

Aktueller Stand: 28.03.18

-Masterthesis-

# Besprochenen ToDos

- Informationen über Feature-based DTW
- Abspeichern der Klassen in rawdata
- CNN implementieren
- Ergebnisse der beiden Klassifizierungen vergleichen

# Value-based DTW

In Distanzmatrix werden die berechneten Distanzen zweier Zeitreihen gespeichert, es wird der kürzeste „Path“ in Matrix bestimmt, dessen Summe, die minimale Distanz der beiden Zeitreihen darstellt.

## Problem:

Da wir die y werte betrachten, ist es problematisch, wenn die beiden Sequenzen Unterschiede in der y-Achse aufweisen. Globale Unterschiede, andere Mittelwerte oder unterschiedliche Skalierungen führen zu sogenannten „Singularities“. Normalisierung hilft hier nur teilweise.

*[Keogh, E. J., & Pazzani, M. J. (2001, April). Derivative dynamic time warping. In Proceedings of the 2001 SIAM International Conference on Data Mining (pp. 1-11). Society for Industrial and Applied Mathematics.]*

# Derivative DTW

Ermittelt die Distanzen zweier Zeitreihen basierend auf der ersten Ableitung der jeweiligen Punkte

Vorteil: Tendenzen der Steigung der Kurven ist mit im Vergleich

Derivative of datapoint  $r_i$ :

$$D(r_i) = \frac{(r_i - r_{i-1}) + ((r_{i+1} - r_{i-1})/2)}{2}$$

[Keogh, E. J., & Pazzani, M. J. (2001, April). Derivative dynamic time warping. In Proceedings of the 2001 SIAM International Conference on Data Mining (pp. 1-11). Society for Industrial and Applied Mathematics.]

Problem:

Globale Trends auch hier nicht mit modelliert und dadurch nicht erkennbar bei der Bestimmung der Distanzen zweier Zeitreihen

[Xie, Y., & Wiltgen, B. (2010). Adaptive feature based dynamic time warping. International Journal of Computer Science and Network Security, 10(1), 264-273.]

# Feature-based DTW

Für Datapoint  $r_i$ :

Local Feature:

$$f_{local}(r_i) = (r_i - r_{i-1}, r_i - r_{i+1})$$

Global Feature:

$$f_{global}(r_i) = \left( r_i - \frac{\sum_{k=1}^{i-1} r_k}{(i-1)}, r_i - \frac{\sum_{k=i+1}^M r_k}{(M-1)} \right)$$

$$r_i \in \text{a sequence } R = r_1, r_2, \dots, r_i, \dots, r_M$$

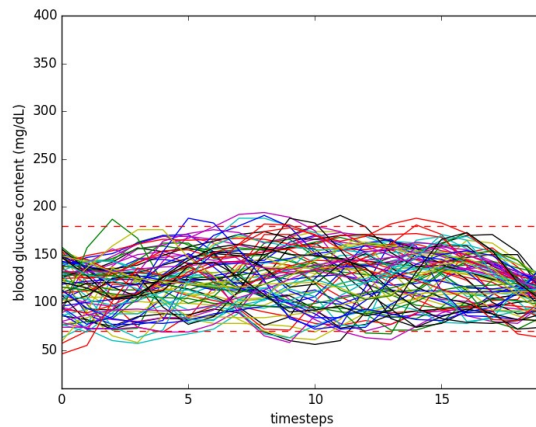
Performanz kann sich je nach Datensatz bei Verwendung von lokalen zu globalen Features unterscheiden

# Adaptive Feature-based DTW

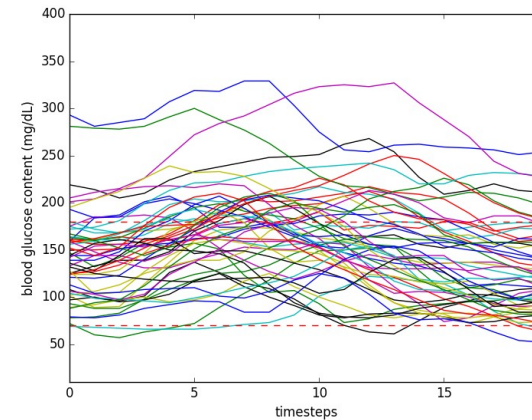
- Deshalb lernen welche Feature zur paarweisen Distanzberechnung verwendet werden soll: *Adaptive Feature-based DTW*
- Berechnung der Gewichten  $w_1$  und  $w_2$ , um globale und lokale Distanz zu gewichten, mittels *In-class range*
- *In-class range*: Die Distanz zw. Betrachteter Sequenz und der weitesten Sequenz der Klasse
- Differenz der Anzahl der Sequenzen, die innerhalb dieser Range liegen und in der eigenen Klasse und der Sequenzen, innerhalb der Range jedoch außerhalb der Klasse, wird aufsummiert

# Vergleich Klasse 1

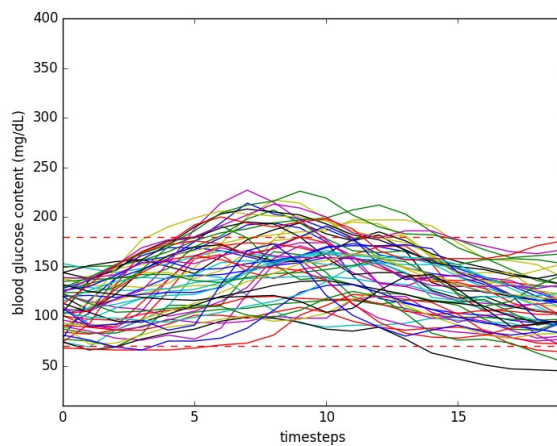
## Class 1: Normal progression



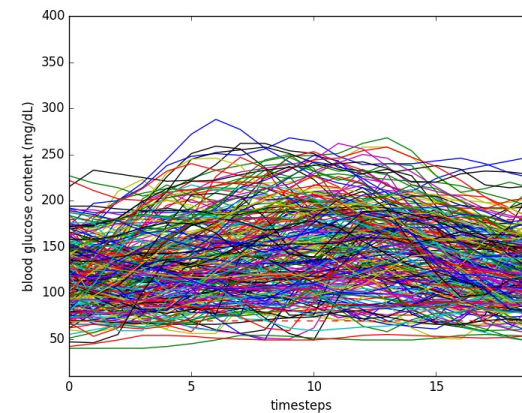
Value-based DTW  
16,9 %



Derivative DTW  
29,44 %



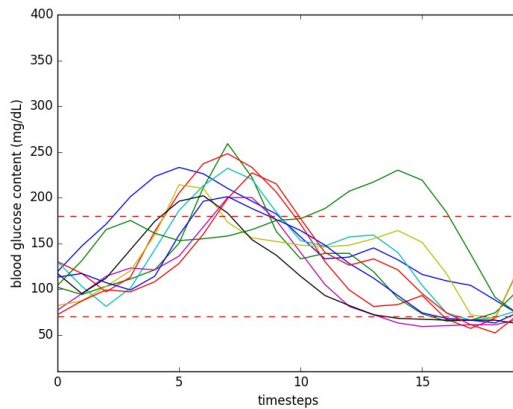
Feature-based DTW  
29,36 %



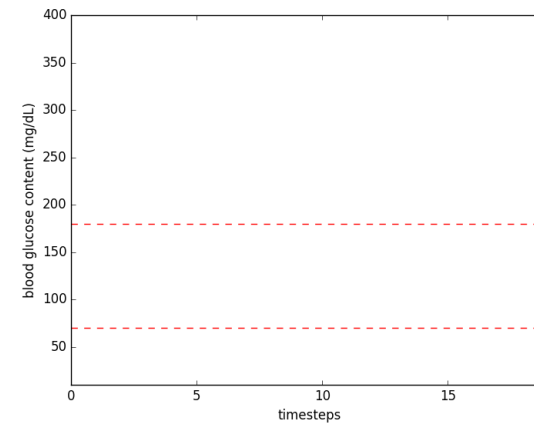
Adaptive Feature-based DTW  
27,96 %

# Vergleich Klasse 2

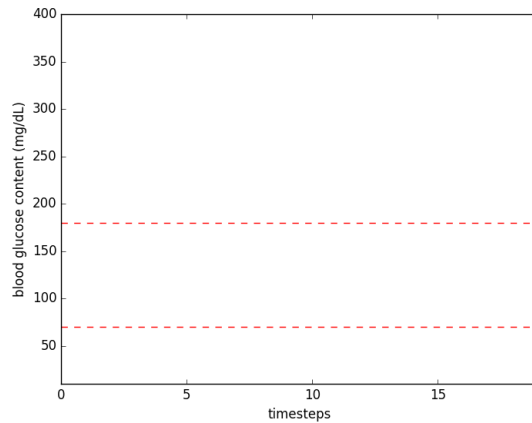
## Class 2: Bolus too high



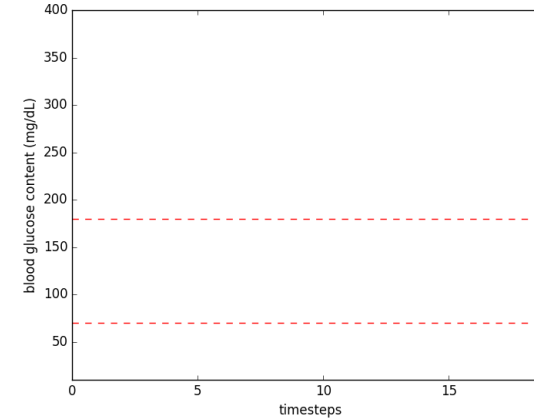
Value-based DTW  
2,77 %



Derivative DTW  
0 %



Feature-based DTW  
0 %

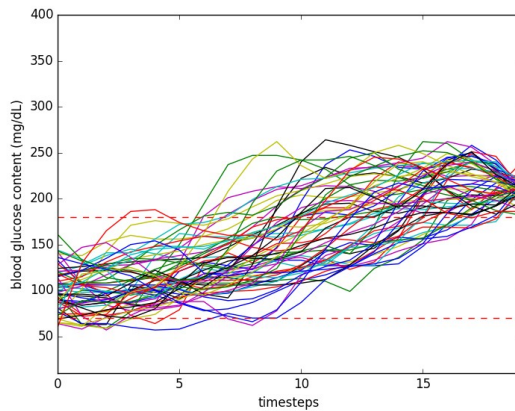


Adaptive Feature-based DTW  
0 %

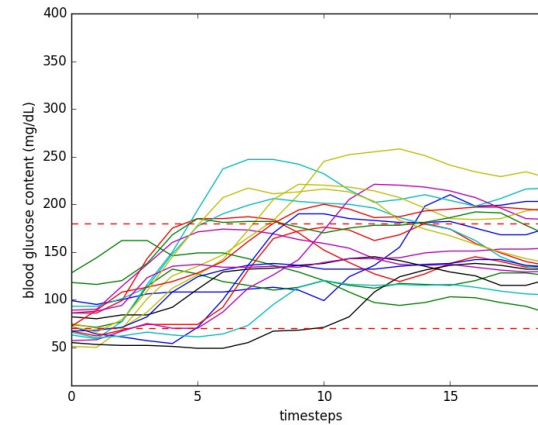


# Vergleich Klasse 4

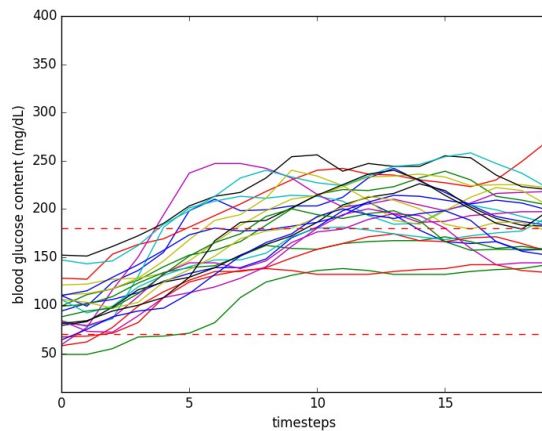
## Class 4: Bolus too small



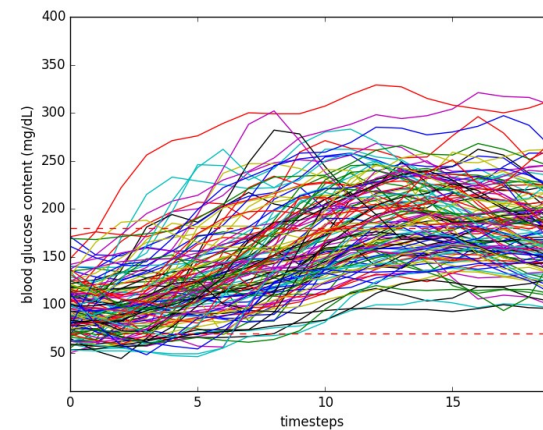
Value-based DTW  
16,34 %



Derivative DTW  
11,17 %



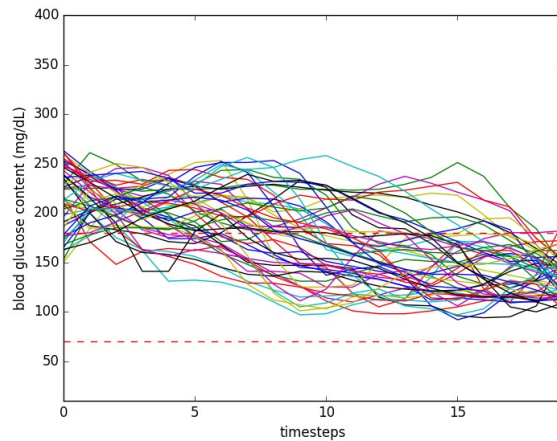
Feature-based DTW  
6,7 %



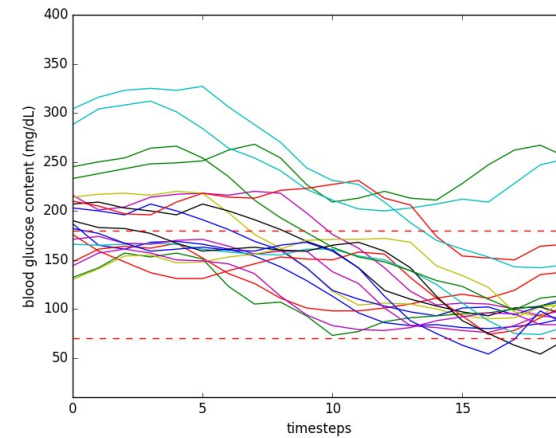
Adaptive Feature-based DTW  
14,09 %

# Vergleich Klasse 6

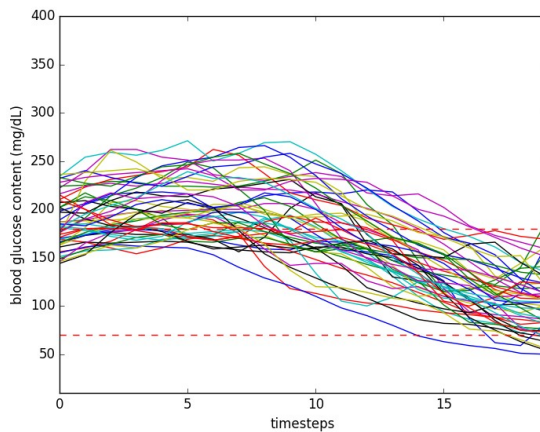
## Class 6: Bolus correction



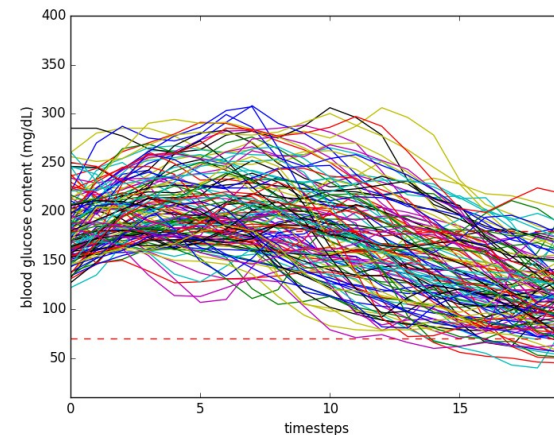
Value-based DTW  
13,85 %



Derivative DTW  
12,18 %



Feature-based DTW  
13,95 %

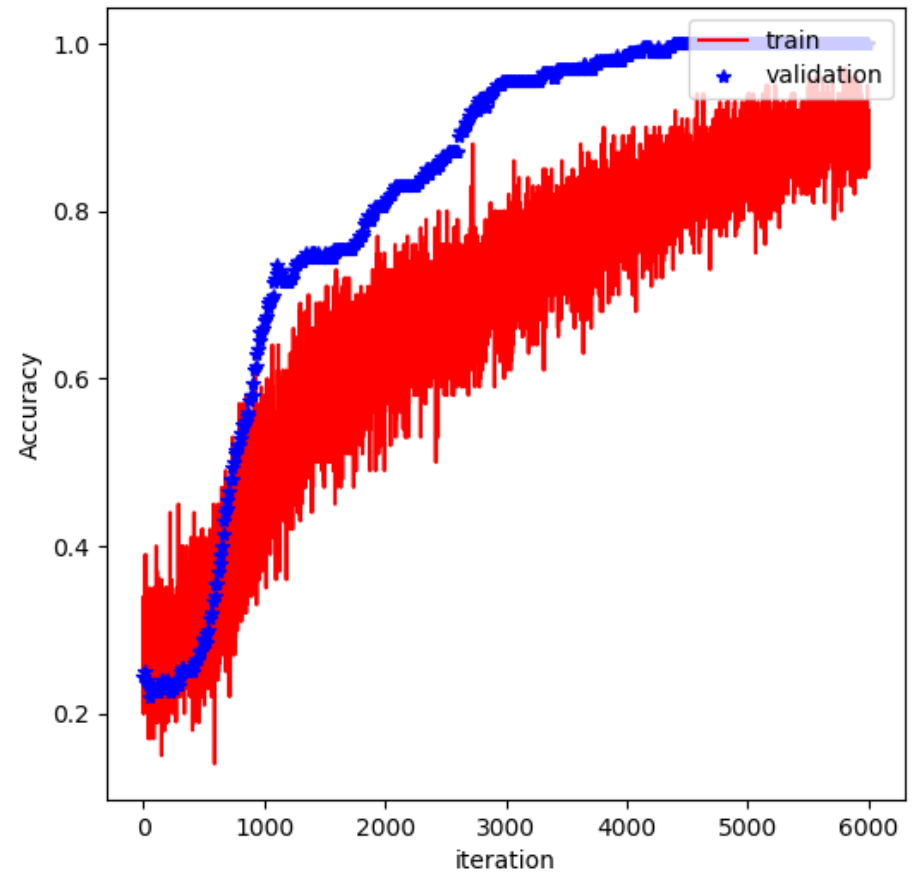
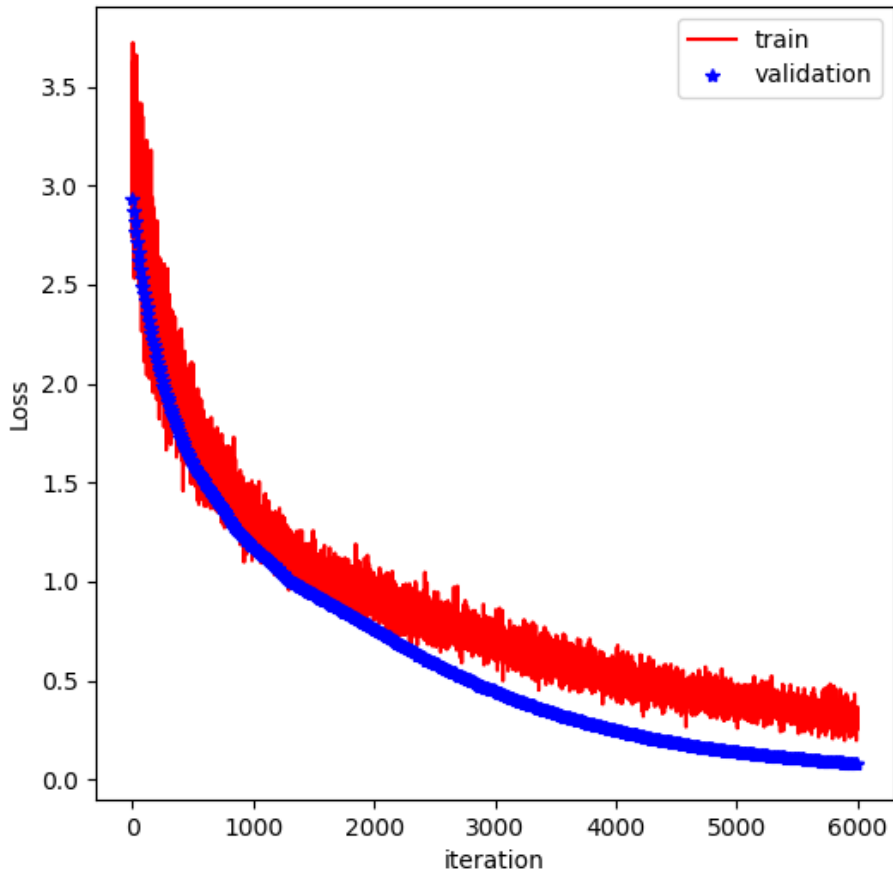


Adaptive Feature-based DTW  
14,09 %

# CNN

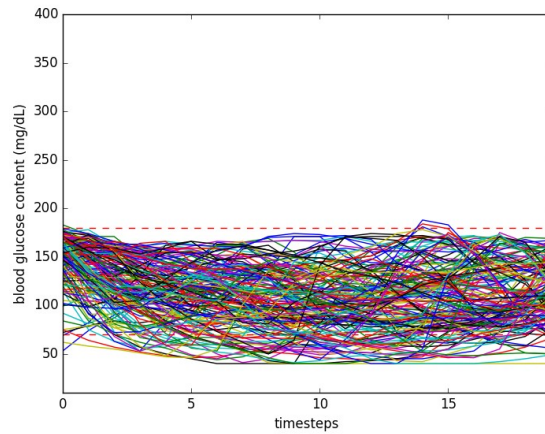
- Implementiert, verwendetes Package: Tensorflow
- 800 Trainings und Testsamples
- 2 Convolutional Layer
- Erste Trainingsergebnisse

# Training des CNNs

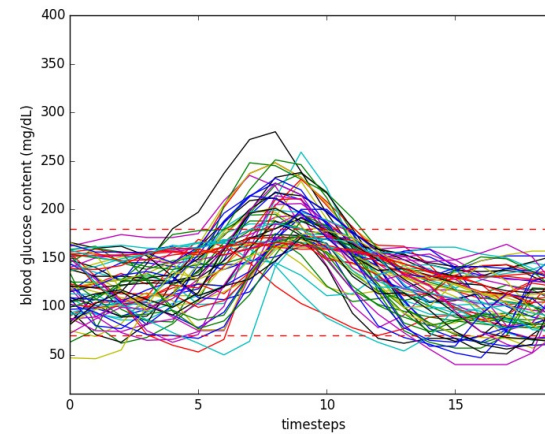


# Klassen

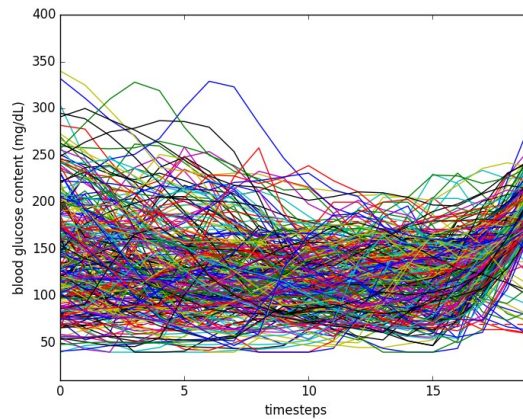
## Convolutional Neural Network



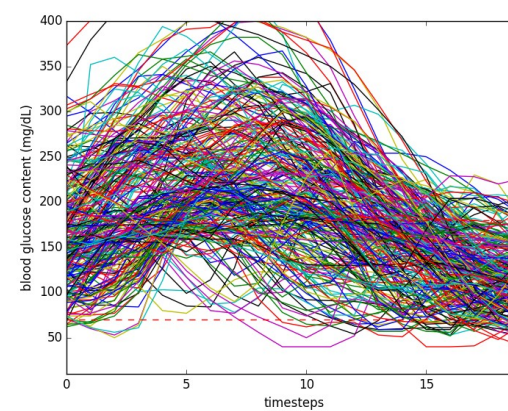
Class1: Normal Progression  
21,34 %



Class2: Bolus too high  
8,61 %



Class4: Bolus too small  
35,22 %



Class6: Bolus correction  
34,83 %

# Übersicht der Klassenzuteilung

Method	Class 1 Normal	Class 2 Too high	Class 4 Too small	Class 6 bolus correction	Class 5 residue class	Discovered curves	Time series ( $M \times TS$ )	Trainings- samples ( $N$ )	Duration and Complexity $O(N)$ ( $TS = 20$ )
VBDTW (threshold > 58)	61 ~ 16,9 %	10 ~ 2,77 %	59 ~ 16,34 %	50 ~ 13,85 %	181 ~ 50,14 %	361	9716	400	1,5 h $O(N \times M \times TS^2)$
DDTW (threshold > 21)	116 ~ 29,44 %	0 0%	44 ~ 11,17 %	48 ~ 12,18 %	186 ~ 47,1%	394	9716	400	1,5 h $O(N \times M \times TS^2)$
FBDTW (threshold > 32)	101 ~ 29,36 %	0 0 %	23 ~ 6,7 %	48 ~ 13,95 %	172 ~ 50 %	344	9716	400	4h $O(N \times M \times TS^2)$
AFBDTW (threshold > 26)	248 ~ 27,96 %	0 0 %	125 ~ 14,09 %	125 ~ 14,09 %	389 ~ 43,86 %	887	9716	400	5 h $O(N^2 \times M^2) + O(N \times M \times TS^2)$
CNN	166 ~ 21,34 %	67 ~ 8,61%	274 ~ 35,22 %	271 ~ 34,83 %	–	778	9716	800	1min

# Accuracy rate of Classification

Method	Accuracy	Improvement	Sam- ples
Value-based DTW	0.75	—	400
Derivative DTW	0.3425	-0.4075	400
Feature-based DTW	0.87	0.12	400
Adaptive Feature-based DTW	TBA	TBA	400
CNN	0.99375	0.24375	800

# ToDos

- CNN verbessern
- Ergebnisse in Datensatz einlesen → Methode zum Abgleich mit Boluswert, Evaluation der gewählten Klassen