

Maschinelles Lernen

Meilenstein 3

Inhalt

- Featurevektor
- Untersuchte Stationen
- Versuchsaufbau
- Training
- Ergebnisse

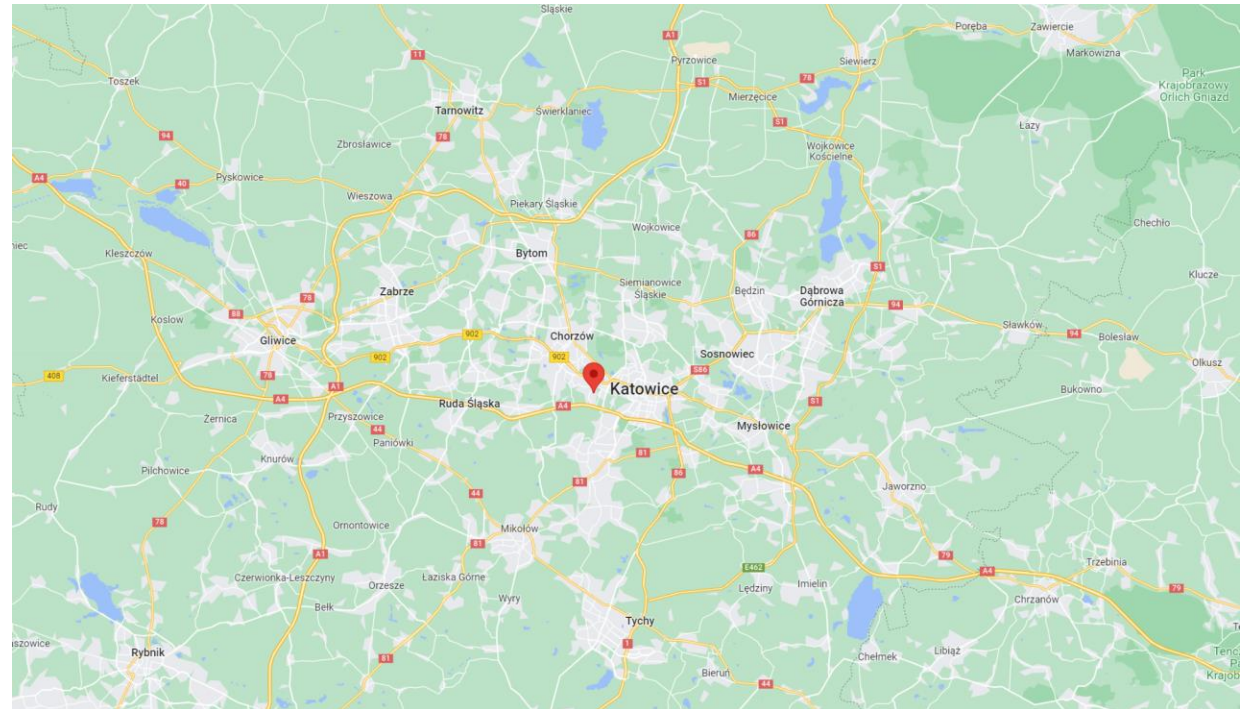
Featurevektor

- Temperatur (°C)
- Luftfeuchtigkeit (%)
- Windgeschwindigkeit (m/s)
- Niederschlag (l/m)
- Windrichtung (sin & cos)
- Tag (sin & cos)
- Jahr (sin & cos)
- PM₁₀

➤ 11 Features

Untersuchte Station

- Station 814
 - PM10 Sensor
 - Katowice
 - Stationärer Container
10 im begrünten urbanen Gebiet



CNN - Modell

Versuchsaufbau

```
model = tf.keras.models.Sequential()
# CNN
model.add(tf.keras.layers.Normalization())
model.add(tf.keras.layers.Lambda(lambda x: x[:, -CONV_WIDTH:, :]))
model.add(tf.keras.layers.Conv1D(128, activation="relu", kernel_size=(CONV_WIDTH), padding="same"))
model.add(tf.keras.layers.MaxPooling1D())
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Conv1D(256, activation="relu", kernel_size=(CONV_WIDTH), padding="same"))
model.add(tf.keras.layers.MaxPooling1D())
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Conv1D(256, activation="relu", kernel_size=(CONV_WIDTH), padding="same"))
model.add(tf.keras.layers.Reshape((-1, 256)))

# LSTM
model.add(tf.keras.layers.LSTM(64, return_sequences=True))
model.add(tf.keras.layers.LSTM(64, return_sequences=False))

# Output
model.add(tf.keras.layers.Dense(512))
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(OUT_STEPS * num_features, kernel_initializer=tf.initializers.zeros()))
model.add(tf.keras.layers.Reshape([OUT_STEPS, num_features]))
```

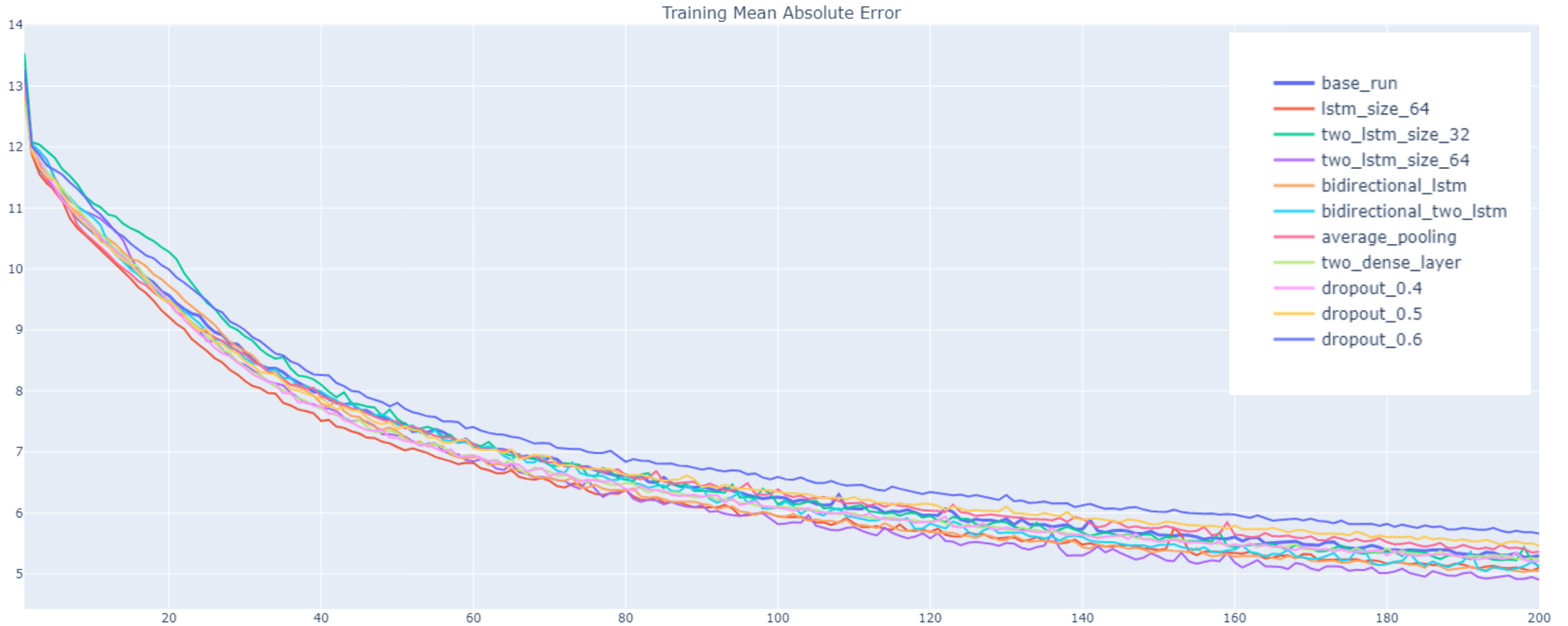
```
CONV_WIDTH = 6
num_features = 1
OUT_STEPS = 24
```

Versuchsaufbau

- Modell-Variationen
 - zwei LSTMs verwenden
 - Größe der LSTMs variieren
 - LSTM in BiLSTM ändern
 - MaxPooling -> AveragePooling
 - Kernel Size (Conv) ändern (24, 12, 6)
 - Fully Connected Layer hinzufügen
 - DenseLayer hinzufügen
 - Dropout erhöhen (0.3, 0.4, 0.6)
- Daten-Variationen
 - Mehr History (7-1) (14-1)
 - Loss variieren (MSE, huber)

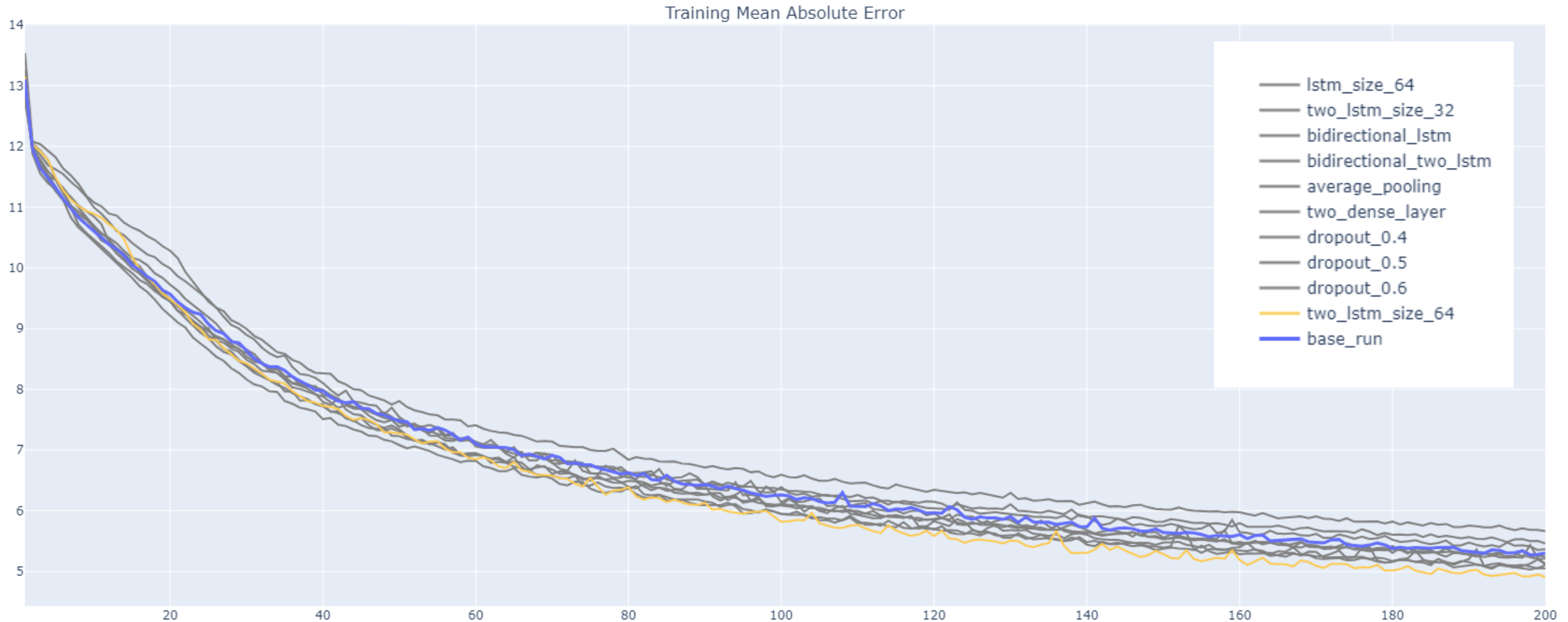
Ergebnisse

Versuchsergebnisse – Training MAE aller Modelle



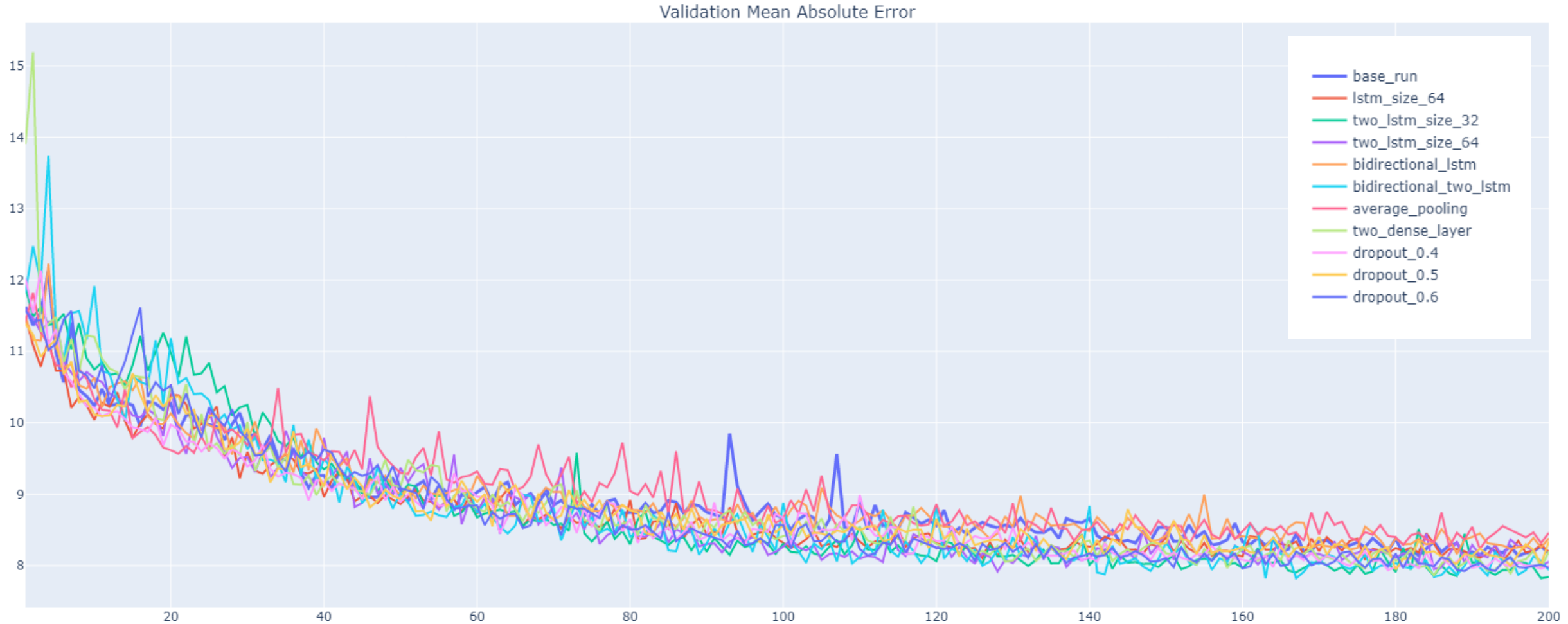
Ergebnisse

Versuchsergebnisse – Training MAE aller Modelle



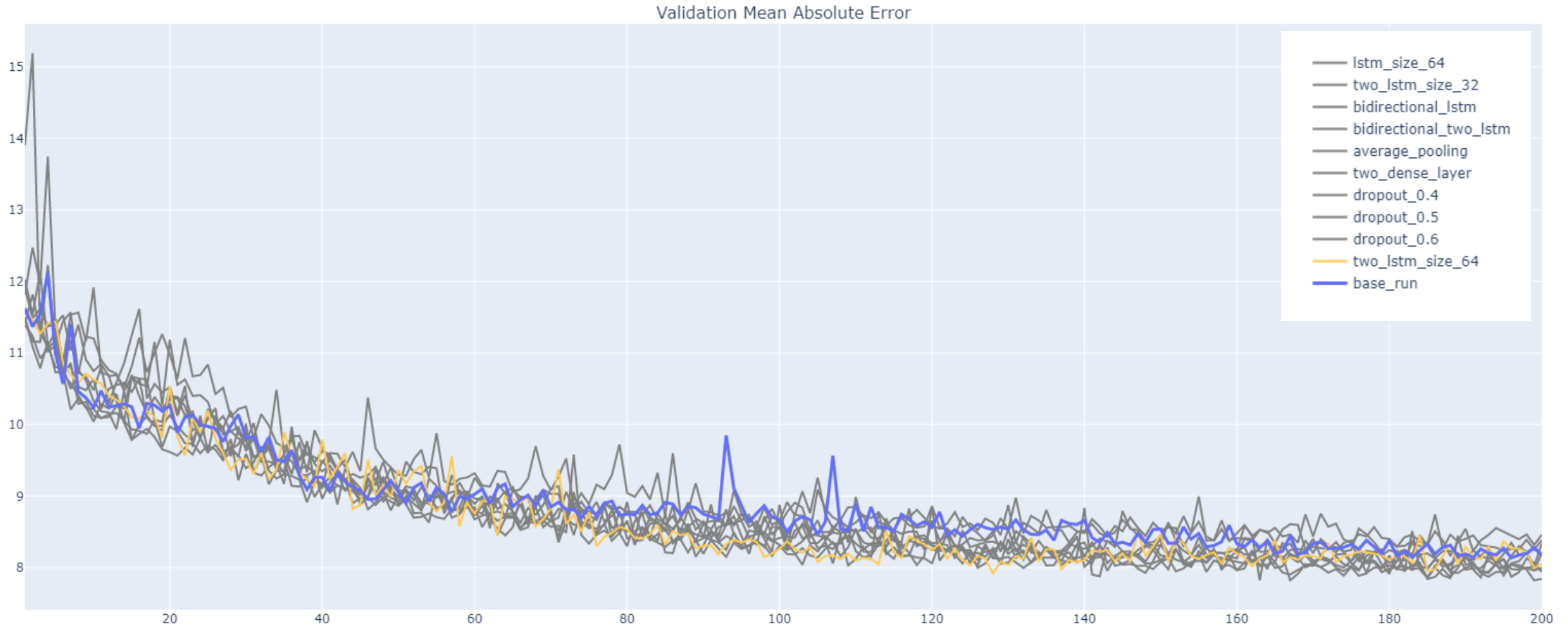
Ergebnisse

Versuchsergebnisse – Validation MAE aller Modelle



Ergebnisse

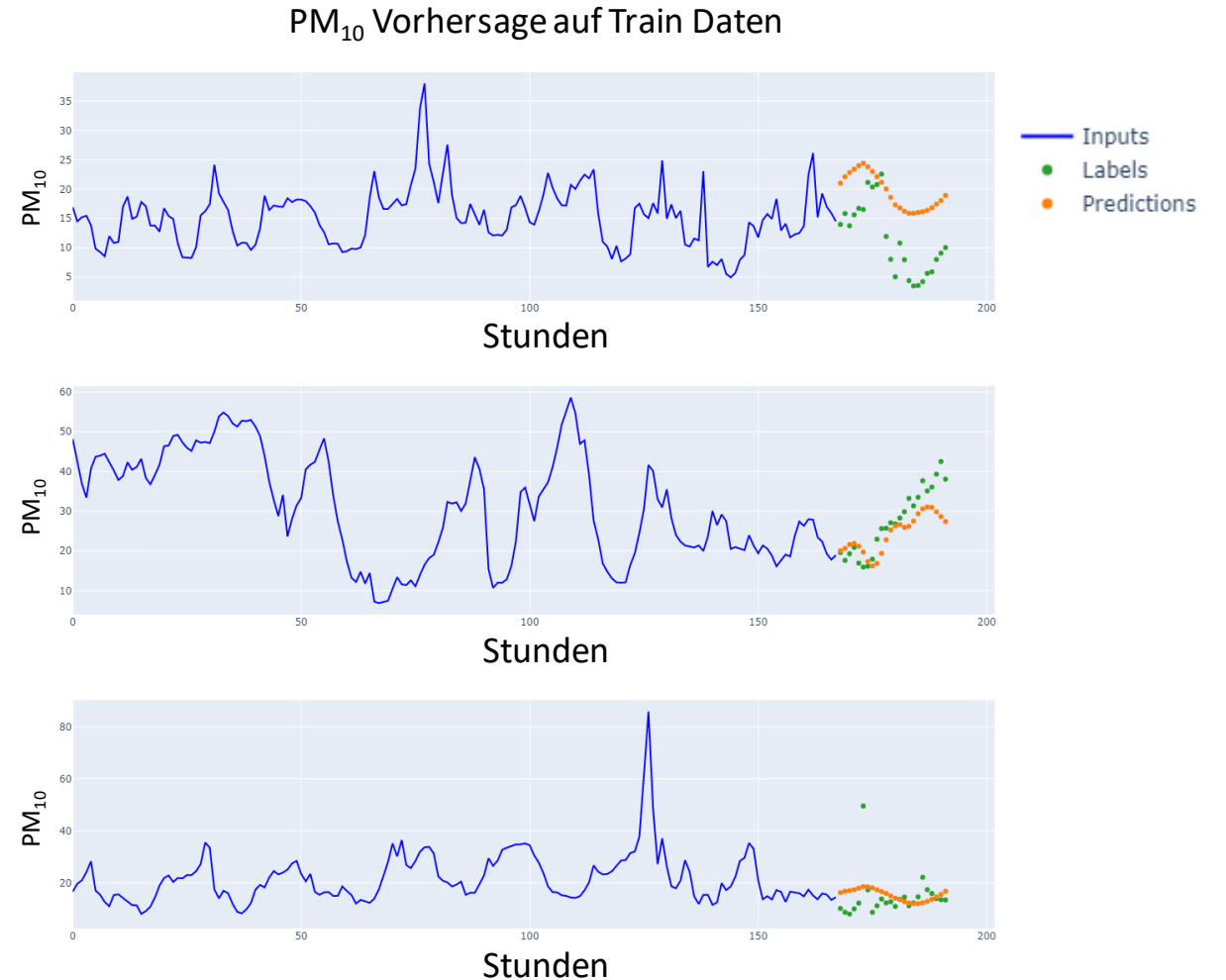
Versuchsergebnisse – Validation MAE aller Modelle



Ergebnisse

Bestes Modell

- Parametrisierung von IMT-14:
 - zwei LSTM-Layer mit Größe 64
 - 200 Epochen
 - Dropout 0.2
 - 7 Tage Eingabe
 - 24 Stunden Vorhersage
- Auswertung
 - Train MAE = 4,91
 - Val MAE = 8,05
 - Test MAE = 10,21



Ergebnisse

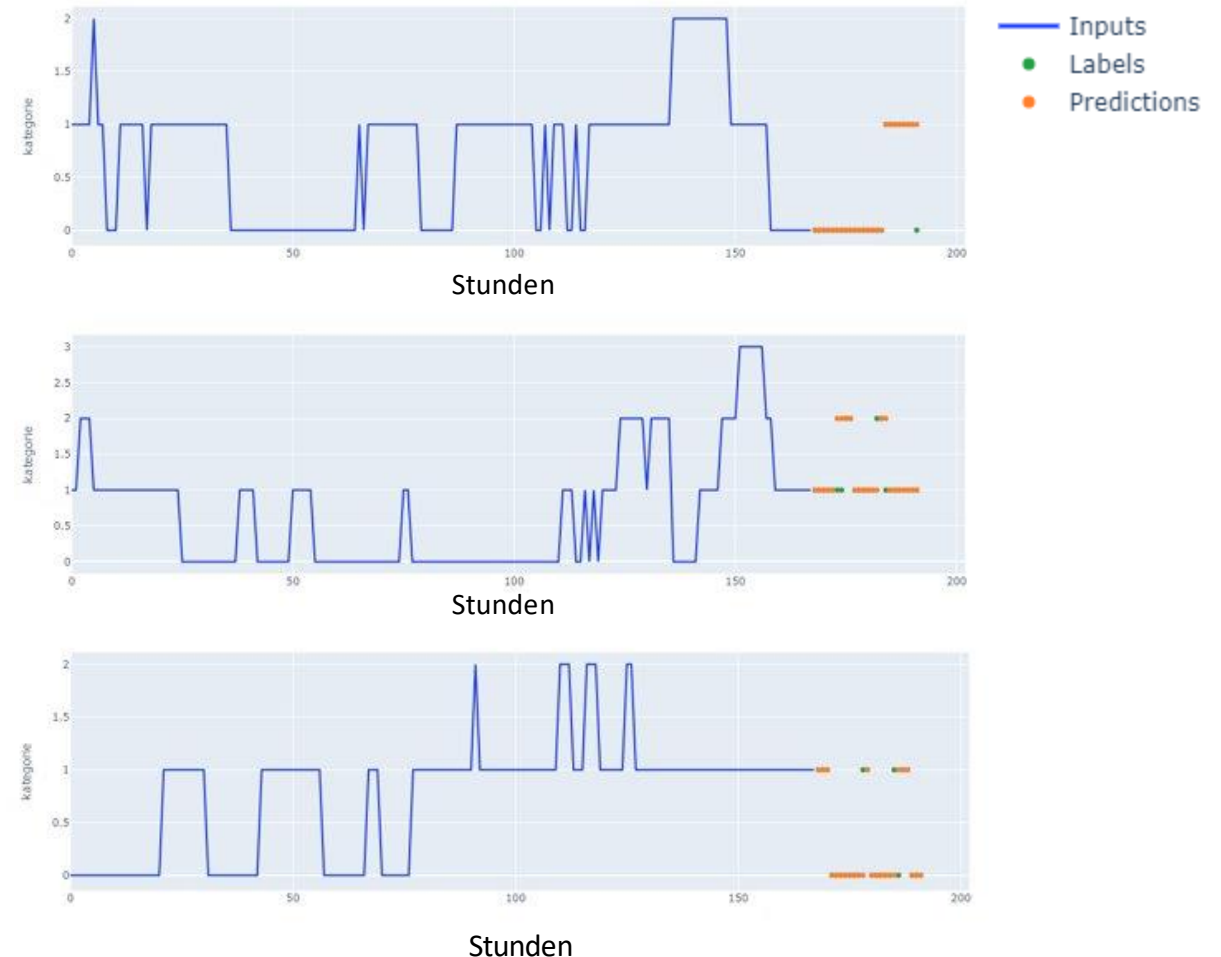
Klassifikationsmodell

- Aufteilung der PM10 Werte in 6 Gruppen
- Feature Vector um Gruppe ergänzt
- Änderungen im Vergleich zum besten Modell

Very good	0 – 70 [$\mu\text{g}/\text{m}^3$]
Good	70.1 – 120 [$\mu\text{g}/\text{m}^3$]
Moderate	120.1 – 150 [$\mu\text{g}/\text{m}^3$]
Sufficient	150.1 – 180 [$\mu\text{g}/\text{m}^3$]
Bad	180.1 – 240 [$\mu\text{g}/\text{m}^3$]
Very bad	> 240 [$\mu\text{g}/\text{m}^3$]

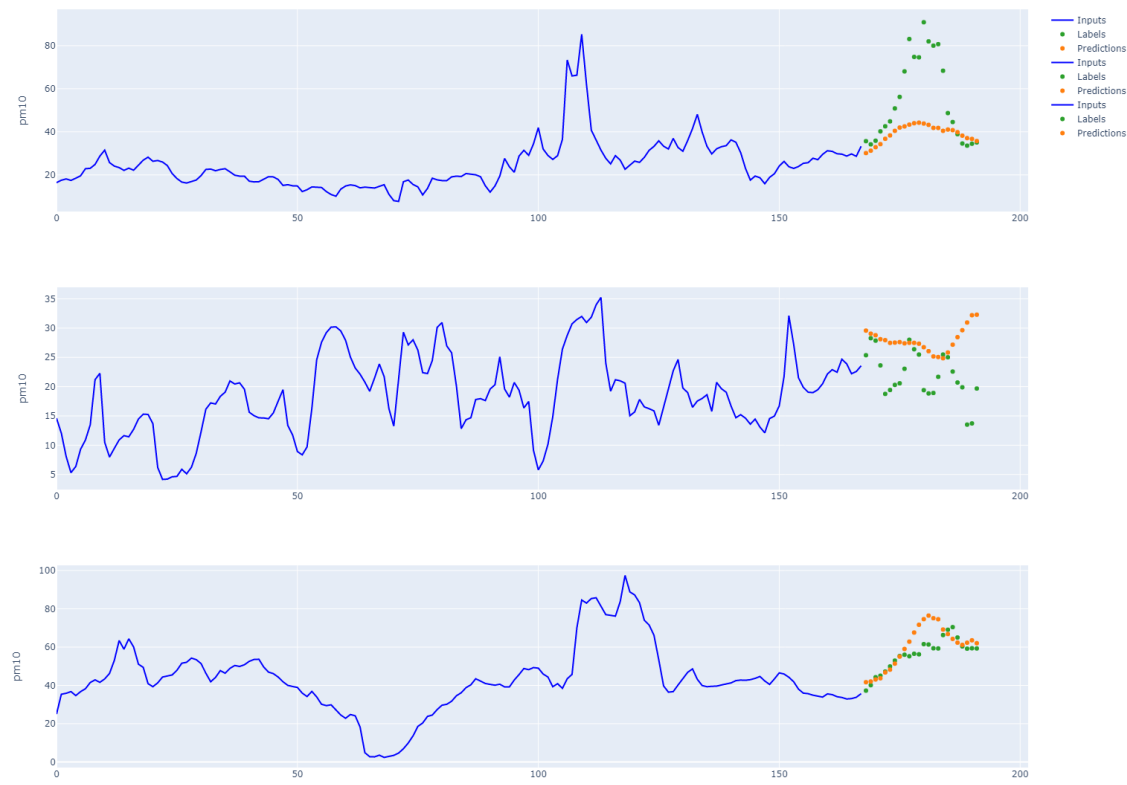
- ⑩ Dropout 0.4
 - ⑩ Letzter Dense Layer angepasst auf Klassifikation (softmax Aktivierungsfunktion)
- Metriken:
Train Accuracy: 0.97
Validation Accuracy: 0.96
Test Accuracy: 0.59

PM₁₀ Vorhersage auf Train Daten (Station 538)



Ergebnisse

Voraussage durch Modell einer naheliegenden Station



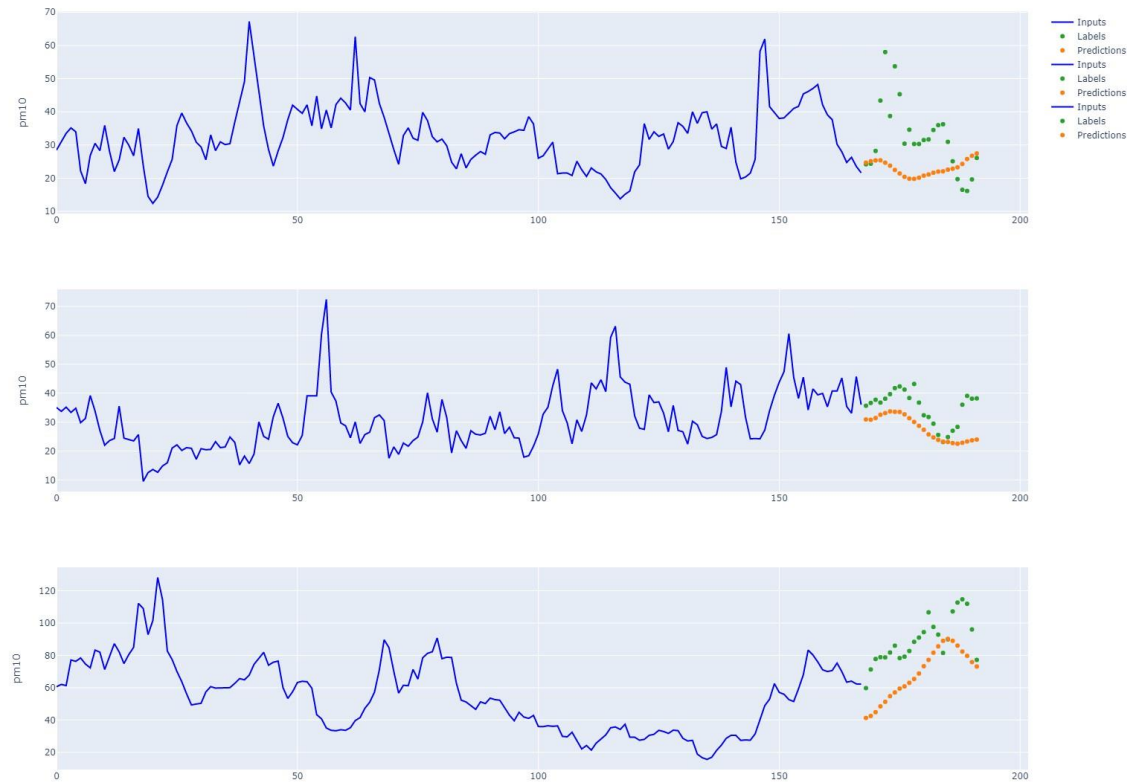
Vorhersage auf Train Daten

- Für Station 538 trainiertes Modell
- Voraussagen mit Daten von 530



Ergebnisse

Voraussage durch Modell einer naheliegenden Station



Vorhersage auf Train Daten

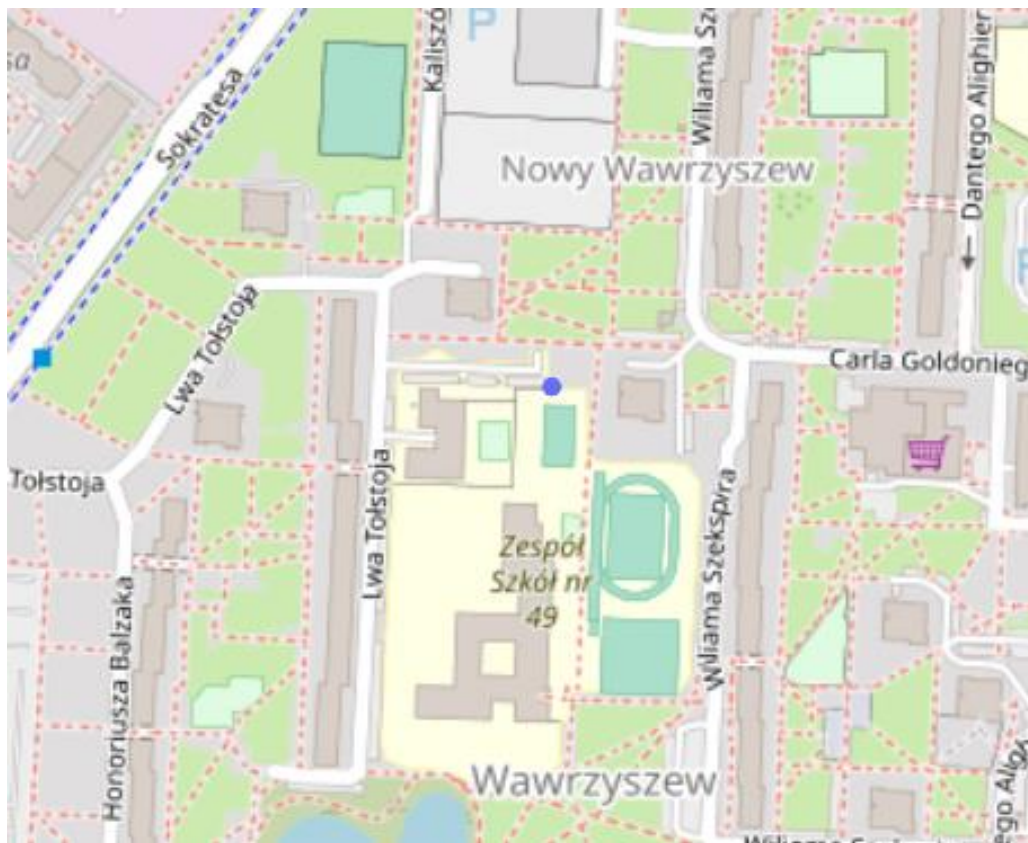
- Für Station 530 trainiertes Modell
- Voraussagen mit Daten von 538



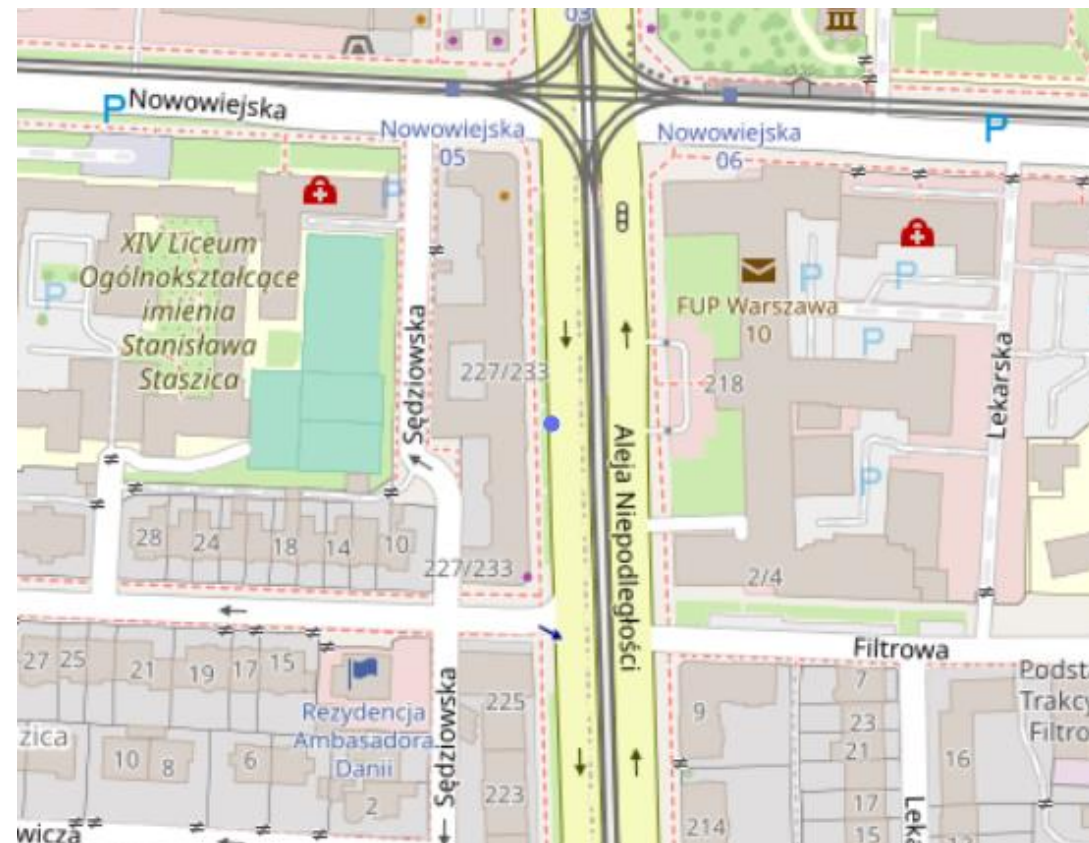
Ergebnisse

Lage der Sensoren 530 und 538

Sensor 538



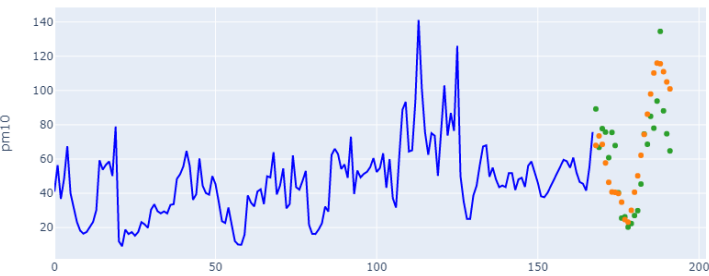
Sensor 530



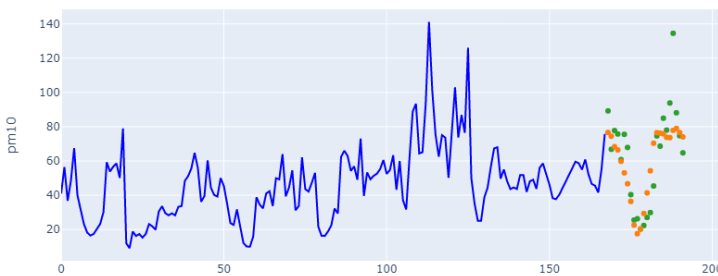
Ergebnisse

Messdaten der Station 530

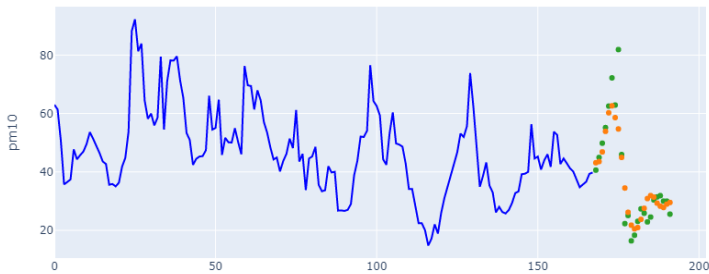
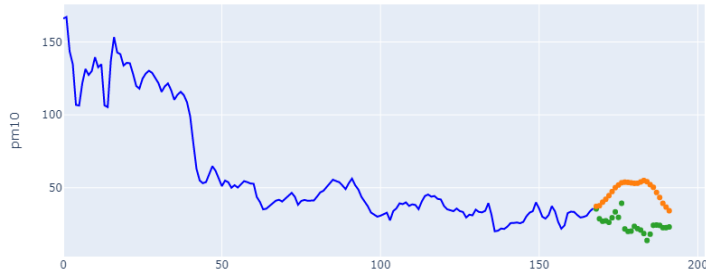
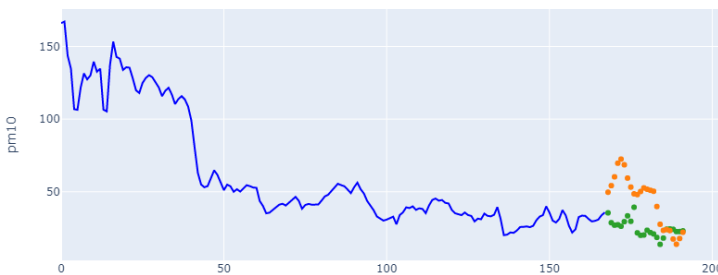
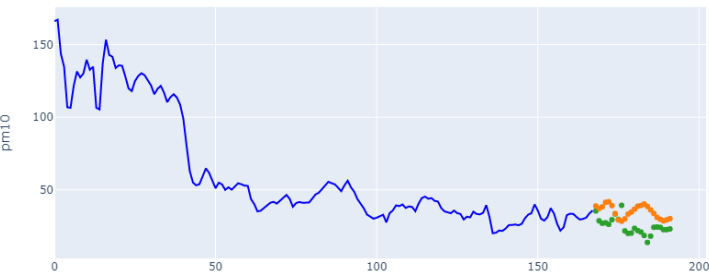
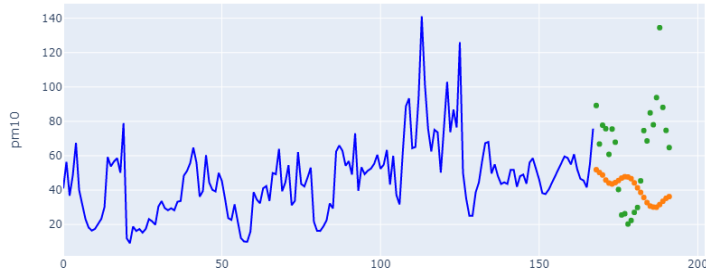
Modell trainiert auf 530



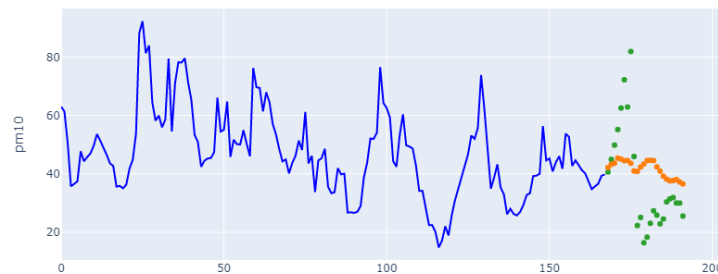
Modell trainiert auf 530 + 538



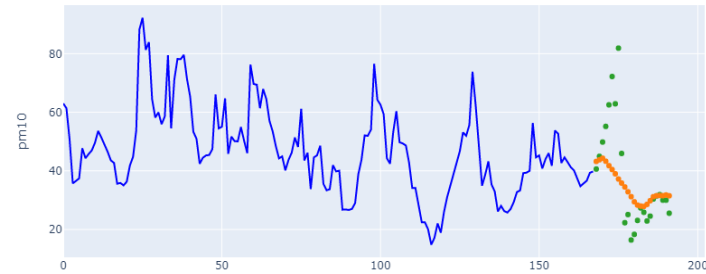
Modell trainiert auf 538



Vorhersage auf Train Daten



Vorhersage auf Train Daten

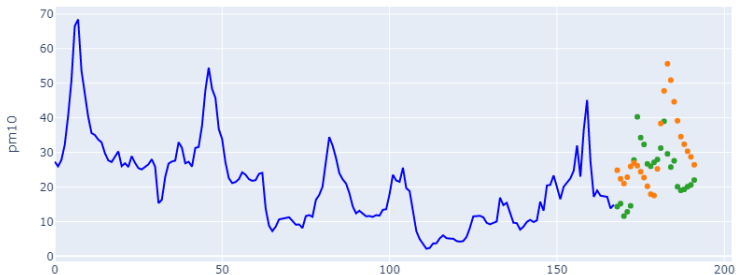


Vorhersage auf Train Daten

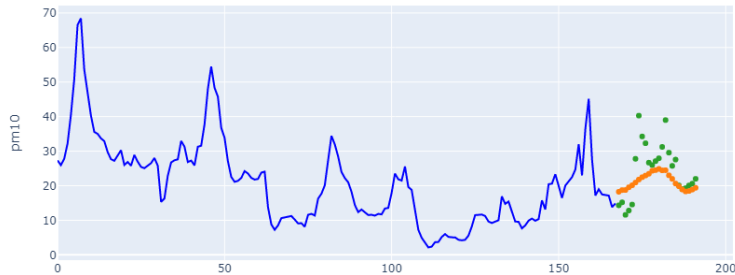
Ergebnisse

Messdaten der Station 538

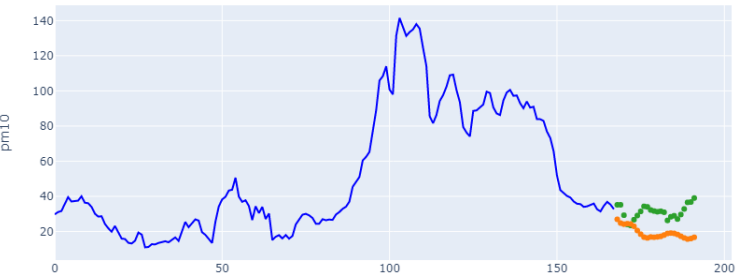
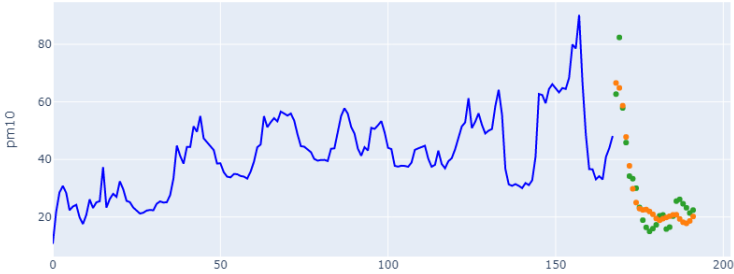
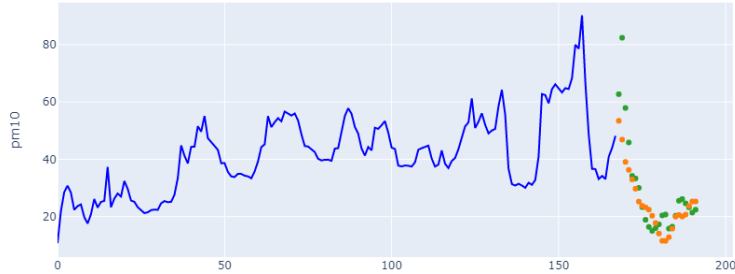
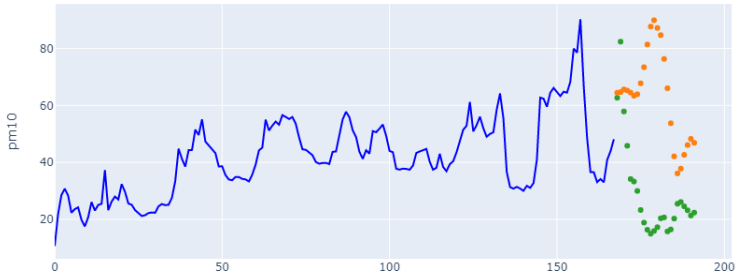
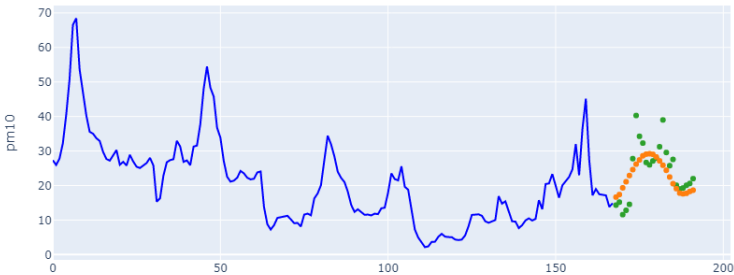
Modell trainiert auf 530



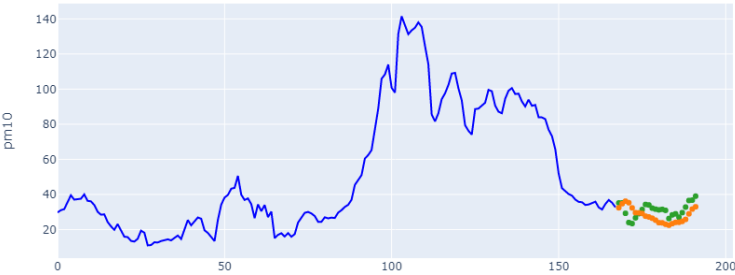
Modell trainiert auf 530 + 538



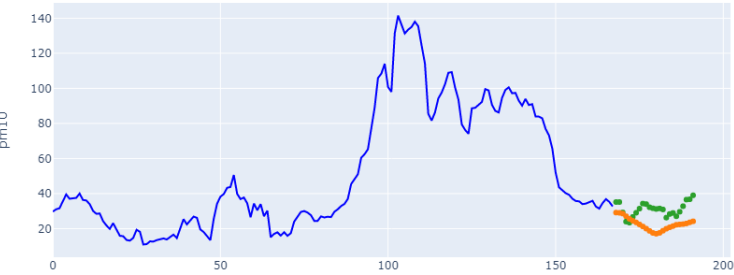
Modell trainiert auf 538



Vorhersage auf Train Daten



Vorhersage auf Train Daten



Vorhersage auf Train Daten

Ergebnisse

Zusammenfassung

- 14 Tage als Prädiktionszeitraum zu lang
 - 24 Stunden Prädiktion liefern akzeptable Ergebnisse
- Modellvariationen zeigen keine deutlichen Prädiktionsverbesserungen
- Modelle lassen sich nur für die trainierte Station verwenden
- Klassifikationsmodell liefert auf Trainings- und Validierungsdaten sehr gute Ergebnisse
 - ⑩ Performanz auf Testdaten ist schlecht → Overfitting des Modells
- Problemstellung zu komplex für die getesteten Modelle