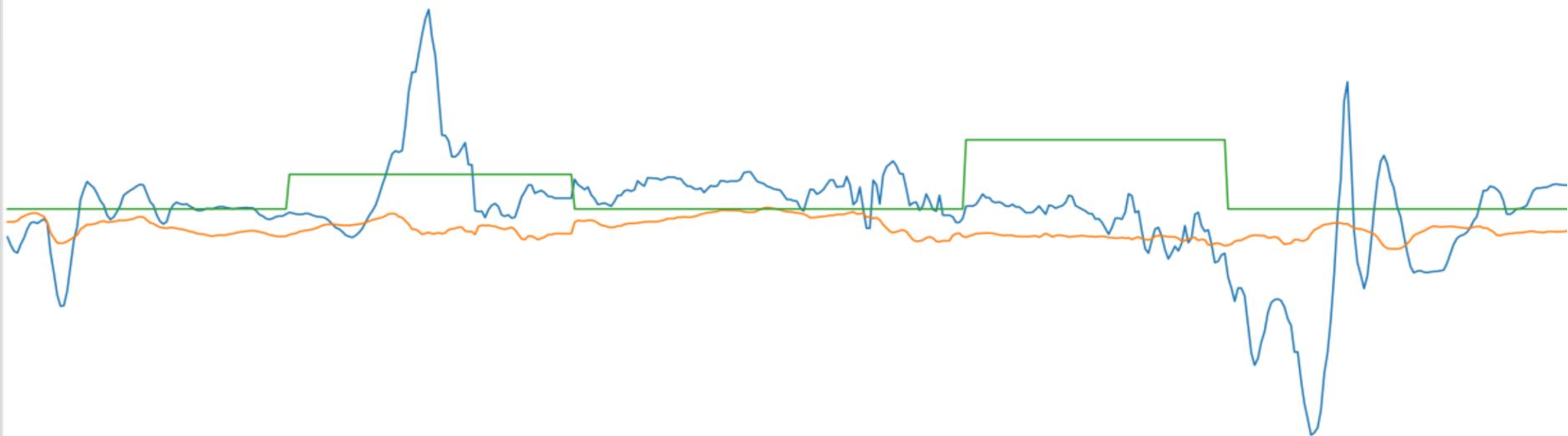


Smart Data Analytics

Kinemic – Klassifikation von Bewegungsdaten

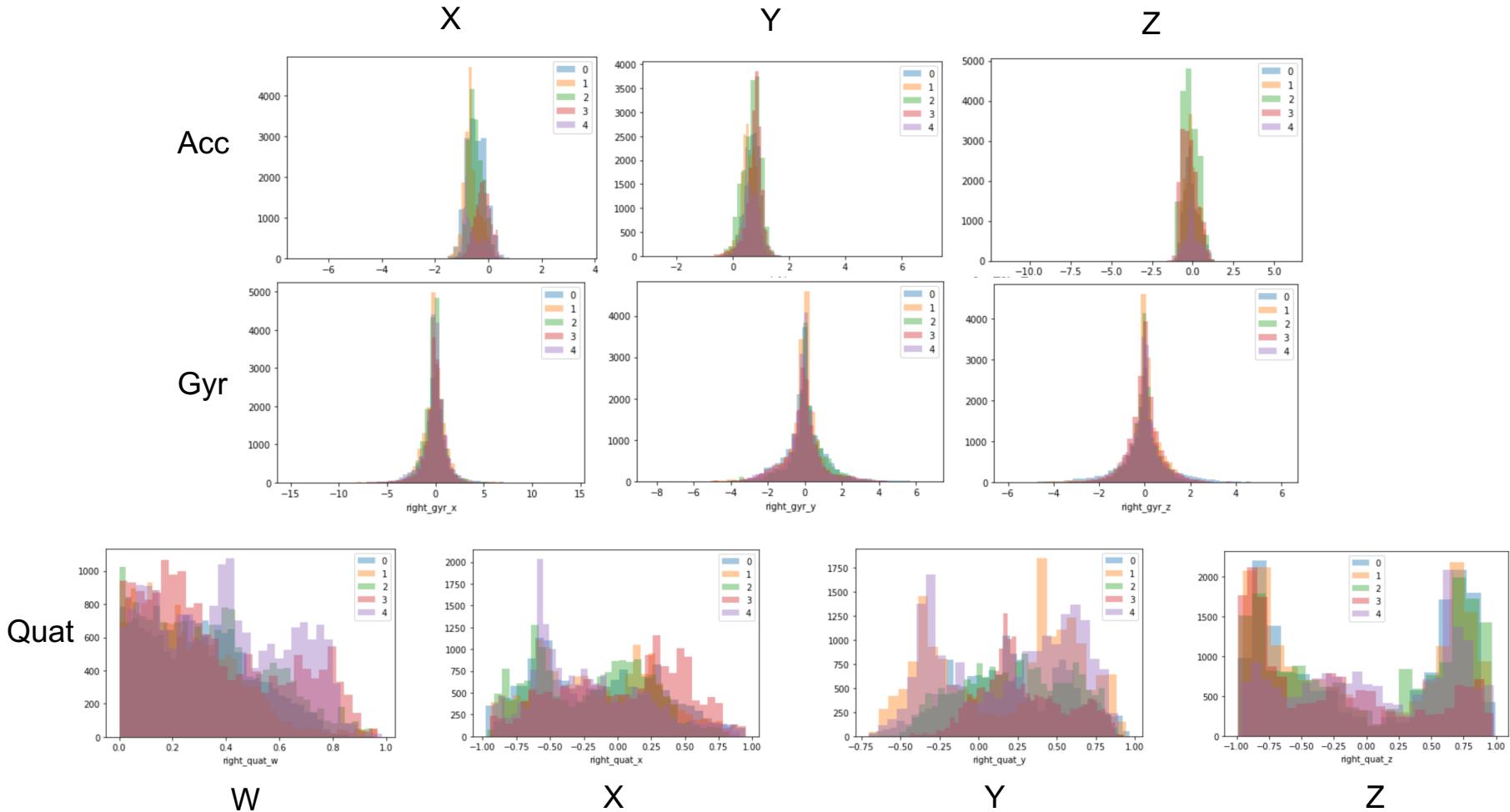
Zihan Chen, Thassilo Helmold, Gerrit Merz, Robin Schnaitt – Team 4



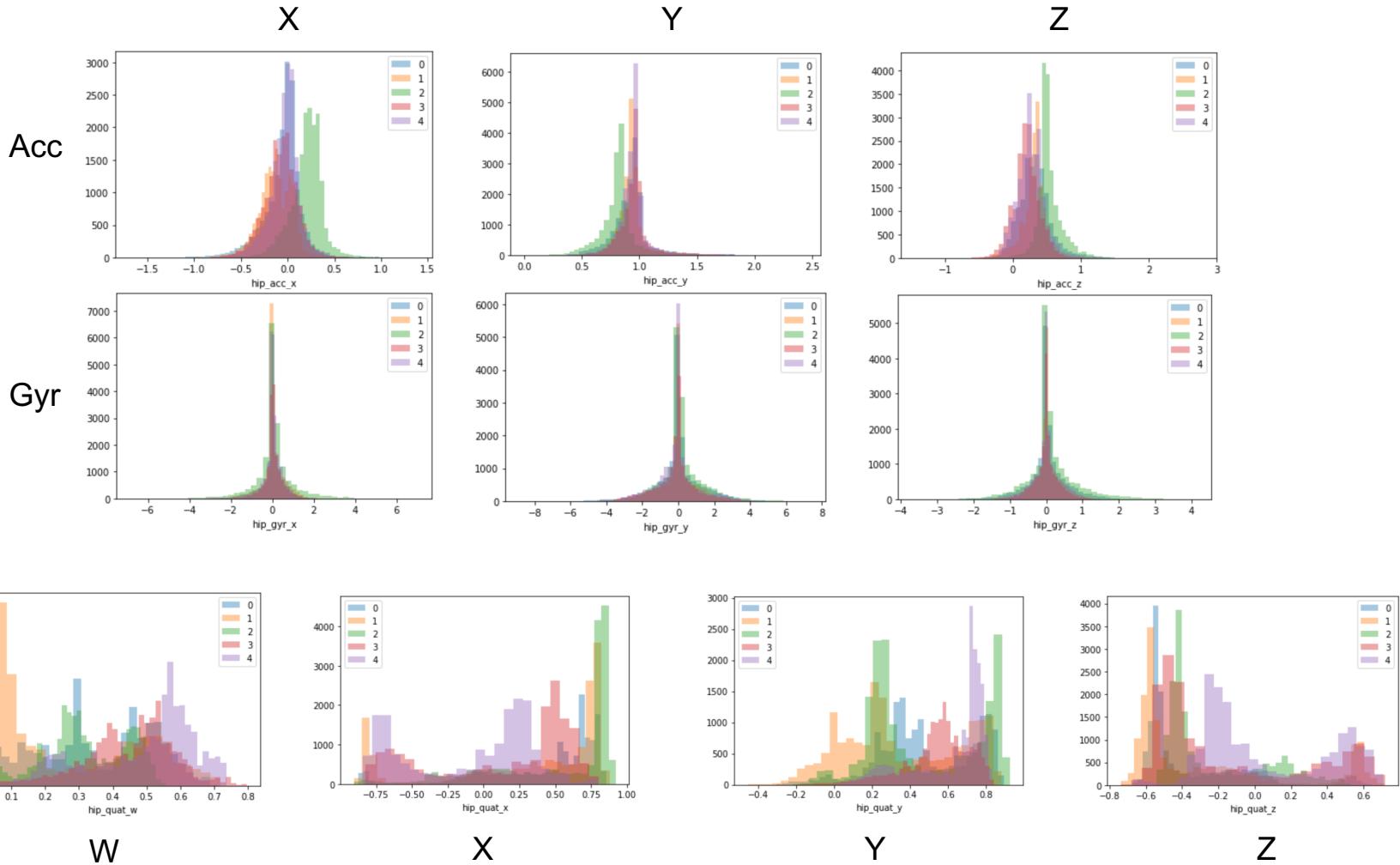
Inhalt

1. Data Exploration
2. Pre-Processing und Data-Pipelines
3. Convolutional Neural Nets
 - Simplenet
 - Stacknet
 - Multinet
 - Sepnet
4. Recurrent Neural Nets
 - LSTM
5. Random Forest
6. Kombination von Modellen: Hybridnet
7. Ergebnisübersicht und Zusammenfassung

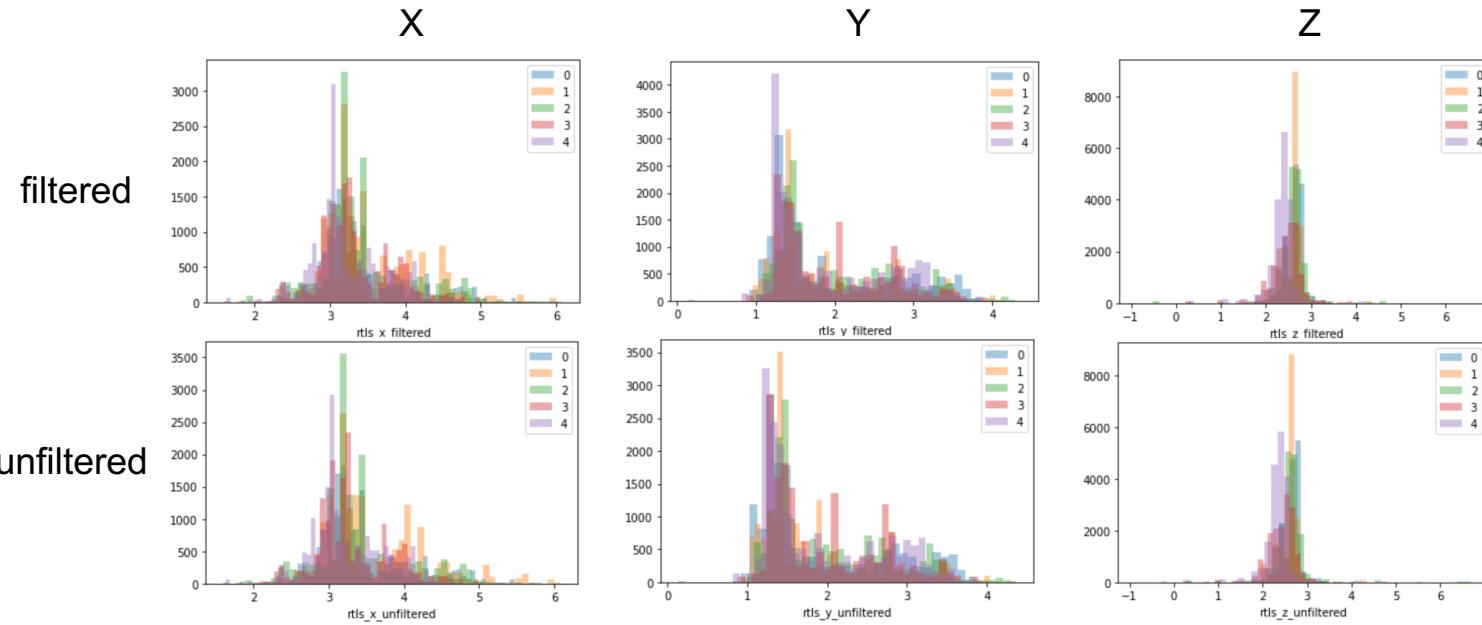
Data Exploration – Right



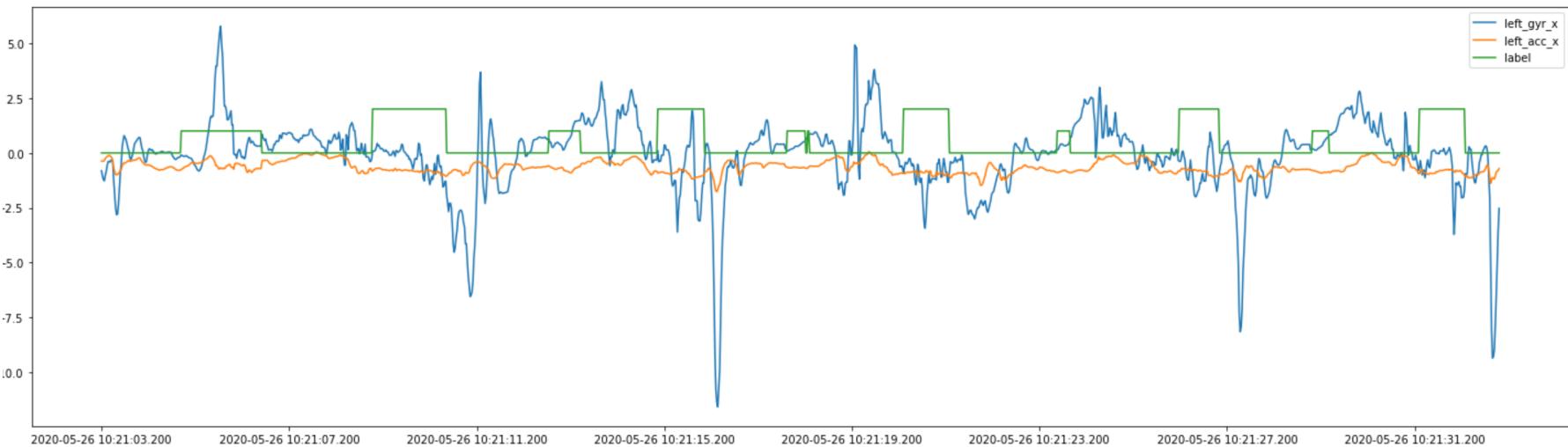
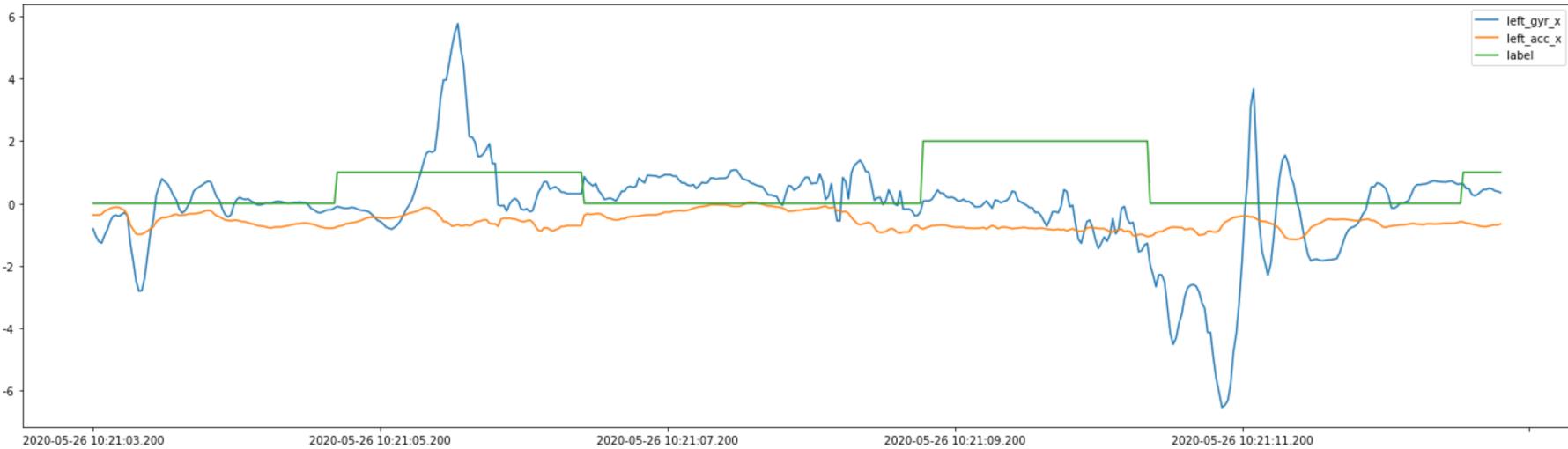
Data Exploration - Hip



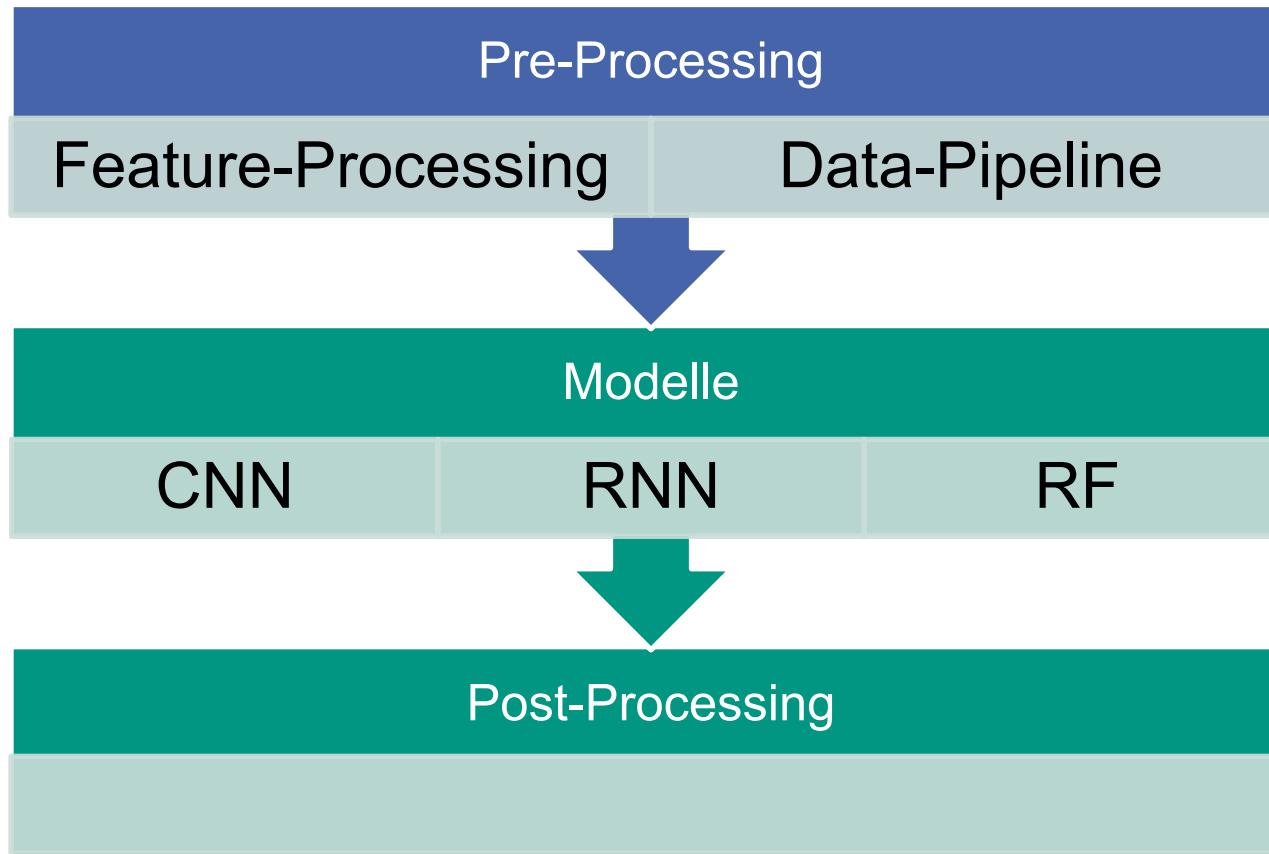
Data Exploration - RTLS



Data Exploration

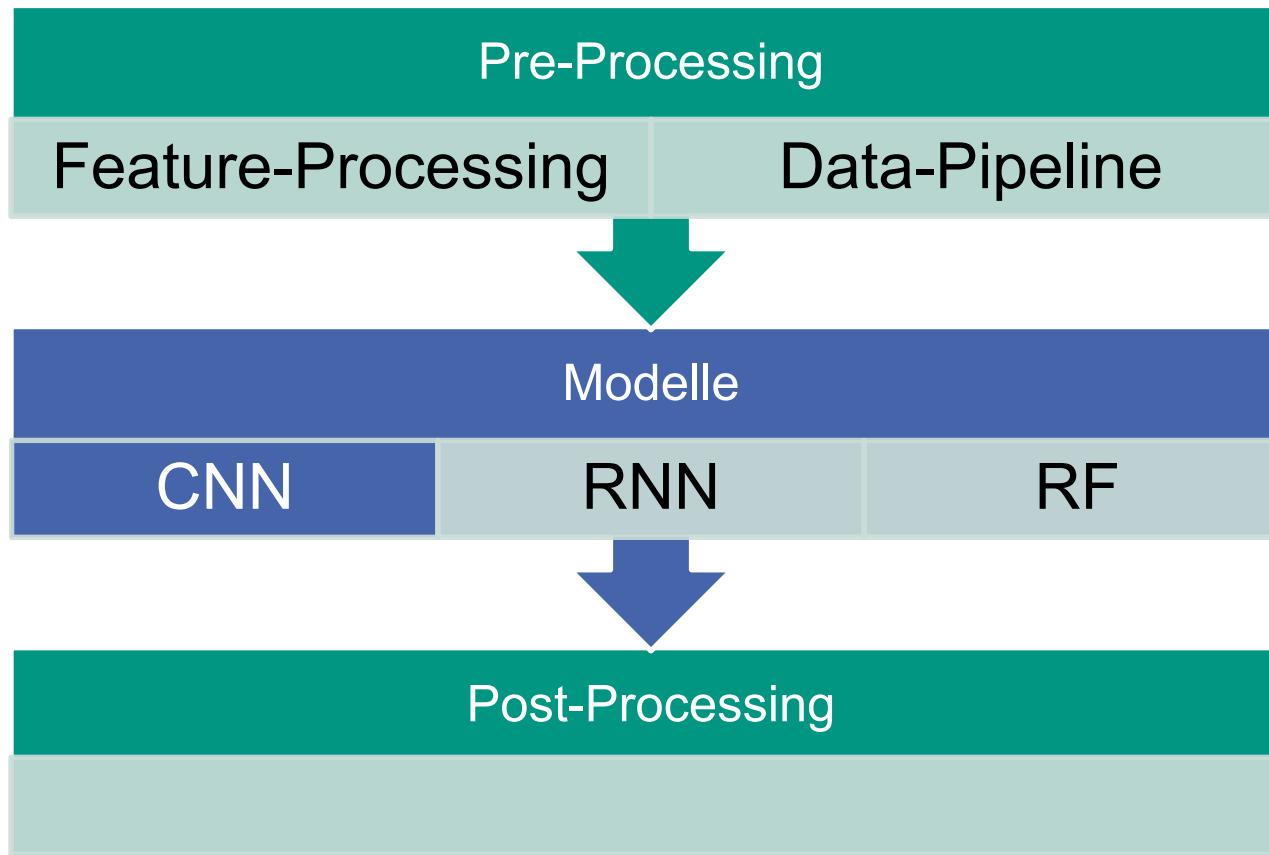


Klassifizierung: Prozess und Pipeline



Data Pipeline

- Herausforderung: Windowing
- Targets müssen passend festgelegt werden
- Verwendung von *Tensorflow-Datasets*
- Vorteile:
 - Effizienter (ca. 4x schneller)
 - Alle Daten in einer Epoche
 - Gleichmäßiges Interleaving der einzelnen Aufnahmen
 - Auch für die Prediction einsetzbar
- Unbalancierte Daten => Gewichtung der Klassen

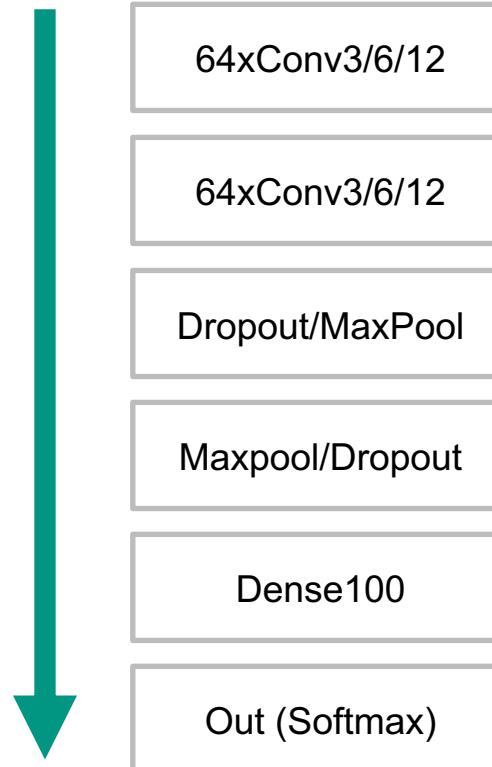


CNN

- 1D-Convolutions über die Zeit
 - Auch: Time-Delay-Neural-Network (Waibel)
 - „Natürliche Architektur“ für temporale Daten: Betrachte den Kontext eines Samples, nicht nur es selbst
 - Erlaubt die Eingabe verschieden langer Aufnahme-Windows

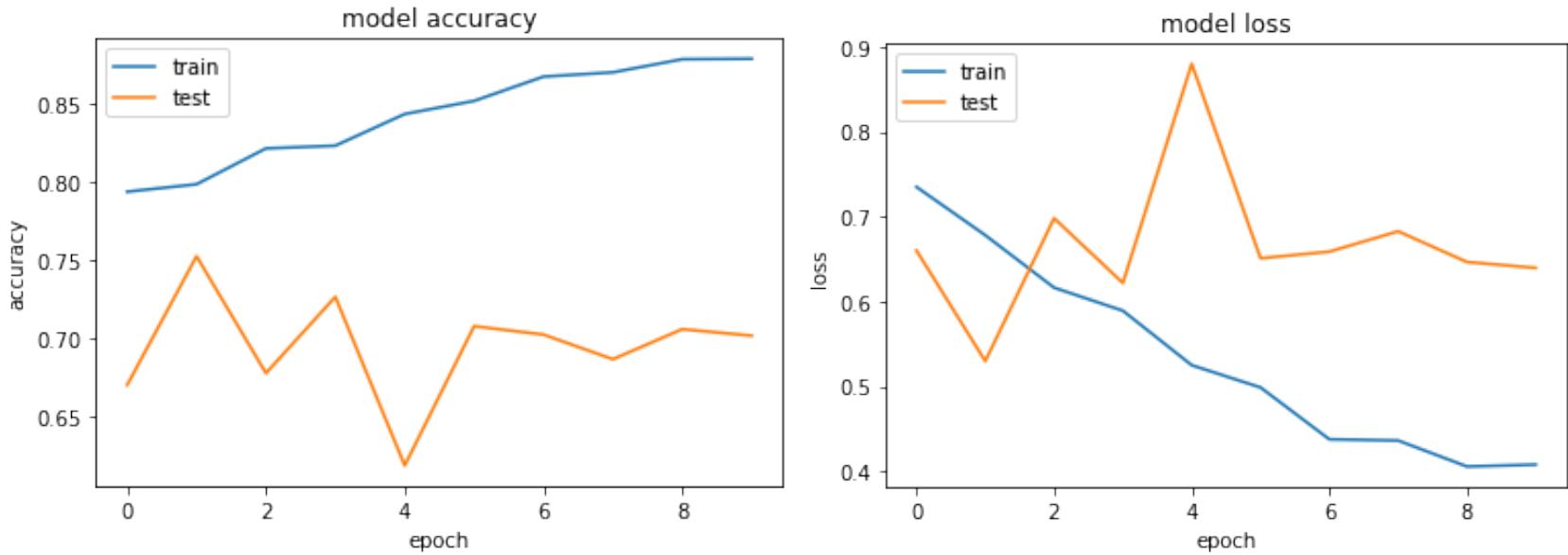
Simplenet

- Orientiert sich an der vorgeschlagenen Beispiel-Architektur
- Modifikationen an Convolution- und Regularisierungslayern



Simplenet: Training

- Beispiel, Simplenet mit Kernelgröße 6:



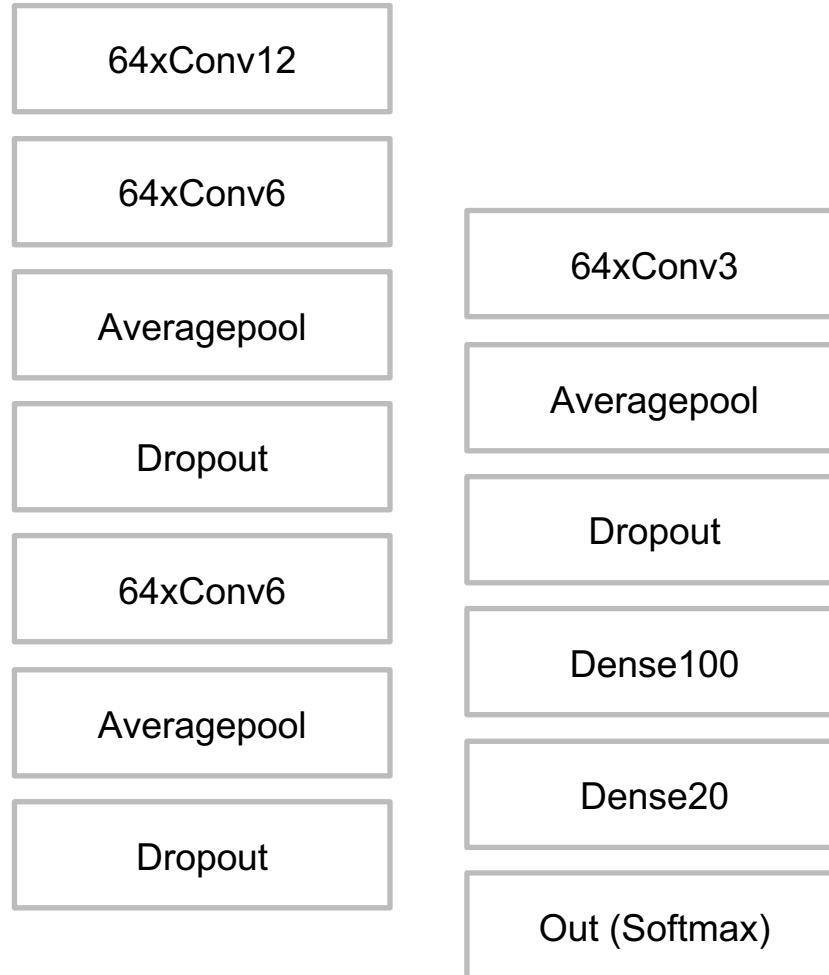
- Neigt zum Overfitting, schwer über ~70% Accuracy zu erreichen
=> Architekturen mit mehr Lernkapazität und Regularisierung
- LR-Scheduling und tuning der Hyperparameter: Nur geringe Verbesserung

Simplenet: Performance

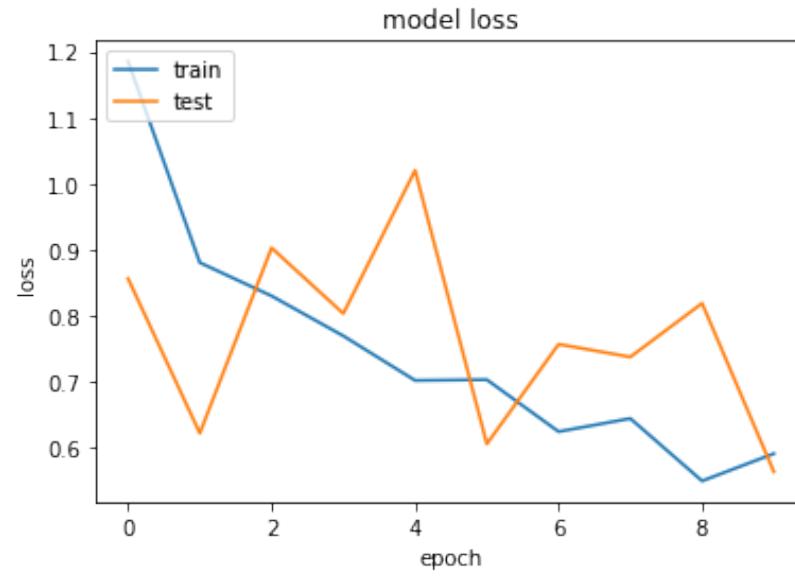
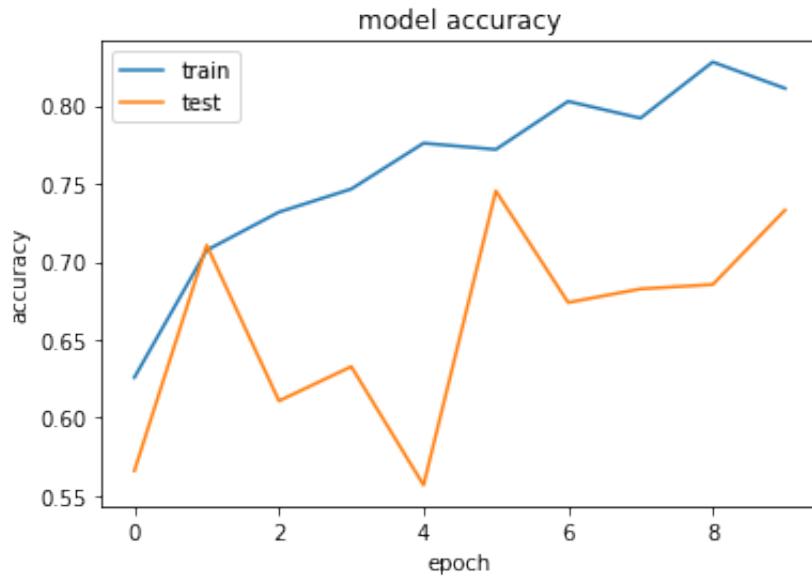
Modellname	Half Window	Accuracy Validation
Baseline	0	0.77
Simplenet-k3	15	0.63
Simplenet-k3	15	0.53
Simplenet-k3	15	0.56
Simplenet-k3	15	0.63
Simplenet-k3	100	0.68
Simplenet-k6	15	0.69
Simplenet-k6	50	0.68
Simplenet-k6-pooldrop	50	0.7
Simplenet-k6-pooldrop0.3	50	0.7
Simplenet-k6-pooldrop0.3	50	0.66
Simplenet-k6-pooldrop0.3	50	0.7
Simplenet-k6-pooldrop0.2	50	0.7
Simplenet-k6-pooldrop0.7	50	0.69
Simplenet-k6	50	0.7
Simplenet-k12	100	0.7

Stacknet

- Nutze die verschiedenen getesteten Simplenets als Layers eines CNNs



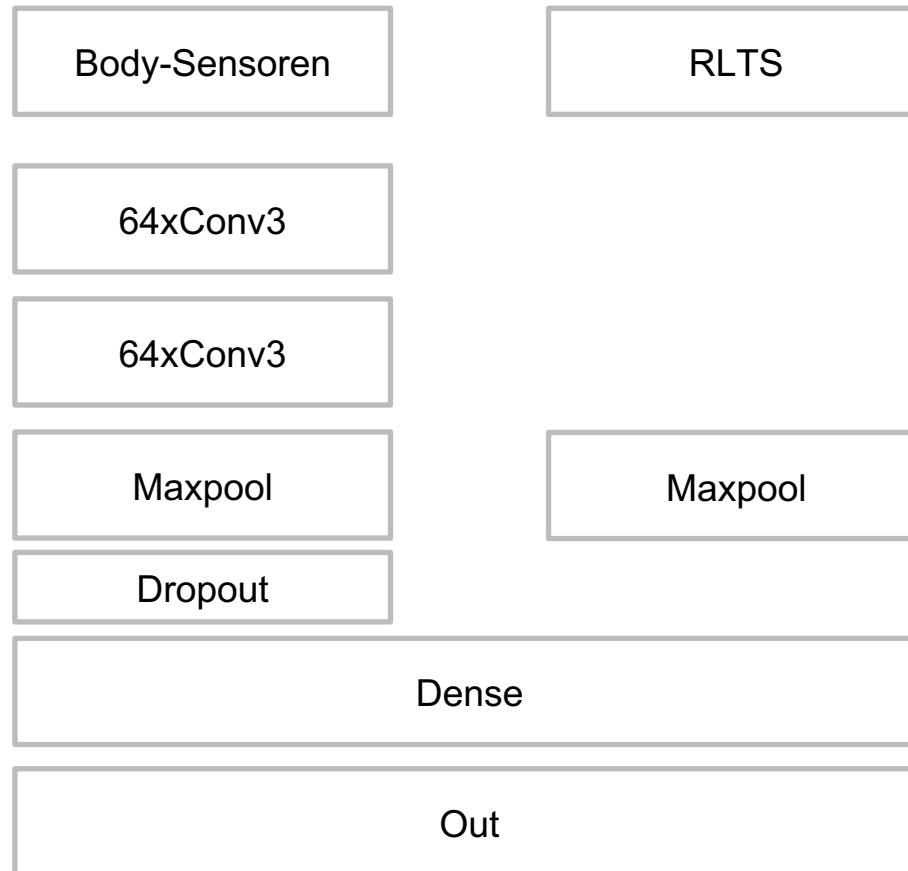
Stacknet: Training



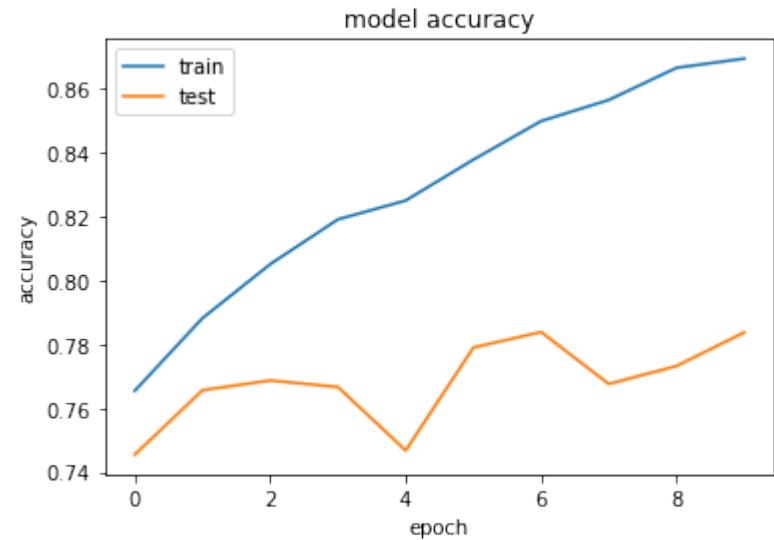
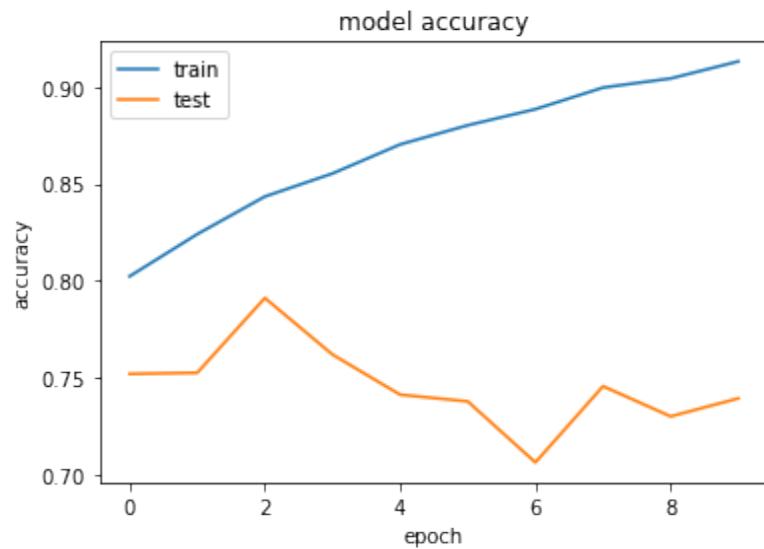
- Etwas leichter, Overfitting zu verhindern
- LR-Scheduling (Exponential Decay) essentiell
- Wir suchen nach weiteren Architekturen, die besser geeignet sind als das Stacking der einfachen CNNs

Multinet

Idee: Körpersensoren separat von RTLS-Werten verarbeiten



Ergebnisse



	precision	recall	f1-score	support
0.0	0.90	0.74	0.82	64014
1.0	0.51	0.69	0.58	15557
2.0	0.34	0.85	0.49	3828
accuracy			0.74	83399

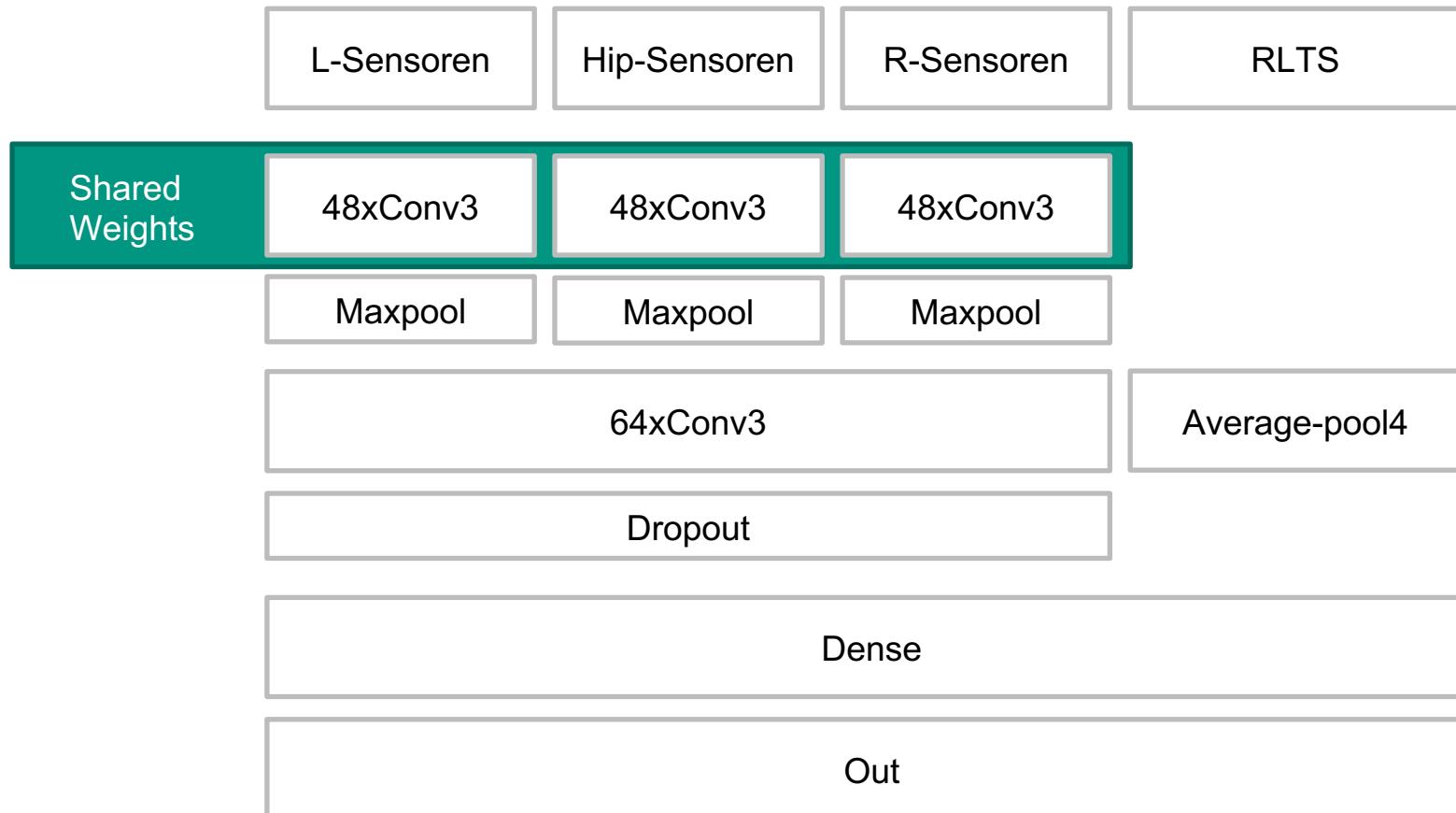
	precision	recall	f1-score	support	
0.0	0.0	0.90	0.81	0.85	64314
1.0	1.0	0.59	0.73	0.66	15557
2.0	2.0	0.31	0.48	0.37	3828
accuracy			0.78	83699	

Multinet-{1,2,3}

- Verschiedene Versionen:
 - Mehr Dropout
 - Mehr Filterkernel, größere Filterkernel
 - AveragePooling im RTLS
 - Zusätzliche Filterschicht im RTLS
 - Mehr Dense Layers

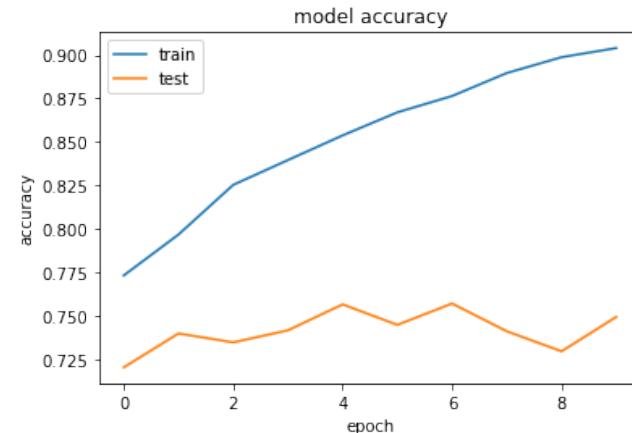
Sepnet

Idee: Alle Sensoren separat verarbeiten



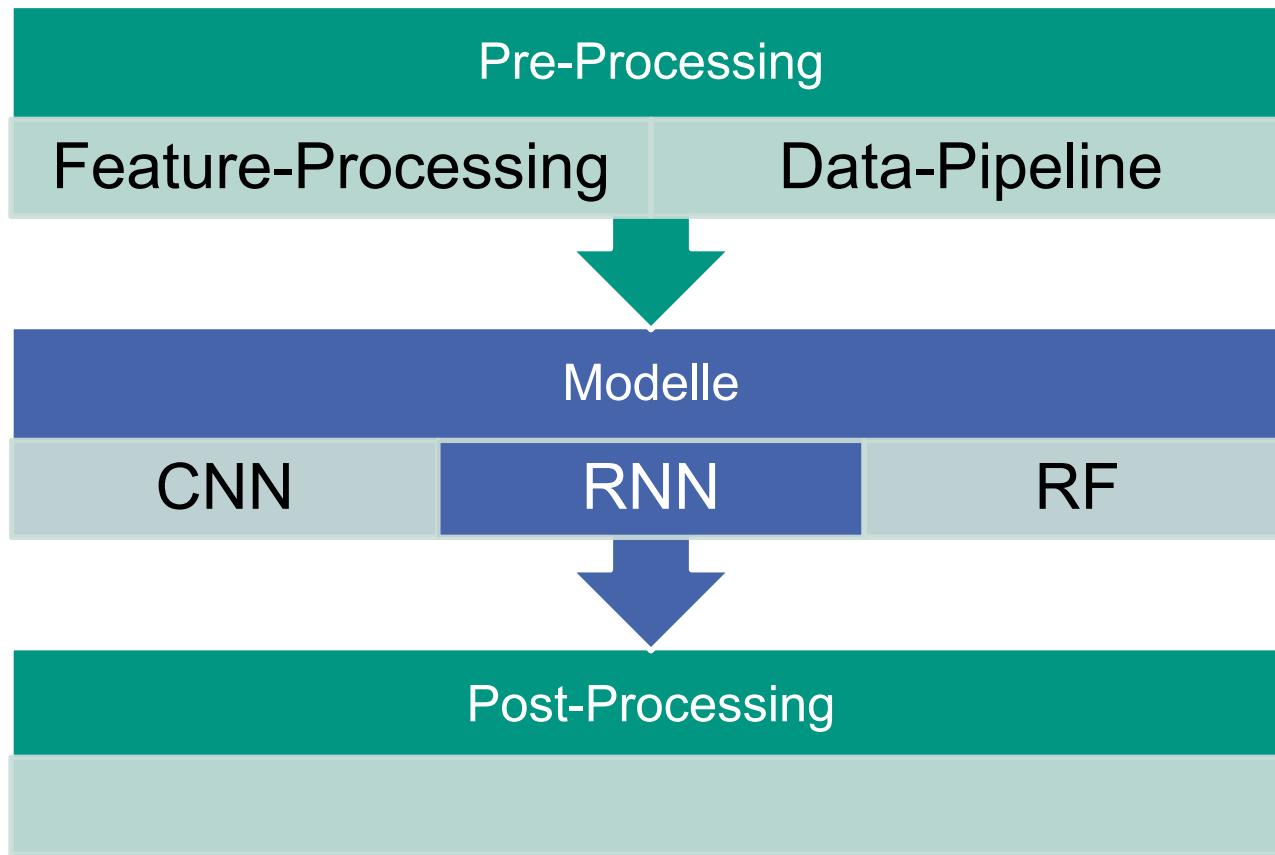
Sepnet

- Auch hier wieder verschiedene Versionen
- Ergebnisse vergleichbar mit Multinet, aber Trainingsdauer 3-5x so lange

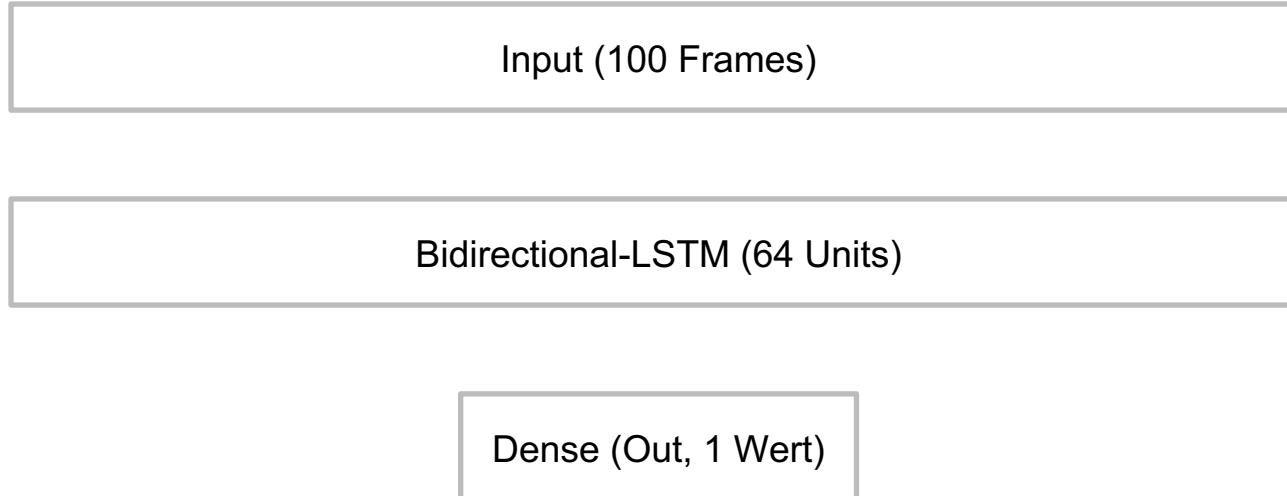


Deepnet

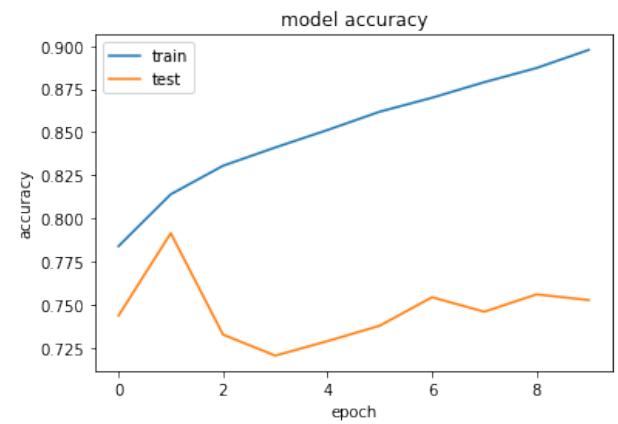
- Sensoren: Conv(64)->Pool->Conv(96)->Pool->Conv(128)->Pool
- RTLS: Pool->Conv(32)->Pool
- Dauert lange, bringt nix



Many-To-One Ansatz



- Nachteil: Training dauert *ewig* (5 Min / Epoche)



Many-To-Many Ansatz

Input (500 Frames)

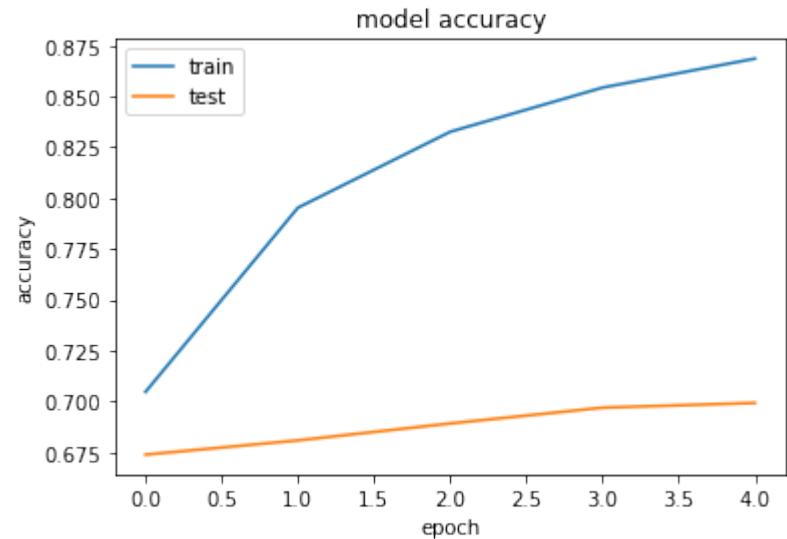
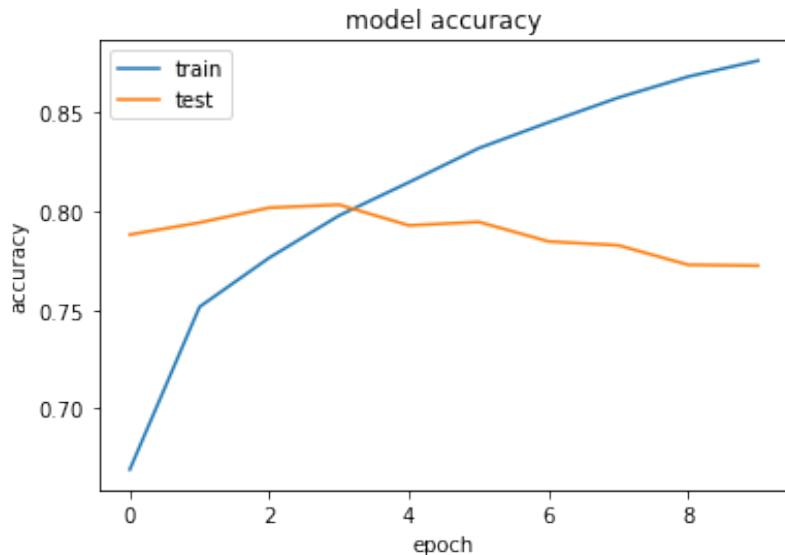
LSTM (64 Units)

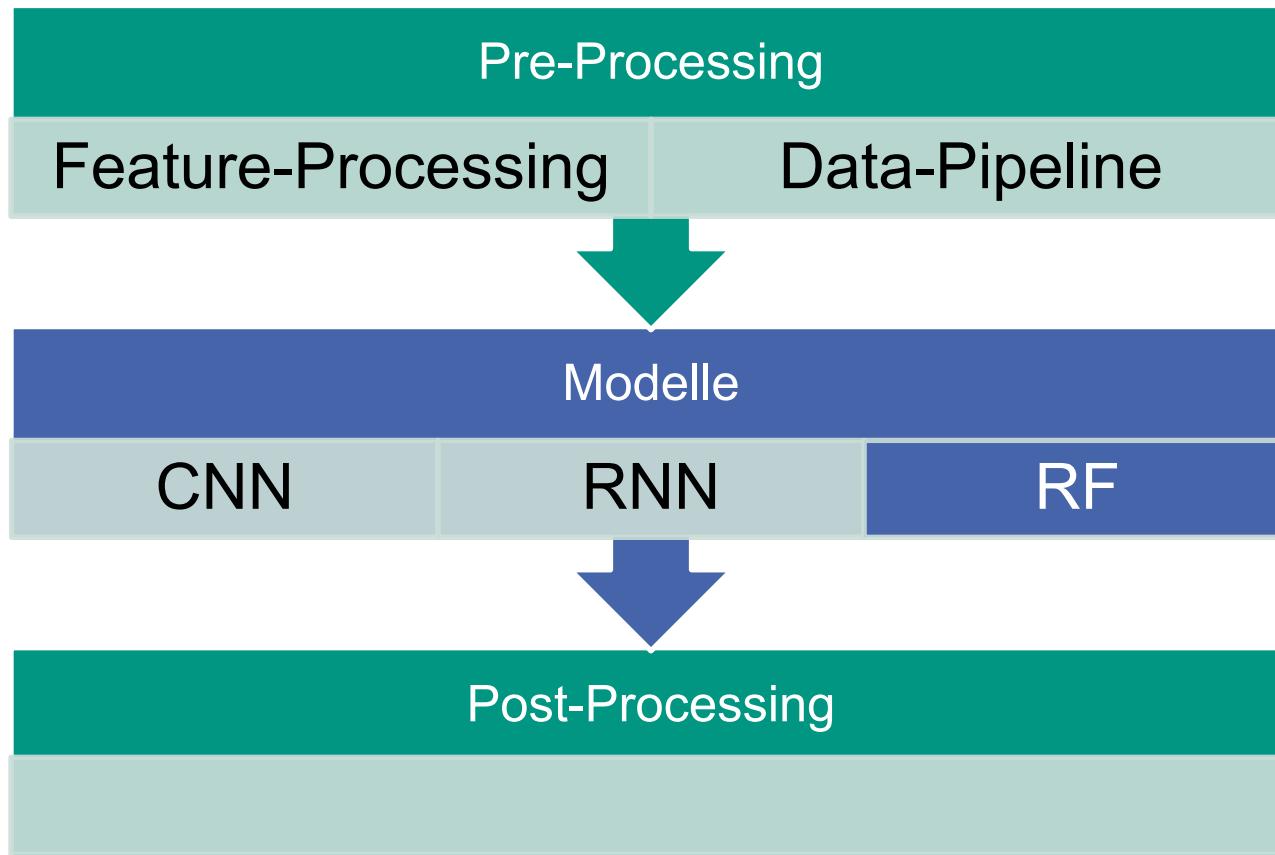
Dense (Out, 500 Werte)

- Training: Stride 50, jeweils alle Outputs trainieren
- Prediction: Stride 300, jeweils nur mittlere 300 Werte nehmen

Ergebnisse

- Zunächst vielversprechend
- Starke Probleme mit Overfitting





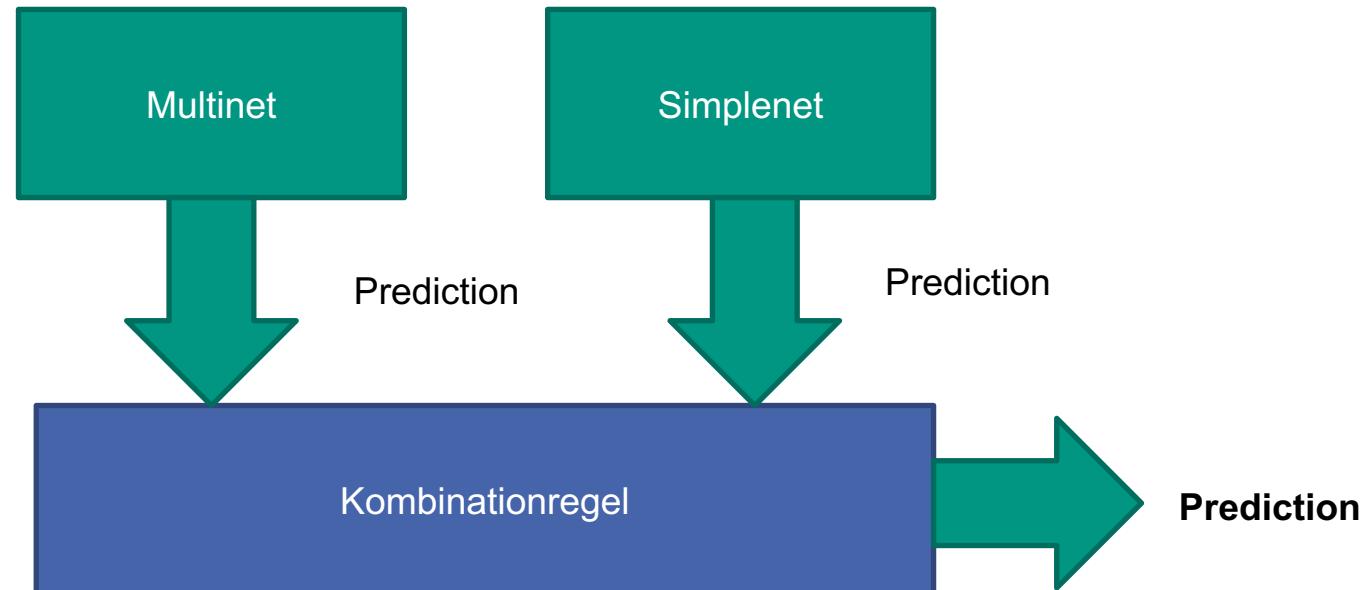
Random Forest Classifier

- Arbeitet direkt auf Sliding Windows, Kontext 15 Samples
 - Konversion nach Numpy-Arrays, hier funktionieren keine TF-Datenframes
 - Scikit-Implementierung: RandomForestClassifier
 - Estimators: 20, Max. Features: 30 => Validation Accuracy 84%
 - Estimators: 100, Max. Features: 30 => Validation Accuracy 85%

		Accuracy		Precision			Recall			F-score		
RF-20-30	15	0.84	0.87	0.72	0.45	0.92	0.61	0.3	0.9	0.66	0.36	
RF-100-30	15	0.85	0.87	0.77	0.47	0.94	0.64	0.2	0.91	0.7	0.28	

Experiment: Hybridnet

- Verknüpfe die Predictions mehrerer Modelle, deren Precision und Recall sich ergänzt (z.B. Multinet und Simplenet-Variante)
 - Trainiere separat
 - Predictions zusammenführen und verknüpfen, basierend auf Aprioriwissen der Performance-Charakteristiken. + ~1-2%-Punkte Acc.
 - Erweiterung: Domänenwissen einbinden, Sanitization



Ergebnisübersicht

Modellname	Half Window	Accuracy Validation	Precision 0 Val	Precision 1 Val	Precision 2 Val	Recall 0 Val	Recall 1 Val	Recall 2 Val	f1-score 0 Val	f1-score 1 Val	f1-score 2 Val
Baseline	0	0.77	0.77	0	0	1	0	0	0.87	0	0
Simplenet-k3	15	0.63									
Simplenet-k3	15	0.53									
Simplenet-k3	15	0.56									
Simplenet-k3	15	0.63									
Simplenet-k3	100	0.68	0.94	0.49	0.26	0.62	0.84	0.97	0.75	0.62	0.41
Simplenet-k6	15	0.69	0.91	0.53	0.18	0.67	0.79	0.63	0.77	0.64	0.28
Simplenet-k6	50	0.68	0.92	0.47	0.27	0.64	0.77	0.94	0.76	0.59	0.42
Simplenet-k6-pooldrop	50	0.7	0.91	0.51	0.26	0.68	0.72	0.92	0.77	0.6	0.41
Simplenet-k6-pooldrop0.3	50	0.7	0.9	0.5	0.25	0.68	0.72	0.81	0.78	0.59	0.39
Simplenet-k6-pooldrop0.3	50	0.66	0.97	0.5	0.22	0.57	0.92	0.99	0.72	0.65	0.36
Simplenet-k6-pooldrop0.3	50	0.7	0.87	0.47	0.28	0.71	0.62	0.75	0.78	0.53	0.41
Simplenet-k6-pooldrop0.2	50	0.7	0.9	0.48	0.28	0.69	0.72	0.82	0.78	0.58	0.42
Simplenet-k6-pooldrop0.7	50	0.69	0.92	0.49	0.26	0.65	0.77	0.93	0.76	0.6	0.4
Simplenet-k6	50	0.7	0.9	0.53	0.25	0.69	0.7	0.87	0.78	0.6	0.39
Simplenet-k12	100	0.7	0.95	0.49	0.29	0.64	0.87	0.97	0.76	0.63	0.44
Stacknet	50	0.73	0.95	0.52	0.31	0.69	0.89	0.85	0.8	0.66	0.46
Multinet-Base	15	0.76	0.9	0.61	0.28	0.78	0.69	0.74	0.84	0.65	0.41
Multinet-2-pooldrop	15	0.77	0.87	0.64	0.29	0.82	0.57	0.71	0.85	0.6	0.41
Multinet-2-pooldrop	25	0.78	0.9	0.59	0.31	0.81	0.73	0.48	0.85	0.66	0.37
Multinet-3-multidrop	25	0.77	0.88	0.68	0.3	0.82	0.56	0.85	0.85	0.61	0.45
Sepnet-1	25	0.77	0.91	0.6	0.31	0.78	0.68	0.88	0.84	0.64	0.46
Sepnet-2	25	0.75	0.9	0.59	0.28	0.77	0.64	0.83	0.83	0.62	0.41
Sepnet-3	25	0.75	0.89	0.62	0.27	0.77	0.64	0.85	0.83	0.63	0.41
Deepnet	50	0.74	0.9	0.51	0.34	0.74	0.69	0.85	0.82	0.58	0.49
Hybridnet	25	0.76	0.86	0.58	0.32	0.83	0.5	0.68	0.85	0.54	0.43
RF-20-30	15	0.84	0.87	0.72	0.45	0.92	0.61	0.3	0.9	0.66	0.36
RF-100-30	15	0.85	0.87	0.77	0.47	0.94	0.64	0.2	0.91	0.7	0.28
LSTM-Many-to-One	50	0.75	0.91	0.58	0.3	0.75	0.75	0.82	0.82	0.65	0.43
LSTM-One-to-One	500	0.8	0.91	0.65	0.38	0.82	0.7	0.86	0.86	0.67	0.53

Ergebnisübersicht

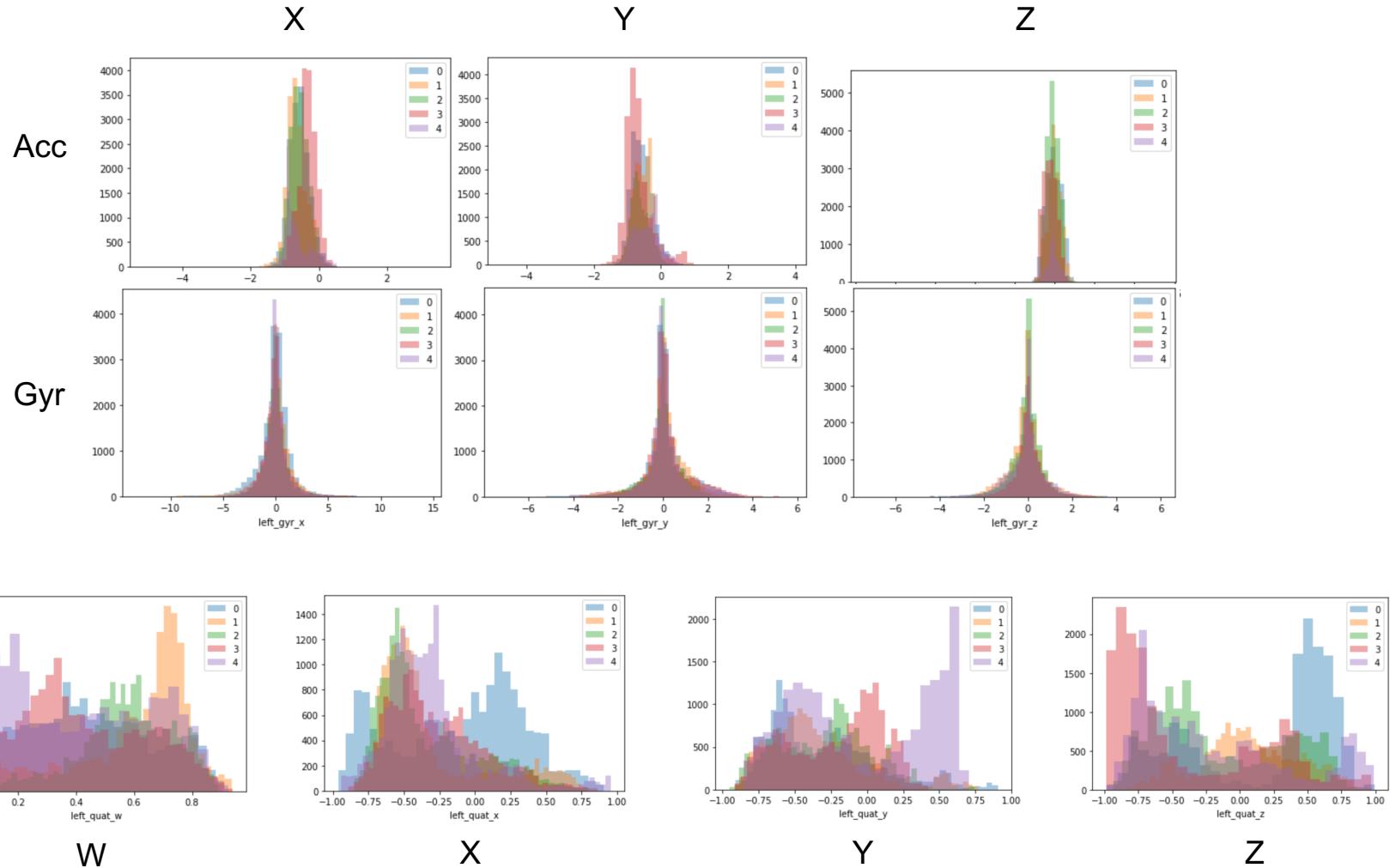
- Accuracy ist nicht alles! Hier die Bewertungsmetrik, im praktischen Einsatz allerdings nicht zwangsläufig am wichtigsten (vgl. Baseline)
 - Auch einfache Methoden können je nach Anforderung gute Performance liefern
- Komplexe Modelle wie LSTMs schwer zu trainieren – Keep it simple
- Bewegungsdaten interessantes Forschungsthema

Thank you for your attention!



<https://xkcd.com/1838/>

Data Exploration – Left (Backup Slide)



Feature Processing

- Standardisierung der gegebenen Sensorwerte
- RTLS-Werte direkt übernehmen
- Konstante Features entfernen