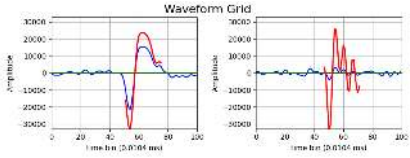


Trennung von Signal und Hintergrund
mit Mitteln des maschinellen Lernens

Signal oder Hintergrund - das ist die Frage

Andreas Zeh-Marschke

09.06.2026

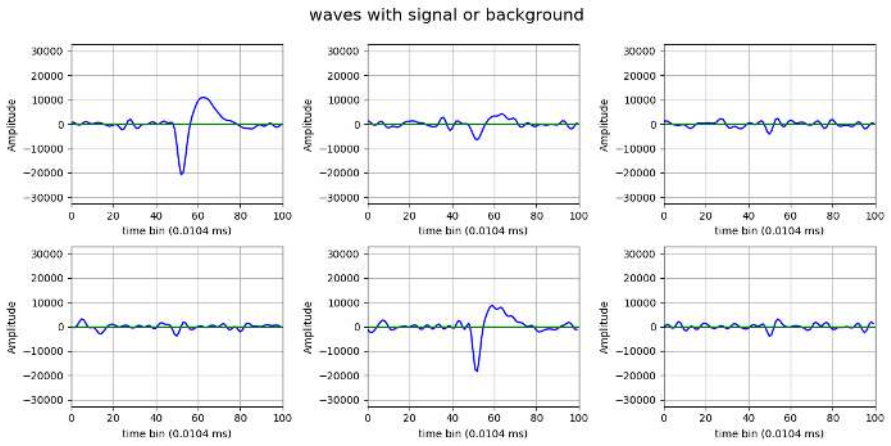
Inhaltsverzeichnis

1. Signal und Hintergrund
2. Daten
3. Maschinelles Lernen
4. Anwendung
5. Zusammenfassung

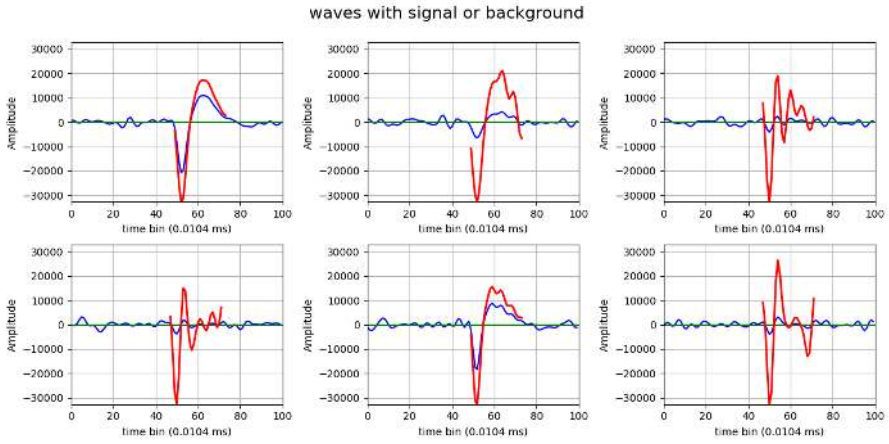
Inhalt

1. Signal und Hintergrund
2. Daten
3. Maschinelles Lernen
4. Anwendung
5. Zusammenfassung

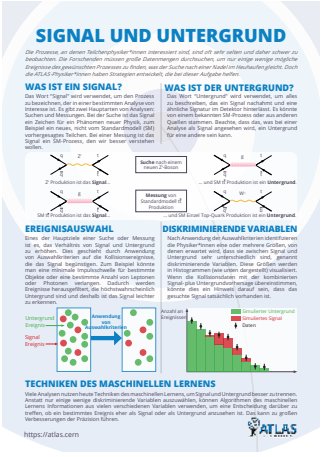
Bilder von Impulsen - Signal oder Untergrund?



Bilder von Impulsen mit Kontur - besser unterscheidbar?



Trennung Signal und Untergrund - wichtige Aufgabe



LHC-ATLAS-Daten

- 600 Mio Kollisionen / s
- 1 MB Daten pro Kollision
- Speicherung: 200 Kollisionen / s
- 4 PB Rohdaten pro Jahr

LHC gesamt: 15 PB / Jahr

- circa 1 Mrd Kollisionen für 1 Higgs-Boson
- effektiv weit mehr, um klare Signale von Higgs-Bosonen zu bekommen

https://cds.cern.ch/record/2814460/files/SignalAndBackground_PhysicsCheatSheet_GER.pdf
https://www.lhc-closer.es/taking_a_closer_look_at_lhc/0.lhc_data_analysis

Science Camp und Workshop Maschinelles Lernen

- Teilnehmer: engagierte, Physik-interessierte Schülerinnen und Schüler (kurz) vor dem Abitur
- Dauer: circa 1 Woche mit Vorträgen zur Teilchenphysik, Workshops und Besuchen von Einrichtungen (Teilchenphysik ganz nah)
- Inhalt im Workshop: Schulungen / Informationen, Diskussionen und Programmierung / Implementierungen in vorgefertigten Programmen mit Lücken.

Bearbeitung von Aufgaben mit den Werkzeugen

- **Python** mit den ergänzenden Modulen: **NumPy**, **MatPlotLib** und **Pandas** - dazu zuerst eine entsprechende Einführung.
- **Jupyter-Notebook**s als Arbeitsumgebung
- **sk-learn** für das maschinelle Lernen
- Technisch genügen Raspberry Pi

Inhalt

1 Signal und Hintergrund

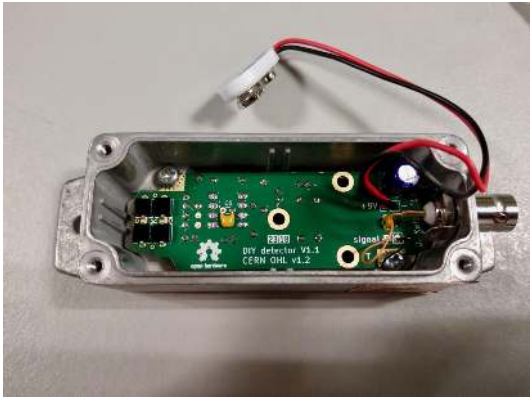
2 Daten

3 Maschinelles Lernen

4 Anwendung

5 Zusammenfassung

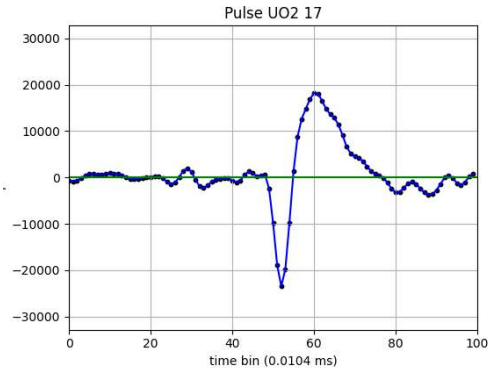
Der CERN DIY Detektor



https://github.com/ozel/DIY_particle_detector

- Aufnahme der Impulse mit Soundkarte
- Abtastrate 96 MHz (0.0104 ms)
- Ausschlag im Bereich 16 bit $[-32768, 32767]$
- Speicherung als 100-Pixel-Impuls, so dass maximaler negativer Ausschlag in der Mitte ist
  Software: *Günter Quast*

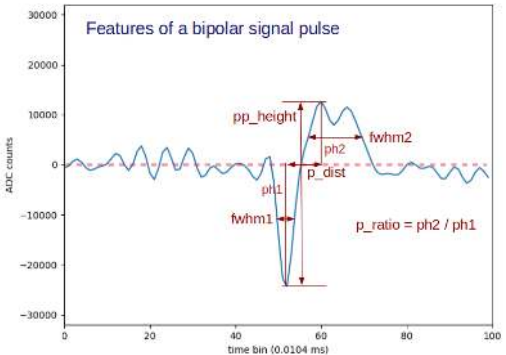
Ein Impuls



- 100 Pixel, Anzahl der Detektionen in einem Zeitintervall von 0.0104 *ms*.
- Verlaufskurve (blaue Linie) und Nulllinie (grüne Linie) nur als Hilfslinie für den Menschen zur besseren Orientierung

Aufgabe für die Teilnehmenden: Implementierung des Plotten eines Impulses in verschiedenen Variationen.

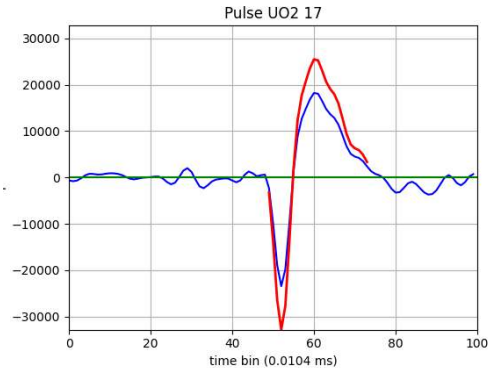
Bipolarer Puls und wichtige Kenndaten



- **event_number**, **event_time**
- **pp_height** peak-to-peak Pulshöhe
- **p_ratio** das Verhältnis der Höhe des (kleineren) zweiten Pulses und des ersten
- **p_dist** der zeitliche Abstand zwischen Maximum und Minimum
- **fwhm_1** und **fwhm_2** die volle Breite bei halber Höhe des ersten (kürzeren) und des zweiten (längeren) Pulsanteils
- **0** . . . **99** die 100 samples im Abstand von etwa 10 *ms*

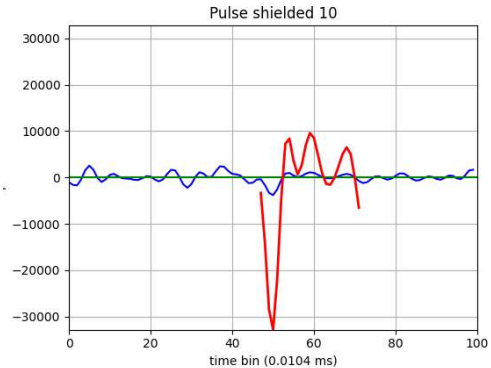
`https://gitlab.kit.edu/kit/etp-lehre/mlsignalanalysis`

Impuls von UO2-Daten



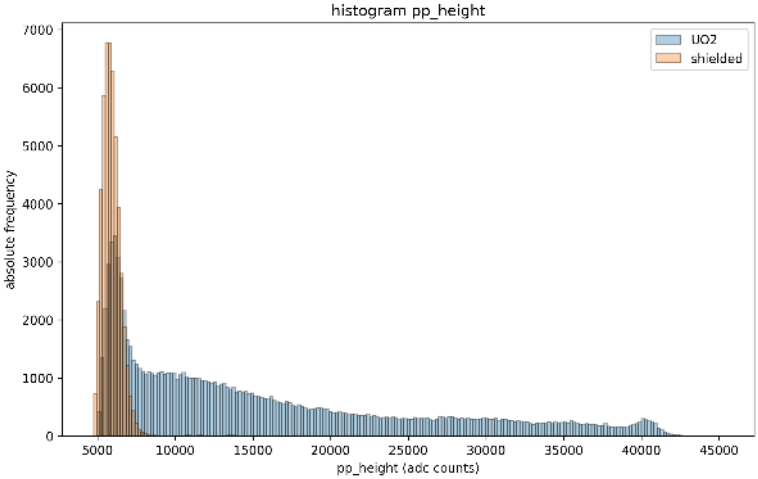
- Aufnahme mit einer Uranoxid-(UO2-)Quelle in der Nähe des Detektors, also viele Impulse
- 25-Pixel-Kontur hervorgehoben und normiert
- deutlicher Verlauf eines Impulses

Impuls von shielded-Daten



- Aufnahme mit einem weitgehend abgeschirmten Detektor, also wenige Impulse, natürliche Impulse aus der Umgebung
- 25-Pixel-Kontur hervorgehoben und normiert
- Overshot ähnelt nicht einem Impuls

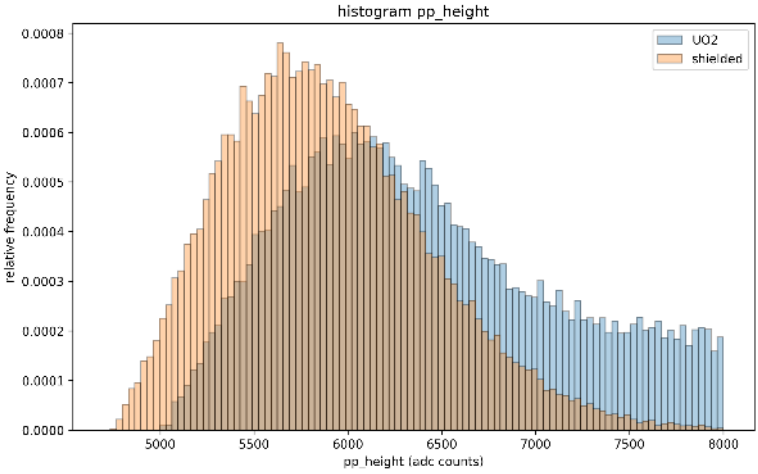
Verteilung von Daten pp_height



UO2-Daten viele Signale mit höherem Ausschlag. shielded-Daten kleinere Ausschläge, aber auch Signale.
Überlappender Bereich!

Aufgabe für die Teilnehmenden: Histogramme mit verschiedenen Variationen erstellen.

Verteilung von Daten `pp_height` - Überlappungsbereich



Welche UO2-Daten sind keine Signale, welche shielded-Daten sind Signale.

`pp_height` (und `ph_1`) nicht geeignet, um Signal und Untergrund zu trennen.

Inhalt

1 Signal und Hintergrund

2 Daten

3 Maschinelles Lernen

4 Anwendung

5 Zusammenfassung

Informationsblock Maschinelles Lernen

Ausgangspunkt: Keine Kenntnisse von maschinellem Lernen!
Daher Informations- / Schulungsblock:

- Was ist maschinelles Lernen (ML)?
- Arten von ML (überwachtes Lernen, nicht überwachtes Lernen)
- Aufgaben: Klassifikation, Regression, Clustern, Dimensionsreduktion
- Sprache des ML: Lerndaten, Trainings- und Testdaten
- Ablauf beim ML: trainieren, vorhersagen und bewerten

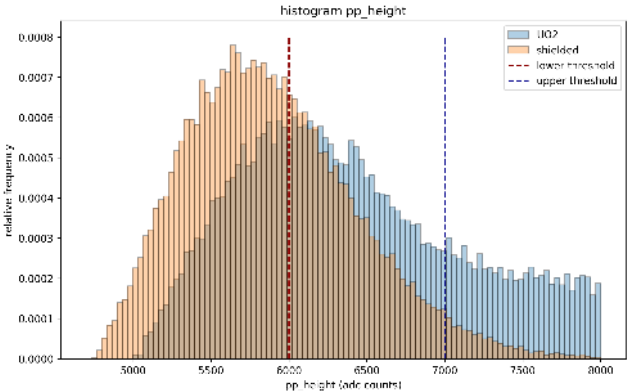
Binäre Klassifikation

- Zwei mögliche Klassen (hier Signal und Untergrund)
- Trainingsdaten für beide Klassen notwendig.
- Wenn Label nicht vorhanden, Label bestimmen, auf Basis der vorhandenen Daten.
- Kennzahlen bestimmen: Genauigkeit (accuracy), Trefferquote (recall), Präzision (precision), ROC, AUC

Aufgabe Lerndaten bestimmen

Diskussion, wie die Lerndaten bestimmt werden, und Implementierung der Routinen.

- Kriterien für die Auswahl der Signale und der Untergrunddaten für die Lerndaten.
- Auswahl durchführen
- Bereinigung der Daten (überzählige Spalten entfernen)
- Lerndaten speichern

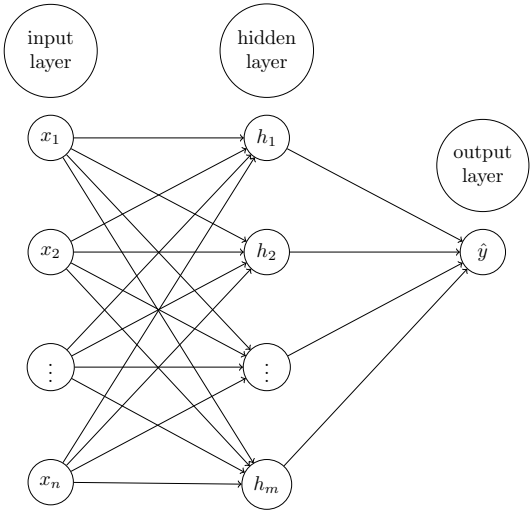


ML-Modell wählen

Information und Diskussion, über ein einfaches neuronales Netz, ein MLP (Multi Layer Perceptron).

- Erläuterung des Aufbaus
- Diskussion der freien Wahlmöglichkeiten
- Bestimmung der Anzahl der Parameter
- Klären Parameter / Hyperparameter
- Erläuterung des Lernprozesses

Wahl für angewendetes ML-Modell: Pixel der Kontur als Eingabelayer, 1 versteckter Layer mit 15 Knoten



Ablauf beim ML

ML durchführen

- splitten der Daten in Trainings- und Testdaten
- skalieren der Daten
- ML-Modell trainieren
- ML-Modell auf den Testdaten anwenden und Bewertung durchführen

Hilfswerkzeug: **sk-learn**

```
# — splitt
train_test_split(X, y, test_size=0.3, ...)

# — select model
model =
MLPClassifier(hidden_layer_sizes=(15,), ...)

# — train
model.fit(X_train, y_train)

# — predict
y_pred = model.predict(X_test)
```

Aufgabe Bewertung implementieren

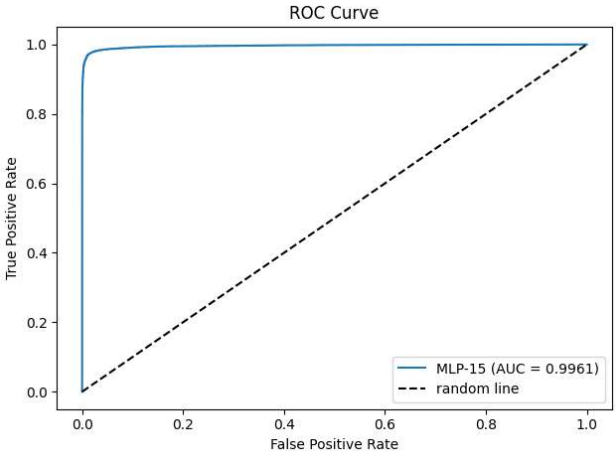
```
 Confusion Matrix       |     predicted label     |
                        | bck         | sig       |
------------------------+-------------+-----------+
             bck        | TN=   9018  | FP=   257 |
 true label  -----------+-------------+-----------+
             sig        | FN=    458  | TP= 24091 |
------------------------+-------------+-----------+

   accuracy    = 0.9789
   precision   = 0.9894    purity
   recall      = 0.9813    sensitivity / efficiency
   FP-rate     = 0.0277    alpha-error
   TP-rate     = 0.9813    1 - beta-error
   F1-score    = 0.9854
   AUC         = 0.9961    AUC (area under curve)
```

Arbeitsauftrag

- Werte von TP (true positive), TN (true negative), FP (false positive) und FN (false negative) mit Hilfe von **sk-learn** bestimmen
- darauf aufbauend andere Kennzahlen ermitteln und ausgeben

Begriffe in der Physik unterscheiden sich manchmal von den Begriffen des Maschinellen Lernens!

Durchführen und Ergebnis betrachten



Das Ergebnis ist gut, obwohl überhaupt nicht optimiert wurde.
Das trainierte Model erzielt auf den Testdaten eine sehr hohe Genauigkeit im Trennen von Signal und Untergrund!

Inhalt

1 Signal und Hintergrund

2 Daten

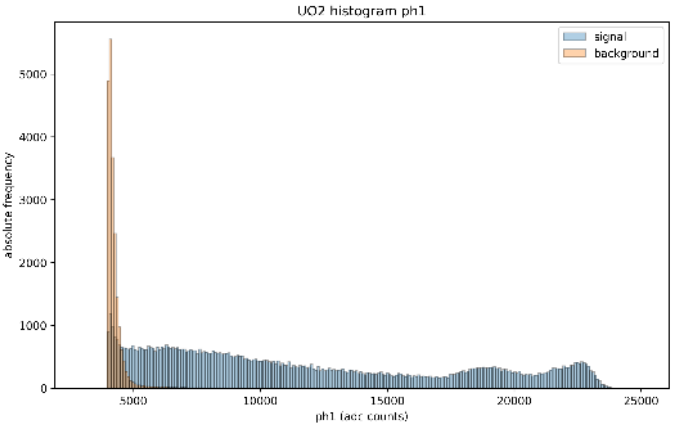
3 Maschinelles Lernen

4 Anwendung

5 Zusammenfassung

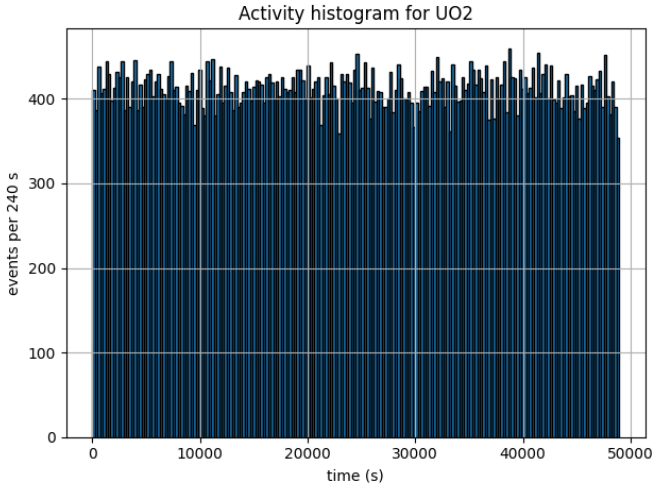
Modell anwenden

- Modell ist trainiert
- Daten einlesen und aufbereiten (Kontur selektieren)
- Modell anwenden und Label (Signal / Untergrund) eintragen.
- Histogramm für Signal / Untergrund bestimmen und Histogramm plotten



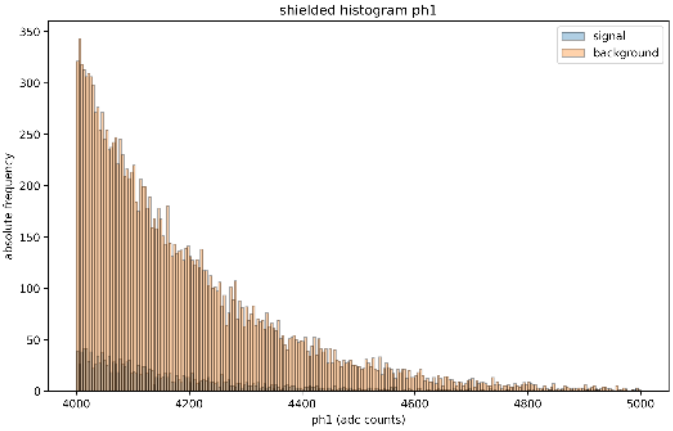
Aktivität bestimmen

- Label gesetzt, also Trennung Signal / Untergrund gemäß dem Modell gesetzt
- Aktivität bestimmen, also radioaktive Zerfälle je Zeit
- Histogramm plotten mit Zeitachse
- Berechnete Aktivität für UO2-Daten: $1.72 \pm 0.08\,Bq$



Modell anwenden - shielded-Daten

- Auch bei shielded Daten wurden Signale erkannt
- Maximalen negativer Ausschlag von Rauschen teilweise nicht zu unterscheiden
- Berechnete Aktivität für shielded-Daten: $0.11 \pm 0.02\,Bq$



Inhalt

1 Signal und Hintergrund

2 Daten

3 Maschinelles Lernen

4 Anwendung

5 Zusammenfassung

Zusammenfassung - was wurde gemacht

- Realisiert mit **python**, **numpy**, **pandas**, **matplotlib** und **sklearn** auf Basis von vorgefertigten Jupyter Notebooks.
- Python-Einführung gemeinsam mit der Higgs-Gruppe.
- Vorausgefüllte Jupyter Notebooks mit Lücken, die fertig gestellt wurden, um verschiedene Aufgaben umzusetzen. (Histogramme erstellen, Impulse plotten, ML-Modelle trainieren und anwenden)
- Parallel dazu inhaltlicher Komponenten um ML zu lernen und damit die Bedeutung der Implementierungen zu verstehen.
- Geeignet für die Arbeit mit Studienanfängern und engagierten Schülerinnen und Schüler.
- Viele Möglichkeiten für erweiterte Fragestellungen und Umsetzungen.

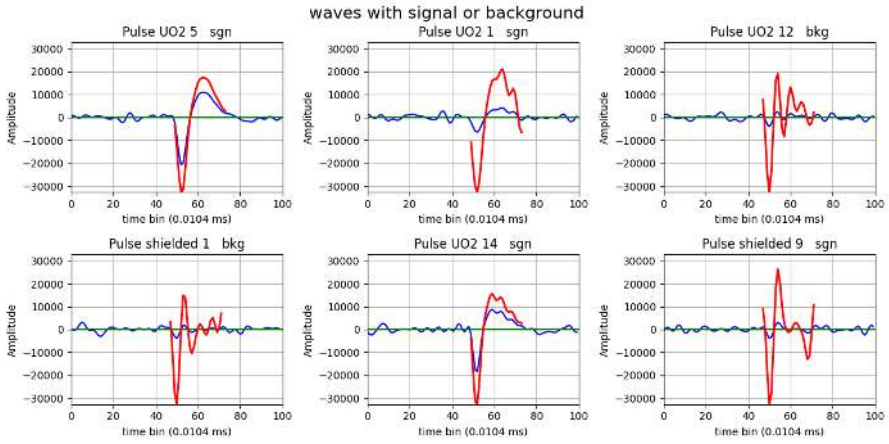
Zusammenfassung - was wurde mitgenommen

- Einfaches Verfahren zur **1-dimensionalen Bilderkennung** mit einer hohen Erkennungsquote zur **Signal-Untergrund-Trennung**, einer wichtigen Aufgabe in der Physik.
- Programmieren lernen mit einer **Python-Umgebung** (Python, numpy, matplotlib, pandas, sk-learn und Jupyter-Notebook).
- **Maschinelles Lernen** kennen lernen (nur ein sehr kleiner Ausschnitt: Binäre Klassifikation, einfaches neuronales Netz).

Ergänzende Aufgaben als Ausgangspunkt für weitere physikalische und technische Fragestellungen:

- andere Messdaten untersuchen (Umgebung, Stein aus dem Schwarzwald, Radon)
- weitere ML-Verfahren untersuchen;
- Untersuchung welche Impulse werden stets als Signal oder Untergrund erkannt werden, welche von unterschiedlichen ML-Verfahren unterschiedlich gekennzeichnet werden.

Zusammenfassung - wo sind Unterlagen

- Unterlagen verfügbar, die verwendet und angepasst werden können. Es ist eine Basis, für dynamische und kreative Anpassungen.
- Für die Workshops können Teile der Implementierung entfernt werden, so dass die Schülerinnen und Schüler diese Füllen können.
- Unterlagen liegen auf `https://github.com/Zeh-Marschke/SigUnt`,
  die Daten in `https://gitlab.kit.edu/kit/etp-lehre/mlsignalanalysis`
  Unterverzeichnis: data
- Workshop basiert auf der Bachelorarbeit:
  Zeh-Marschke, Andreas. (2025, January 20). Einsatz von Methoden des Maschinellen Lernens für eine Binäre Klassifikation zur Signal-Hintergrund-Trennung.

Bilder von Impulsen mit Kontur und Vorhersage