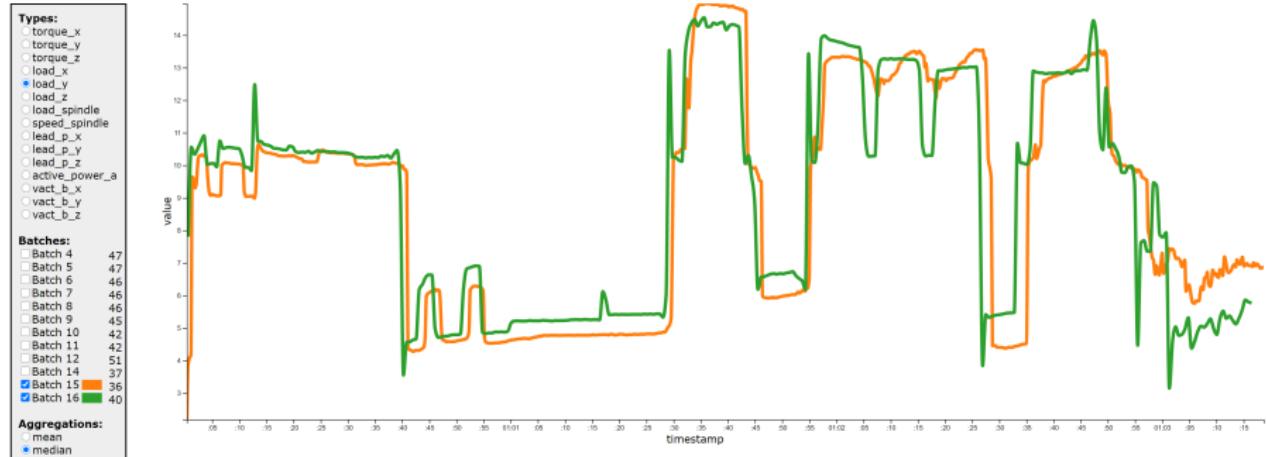
Results



Results



Results





-  Michael Phi - Illustrated Guide to LSTM's and GRU's: A step by step explanation,
<https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>