Angewandtes Maschinelles Lernen:
Erkennung und Lokalisierung von Leckstellen in Wassernetzen

Andrea Maldonado

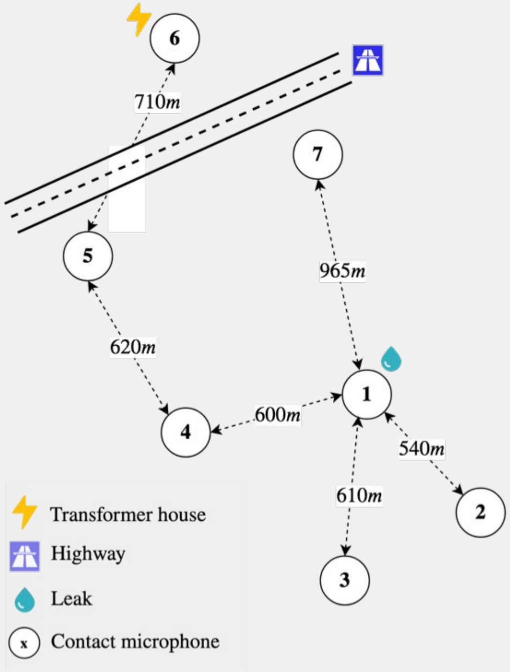
@andreamalhera

Praktikum Innovative Mobile Applications: "Gruppe A wie Anomalie"

Motivation

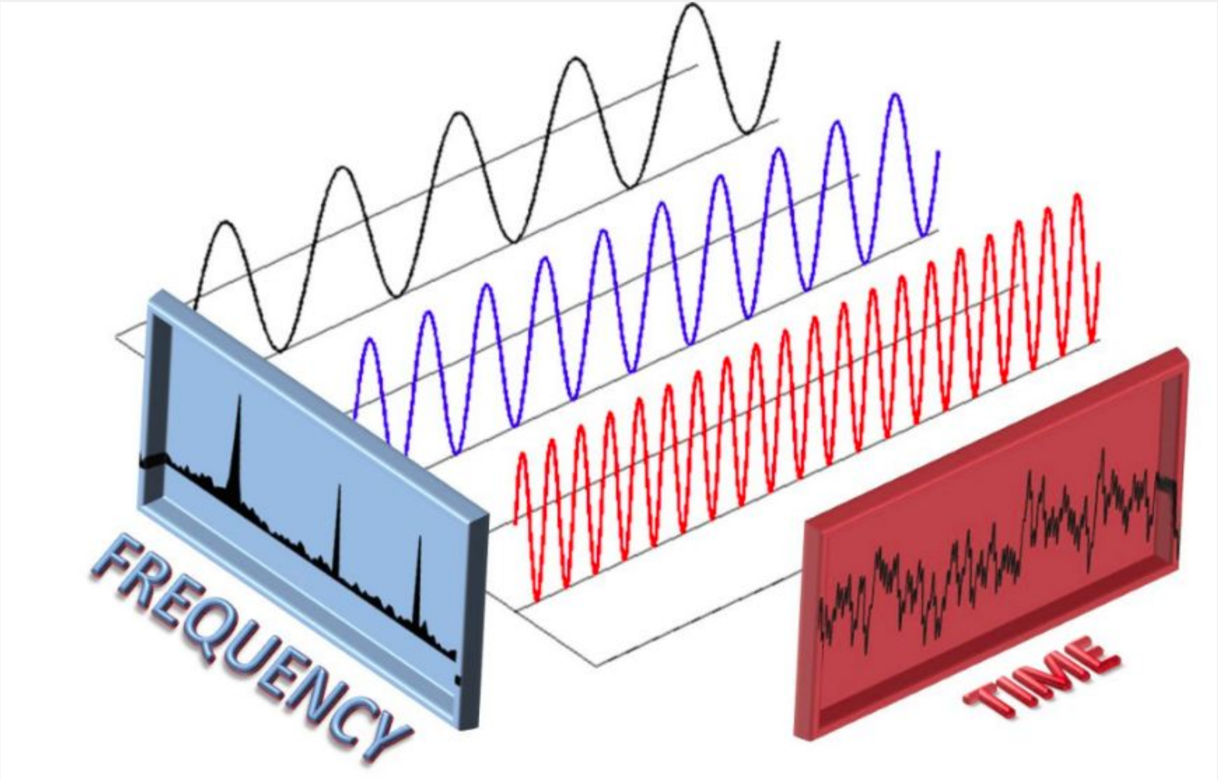
[1-3] Sources: https://www.wasser-macht-schule.de/trinkwasser...; https://www.br.de/radio/...; https://www.bund-naturschutz.de/alpen/...

Leckstellenerkennung in Wassernetzen



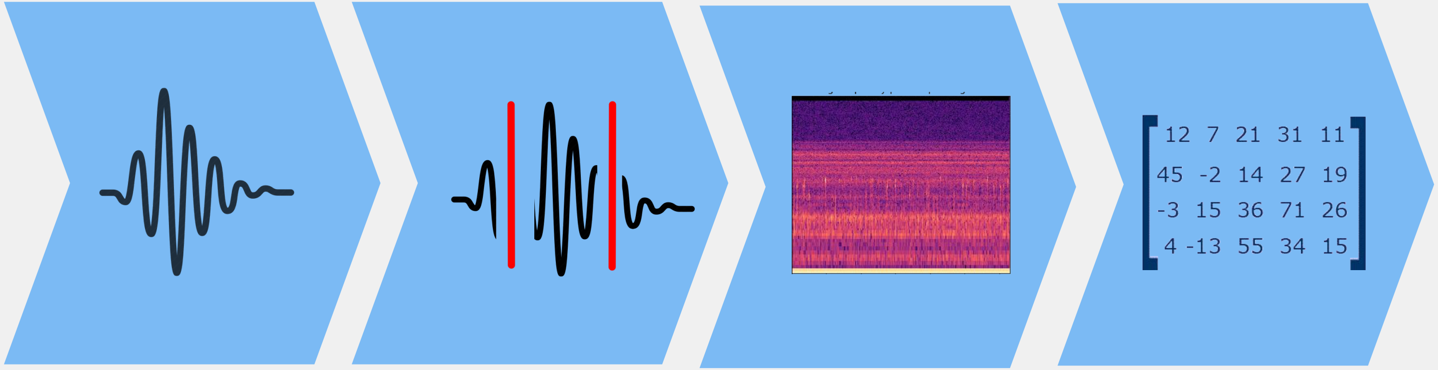
[4]: Quelle http://www.mobile.ifi.lmu.de/lehrveranstaltungen/praktikum-innovative-mobile-applications-sose19/

Was ist Ton und wie arbeiten wir damit?



[5] Quelle: https://steemit.com/steemstem/@wilians/fourier-series-and-transforms-applications-part-2

Preprocessing Pipeline



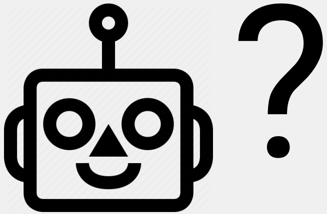
raw data | snippets | transformations | AE input

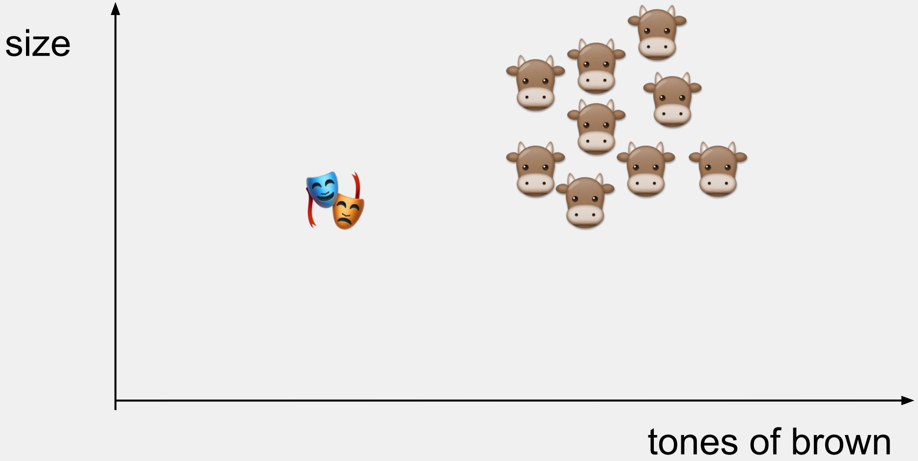
How does Anomaly Detection work in Machine Learning?

[4] Source: https://twitter.com/pigeonjon/status/708412176306987008

How does Anomaly Detection work in Machine Learning?

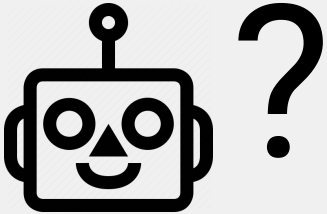
[4] Source: https://twitter.com/pigeonjon/status/708412176306987008

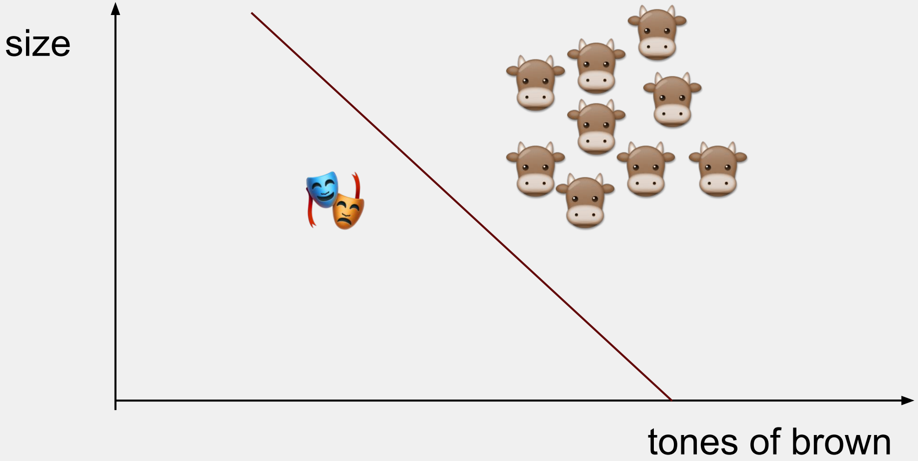




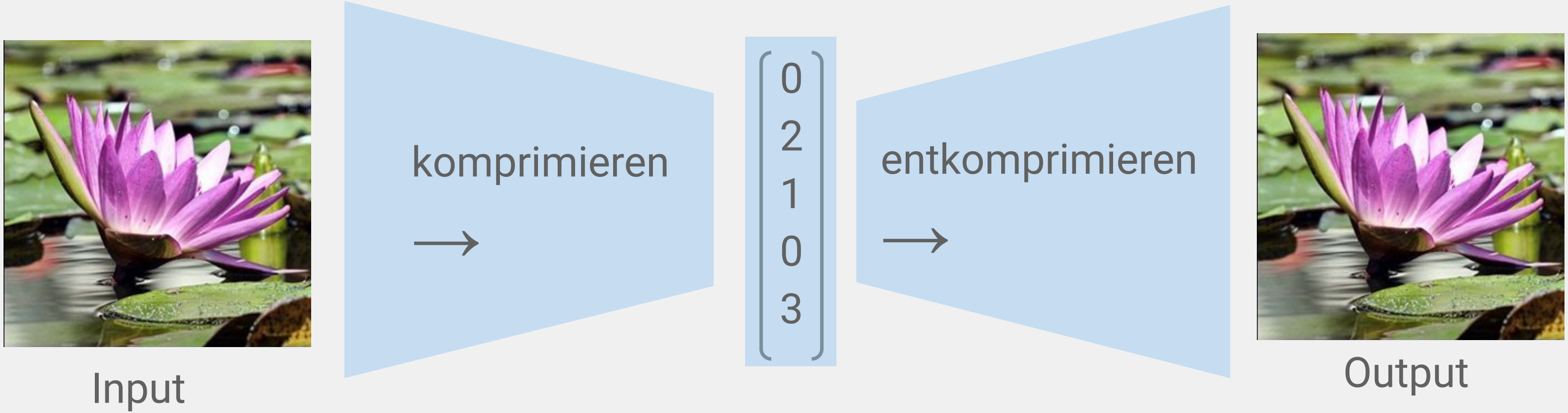
How does Anomaly Detection work in Machine Learning?

[4] Source: https://twitter.com/pigeonjon/status/708412176306987008

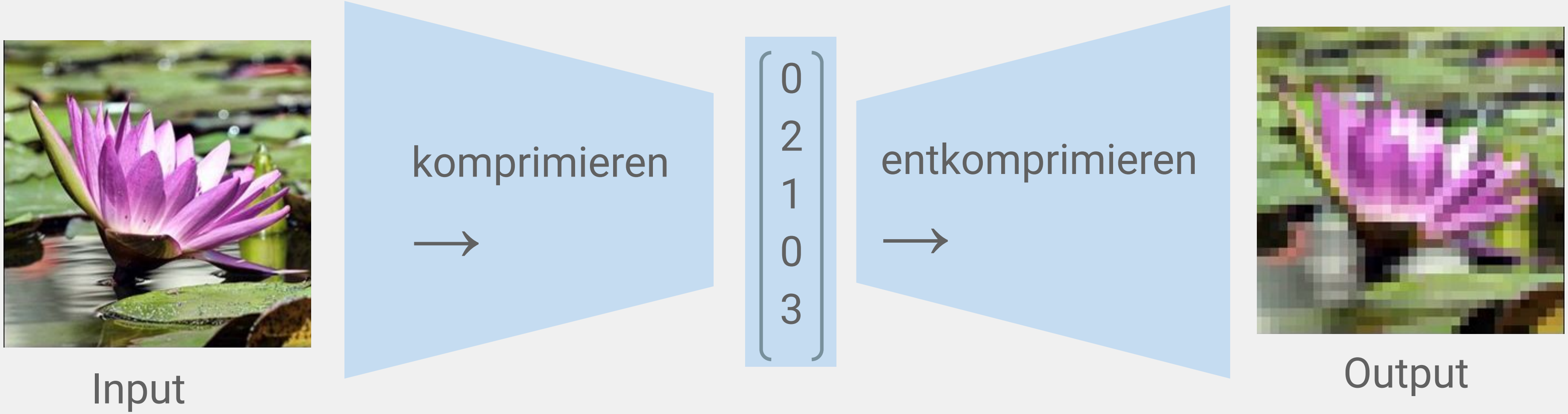




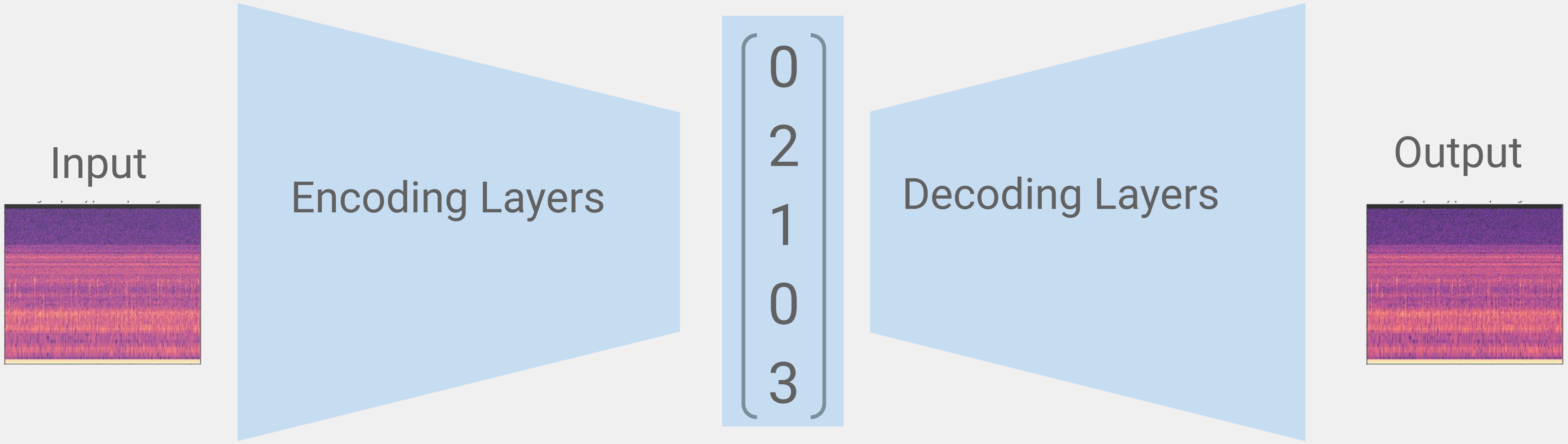
Prinzip des Autoencoders



Prinzip des Autoencoders



Architektur des Autoencoders



Architekturen des Autoencoders

Simple Autoencoder

Einfache Dense-Layers

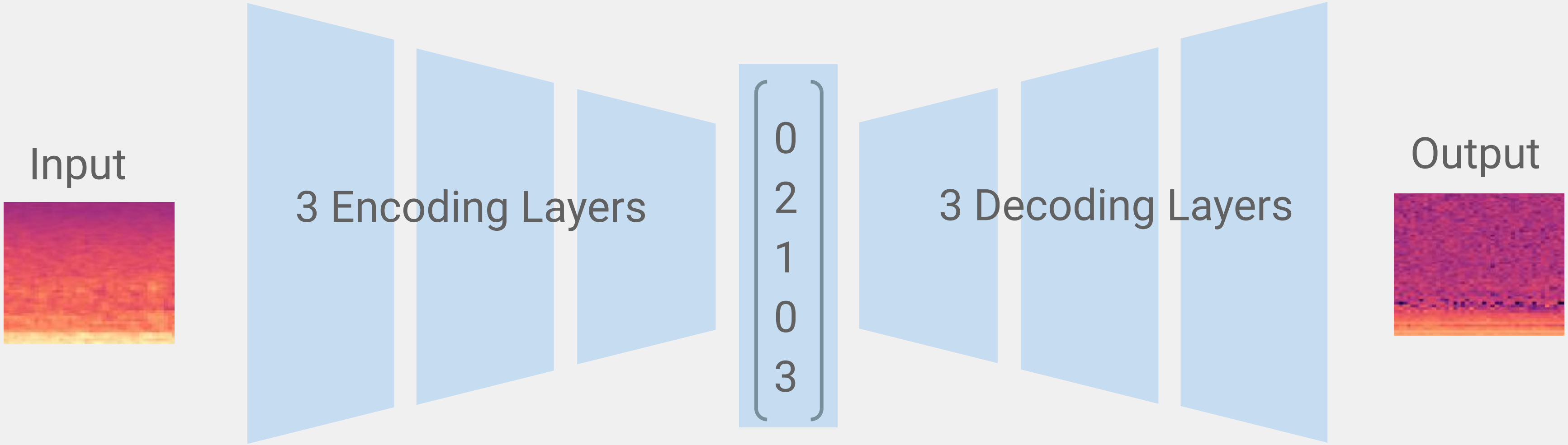
Convolutional Autoencoder

Convolutions, Max Pooling, Dense Layers

Variational Autoencoder

Lernt eine Verteilung der Daten

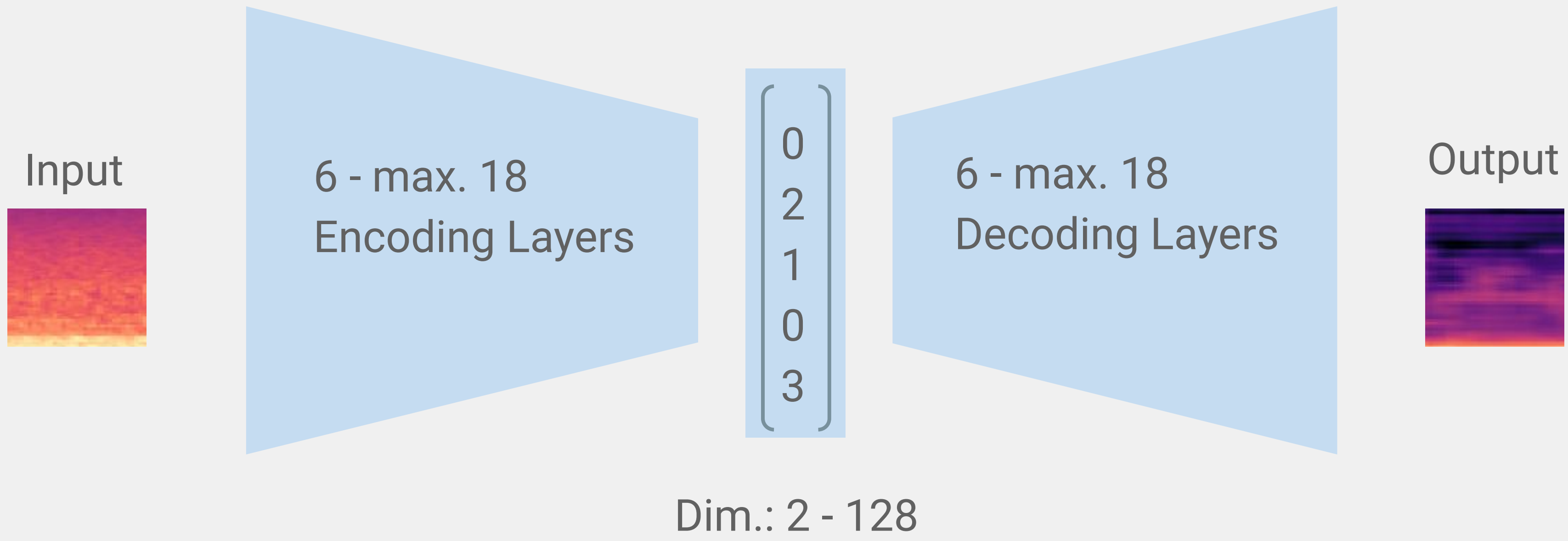
Autoencoder Architekturen: Simple Autoencoder



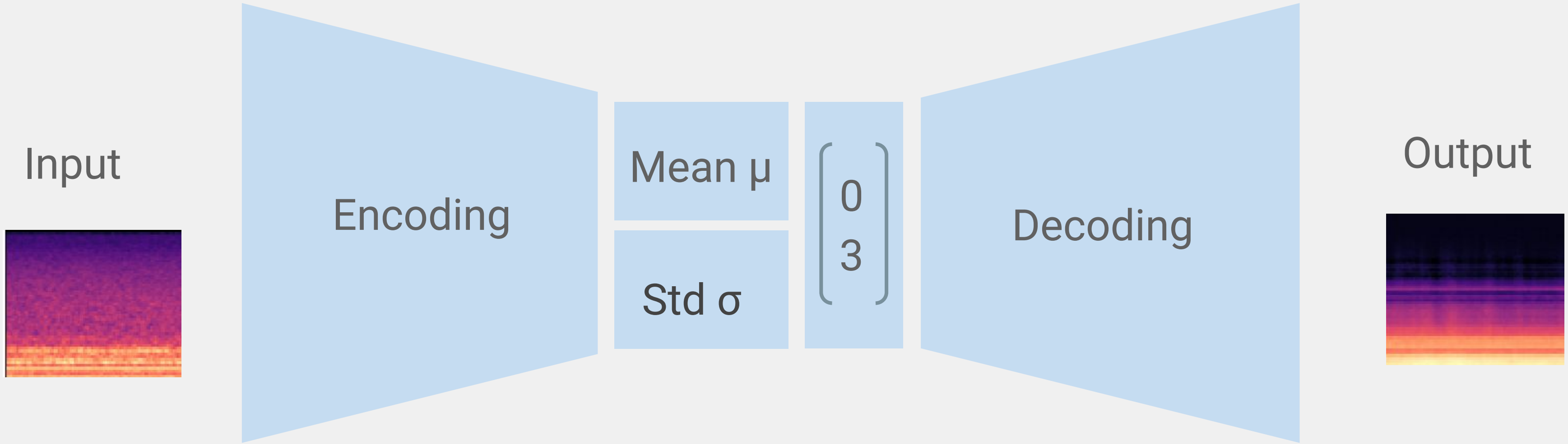
Dim.: 3 - 300

Nur Dense Layers

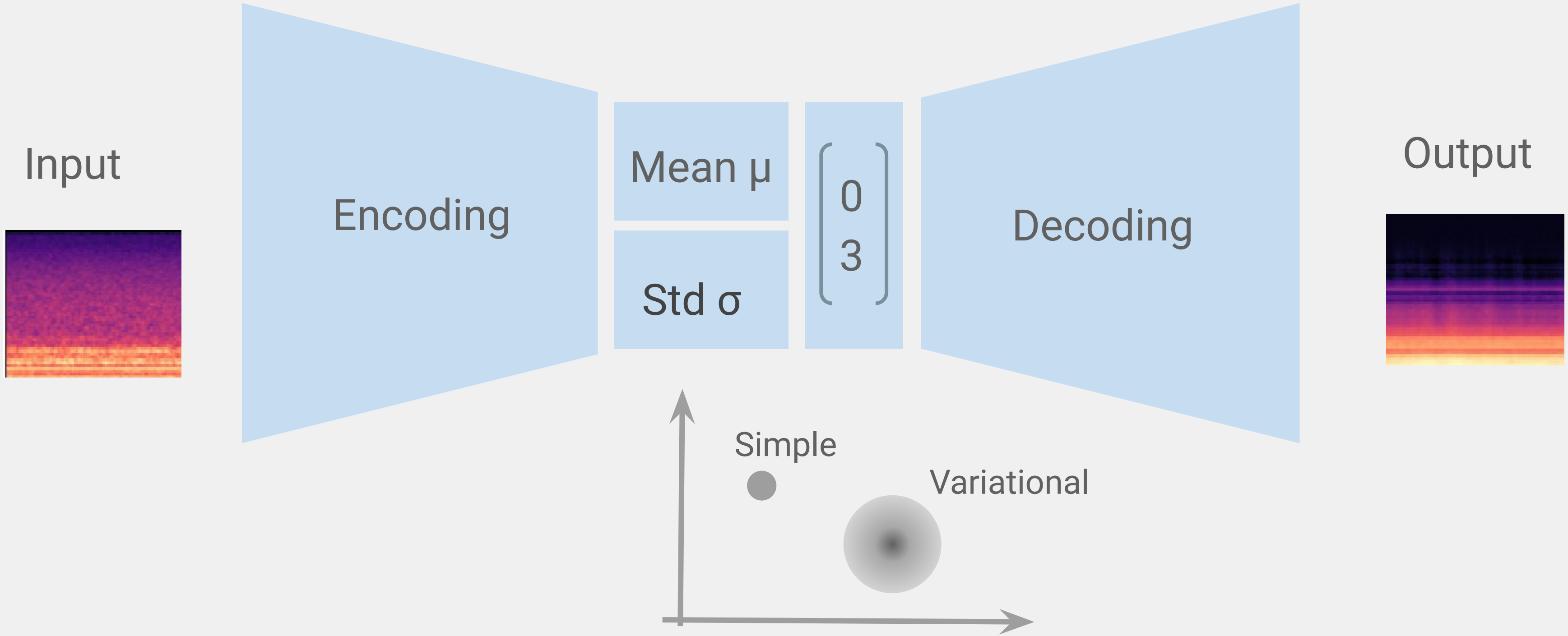
Autoencoder Architekturen: Convolutional Autoencoder (CNN)



Autoencoder Architekturen: Variational Autoencoder



Autoencoder Architekturen: Variational Autoencoder

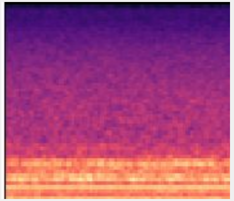


Autoencoder Architekturen: Training

vs.

Input

$$\begin{bmatrix} 0 & 2 & 8 & 6 \\ 3 & 7 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$



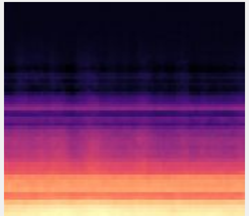
Verlust

neue Gewichte

neues Modell

Output

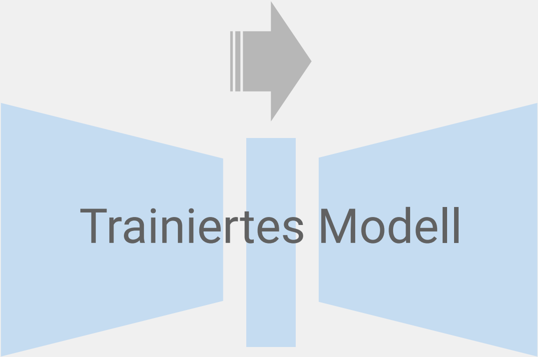
$$\begin{bmatrix} 0 & 2 & 8 & 6 \\ 0 & 0 & 1 & 2 \\ 3 & 7 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$



Data Classification: Reconstruction Error (RE)

Input: Eine Datei

$$\begin{bmatrix} 0 & 2 & 8 & 6 \\ 3 & 7 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 8 & 6 \\ 0 & 0 & 1 & 2 \\ 3 & 7 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 4 & 3 \\ 3 & 7 & 1 & 0 \\ 0 & 0 & 2 & 4 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$



Prediction

$$\begin{bmatrix} 0 & 2 & 8 & 6 \\ 0 & 0 & 1 & 2 \\ 3 & 7 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 4 & 3 \\ 3 & 7 & 1 & 0 \\ 0 & 0 & 2 & 4 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 8 & 6 \\ 3 & 7 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

RE: euklidische Distanz pro Frame

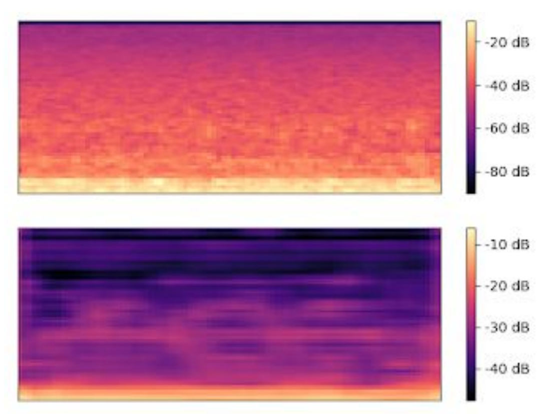
Klassifikation mit RE Threshold

Daten Klassifikation: Bsp. CNN AE

Original

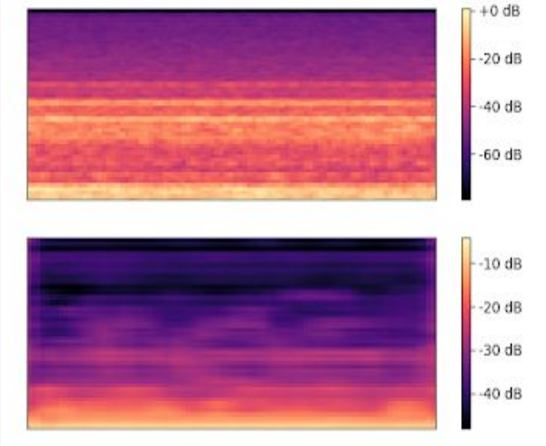
Rekonstruktion

kein Leck



2 Sek

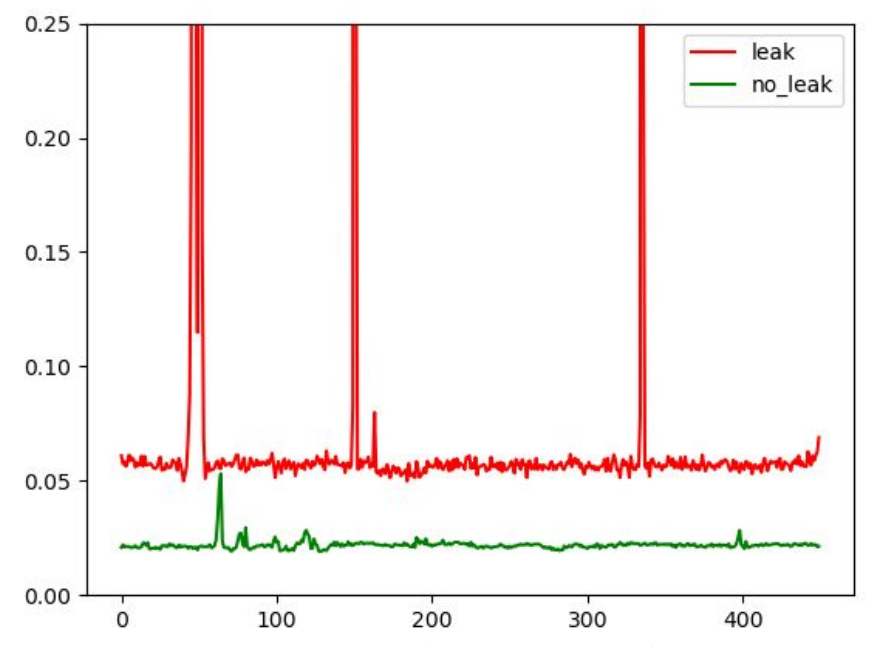
Leck



2 Sek

Daten Klassifikation: Reconstruction Error von zwei Dateien

Reconstruction Error



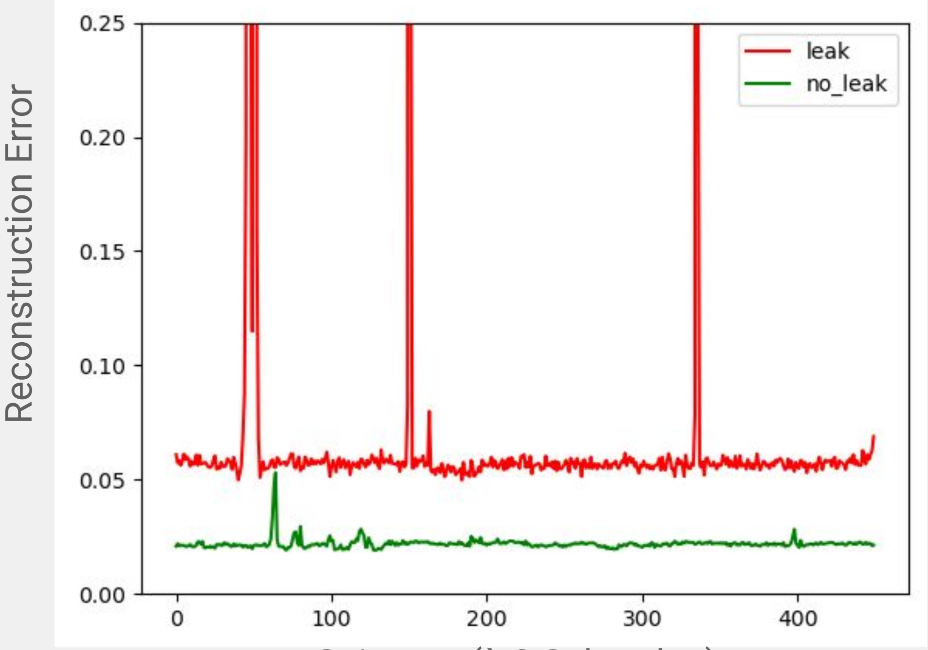
Snippets (à 2 Sekunden)

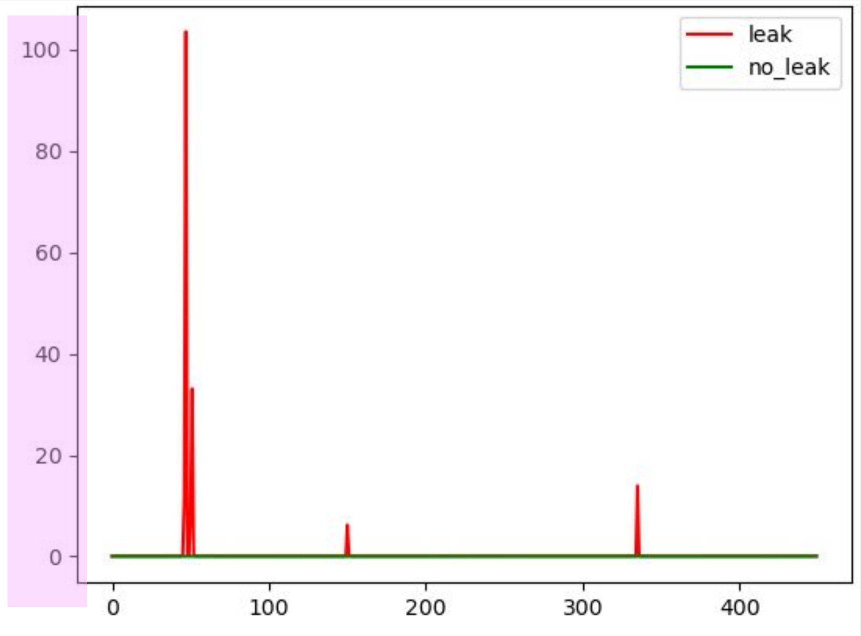
CNN-Autoencoder mit 6 Layers,
Encoding-Dim. 2,
30 Epochen lang trainiert

Jeweils eine Beispiel-Audiodatei mit Leck (rot) vs. eine ohne Leck (grün)

Daten Klassifikation: Reconstruction Error

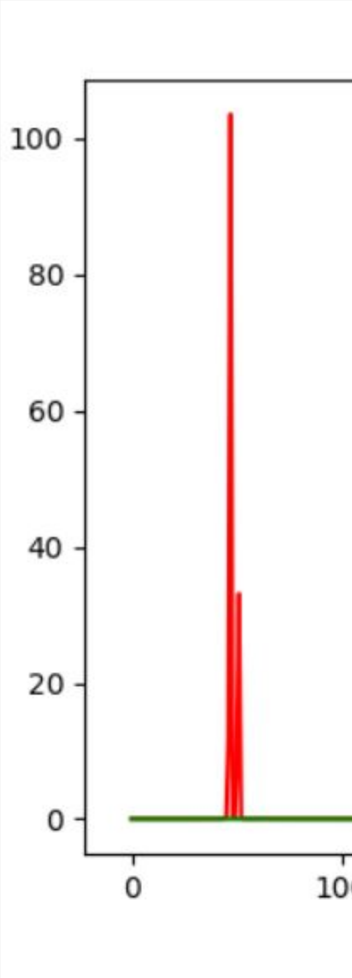
Herausgezoomte y-Achse:





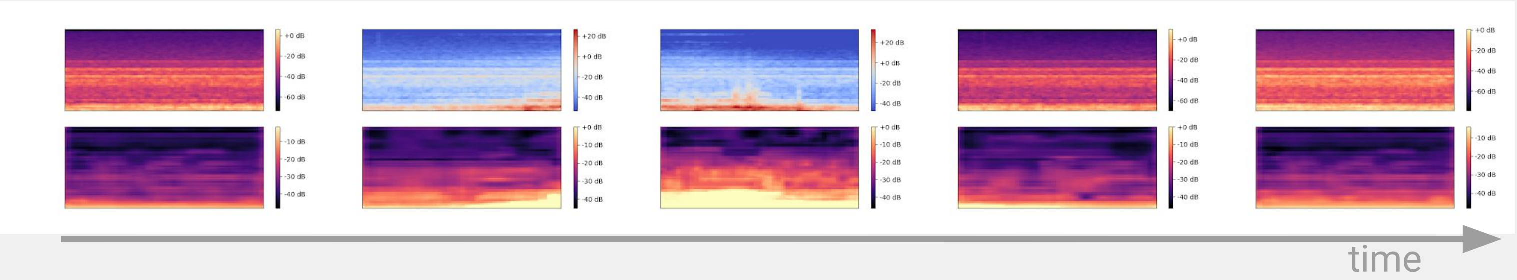
Snippets (à 2 Sekunden)

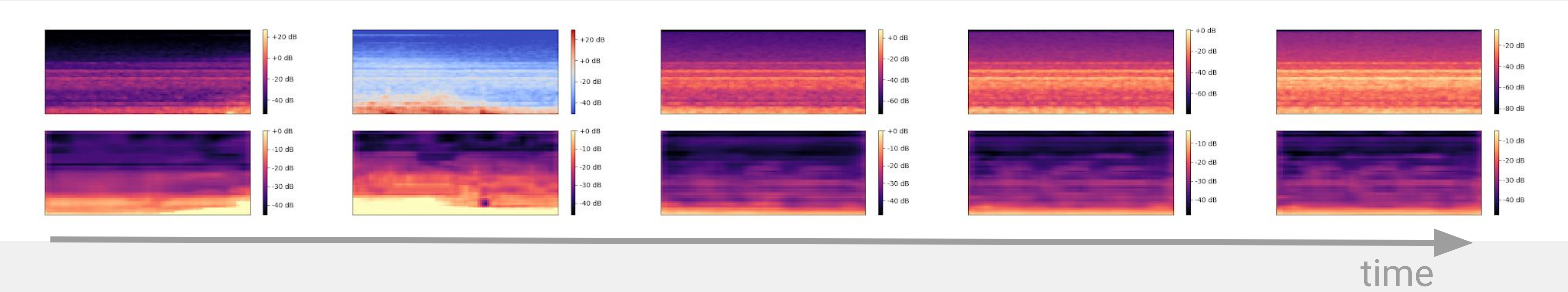
Daten Klassifikation: Reconstruction Error

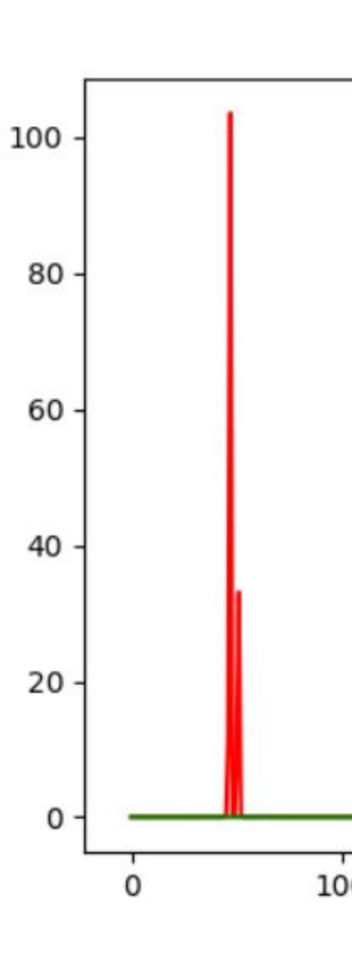


Daten Klassifikation: Reconstruction Error

Leck-Spektrogramme der Snippets: Input (o.) vs. Prediction (u.)

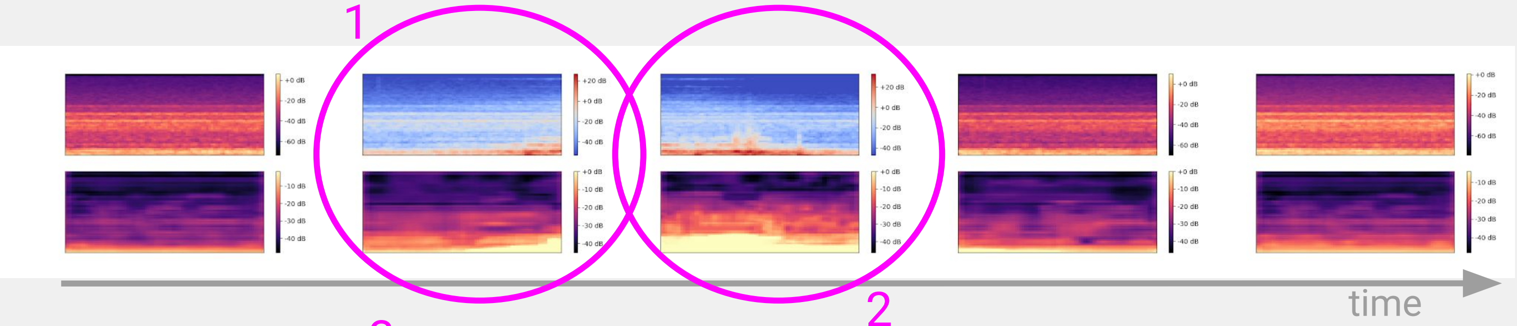


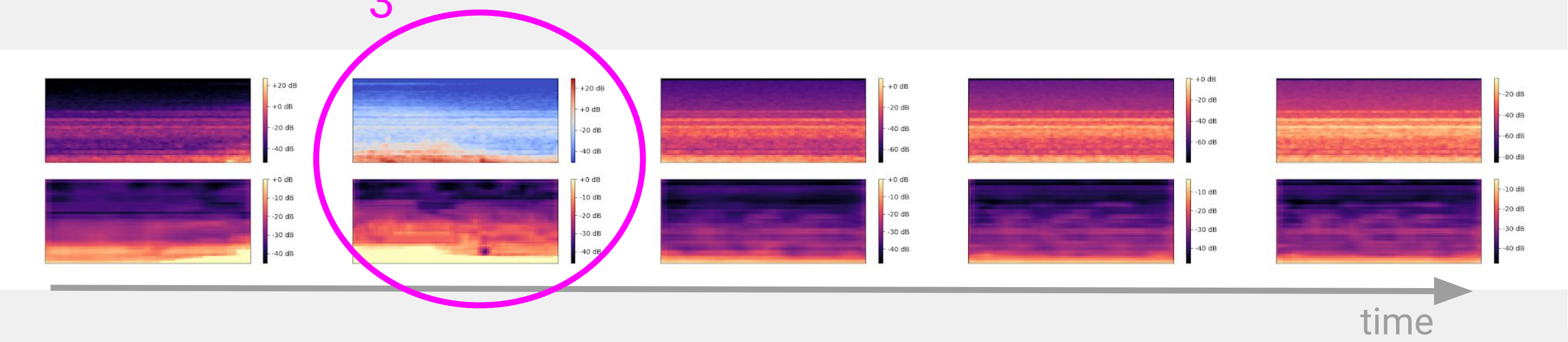


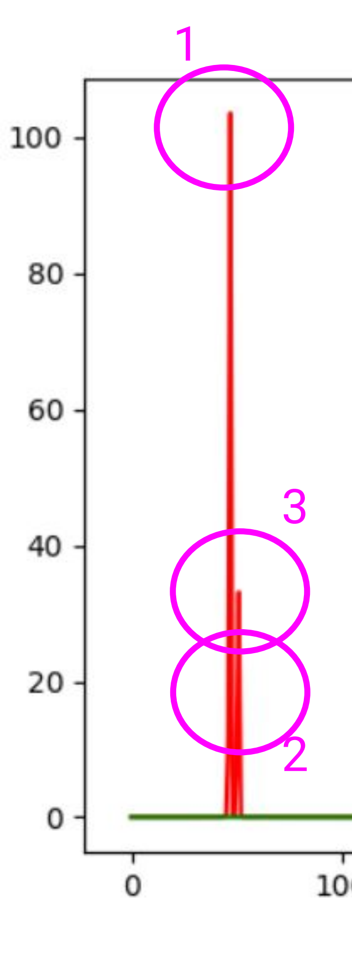


Daten Klassifikation: Reconstruction Error

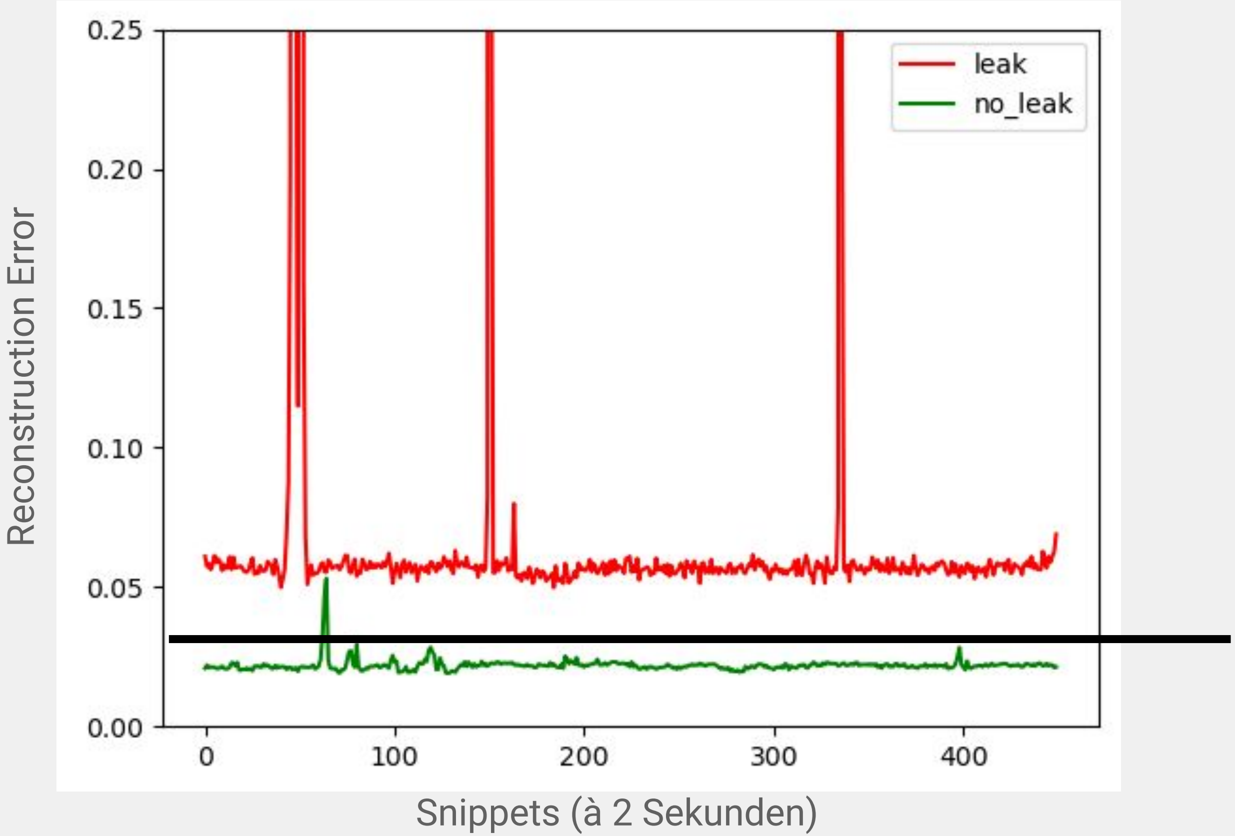
Leck-Spektrogramme der Snippets: Input (o.) vs. Prediction (u.)





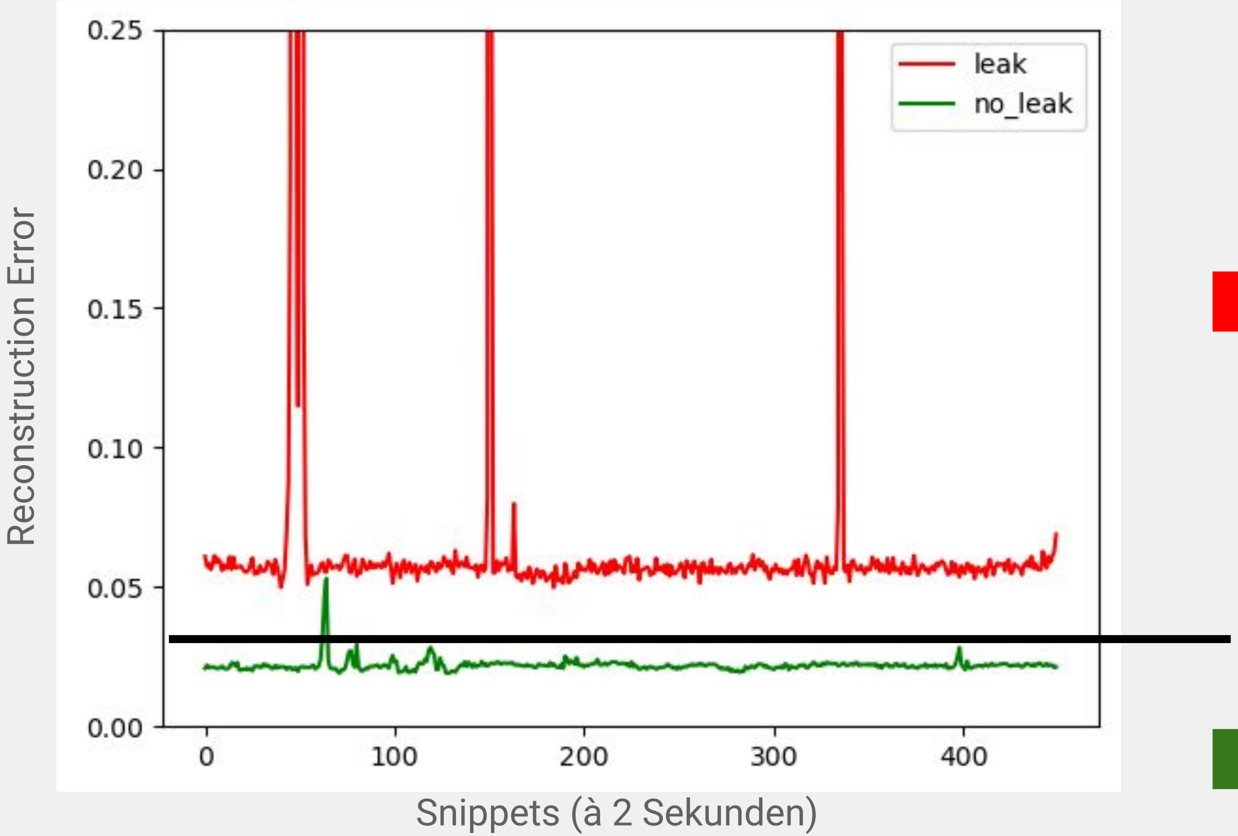


Klassifikation anhand eines Threshold



Bsp. für Threshold:
Mean + Std
von Leak data

Klassifikation anhand eines Threshold



Leck

Bsp. für Threshold:
Mean + Std
von Leak data

Kein Leck

Klassifikation anhand eines Threshold

$$\begin{bmatrix} 0 & 2 & 8 & 6 \\ 0 & 0 & 1 & 2 \\ 3 & 7 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 4 & 3 \\ 3 & 7 & 1 & 0 \\ 0 & 0 & 2 & 4 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 8 & 6 \\ 3 & 7 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Input Datei
in Snippets

3.425 6.671 7.425

Reconstruction
errors

Klassifikation anhand eines Threshold

$$\begin{bmatrix} 0 & 2 & 8 & 6 \\ 0 & 0 & 1 & 2 \\ 3 & 7 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 4 & 3 \\ 3 & 7 & 1 & 0 \\ 0 & 0 & 2 & 4 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 8 & 6 \\ 3 & 7 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Input Datei in Snippets

3.425 6.671 7.425

Reconstruction Errors

0 1 1

Scores von Snippets

Bsp. Threshold: 0.4

Klassifikation anhand eines Threshold

$$\begin{bmatrix} 0 & 2 & 8 & 6 \\ 0 & 0 & 1 & 2 \\ 3 & 7 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 4 & 3 \\ 3 & 7 & 1 & 0 \\ 0 & 0 & 2 & 4 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 8 & 6 \\ 3 & 7 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Input Datei in Snippets

3.425 6.671 7.425

Reconstruction Errors

0 1 1

Scores von Snippets

Mean

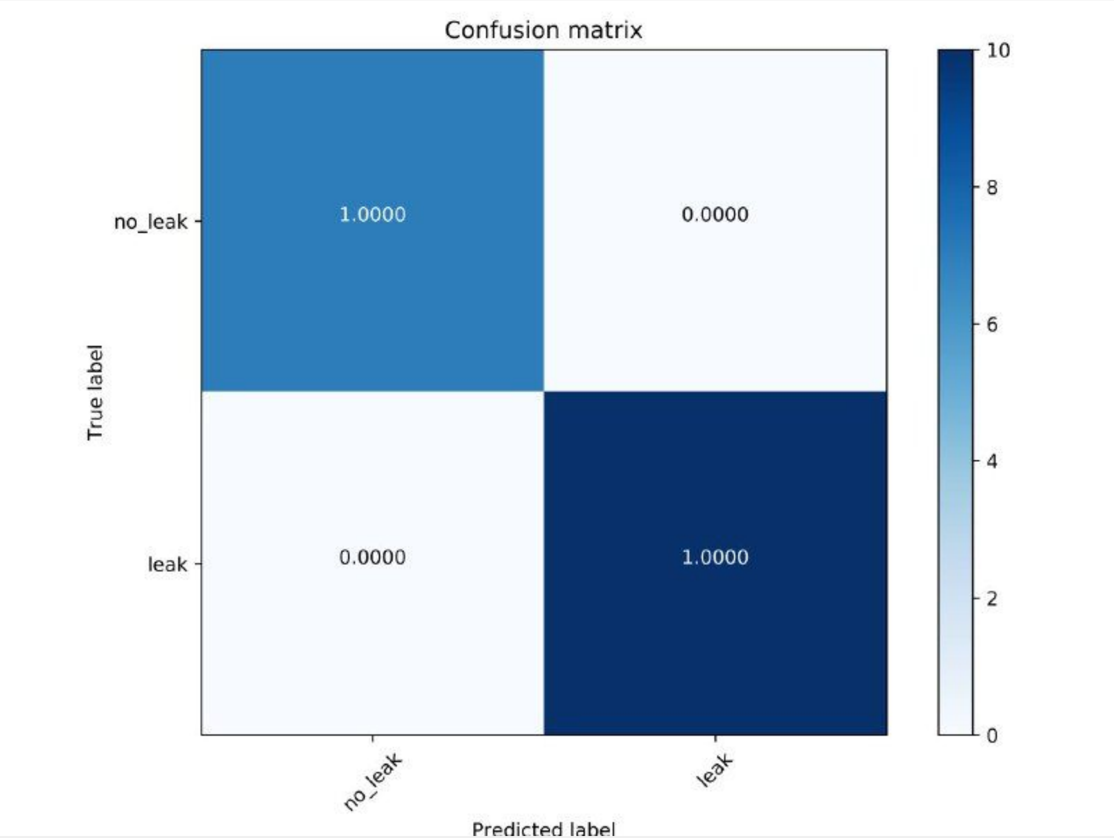
Bsp. Threshold: 0.4

0.666 Leck!

Score von Input Datei

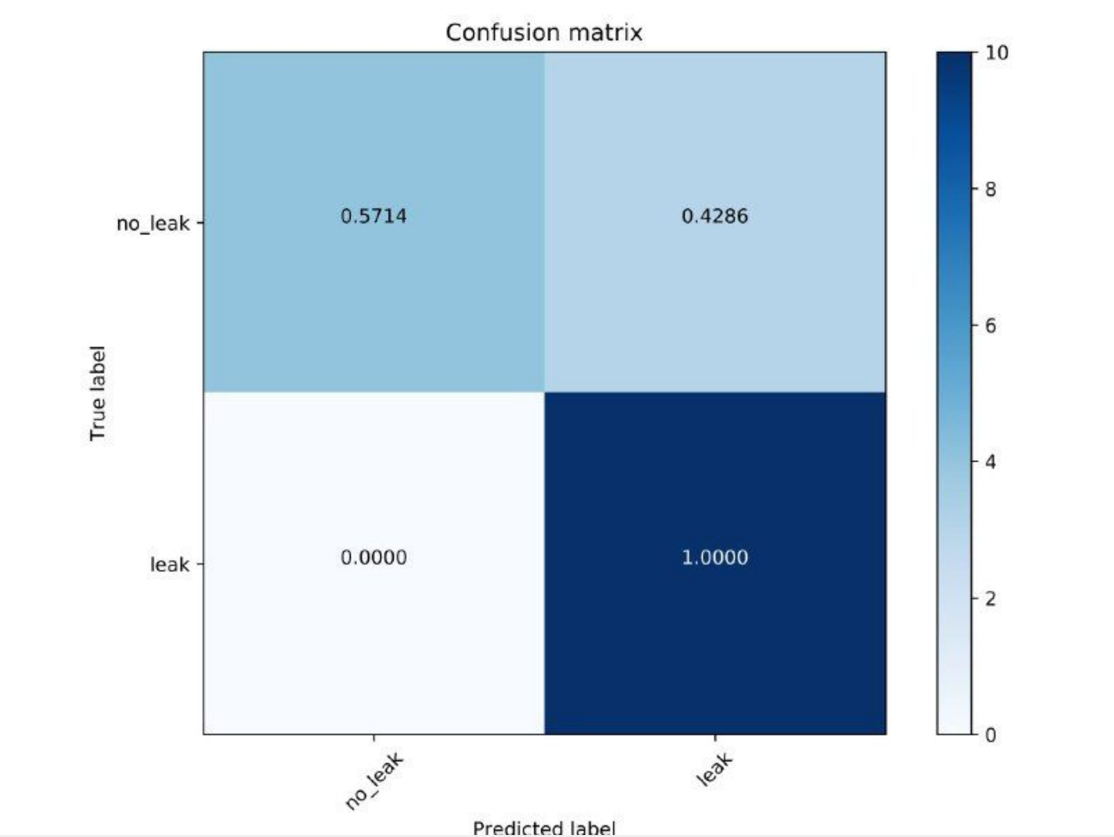
Evaluation: Konfusionsmatrix

CNN mit 2D



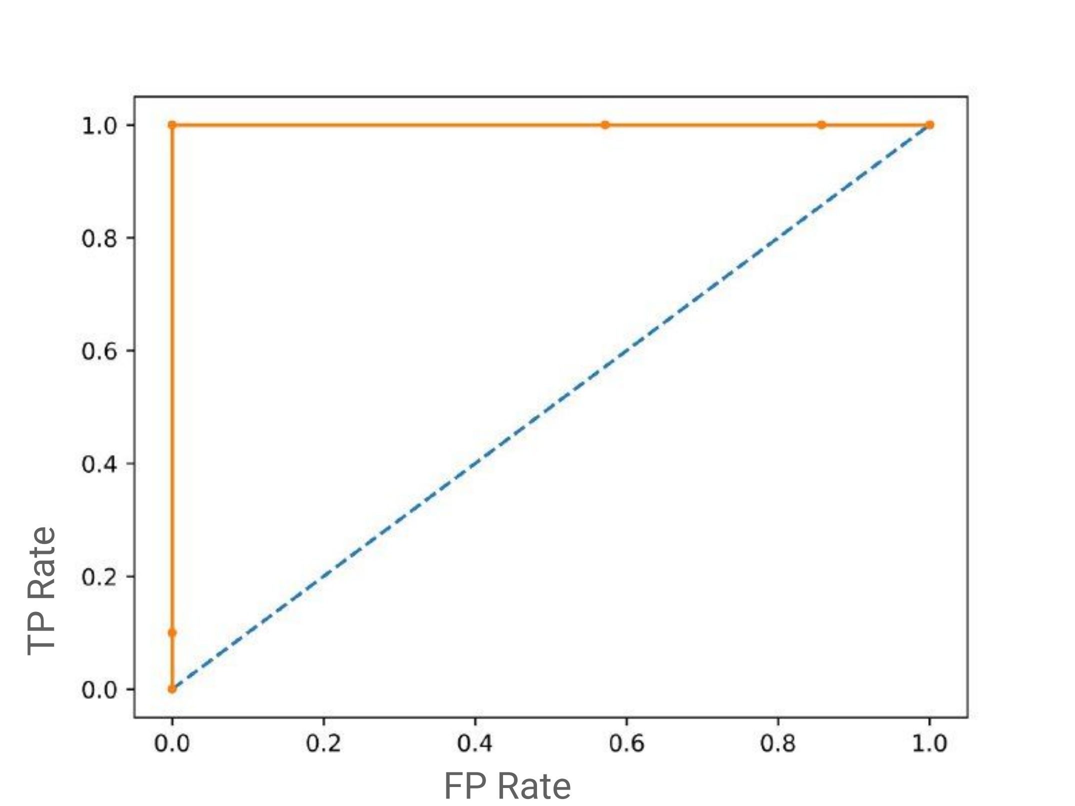
Evaluation: Konfusionsmatrix

SAE mit 10D



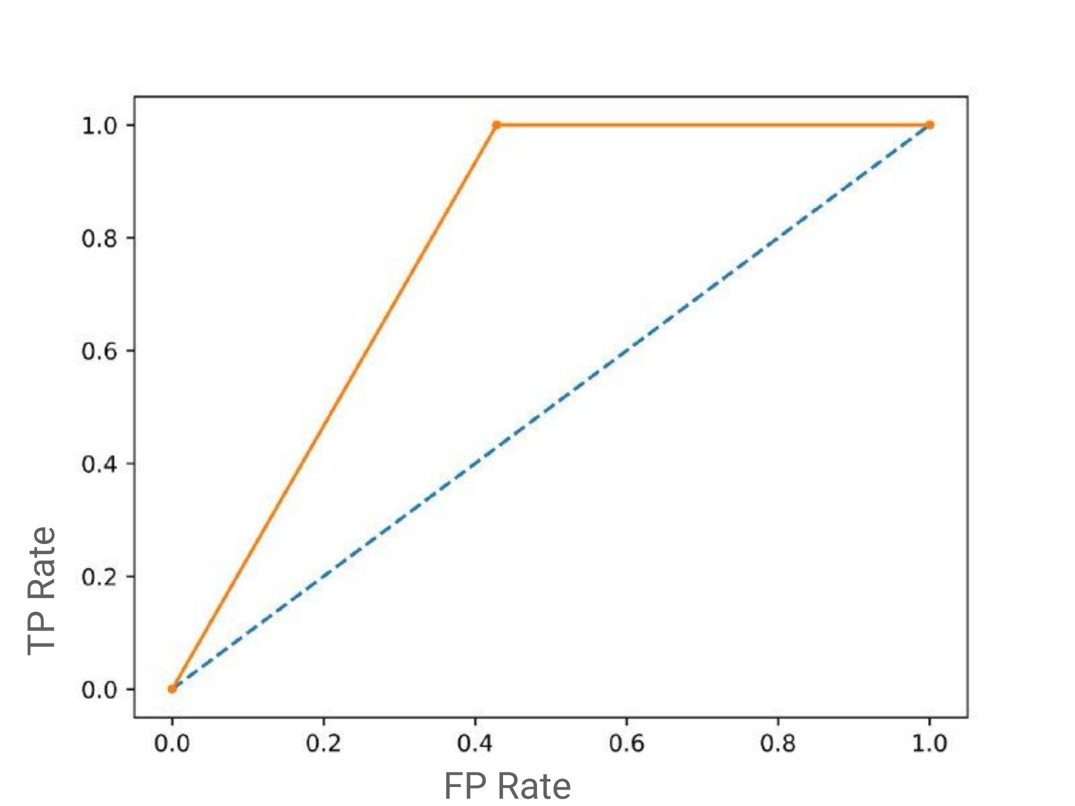
Evaluation: ROC AUC Kurve

CNN mit 2D



Evaluation: ROC AUC Kurve

SAE mit 10D



Fazit

Erfolge:

- Pipeline Setup auf unterschiedliche Systeme
- Deep-Learning-Methoden vielversprechend
- Spannende Erkenntnisse durch Analyse

Ausblick:

- Vereinzelte Fehlklassifikationen verbessern
- Mehr Experimente