

```
1 /home/mtnx01/.virtualenvs/Dataspell/bin/python /home/
  mtnx01/.pycharm_helpers/pydev/pydevconsole.py --mode=
    client --host=127.0.0.1 --port=45403
2
3 import sys; print('Python %s on %s' % (sys.version, sys
  .platform))
4 sys.path.extend(['/home/mtnx01/.virtualenvs/Dataspell
  '])
5
6 PyDev console: starting.
7
8 Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.
  0] on linux
9 >>> import matplotlib.pyplot as plt
10 ... import numpy as np
11 ... import pandas as pd
12 ... import scipy.signal as signal
13 ... import tensorflow as tf
14 ... import keras
15 ...
16 ...
17 ... # def normalize(data):
18 ... #     envlength = int(data.shape[0] / 24)
19 ... #     envmax = np.zeros((envlength, data.shape[1
  ]), float)
20 ... #     envmin = np.zeros((envlength, data.shape[1
  ]), float)
21 ... #     for i in range(envlength):
22 ... #         envmax[i, :] = np.ma.max(data[24 * i:24
  * (i + 1), :], axis=0)
23 ... #         envmin[i, :] = np.ma.min(data[24 * i:24
  * (i + 1), :], axis=0)
24 ... #     envamp = envmax - envmin
25 ... #     envoffs = 0.5 * (envmax - envmin) + envmin
26 ... #     signorm = data
27 ... #     for i in range(envlength):
28 ... #         for j in range(24):
29 ... #             signorm[(i * 24 + j), :] = (data[(i
```

```
29 * 24 + j), :] - envoffs[i, :]) / envamp[i, :]
30 ... #     return signorm, envamp, envoffs
31 ...
32 ...
33 ... def normalize(data, train_split):
34 ...     data_mean = data[train_split:][:].mean(axis=0)
35 ...     data_std = data[train_split:][:].std(axis=0)
36 ...     normal_data = (data - data_mean) / data_std
37 ...     return normal_data, data_mean, data_std
38 ...
39 ...
40 ... # def normalize(data):
41 ... #     envlength = int(data.shape[0] / 24)
42 ... #     envmax = np.zeros((envlength, data.shape[1]), np.complex64)
43 ... #     envmin = np.zeros((envlength, data.shape[1]), np.complex64)
44 ... #     for i in range(envlength):
45 ... #         maxmag = np.ma.max(data[24 * i:24 * (i + 1), :], axis=0)
46 ... #         minmag = np.ma.min(data[24 * i:24 * (i + 1), :], axis=0)
47 ... #         maxphase = np.exp(1j*np.pi*(1.0/12.0)*np.argmax(data[24 * i:24 * (i + 1), :], axis=0))
48 ... #         minphase = np.exp(1j*np.pi*(1.0/12.0)*np.argmin(data[24 * i:24 * (i + 1), :], axis=0))
49 ... #         envmax[i, :] = maxmag*maxphase
50 ... #         envmin[i, :] = minmag*minphase
51 ... #     envamp = envmax - envmin
52 ... #     envoffs = 0.5 * (envmax - envmin) + envmin
53 ... #     signorm = data
54 ... #     for i in range(envlength):
55 ... #         for j in range(24):
56 ... #             signorm[(i * 24 + j), :] = (data[(i * 24 + j), :] - envoffs[i, :]) / envamp[i, :]
57 ... #     return signorm, envmax, envmin, envamp,
58 ...     envoffs
```

```
59 ...
60 ... # def denormalize(data, envamp, envoffs):
61 ... #     sigdenorm = data
62 ... #     for i in range(len(envamp)):
63 ... #         for j in range(24):
64 ... #             sigdenorm[(i*24 + j), :] = (data[(i*
24 + j), :]*envamp[i, :] + envoffs[i, :])
65 ...
66 ... def combFilt(xsig, invert=False, feedback=False):
67 ...     Kpls = 25
68 ...     qfilt = 0.9
69 ...     bcoeffs = np.zeros(Kpls)
70 ...     acoeffs = np.zeros(Kpls)
71 ...     if feedback:
72 ...         bcoeffs[0] = 1
73 ...         acoeffs[0] = 1
74 ...         acoeffs[-1] = -1*qfilt
75 ...     else:
76 ...         bcoeffs[0] = 1
77 ...         bcoeffs[-1] = qfilt
78 ...         acoeffs[0] = 1
79 ...
80 ...     if invert:
81 ...         return signal.lfilter(b=acoeffs, a=bcoeffs
, x=xsig, axis=0)
82 ...     else:
83 ...         return signal.lfilter(b=bcoeffs, a=acoeffs
, x=xsig, axis=0)
84 ...
85 ...
86 ... def visualize_loss(history, title):
87 ...     loss = history.history["loss"]
88 ...     val_loss = history.history["val_loss"]
89 ...     epochs = range(len(loss))
90 ...     plt.figure()
91 ...     plt.plot(epochs, loss, "b", label="Training
loss")
92 ...     plt.plot(epochs, val_loss, "r", label="
```

```
92 Validation loss")
93 ...     plt.title(title)
94 ...     plt.xlabel("Epochs")
95 ...     plt.ylabel("Loss")
96 ...     plt.legend()
97 ...     plt.show()
98 ...
99 ...
100 ... def show_plot(plot_data, delta, title):
101 ...     labels = ["History", "True Future", "Model
102 ...     Prediction"]
103 ...     marker = [".-", "rx", "go"]
104 ...     time_steps = list(range(-(plot_data[0].shape[0]
105 ...     ]), 0))
106 ...     if delta:
107 ...         future = delta
108 ...     else:
109 ...         future = 0
110 ...     plt.title(title)
111 ...     for i, val in enumerate(plot_data):
112 ...         if i:
113 ...             plt.plot(future, plot_data[i], marker[
114 ...                 i], markersize=10, label=labels[i])
115 ...         else:
116 ...             plt.plot(time_steps, plot_data[i].
117 ...                 flatten(), marker[i], label=labels[i])
118 ...     plt.legend()
119 ...     plt.xlim([time_steps[0], (future + 5) * 2])
120 ...     plt.xlabel("Time-Step")
121 ...     plt.show()
122 ...     return
123 ...
```

```
121 ... def model_train(dsnp, past, future, learning_rate
122 ... , batch_size, epochs, nlstm, nDeep, train_split, name
123 ... ):
```

```
122 ...     train_data = dsnp[:train_split]::
123 ...     val_data = dsnp[train_split:]::
```

```
124 ...
125 ...     start = past + future
126 ...     end = start + train_split
127 ...
128 ...     x_train = train_data
129 ...     y_train = dsnp[start:end][:]
130 ...     sequence_length = int(past)
131 ...
132 ...     dataset_train = keras.preprocessing.
133 ...         timeseries_dataset_from_array(
134 ...             x_train,
135 ...             y_train,
136 ...             sequence_length=sequence_length,
137 ...             sampling_rate=1,
138 ...             batch_size=batch_size,
139 ...
140 ...             )
141 ...
142 ...     x_end = val_data.shape[0] - past - future
143 ...
144 ...     label_start = train_split + past + future
145 ...
146 ...
147 ...     dataset_val = keras.preprocessing.
148 ...         timeseries_dataset_from_array(
149 ...             x_val,
150 ...             y_val,
151 ...             sequence_length=sequence_length,
152 ...             sampling_rate=1,
153 ...             batch_size=batch_size,
154 ...
155 ...     for batch in dataset_train.take(1):
156 ...         inputs, targets = batch
157 ...
158 ...
159 ...     print("Input shape:", inputs.numpy().shape)
```

```
160 ...     print("Target shape:", targets.numpy().shape)
161 ...     inputs = keras.layers.Input(shape=(inputs.
162 ...         shape[1], inputs.shape[2]))
163 ...     lstm_out = keras.layers.LSTM(nlstm)(inputs)
164 ...     outputs = keras.layers.Dense(8)(lstm_out)
165 ...
166 ...     model = keras.Model(inputs=inputs, outputs=
167 ...         outputs)
168 ...     model.compile(optimizer=tf.keras.optimizers.
169 ...         Nadam(learning_rate=learning_rate), loss=tf.keras.
170 ...         losses.MeanAbsolutePercentageError())
171 ...     model.summary()
172 ...     path_checkpoint = name + "model_checkpoint.
173 ...         weights.h5"
174 ...     es_callback = keras.callbacks.EarlyStopping(
175 ...         monitor="val_loss", min_delta=0, patience=5)
176 ...     modelckpt_callback = keras.callbacks.
177 ...         ModelCheckpoint(
178 ...             monitor="val_loss",
179 ...             filepath=path_checkpoint,
180 ...             verbose=1,
181 ...             save_weights_only=True,
182 ...             save_best_only=True,
183 ...         )
184 ...
185 ...     history = model.fit(
186 ...         dataset_train,
187 ...         epochs=epochs,
188 ...         validation_data=dataset_val,
189 ...         callbacks=[es_callback, modelckpt_callback
190 ... ],
191 ...     )
192 ...
193 ...     visualize_loss(history, "Training and
194 ...         Validation Loss")
195 ...
196 ...     model.save(name + "model.keras")
197 ...
198 ...
```

```
189 ...         return model, history, modelckpt_callback
190 ...
191 ...
192 ... def grid_search_hyperparameters(dsnp, train_split
193     , param_grid, name):
194 ...     best_loss = float('inf')
195 ...     best_params = None
196 ...
197 ...     for past in param_grid['past']:
198 ...         for future in param_grid['future']:
199 ...             for learning_rate in param_grid['
200 ...                 learning_rate']:
201 ...                 for batch_size in param_grid['
202 ...                     batch_size']:
203 ...                     for epochs in param_grid['
204 ...                         epochs']:
205 ...                         for nlstm in param_grid['
206 ...                             nlstm']:
207 ...                             print(f"Testing
208 ...                                 hyperparameters: past={past}, future={future},
209 ...                                 learning_rate={learning_rate}, batch_size={batch_size}
210 ...                                 }, epochs={epochs}, nlstm={nlstm}, nDeep={param_grid['
211 ...                                     nDeep'][0]}")
212 ...
213 ...         # Train model
214 ...         model, history, _ =
215 ...             model_train(dsnp, past, future, learning_rate,
216 ...                         batch_size, epochs, nlstm, nDeep, train_split, name)
217 ...
218 ...         # Get validation loss
219 ...         val_loss = history.
220 ...             history['val_loss'][-1]
221 ...
222 ...         # Check if current
223 ...         combination is better
224 ...         if val_loss <
225 ...             best_loss:
226 ...             best_loss =
```

```
212 val_loss
213 ...
214 ...
215 ...
216 ...
217 ...
218 ...
219 ...
220 ...
221 ...
222 ...
223 ...
224 ...
225 ...
226 ...
227 ...
228 ...
229 ...
230 >>> #Database connection
231 ...
232 ...
233 ...
234 ...
235 ...
236 ...
237 ...
238 ...
239 ...
240 ...
241 ...  
best_params = {
    'past': past,
    'future': future,
    'learning_rate': learning_rate,
    'batch_size': batch_size,
    'epochs': epochs,
    'nlstm': nlstm,
    'nDeep': nDeep
}  
print("Grid search complete.")  
print("Best hyperparameters:")  
print(best_params)  
print("Best validation loss:", best_loss)  
return best_params  
# declares the variables and defines them for the  
# server connections, along with the table names that  
# are going to be assigned  
SERVER_NAME = 'tcp:tecafs.database.windows.net,  
1433'  
DATABASE_NAME = 'TecafsSqlDatabase'  
TABLE_NAME = 'clean_data'
```

```
242 ...
243 ... #####
244 ...
245 ... # makes the connection to the database with the
      connection string; has the driver, server name,
      database name, id, and password
246 ... connection_string = f"""
247 ...     DRIVER={{ODBC Driver 18 for SQL Server}};
248 ...     SERVER={{SERVER_NAME}};
249 ...     DATABASE={{DATABASE_NAME}};
250 ...     Uid={{'tecafs2023'}};
251 ...     Pwd={{'Capstone50'}};
252 ... """
253 ...
254 ...
255 ...
256 ... #####
257 ...
258 ... # attempted connection string, didn't work though
259 ...
260 ... #connection_string1 = pypyodbc.connect("Driver={{ODBC Driver 18 for SQL Server}};Server=tcp:tecafs.database.windows.net,1433;Database=TecafsSqlDatabase;Uid=tecafs2023;Pwd={{Capstone50}};")
261 ...
262 ... #####
263 ...
264 ...
265 ... connection_url = URL.create('mssql+pyodbc', query
      ={'odbc_connect': connection_string})
266 ... engine = create_engine(connection_url, module=
      pypyodbc)
267 ...
268 ...
269 ... metadata = MetaData()
270 ... metadata.reflect(bind=engine)
271 ...
272 ... # Print the names of tables
```

```
273 ... # table_list = {}
274 ... # print("Tables in the database:")
275 ... # for table in metadata.tables.values():
276 ... #     print(table.name)
277 ... #     if table.name != "clean_data":
278 ... #         table_list[table.name] = pd.
279 ... #             read_sql_table(table.name, engine)
280 ... ...
281 ... #hist_data = pd.read_sql_table("hist_data", engine
282 ... , index_col="Time", parse_dates={"Time": '%Y-%m-%d %H
283 ... :%M:%S'})
284 ... hist_data = pd.read_sql_table("hist_data", engine)
285 ... hist_data.index = pd.to_datetime(hist_data.pop("Time"),
286 ... format='%Y-%m-%d %H:%M:%S', unit='s')
287 ... feature_titles = hist_data.columns.tolist()
288 ... #timestamp_range = pd.Series(np.arange(hist_data.
289 ... index.min(), hist_data.index.max(), step=np.
290 ... timedelta64(1, "h")))
291 ... region_titles = ['COAST', 'EAST', 'FWEST', 'NORTH
292 ... ', 'NCENT', 'SOUTH', 'SCENT', 'WEST', 'ERCOT']
293 ... regions = {title: hist_data[hist_data['Location
294 ... '] == title] for title in region_titles}
295 ... selected_regions = [region_titles[i] for i in [0,
296 ... 1, 3, 7]]
297 ... selected_features = hist_data.select_dtypes(
298 ... include='float64').columns.to_numpy(dtype=str).tolist
299 ... ()
300 ... ...
301 ... selected_data = {i: pd.DataFrame(regions[i][:]
302 ... selected_features], index=regions[i].index, columns=
303 ... selected_features) for i in selected_regions}
304 ... ...
305 ... 2024-04-18 01:23:33.390867: I tensorflow/core/util/
306 ... port.cc:113] oneDNN custom operations are on. You may
307 ... see slightly different numerical results due to
308 ... floating-point round-off errors from different
309 ... computation orders. To turn them off, set the
310 ... environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
```

```
293 2024-04-18 01:23:33.420092: I tensorflow/core/platform  
/cpu_feature_guard.cc:210] This TensorFlow binary is  
optimized to use available CPU instructions in  
performance-critical operations.  
294 To enable the following instructions: AVX2 AVX_VNNI  
FMA, in other operations, rebuild TensorFlow with the  
appropriate compiler flags.  
295 2024-04-18 01:23:33.820101: W tensorflow/compiler/  
tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning:  
Could not find TensorRT  
296 >>> # Data preprocessing  
297 ... #print("Num GPUs Available: ", len(tf.config.  
list_physical_devices('GPU')))  
298 ... RegionDict = {}  
299 ... #reg_data = selected_data  
300 ... load_model = False  
301 ... for i in selected_regions:  
302 ...     #power data  
303 ...     reg_data = selected_data[i]  
304 ...     reg_data = reg_data.sort_index(axis=0)  
305 ...     reg_columns = reg_data.columns  
306 ...     reg_timestamps = reg_data.index  
307 ...     split_fraction = 0.75  
308 ...     train_split = int(split_fraction * reg_data.  
shape[0])  
309 ...     reg_data = selected_data[i].interpolate(method  
='time', axis=0)  
310 ...     reg_data, reg_mean, reg_std = normalize(  
reg_data, train_split)  
311 ...     reg_data = pd.DataFrame(combFilt(reg_data),  
index=reg_timestamps, columns=reg_columns)  
312 ...     #envelope_data = pd.DataFrame(data=dict({'  
amplitude': reg_amp, 'offset': reg_off}), index=  
regions[i].index, columns=selected_features)  
313 ...     ts_hr = reg_data  
314 ...     #general data  
315 ...     array = ts_hr  
316 ...     name = i
```

```
317 ...     past = 720
318 ...     future = 336
319 ...     nDeep = array.shape[1]
320 ...
321 ...     # param_grid = {
322 ...         #     'past': [720],
323 ...         #     'future': [336],
324 ...         #     'learning_rate': [0.001, 0.0025, 0.005],
325 ...         #     'batch_size': [24, 48, 72, 84, 96],
326 ...         #     'epochs': [10],
327 ...         #     'nlstm': [24, 48, 72, 84, 96],
328 ...         #     'nDeep': [nDeep]
329 ...     # }
330 ...
331 ...     #best_params = grid_search_hyperparameters(
332 ...         selected_data[i], train_split, param_grid, i)
333 ...         #(array, past, future, learning_rate,
334 ...         batch_size, epochs, nlstm, nDeep, train_split)
335 ...     if load_model:
336 ...         model = keras.saving.load_model(name + "
337 ...             model.keras")
338 ...     else:
339 ...         model, history, chkpt_callback =
340 ...             model_train(array, past, future, 0.0025, 24, 10, 42, 8
341 ...             , train_split, name)
342 ...
343 2024-04-18 01:23:45.687569: [I external/local_xla/xla/
344 ... stream_executor/cuda/cuda_executor.cc:984] could not
345 ... open file to read NUMA node: /sys/bus/pci/devices/0000
```

```
343 :01:00.0/numa_node
344 Your kernel may have been built without NUMA support.
345 2024-04-18 01:23:45.700512: W tensorflow/core/
    common_runtime/gpu/gpu_device.cc:2251] Cannot dlopen
    some GPU libraries. Please make sure the missing
    libraries mentioned above are installed properly if
    you would like to use GPU. Follow the guide at https
    ://www.tensorflow.org/install/gpu for how to download
    and setup the required libraries for your platform.
346 Skipping registering GPU devices...
347 2024-04-18 01:23:45.889445: W tensorflow/core/
    framework/local_rendezvous.cc:404] Local rendezvous is
    aborting with status: OUT_OF_RANGE: End of sequence
348 Input shape: (24, 720, 8)
349 Target shape: (24, 8)
350 Model: "functional_1"
351
352 | Layer (type)           | Param # | Output Shape
353 |
354 | input_layer (InputLayer)| 0       | (None, 720, 8)
355 |
356 | lstm (LSTM)            | 8,568   | (None, 42)
357 |
358 | dense (Dense)          | 344     | (None, 8)
359 |
360 Total params: 8,912 (34.81 KB)
361 Trainable params: 8,912 (34.81 KB)
362 Non-trainable params: 0 (0.00 B)
363 Epoch 1/10
```

```
364 1335/1335 ━━━━━━ 0s 79ms/step - loss:  
133.9010  
365 Epoch 1: val_loss improved from inf to 88.91463,  
saving model to COASTmodel_checkpoint.weights.h5  
366 1335/1335 ━━━━━━ 116s 86ms/step - loss:  
133.8883 - val_loss: 88.9146  
367 Epoch 2/10  
368 1335/1335 ━━━━━━ 0s 79ms/step - loss:  
113.4131  
369 Epoch 2: val_loss improved from 88.91463 to 84.50244,  
saving model to COASTmodel_checkpoint.weights.h5  
370 1335/1335 ━━━━━━ 115s 86ms/step - loss:  
113.4094 - val_loss: 84.5024  
371 Epoch 3/10  
372 1335/1335 ━━━━━━ 0s 80ms/step - loss:  
106.7273  
373 Epoch 3: val_loss did not improve from 84.50244  
374 1335/1335 ━━━━━━ 115s 86ms/step - loss:  
106.7222 - val_loss: 86.2183  
375 Epoch 4/10  
376 1335/1335 ━━━━━━ 0s 80ms/step - loss: 99  
.8943  
377 Epoch 4: val_loss did not improve from 84.50244  
378 1335/1335 ━━━━━━ 116s 87ms/step - loss:  
99.8903 - val_loss: 84.8856  
379 Epoch 5/10  
380 1335/1335 ━━━━━━ 0s 79ms/step - loss:  
109.5586  
381 Epoch 5: val_loss improved from 84.50244 to 83.37206,  
saving model to COASTmodel_checkpoint.weights.h5  
382 1335/1335 ━━━━━━ 115s 86ms/step - loss:  
109.5526 - val_loss: 83.3721  
383 Epoch 6/10  
384 1335/1335 ━━━━━━ 0s 79ms/step - loss: 97  
.9475  
385 Epoch 6: val_loss improved from 83.37206 to 82.53905,  
saving model to COASTmodel_checkpoint.weights.h5  
386 1335/1335 ━━━━━━ 114s 86ms/step - loss:
```

```
386 97.9446 - val_loss: 82.5390
387 Epoch 7/10
388 1334/1335 ━━━━━━━━ 0s 79ms/step - loss:
106.4034
389 Epoch 7: val_loss improved from 82.53905 to 81.70446,
saving model to COASTmodel_checkpoint.weights.h5
390 1335/1335 ━━━━━━━━ 115s 86ms/step - loss:
106.3896 - val_loss: 81.7045
391 Epoch 8/10
392 1335/1335 ━━━━━━━━ 0s 79ms/step - loss: 89
.0851
393 Epoch 8: val_loss did not improve from 81.70446
394 1335/1335 ━━━━━━━━ 115s 86ms/step - loss:
89.0836 - val_loss: 82.0520
395 Epoch 9/10
396 1335/1335 ━━━━━━━━ 0s 79ms/step - loss: 91
.5194
397 Epoch 9: val_loss did not improve from 81.70446
398 1335/1335 ━━━━━━━━ 115s 86ms/step - loss:
91.5176 - val_loss: 84.0345
399 Epoch 10/10
400 1335/1335 ━━━━━━━━ 0s 80ms/step - loss: 91
.2400
401 Epoch 10: val_loss did not improve from 81.70446
402 1335/1335 ━━━━━━━━ 115s 86ms/step - loss:
91.2387 - val_loss: 83.1393
403 2024-04-18 01:42:56.567634: W tensorflow/core/
framework/local_rendezvous.cc:404] Local rendezvous is
aborting with status: OUT_OF_RANGE: End of sequence
404 Input shape: (24, 720, 8)
405 Target shape: (24, 8)
406 Model: "functional_3"
407 ━━━━━━━━
408 | Layer (type) | Param # | Output Shape |
409 ━━━━━━━━
```

```
410     input_layer_1 (InputLayer)      | (None, 720, 8
411     )                                | 0
412     lstm_1 (LSTM)                   | (None, 42
413     )                                | 8,568
414     dense_1 (Dense)                | (None, 8
415     )                                | 344
416 Total params: 8,912 (34.81 KB)
417 Trainable params: 8,912 (34.81 KB)
418 Non-trainable params: 0 (0.00 B)
419 Epoch 1/10
420 1299/1300 ━━━━━━━━ 0s 79ms/step - loss:
421          108.1074
422 Epoch 1: val_loss improved from inf to 100.79239,
423           saving model to EASTmodel_checkpoint.weights.h5
424 1300/1300 ━━━━━━━━ 113s 86ms/step - loss:
425          108.0968 - val_loss: 100.7924
426 Epoch 2/10
427 1299/1300 ━━━━━━━━ 0s 79ms/step - loss: 88
428          .0663
429 Epoch 2: val_loss improved from 100.79239 to 89.80677
430           , saving model to EASTmodel_checkpoint.weights.h5
431 1300/1300 ━━━━━━━━ 112s 86ms/step - loss:
432          88.0651 - val_loss: 89.8068
433 Epoch 3/10
434 1299/1300 ━━━━━━━━ 0s 79ms/step - loss: 84
435          .0612
436 Epoch 3: val_loss improved from 89.80677 to 84.67481,
437           saving model to EASTmodel_checkpoint.weights.h5
438 1300/1300 ━━━━━━━━ 112s 86ms/step - loss:
439          84.0605 - val_loss: 84.6748
440 Epoch 4/10
441 1299/1300 ━━━━━━━━ 0s 79ms/step - loss: 82
```

```
432 .2358
433 Epoch 4: val_loss improved from 84.67481 to 83.60118,
    saving model to EASTmodel_checkpoint.weights.h5
434 1300/1300 ━━━━━━━━ 112s 86ms/step - loss:
    82.2357 - val_loss: 83.6012
435 Epoch 5/10
436 1299/1300 ━━━━━━━━ 0s 81ms/step - loss: 81
    .0206
437 Epoch 5: val_loss improved from 83.60118 to 82.94233,
    saving model to EASTmodel_checkpoint.weights.h5
438 1300/1300 ━━━━━━━━ 114s 88ms/step - loss:
    81.0206 - val_loss: 82.9423
439 Epoch 6/10
440 1299/1300 ━━━━━━━━ 0s 78ms/step - loss: 80
    .3505
441 Epoch 6: val_loss improved from 82.94233 to 81.45472,
    saving model to EASTmodel_checkpoint.weights.h5
442 1300/1300 ━━━━━━━━ 110s 84ms/step - loss:
    80.3505 - val_loss: 81.4547
443 Epoch 7/10
444 1299/1300 ━━━━━━━━ 0s 77ms/step - loss: 79
    .8497
445 Epoch 7: val_loss improved from 81.45472 to 80.39355,
    saving model to EASTmodel_checkpoint.weights.h5
446 1300/1300 ━━━━━━━━ 109s 84ms/step - loss:
    79.8497 - val_loss: 80.3936
447 Epoch 8/10
448 1299/1300 ━━━━━━━━ 0s 77ms/step - loss: 79
    .7573
449 Epoch 8: val_loss did not improve from 80.39355
450 1300/1300 ━━━━━━━━ 109s 83ms/step - loss:
    79.7576 - val_loss: 81.6954
451 Epoch 9/10
452 1299/1300 ━━━━━━━━ 0s 77ms/step - loss: 79
    .4942
453 Epoch 9: val_loss did not improve from 80.39355
454 1300/1300 ━━━━━━━━ 109s 84ms/step - loss:
    79.4939 - val_loss: 80.9656
```

```
455 Epoch 10/10
456 1299/1300 ━━━━━━━━ 0s 77ms/step - loss: 79
.3576
457 Epoch 10: val_loss did not improve from 80.39355
458 1300/1300 ━━━━━━━━ 109s 84ms/step - loss:
79.3574 - val_loss: 80.7906
459 2024-04-18 02:01:24.579706: W tensorflow/core/
framework/local_rendezvous.cc:404] Local rendezvous is
aborting with status: OUT_OF_RANGE: End of sequence
460 Input shape: (24, 720, 8)
461 Target shape: (24, 8)
462 Model: "functional_5"
463 ━━━━━━━━
464 | Layer (type) | Param # | Output Shape
465 ━━━━━━━━
466 | input_layer_2 (InputLayer) | 0 | (None, 720, 8)
467 ━━━━━━━━
468 | lstm_2 (LSTM) | 8,568 | (None, 42)
469 ━━━━━━━━
470 | dense_2 (Dense) | 344 | (None, 8)
471 ━━━━━━━━
472 Total params: 8,912 (34.81 KB)
473 Trainable params: 8,912 (34.81 KB)
474 Non-trainable params: 0 (0.00 B)
475 Epoch 1/10
476 1335/1335 ━━━━━━━━ 0s 71ms/step - loss:
136.4240
477 Epoch 1: val_loss improved from inf to 120.79106,
saving model to NORTHmodel_checkpoint.weights.h5
```

```
478 1335/1335 ━━━━━━━━ 104s 78ms/step - loss:  
    136.4159 - val_loss: 120.7911  
479 Epoch 2/10  
480 1335/1335 ━━━━━━━━ 0s 70ms/step - loss:  
    105.6296  
481 Epoch 2: val_loss improved from 120.79106 to 109.65707  
    , saving model to NORTHmodel_checkpoint.weights.h5  
482 1335/1335 ━━━━━━━━ 104s 78ms/step - loss:  
    105.6283 - val_loss: 109.6571  
483 Epoch 3/10  
484 1335/1335 ━━━━━━━━ 0s 71ms/step - loss: 96  
    .5191  
485 Epoch 3: val_loss improved from 109.65707 to 93.79003  
    , saving model to NORTHmodel_checkpoint.weights.h5  
486 1335/1335 ━━━━━━━━ 103s 77ms/step - loss:  
    96.5179 - val_loss: 93.7900  
487 Epoch 4/10  
488 1335/1335 ━━━━━━━━ 0s 70ms/step - loss: 88  
    .3486  
489 Epoch 4: val_loss improved from 93.79003 to 93.15907,  
    saving model to NORTHmodel_checkpoint.weights.h5  
490 1335/1335 ━━━━━━━━ 103s 77ms/step - loss:  
    88.3485 - val_loss: 93.1591  
491 Epoch 5/10  
492 1335/1335 ━━━━━━━━ 0s 75ms/step - loss: 88  
    .8285  
493 Epoch 5: val_loss improved from 93.15907 to 87.53237,  
    saving model to NORTHmodel_checkpoint.weights.h5  
494 1335/1335 ━━━━━━━━ 109s 82ms/step - loss:  
    88.8283 - val_loss: 87.5324  
495 Epoch 6/10  
496 1335/1335 ━━━━━━━━ 0s 74ms/step - loss: 85  
    .6670  
497 Epoch 6: val_loss did not improve from 87.53237  
498 1335/1335 ━━━━━━━━ 107s 80ms/step - loss:  
    85.6668 - val_loss: 89.2420  
499 Epoch 7/10  
500 1335/1335 ━━━━━━━━ 0s 70ms/step - loss: 86
```

```
500 .2390
501 Epoch 7: val_loss improved from 87.53237 to 86.30577,
      saving model to NORTHmodel_checkpoint.weights.h5
502 1335/1335 ━━━━━━━━ 103s 77ms/step - loss:
      86.2386 - val_loss: 86.3058
503 Epoch 8/10
504 1335/1335 ━━━━━━━━ 0s 70ms/step - loss: 84
      .4622
505 Epoch 8: val_loss did not improve from 86.30577
506 1335/1335 ━━━━━━━━ 103s 77ms/step - loss:
      84.4621 - val_loss: 86.5642
507 Epoch 9/10
508 1335/1335 ━━━━━━━━ 0s 72ms/step - loss: 83
      .9990
509 Epoch 9: val_loss did not improve from 86.30577
510 1335/1335 ━━━━━━━━ 106s 79ms/step - loss:
      83.9987 - val_loss: 89.0649
511 Epoch 10/10
512 1335/1335 ━━━━━━━━ 0s 73ms/step - loss: 84
      .0942
513 Epoch 10: val_loss improved from 86.30577 to 84.69683
      , saving model to NORTHmodel_checkpoint.weights.h5
514 1335/1335 ━━━━━━━━ 107s 80ms/step - loss:
      84.0940 - val_loss: 84.6968
515 2024-04-18 02:18:53.385056: W tensorflow/core/
      framework/local_rendezvous.cc:404] Local rendezvous is
      aborting with status: OUT_OF_RANGE: End of sequence
516 Input shape: (24, 720, 8)
517 Target shape: (24, 8)
518 Model: "functional_7"
519 ━━━━━━━━
520 | Layer (type) | Param # | Output Shape |
521 ━━━━━━━━
522 | input_layer_3 (InputLayer) | 0 | (None, 720, 8)
```

```
523
524     lstm_3 (LSTM)
525
526     dense_3 (Dense)
527
528 Total params: 8,912 (34.81 KB)
529 Trainable params: 8,912 (34.81 KB)
530 Non-trainable params: 0 (0.00 B)
531 Epoch 1/10
532 1320/1320 ━━━━━━━━ 0s 82ms/step - loss: 101.2363
533 Epoch 1: val_loss improved from inf to 97.33797,
      saving model to WESTmodel_checkpoint.weights.h5
534 1320/1320 ━━━━━━━━ 118s 89ms/step - loss: 101.2300 - val_loss: 97.3380
535 Epoch 2/10
536 1320/1320 ━━━━━━━━ 0s 79ms/step - loss: 85.3369
537 Epoch 2: val_loss did not improve from 97.33797
538 1320/1320 ━━━━━━━━ 114s 86ms/step - loss: 85.3348 - val_loss: 97.4554
539 Epoch 3/10
540 1320/1320 ━━━━━━━━ 0s 80ms/step - loss: 82.1245
541 Epoch 3: val_loss improved from 97.33797 to 90.46409,
      saving model to WESTmodel_checkpoint.weights.h5
542 1320/1320 ━━━━━━━━ 114s 87ms/step - loss: 82.1233 - val_loss: 90.4641
543 Epoch 4/10
544 1320/1320 ━━━━━━━━ 0s 80ms/step - loss: 80.8834
545 Epoch 4: val_loss improved from 90.46409 to 90.07835,
      saving model to WESTmodel_checkpoint.weights.h5
```

```
546 1320/1320 ━━━━━━━━ 115s 87ms/step - loss:  
    80.8823 - val_loss: 90.0784  
547 Epoch 5/10  
548 1320/1320 ━━━━━━━━ 0s 80ms/step - loss: 80  
    .1025  
549 Epoch 5: val_loss did not improve from 90.07835  
550 1320/1320 ━━━━━━━━ 115s 87ms/step - loss:  
    80.1013 - val_loss: 90.1006  
551 Epoch 6/10  
552 1320/1320 ━━━━━━━━ 0s 81ms/step - loss: 79  
    .7480  
553 Epoch 6: val_loss improved from 90.07835 to 85.63578,  
    saving model to WESTmodel_checkpoint.weights.h5  
554 1320/1320 ━━━━━━━━ 143s 88ms/step - loss:  
    79.7472 - val_loss: 85.6358  
555 Epoch 7/10  
556 1320/1320 ━━━━━━━━ 0s 80ms/step - loss: 78  
    .6742  
557 Epoch 7: val_loss improved from 85.63578 to 85.63471,  
    saving model to WESTmodel_checkpoint.weights.h5  
558 1320/1320 ━━━━━━━━ 115s 87ms/step - loss:  
    78.6736 - val_loss: 85.6347  
559 Epoch 8/10  
560 1320/1320 ━━━━━━━━ 0s 80ms/step - loss: 78  
    .0778  
561 Epoch 8: val_loss improved from 85.63471 to 84.48029,  
    saving model to WESTmodel_checkpoint.weights.h5  
562 1320/1320 ━━━━━━━━ 116s 88ms/step - loss:  
    78.0772 - val_loss: 84.4803  
563 Epoch 9/10  
564 1320/1320 ━━━━━━━━ 0s 81ms/step - loss: 77  
    .6813  
565 Epoch 9: val_loss improved from 84.48029 to 81.51317,  
    saving model to WESTmodel_checkpoint.weights.h5  
566 1320/1320 ━━━━━━━━ 116s 88ms/step - loss:  
    77.6809 - val_loss: 81.5132  
567 Epoch 10/10  
568 1320/1320 ━━━━━━━━ 0s 81ms/step - loss: 77
```

```
568 .1003
569 Epoch 10: val_loss improved from 81.51317 to 81.23911
, saving model to WESTmodel_checkpoint.weights.h5
570 1320/1320 ━━━━━━━━ 116s 88ms/step - loss:
77.1003 - val_loss: 81.2391
571
```