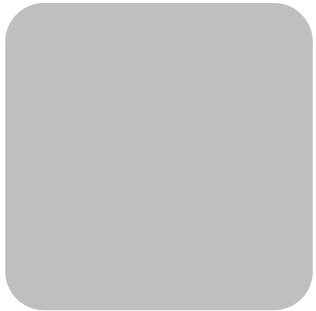


# KI in der Energiewirtschaft - Entwicklung und Implementierung eines Verfahrens zur Eigenverbrauchsoptimierung auf Haushaltsebene mittels Methoden des Maschinellen Lernens



# Gliederung

Motivation und Zielsetzung

Verwendete Methoden des Maschinellen Lernens

Modell

Ergebnisse

Zusammenfassung und Ausblick

# Gliederung

## ► Motivation und Zielsetzung

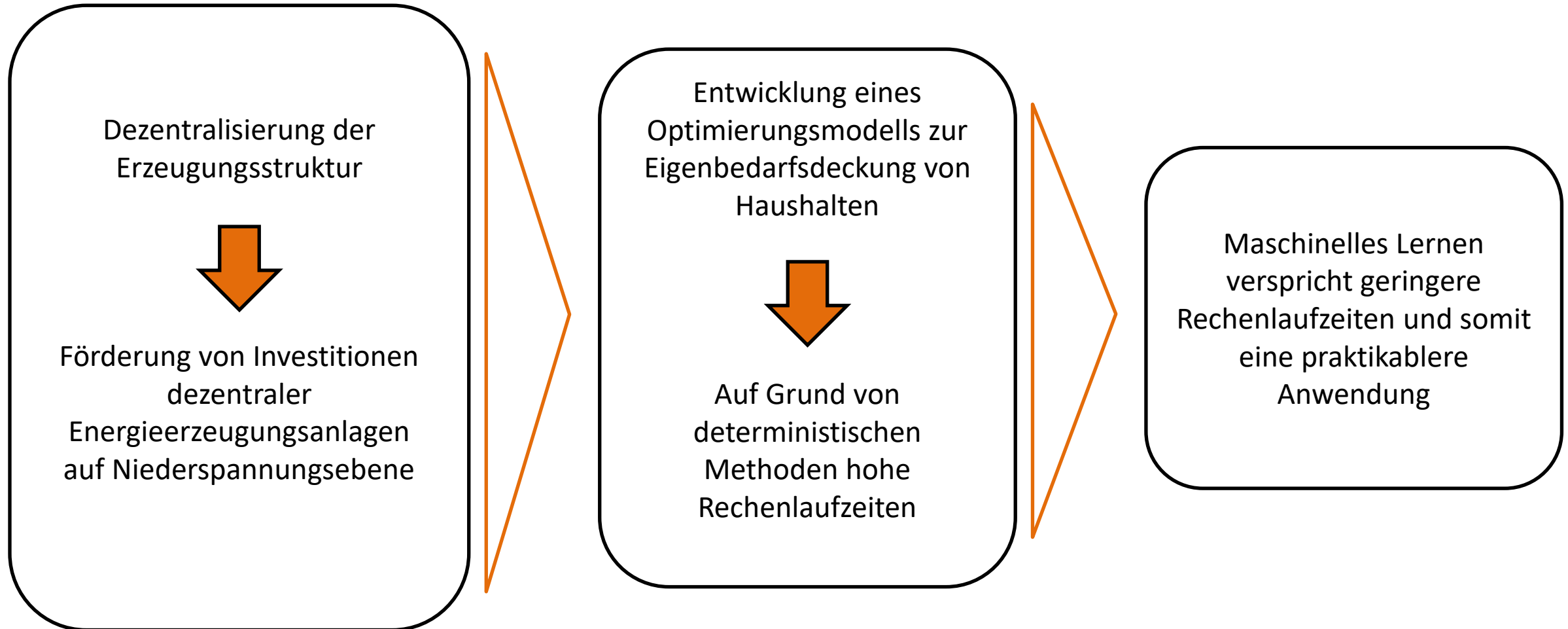
Verwendete Methoden des Maschinellen Lernens

Modell

Ergebnisse

Zusammenfassung und Ausblick

## Motivation



## Zielsetzung

### Hauptziel

Geeignete Methoden des  
Maschinellen Lernens zur  
Approximation eines gemischt-  
ganzzahligen  
Optimierungsproblems  
identifizieren



### Nebenziele

Entscheidungshilfe für den Ausbau  
dezentraler Anlagen schaffen



Neue Lastprofile erschließen



Netzberechnung

# Gliederung

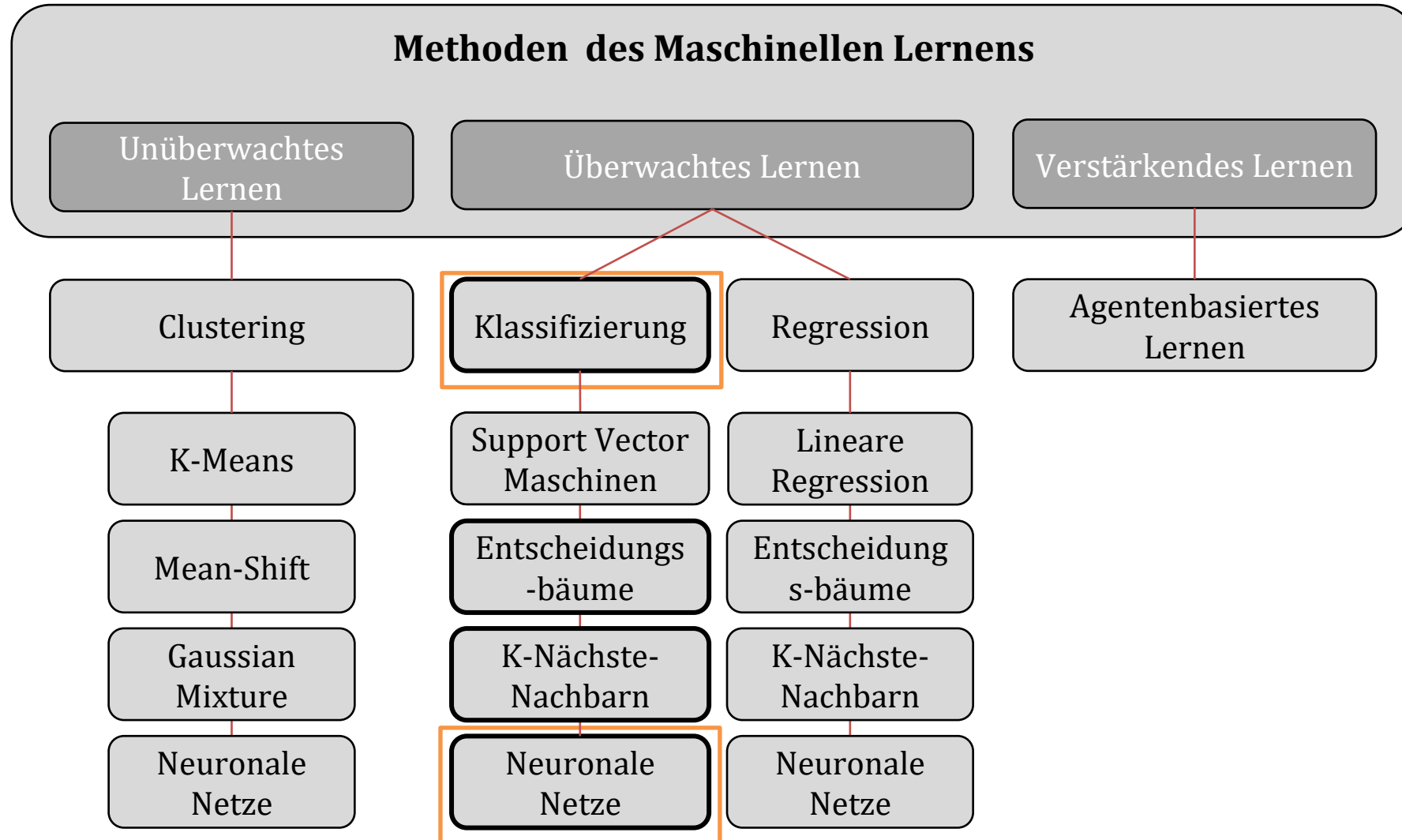
Motivation und Zielsetzung

▶ **Verwendete Methoden des Maschinellen Lernens**

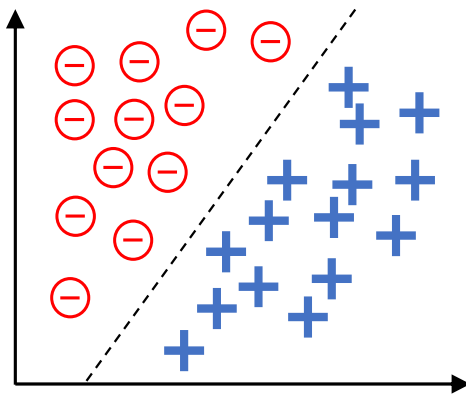
Modell

Ergebnisse

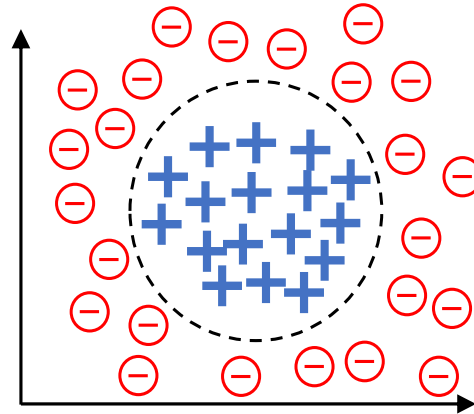
Zusammenfassung und Ausblick



# Klassifizierung



Lineare  
Aufgabenstellung



Nichtlineare  
Aufgabenstellung

Ausbauoptionen	Multi-Label	Multi-Class
	 [1,0,1]	 [1,0,0]
	 [1,1,0]	 [0,1,0]
	 [1,1,1]	 [0,0,1]



# Klassifizierung - Problemtransformationen

## Problemtransformationen

Umwandlung des Multi-Label Problems in Multi- oder Single Class Probleme

Label Powerset

Label Powerset

X	Y1	Y2	Y3	Y4
X(1)	0	1	1	0
X(2)	1	0	1	0
X(3)	0	0	0	1
X(4)	1	0	1	0
X(5)	0	1	1	0



X	Y1
X(1)	1
X(2)	2
X(3)	3
X(4)	2
X(5)	1

Classifier Chain

Classifier Chain

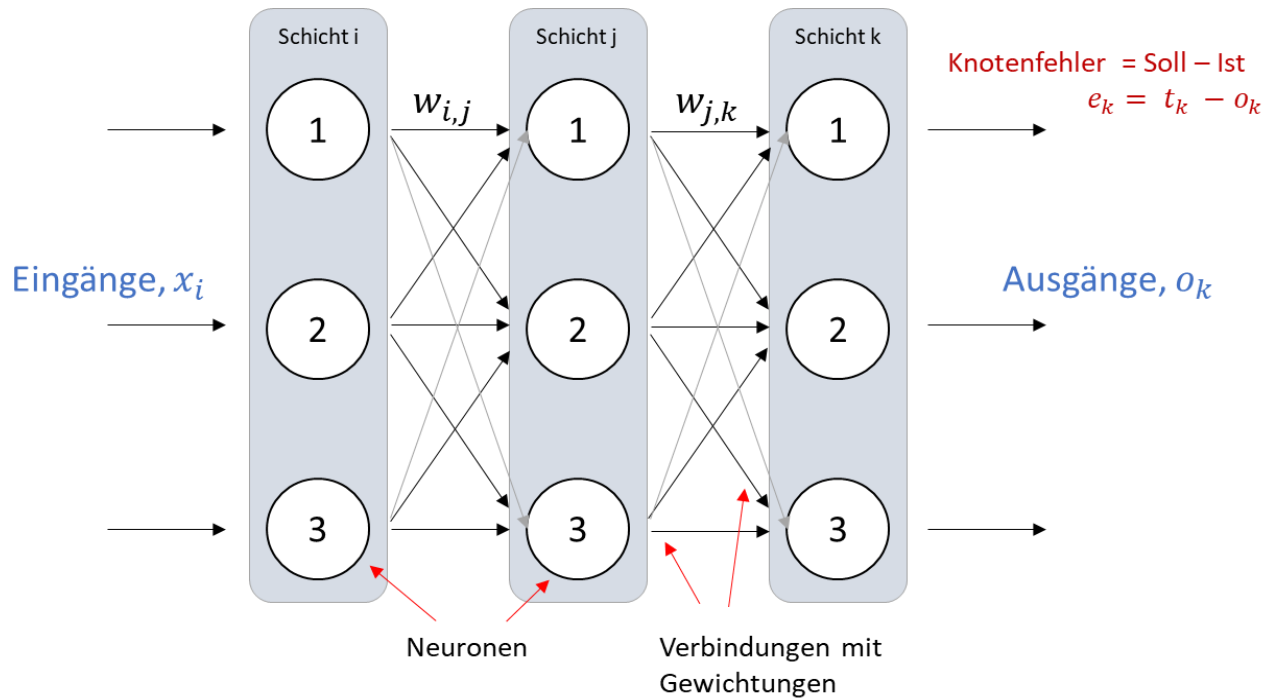
X	Y1	Y2	Y3	...	Y(n)
X(1)	0	1	1	...	...
X(2)	1	0	1	...	...
X(3)	0	0	0	...	...
X(4)	1	0	1	...	...
X(n)	...	...	...	...	...



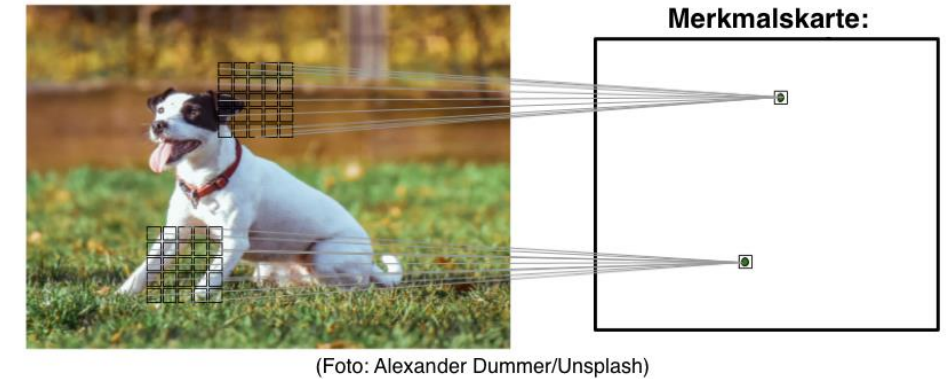
X	Y1	X	Y1	Y2	X	Y1	Y2	Y3	X	Y1	Y2	Y3	...	Y(n)
X(1)	0	X(1)	0	1	X(1)	0	1	1	X(1)	0	1	1	...	...
X(2)	1	X(2)	1	0	X(2)	1	0	1	X(2)	1	0	1	...	...
X(3)	0	X(3)	0	0	X(3)	0	0	0	X(3)	0	0	0	...	...
X(4)	1	X(4)	1	0	X(4)	1	0	1	X(4)	1	0	1	...	...
X(n)	...	X(n)	...	...	X(n)	...	...	...	X(n)	...	...	...	...	...

# Neuronale Netze

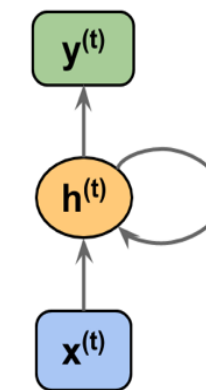
## Feedforward Neuronales Netz



## Konvolutionales Neuronales Netz



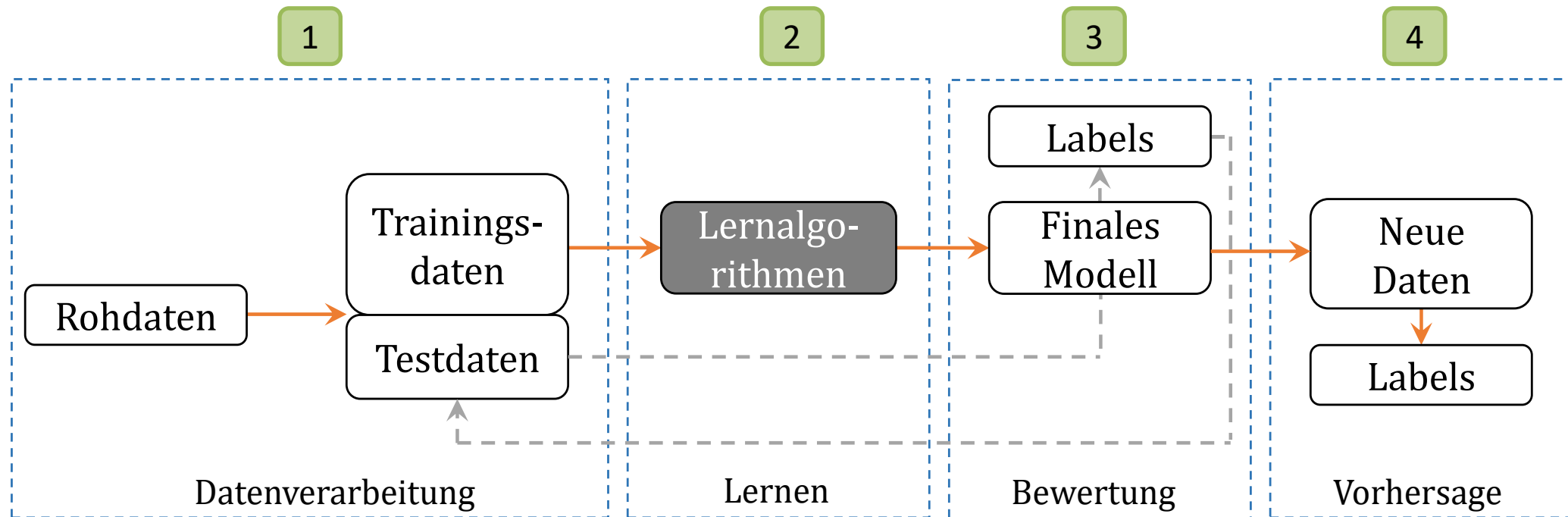
## Rekurrentes Neuronales Netz



Sequenz 1: [6, 0, 1, 2, 3, 6]

Sequenz 2: [9, 0, 1, 2, 3, 9]

# Allgemeines Vorgehen für Modelle aus dem Bereich ML



# Gliederung

Motivation und Zielsetzung

Verwendete Methoden des Maschinellen Lernens

▶ Modell

Ergebnisse

Zusammenfassung und Ausblick

1

2

3

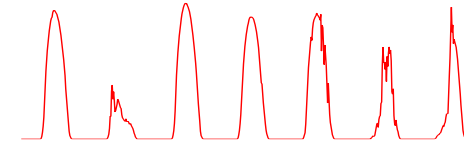
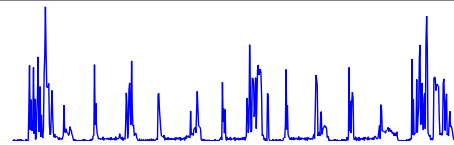
4

## Generierung des Trainings- und Testdatensatzes

Konstante  
Kosten/Vergütungen

Lastverlauf

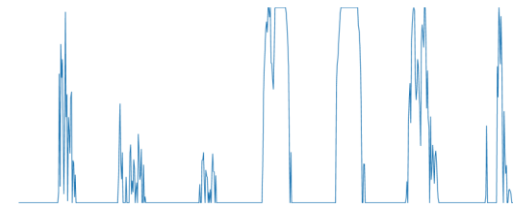
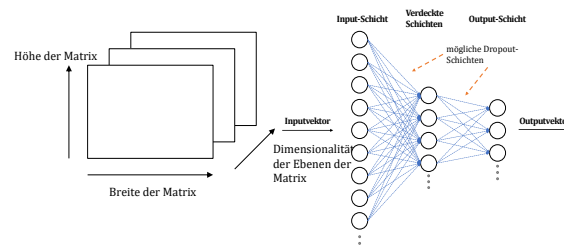
PV-Einspeiselinie

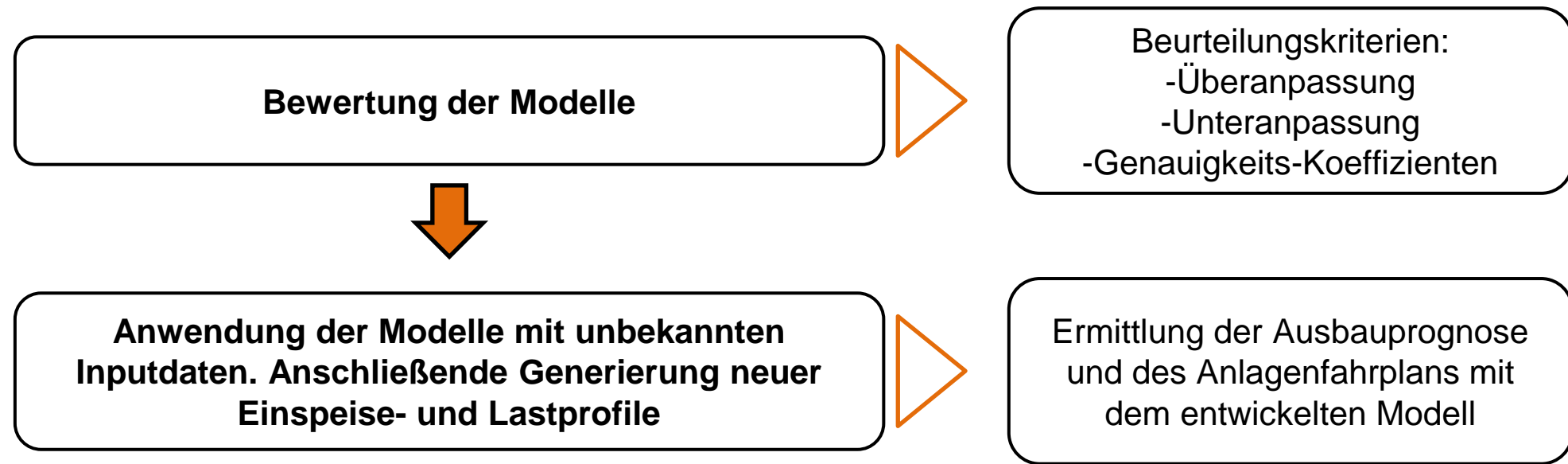


## Entwicklung der Prognosemodelle

Modell zur Prognose der  
Ausbaupfade

Modell zur Prognose des  
Anlagenfahrplans





# Modellierung – Generierung der Trainingsdaten

1

2

3

4

Input des Modells

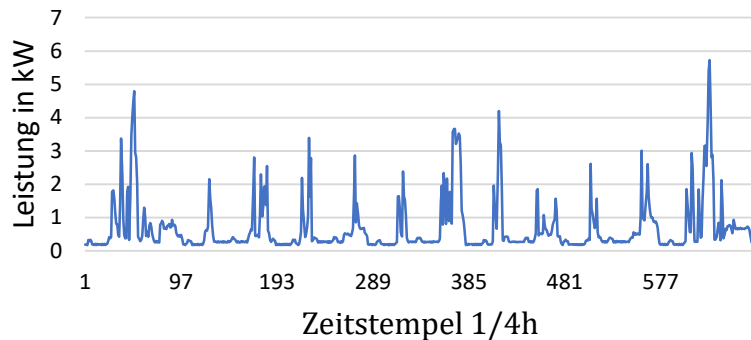
Lastprofile

PV-Einspeiselinie

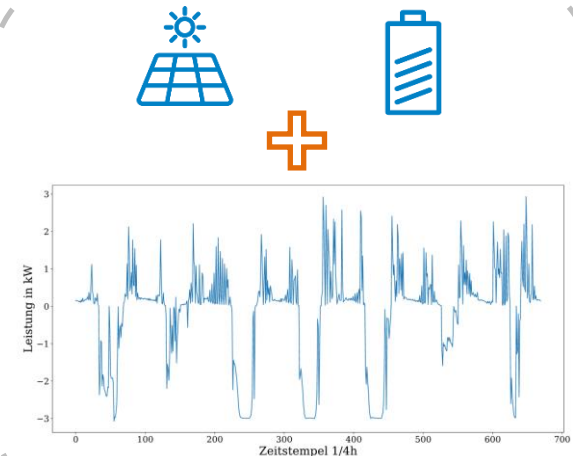
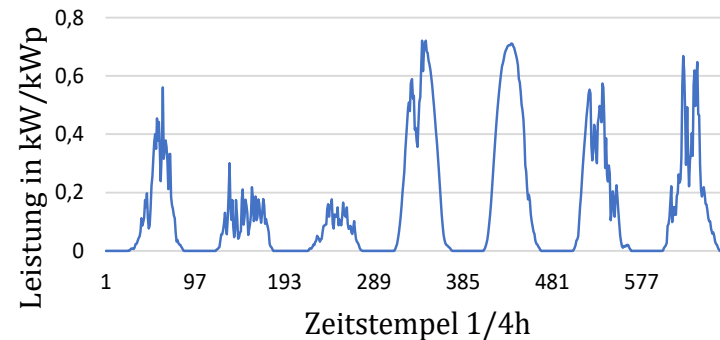
Output

Optimierungsalgorithmus

Lastprofil

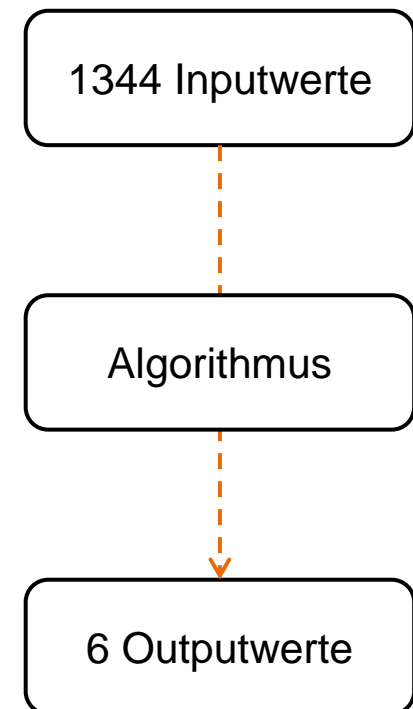
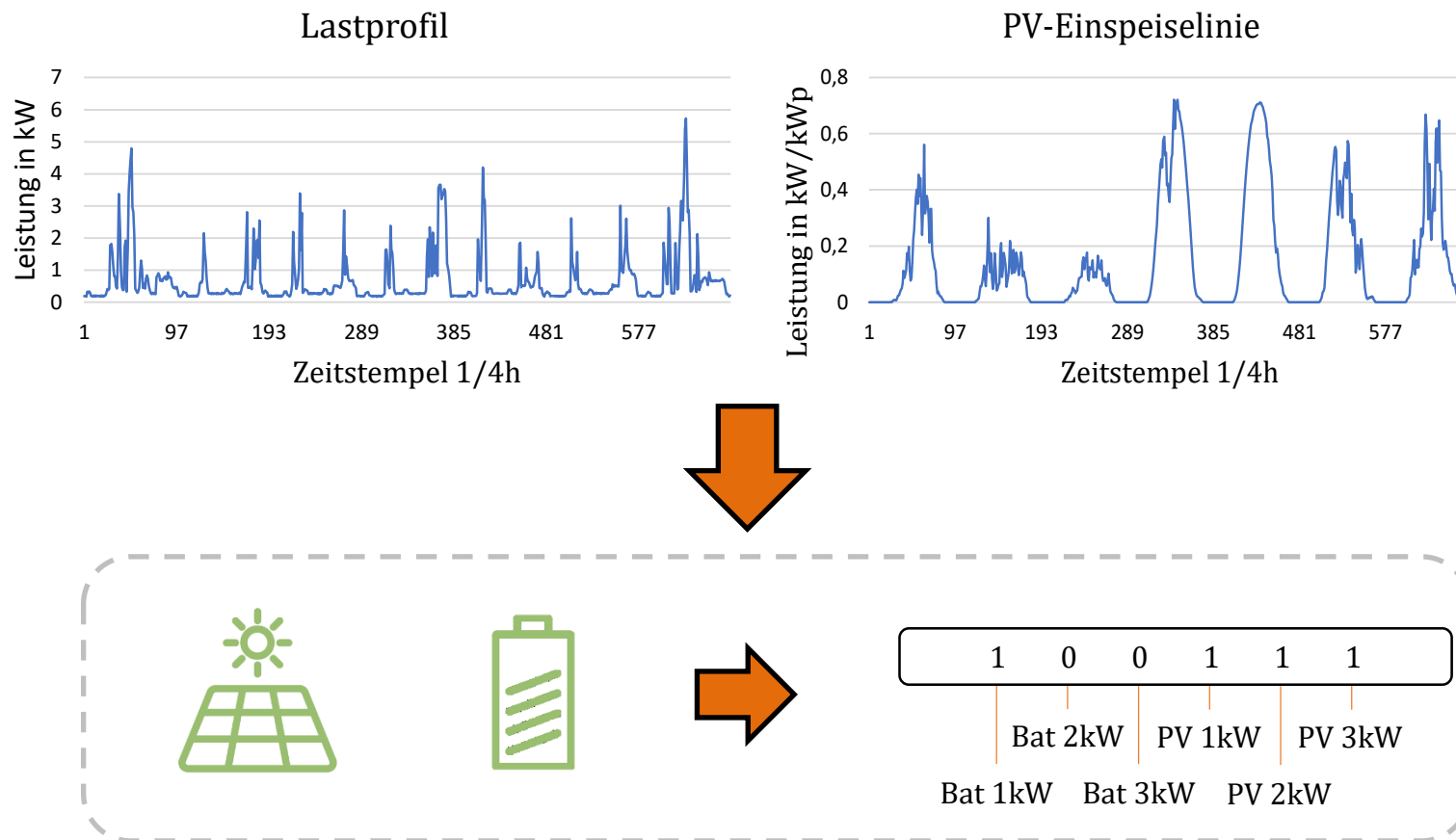


PV-Einspeiselinie



# Modellierung – Modell zur Ausbauprognose

1 2 3 4





# Modell zur Ausbauprognose - Klassifizierungsalgorithmen

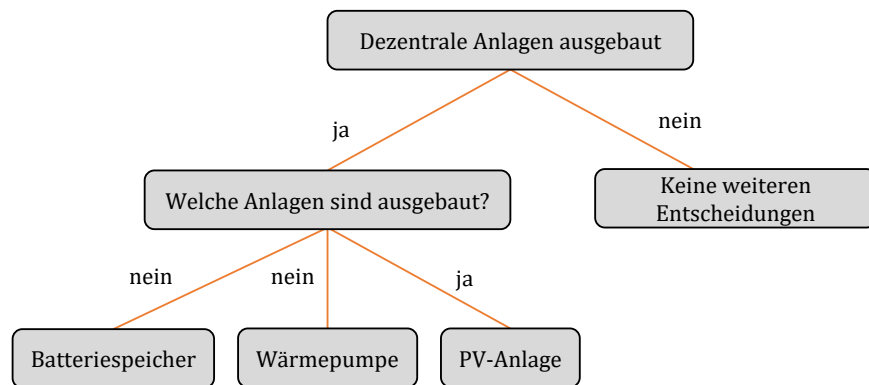
1

2

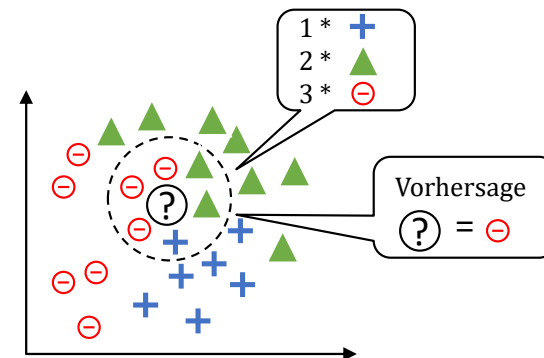
3

4

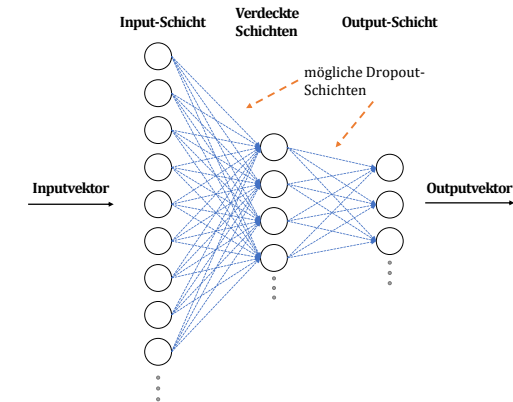
Entscheidungsbäume



k-Nächste-Nachbarn



Neuronale Netze



# Modell zur Ausbauprognose mittels Neuronalen Netzen

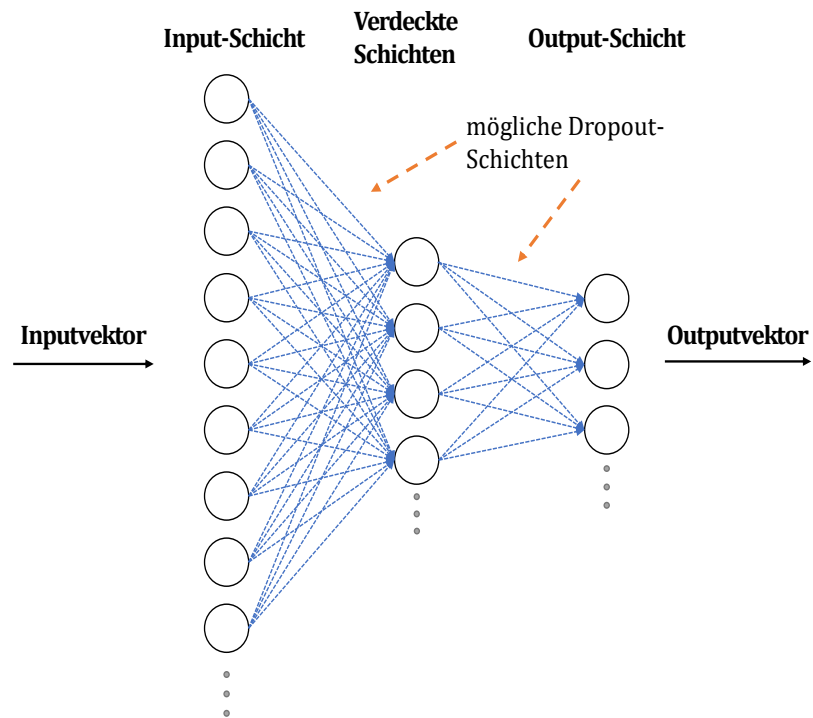
1

2

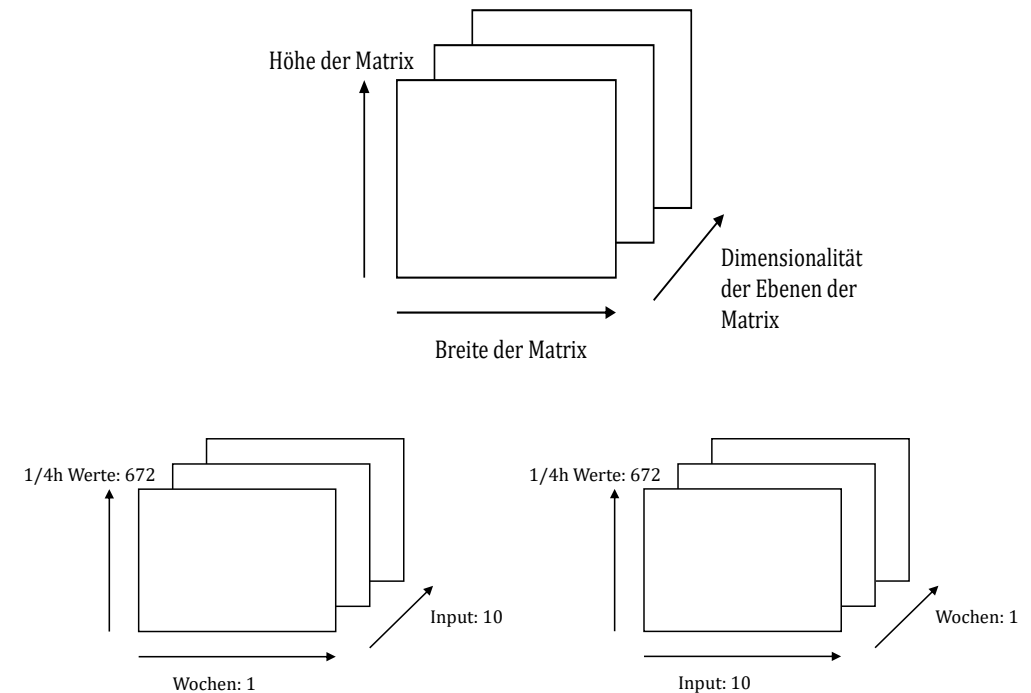
3

4

## Feedforward Neuronales Netz

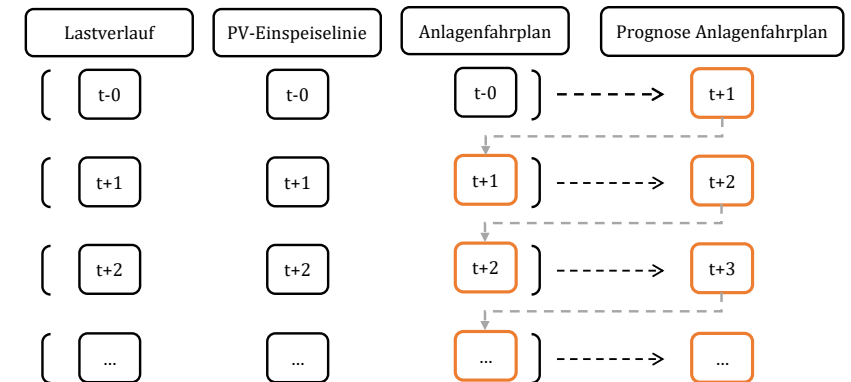
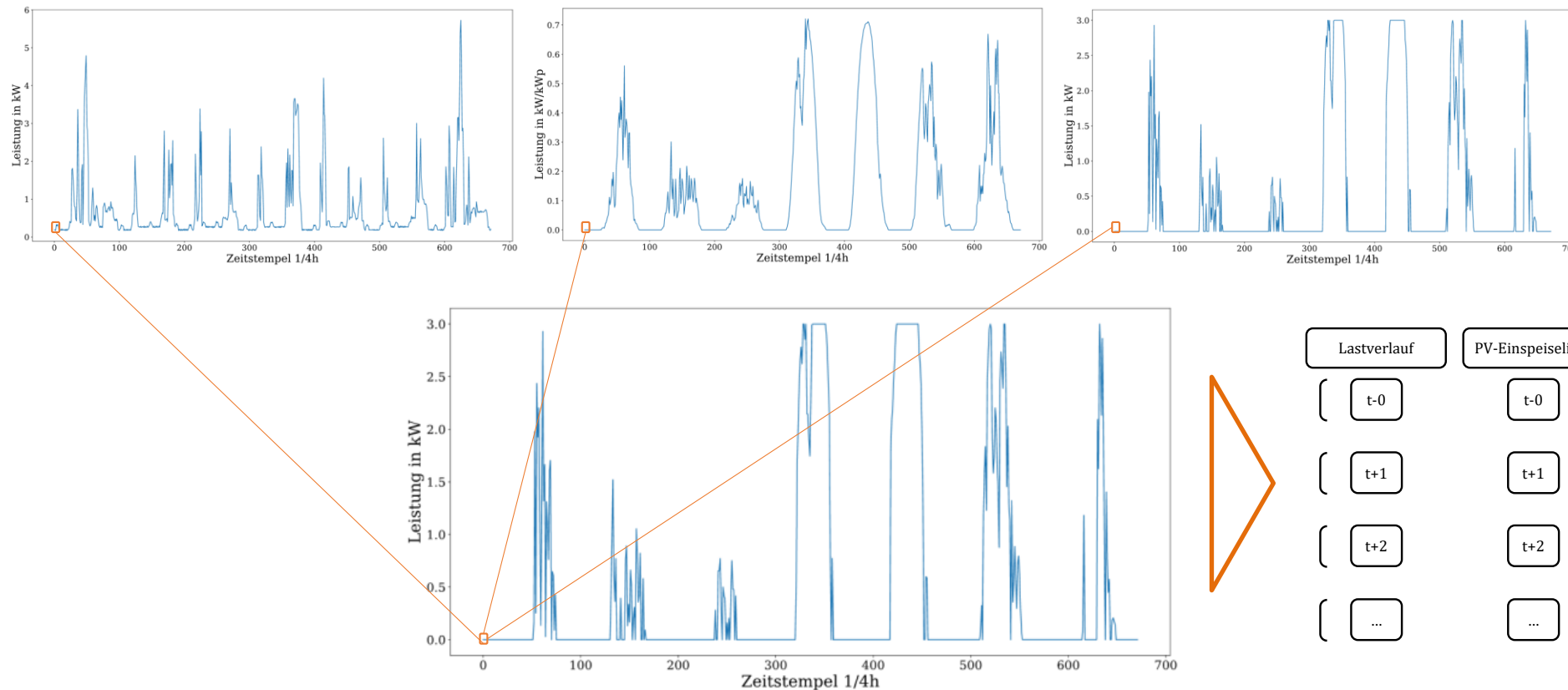


## Konvolutionales Neuronales Netz



# Modell zur Zeitreihenvorhersage mit Rekurrenten NN

1 2 3 4



# Gliederung

Motivation und Zielsetzung

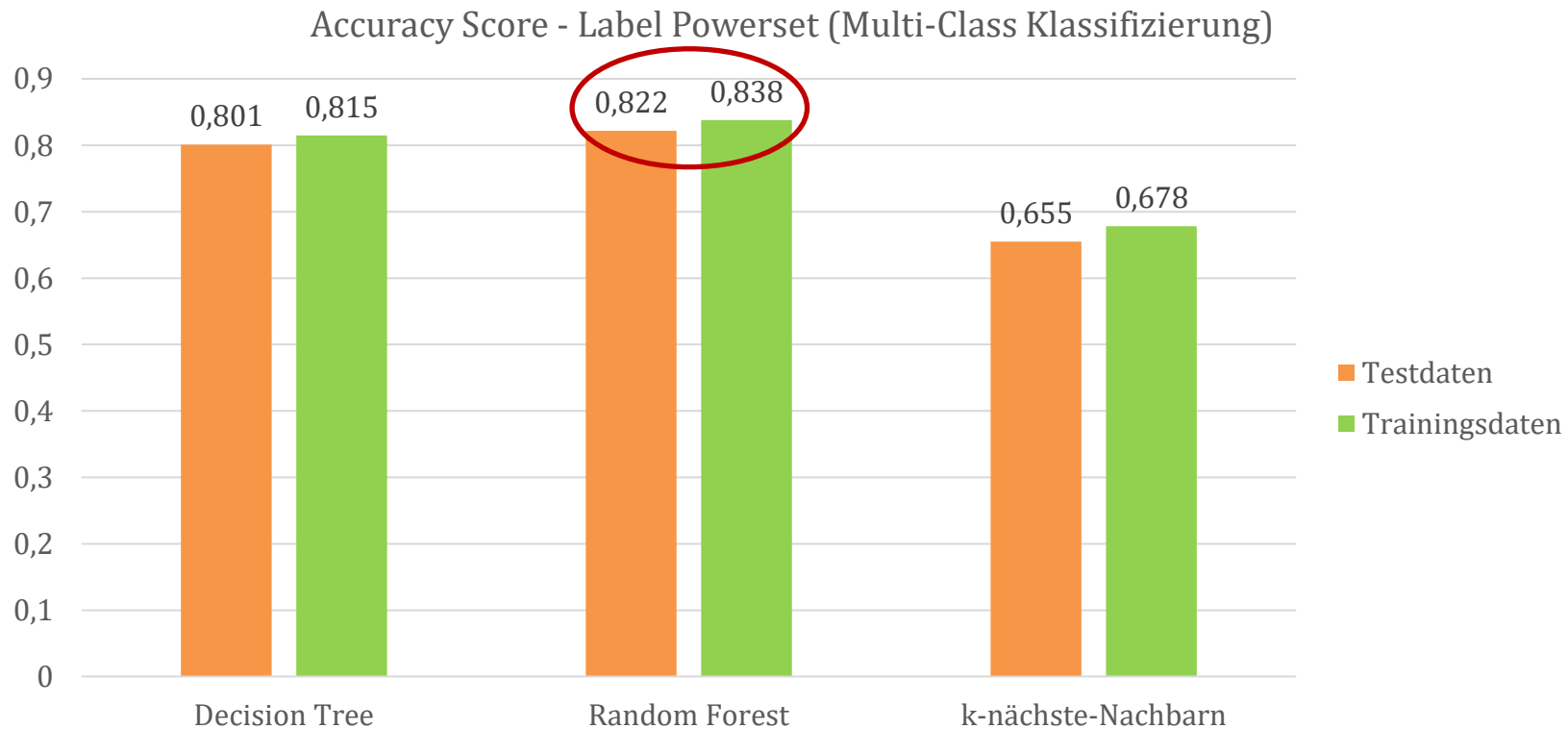
Verwendete Methoden des Maschinellen Lernens

Modell

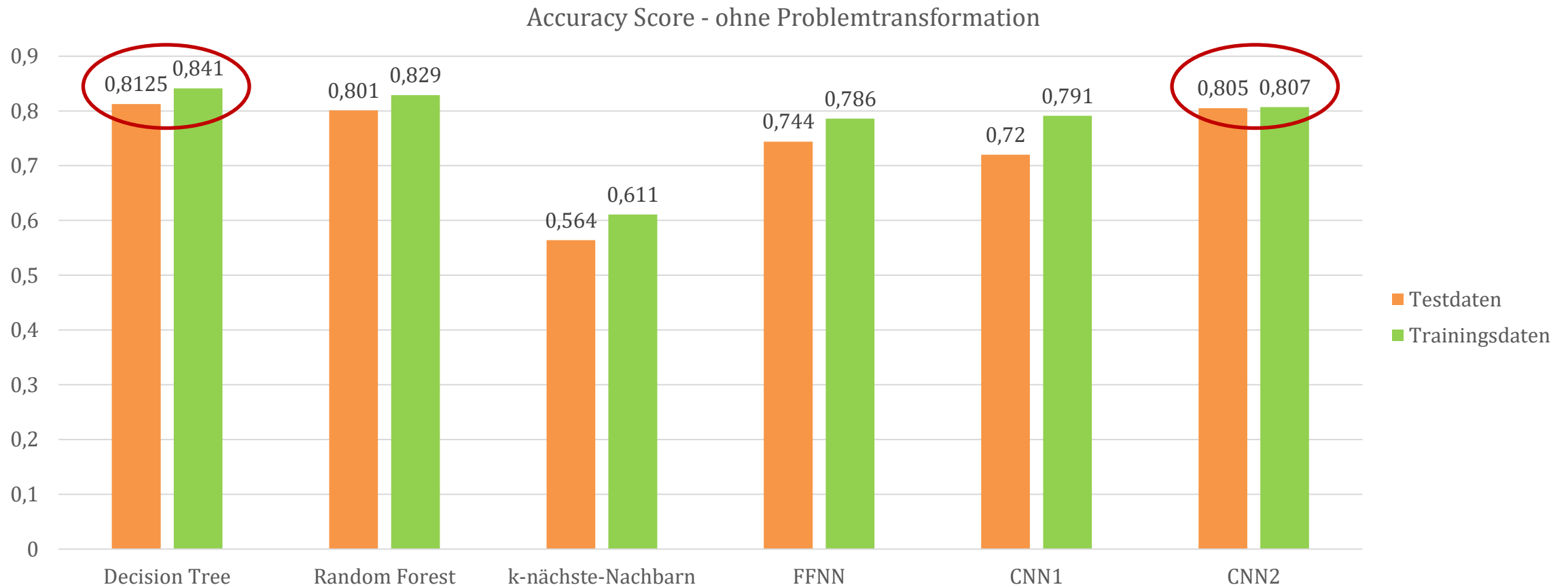
▶ Ergebnisse

Zusammenfassung und Ausblick

## Ergebnisse - Prognose der Ausbaupfade



## Ergebnisse - Prognose der Ausbaupfade



## Ergebnisse – Prognose des Anlagenfahrplans

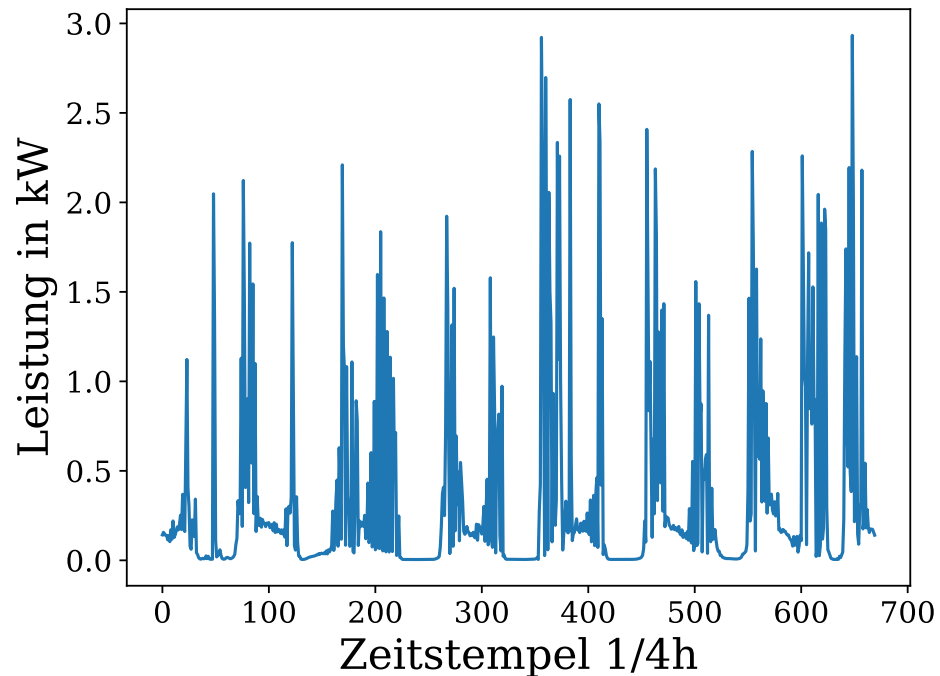
1

2

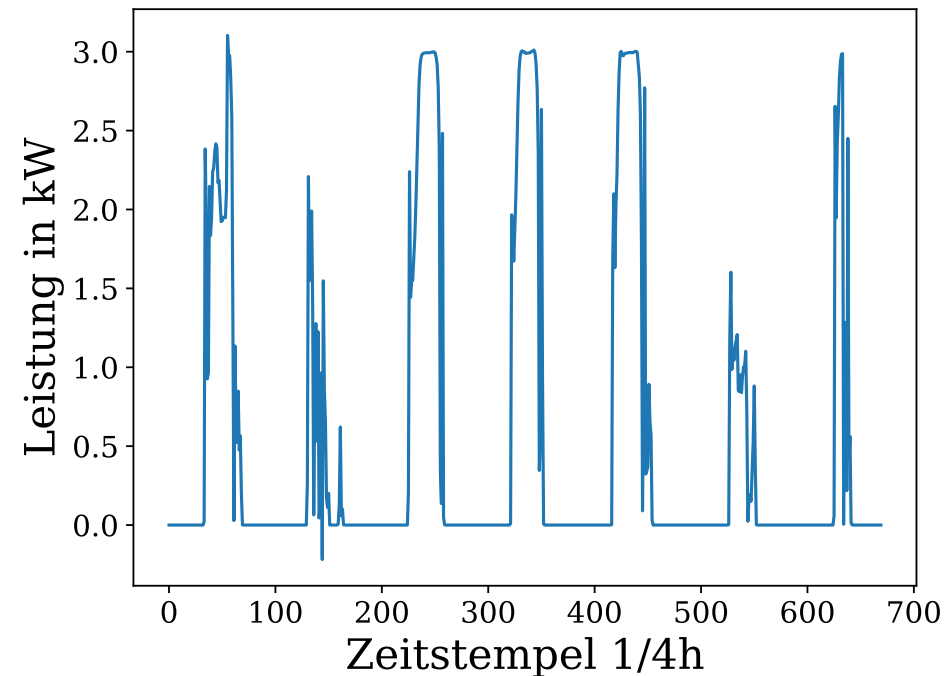
3

4

Bezugsleistung des  
Haushalts



Einspeiseleistung des  
Haushalts



## Ergebnisse – resultierendes Lastprofil

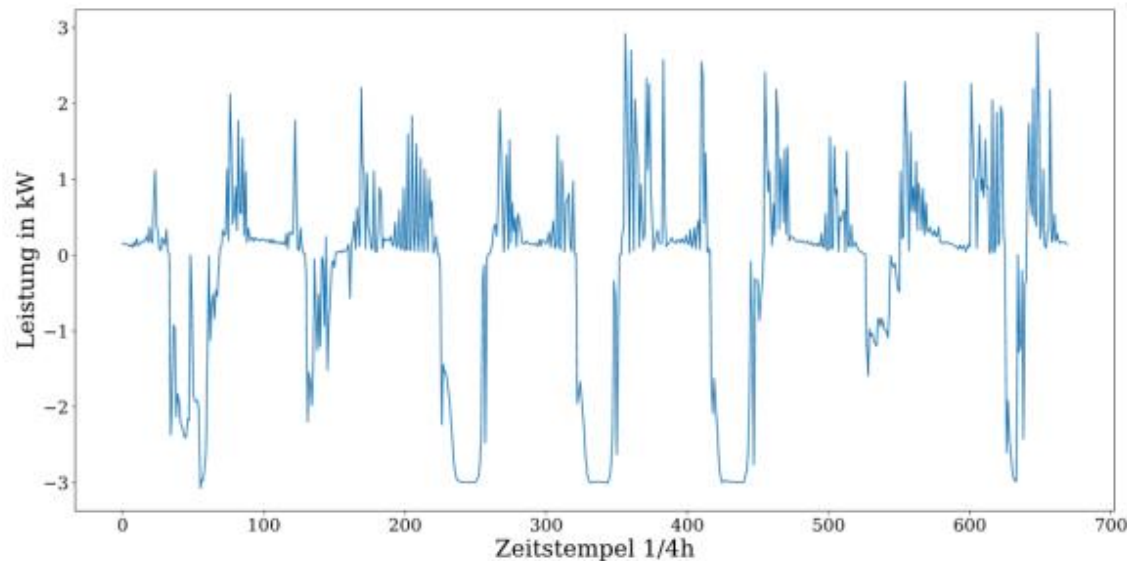
1

2

3

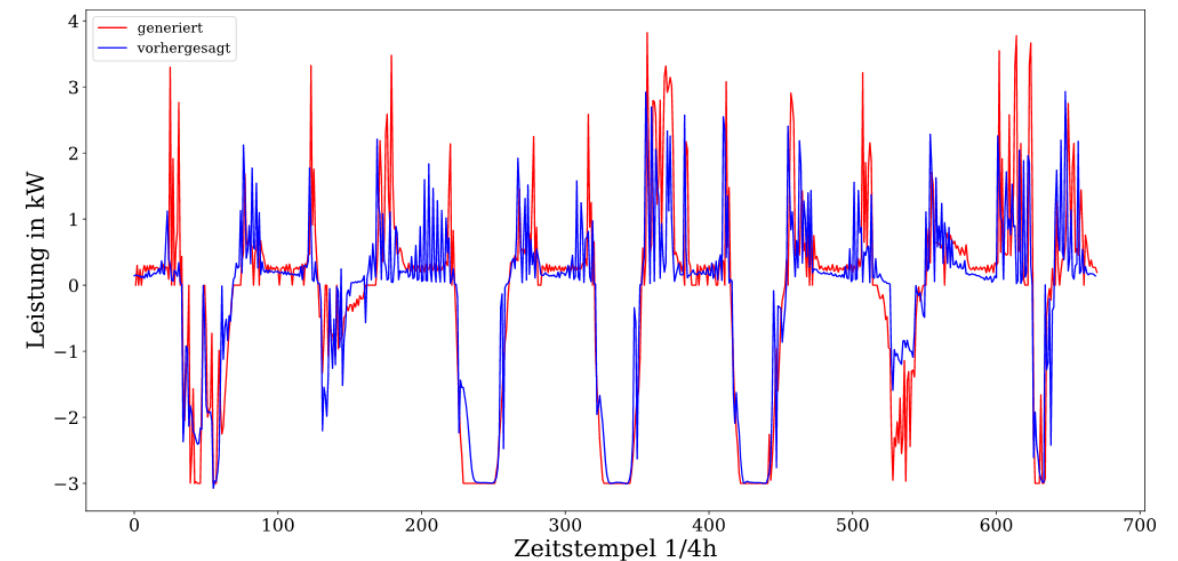
4

Prognostiziertes Lastprofil



Abweichung von prognostiziertem und  
generiertem Lastprofil in kWh: ~12%

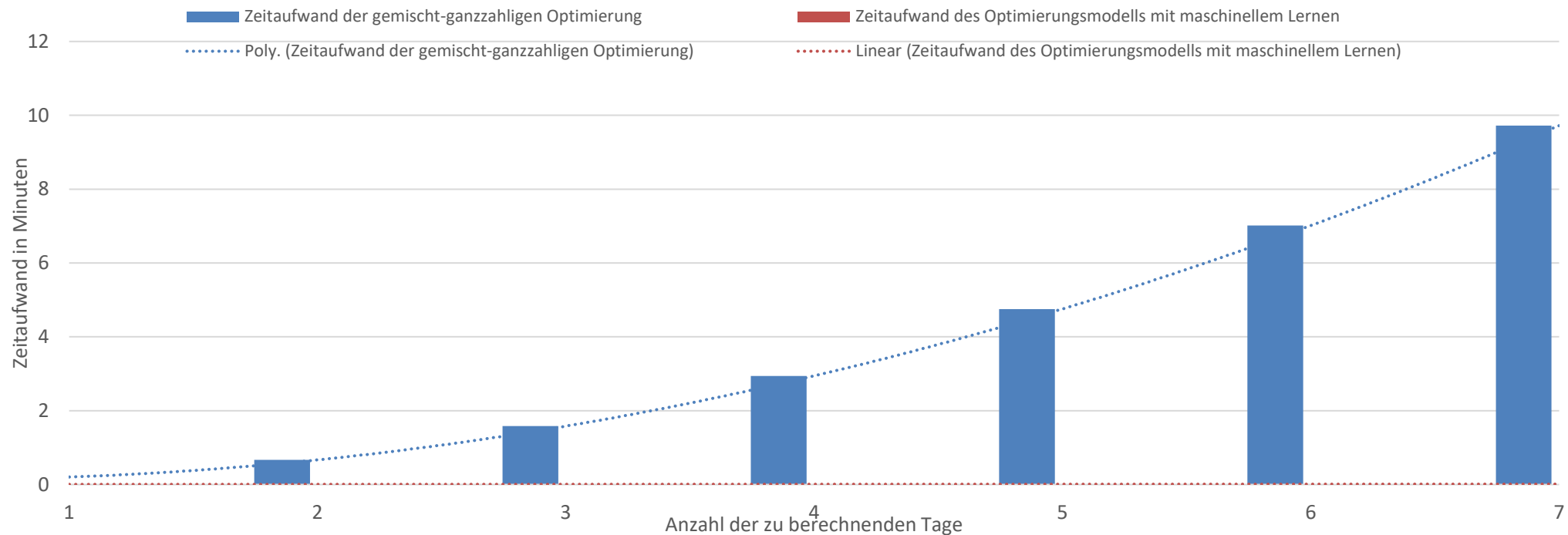
Vergleich Prognose und generiertes  
Lastprofil



Gemischt-ganzzahlige Optimierung: ~600 Sek  
ML-Modell: ~ 0,6 Sek



## Ergebnisse - Vergleich des Zeitaufwands der Modelle



Gemischt-ganzzahlige Optimierung ~10min

ML-Modell ~ 0,6 Sek

# Gliederung

Motivation und Zielsetzung

Verwendete Methoden des Maschinellen Lernens

Modell

Ergebnisse

▶ Zusammenfassung und Ausblick

## Zusammenfassung

- Zugrundeliegendes Optimierungsproblem wurde analysiert
- Auswahl geeigneter Methoden aus dem Bereich ML zur Approximation des Optimierungsproblems getroffen
- Entwicklung des ML-Modells unter Anwendung der ausgewählten Methoden
- Bewertung des Modells mittels verschiedenen Beurteilungskriterien
- Anwendung der Modelle mit unbekannten Inputdaten

## Ausblick

- Betriebs- und Investitionskosten variieren, um weitere Ausbaupfade wie Wärmepumpen und BHKW etc. zu generieren
- Optimierung der Trainingsdaten hinsichtlich der Ausgeglichenheit des Lösungsraums
- Hyperparameter Optimierung zur strukturellen Verbesserung der Algorithmen – insbesondere der Neuronalen Netze
- Szenarioanalysen mit den neuen Lastprofilen



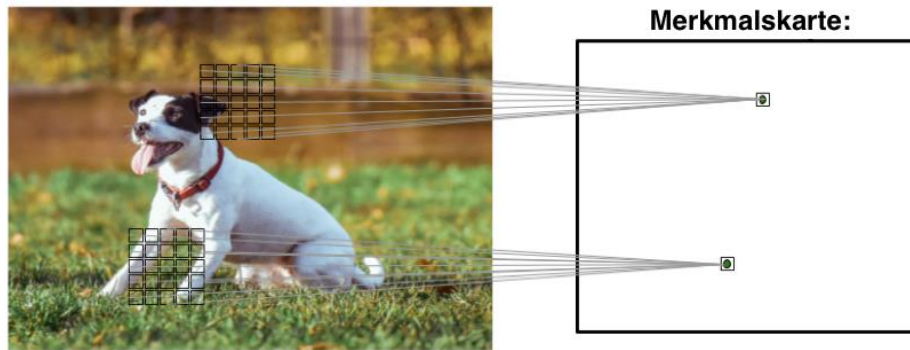
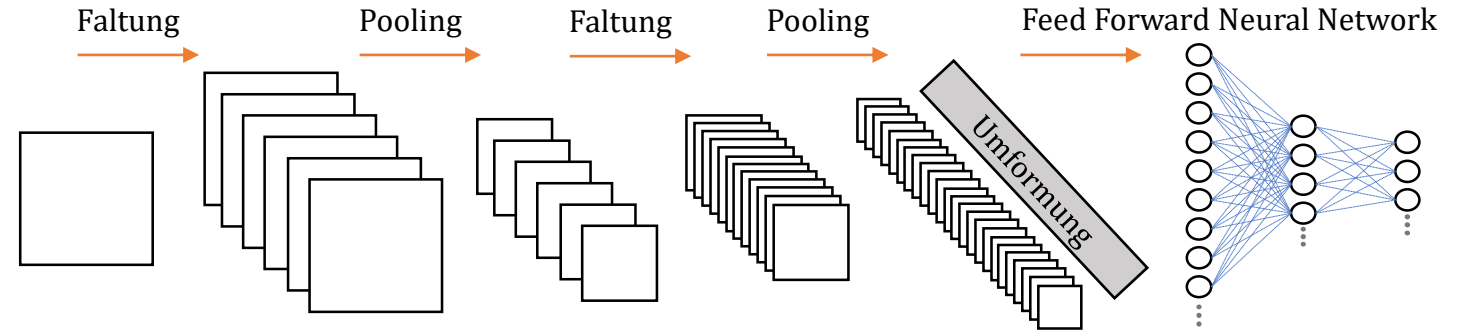
[ie3.tu-dortmund.de](https://ie3.tu-dortmund.de)





# Konvolutionale Neuronale Netze

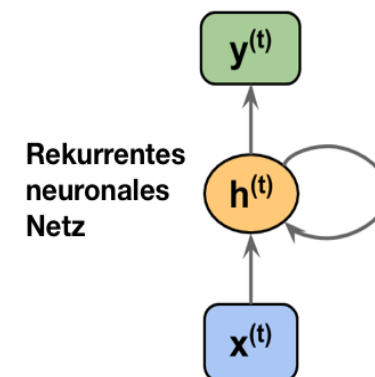
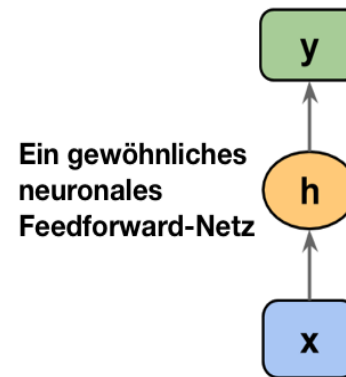
- Konvolutionale Netze sind Neuronale Netze, die speziell für die Merkmalsextrahierung gedacht sind.
- Anwendungsbereiche
  - Bildanalyse, Objekterkennung



(Foto: Alexander Dummer/Unsplash)

# Rekurrente Neuronale Netze

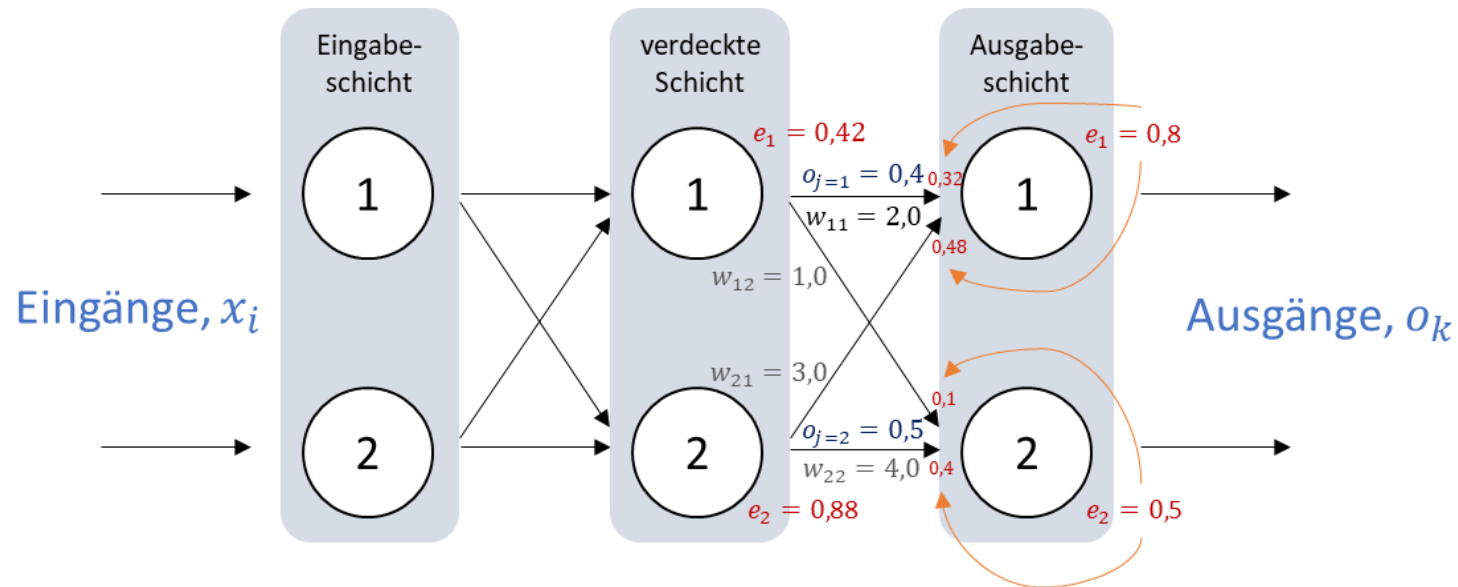
- Rekurrente Netze nutzen die Eigenschaft bestimmte Informationen aus vergangenen Zeitschritten als zusätzlichen Input zu verwenden.
- Anwendungsbereiche
  - Übersetzungsprogramme
  - Textgenerierung
  - Sequenzgenerierung
  - Audio Erzeugung



Sequenz 1  
Sequenz 2

[6, 0, 1, 2, 3, 6]  
[9, 0, 1, 2, 3, 9]

# Backpropagierung

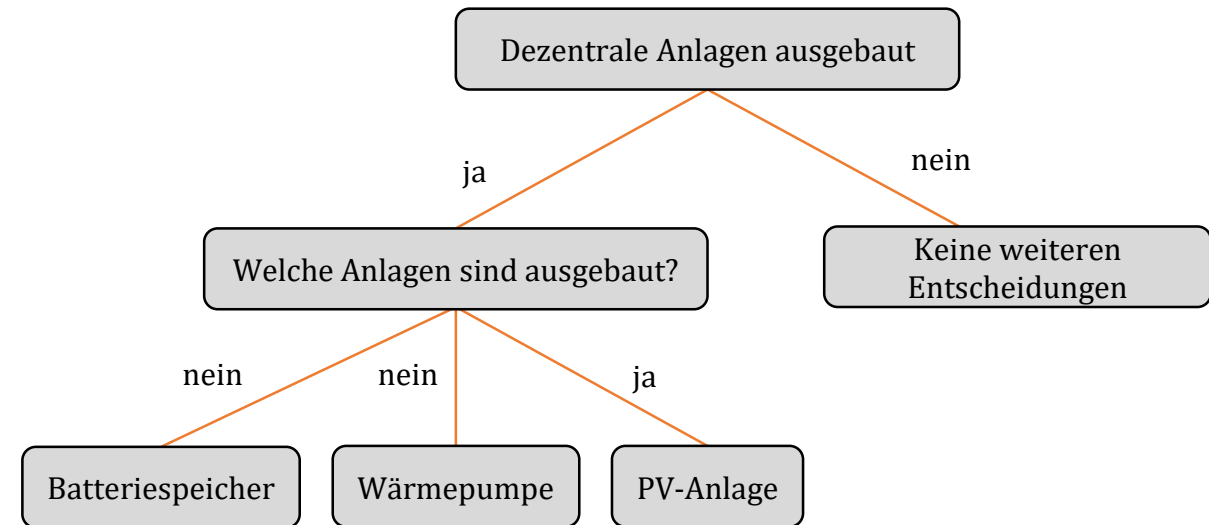


- Aktualisierung der Gewichte zwischen den neuronalen Schichten und somit die Fähigkeit zu „lernen“
- Fehler kann mit verschiedenen Fehlerfunktionen berechnet werden



# Entscheidungsbäume

- Gut geeignet zur Klassifizierung
- Auch geeignet bei nicht kategorialen Daten
- Neigt zur Überanpassung
- Abhängig von Hyperparametern
  - Max Features:
    - -> Wie viele Merkmale werden maximal benutzt
  - Maximale Tiefe:
    - -> Entscheidungsgrenzen werden komplizierter
  - Min samples split
  - Min samples leaf

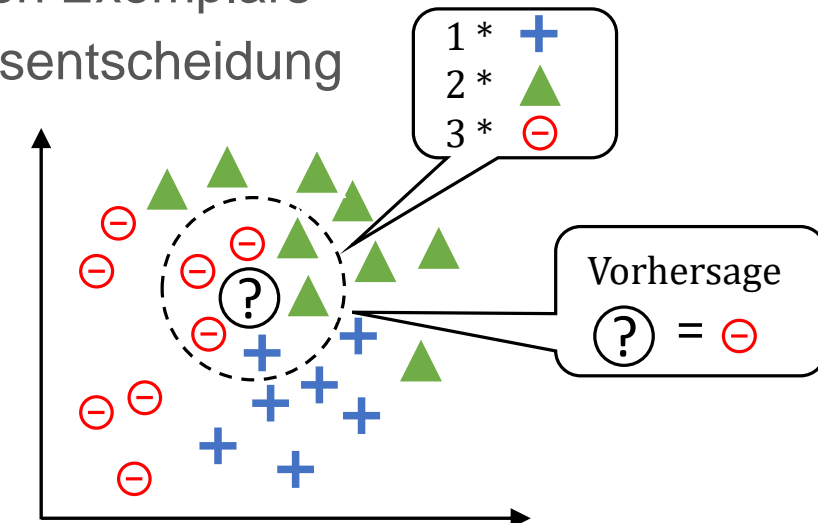


# Random Forest

- mehrere unkorrelierte Entscheidungsbäume
  - Sonderform von Entscheidungsbäumen, eine sogenannte Ensemble Methode
  - Ensemble mehrerer Entscheidungsbäume mit jeweils großer Varianz bilden ein stabileres Modell, welches beim Verallgemeinern besser arbeitet
  - Sehr gut geeignet zur Klassifizierung
  - Neigt weniger zur Überanpassung als einzelne Entscheidungsbäume

## K-nächste-Nachbarn

- Algorithmus speichert die Trainingsdaten
- Funktionsweise:
  - 1. Auswahl der Merkmale und Sammeln von Trainingsdaten
  - 2. Auswahl der k nächsten Nachbarn des zu klassifizierenden Exemplars
  - 3. Zuweisung der Klassenbezeichnung durch eine Mehrheitsentscheidung
- Neigt zur Überanpassung bei hoher Dimensionalität



## Bewertungskriterien

- Berechnung der Binären Kreuz Entropie
  - „Wahrscheinlichkeit der Klasse 1 zu entsprechen“ für jede Ausbauoption
  - → Schwellenwertklassifizierung zur Maximierung der Ergebnisse

	Predicted Class			„Confusion Matrix
		Class = 1	Class = 0	
	Actual Class	Class = 1	Class = 0	
		True Positive	False Negative	
		False Positive	True Negative	

- Multilabel:

$$TP = 2 \sum_{i=1}^n \sum_j^m \mathbb{1}(P_{ij} = 1) \mathbb{1}(G_{ij} = 1)$$

- Analog dazu TN, FP und FN

## Bewertungskriterien

- Accuracy Score
  - Der für ein Sample vorhergesagte Satz von Labels muss *genau* mit dem entsprechenden wirklichen Satz von Labels übereinstimmen.
  - Formel für Multilabel:

$$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN}$$

Oder

$$\text{Acc}(P|G) = \frac{1}{nm} \sum_{i=1}^n \sum_j^m \mathbb{1}(P_{ij} = G_{ij})$$

- → True: [0,1] und [1,1] → Predicted: [1,1], [1,1] → AS = 0.5

## Bewertungskriterien

- Jaccard Similarity
  - Der Jaccard Ähnlichkeitskoeffizient ist definiert als die Größe des Schnittpunktes geteilt durch die Größe der Vereinigung zweier Labelsätze.
  - Formel:

$$\text{Jaccard} = \frac{TP}{TP + FN + FP}$$

- → True: [0,1] und [1,1] → Predicted [1,1], [1,1] → JS = 0.75

# Bewertungskriterien

- F1 Score
  - Der F1-Score kann als gewichteter Mittelwert aus „Precision Score“ und „Recall“ interpretiert werden.
- Precision Score
  - Der „Precision Score“ ist die Fähigkeit des Klassifikators, eine Probe, die negativ ist, nicht als positiv zu kennzeichnen.
- Recall
  - Der „Recall“ ist die Fähigkeit des Klassifikators, alle positiven Proben zu finden.

## Ergebnisse - Bewertungskriterien

- Test auf Über- und Unteranpassung:
  - Modell mit den Trainingsdaten validieren
    - Wenn Trainingsscore ~100% -> Überanpassung
    - Wenn Trainigsscore schlecht -> Unteranpassung
    - Wenn Trainingsscore ähnlich Testscore -> generalisiertes Modell
    - Hier: Exemplarisch anhand der „Baum Tiefe“ eines Entscheidungsbaumes getestet.

**Kernaussage:** Modelle müssen anhand bestimmter Hyperparemeter auf Über-/Unteranpassung getestet und validiert werden.

