

**Institut für Akustik und Sprachkommunikation**  
Professur für Sprachtechnologie und Kognitive Systeme

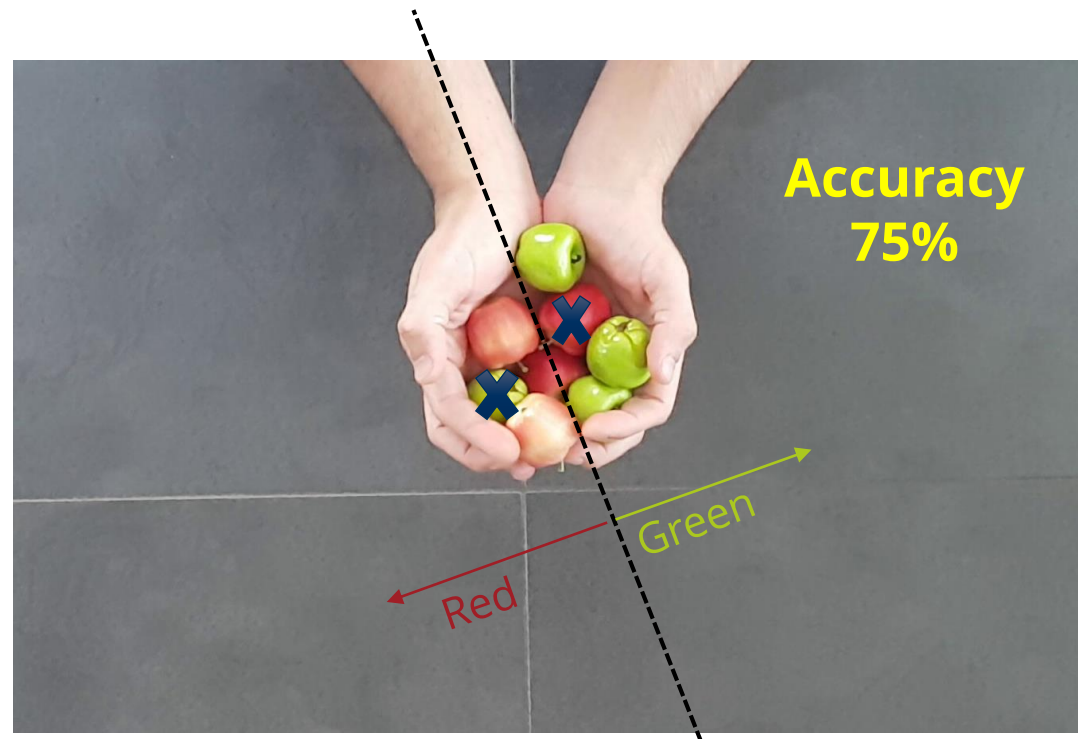
# **Reservoir Computing**

## Einführung und Zeitreihenvorhersage

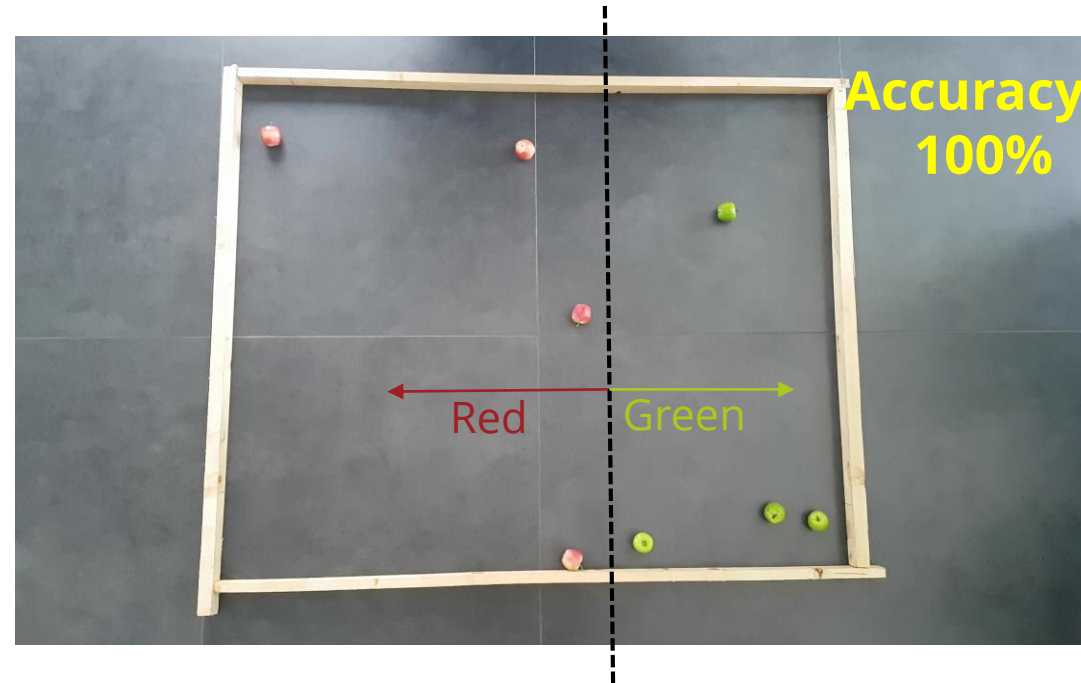
Dipl.-Ing. Peter Steiner  
Dresden // 21.04.2023

# Reservoir Computing

## Einführung



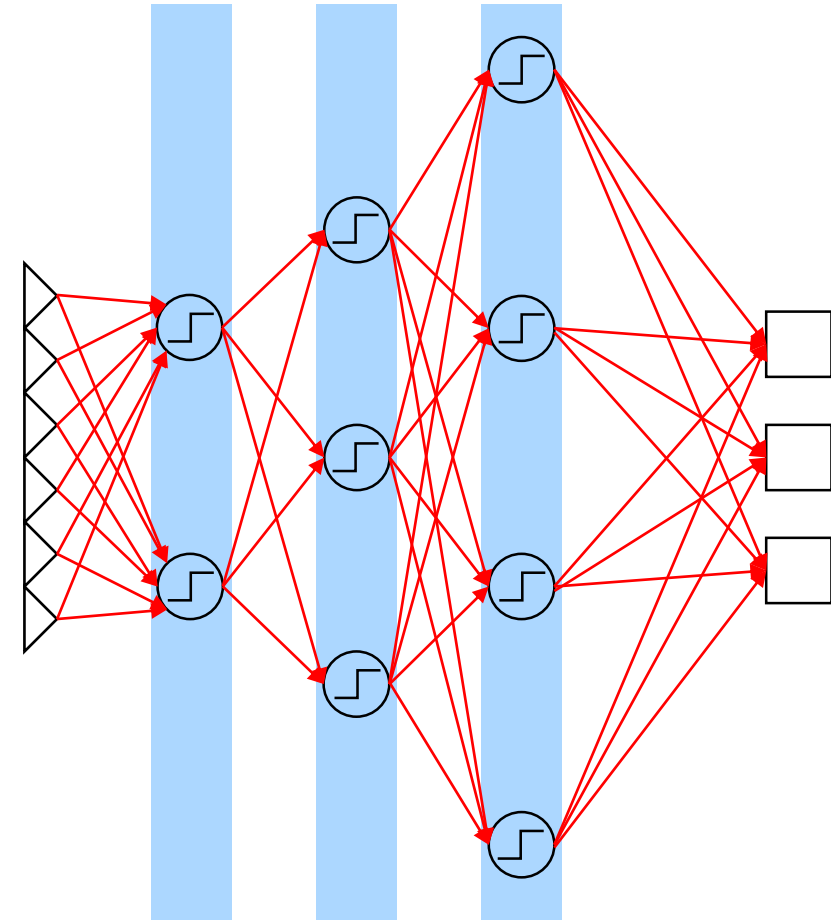
<https://acee.princeton.edu/events/summer-seminar-series-azarakhsh-jalalvand/>



<https://acee.princeton.edu/events/summer-seminar-series-azarakhsh-jalalvand/>

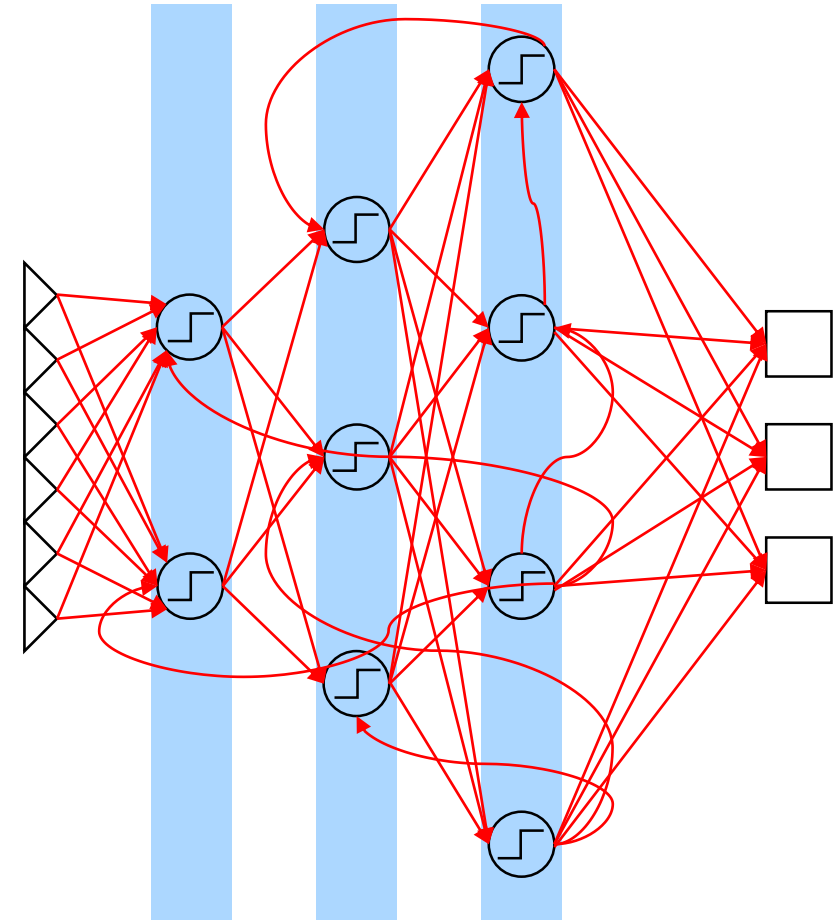
# Das mehrschichtige Perzeptron (multi-layer perceptron, MLP)

- Ein MLP besteht aus beliebig vielen verdeckten Schichten und einer Ausgabe-Schicht
- Jede Schicht abstrahiert den Eingang mehr
- Die Ausgabe-Schicht löst eine Aufgabe (Klassifikation oder Regression)
- Alle Verbindungen werden mittels Backpropagation trainiert
- MLPs können keinen zeitlichen Kontext berücksichtigen



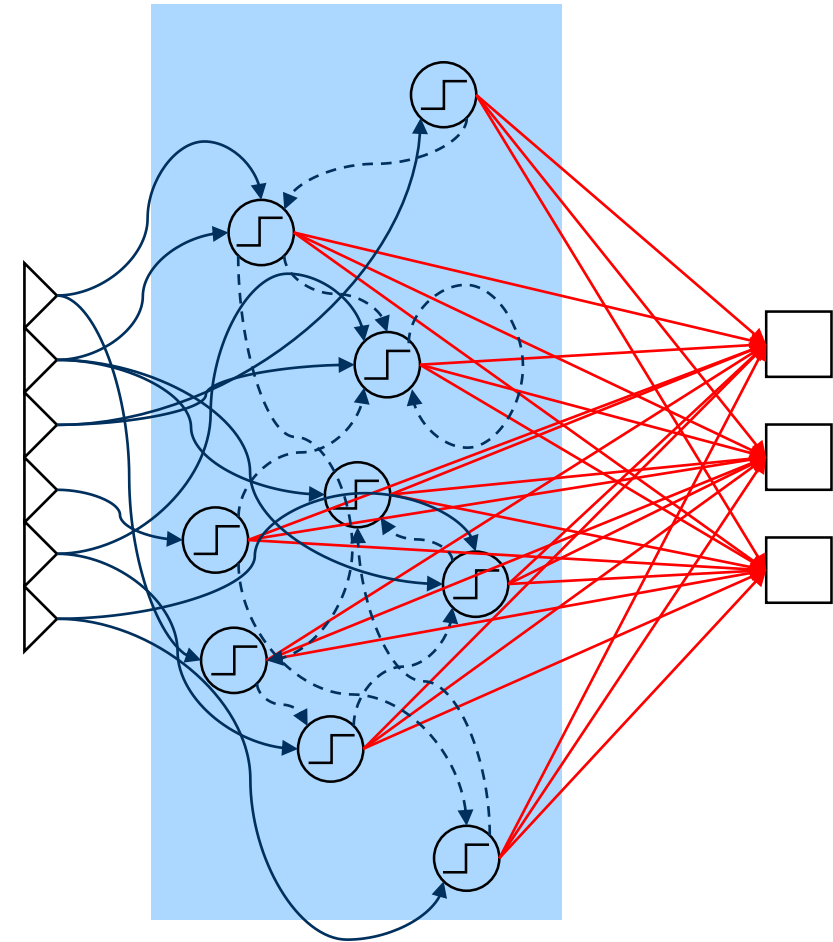
# Rekurrentes Neuronale Netzwerk (RNN)

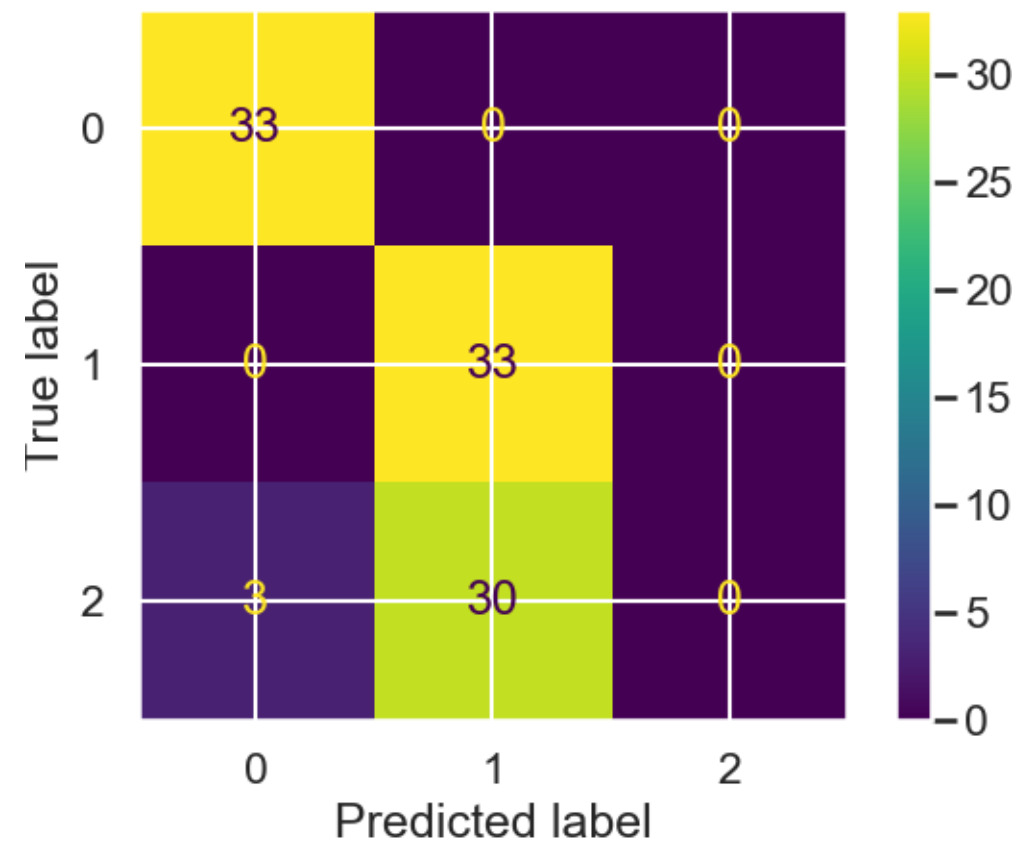
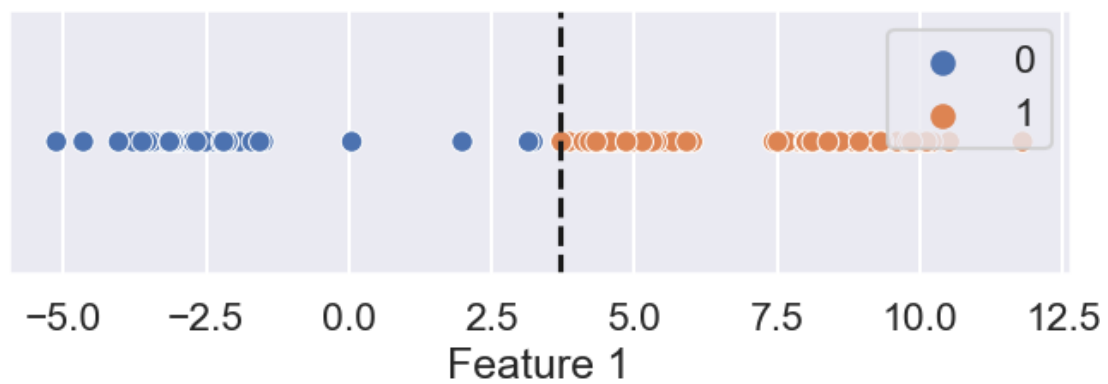
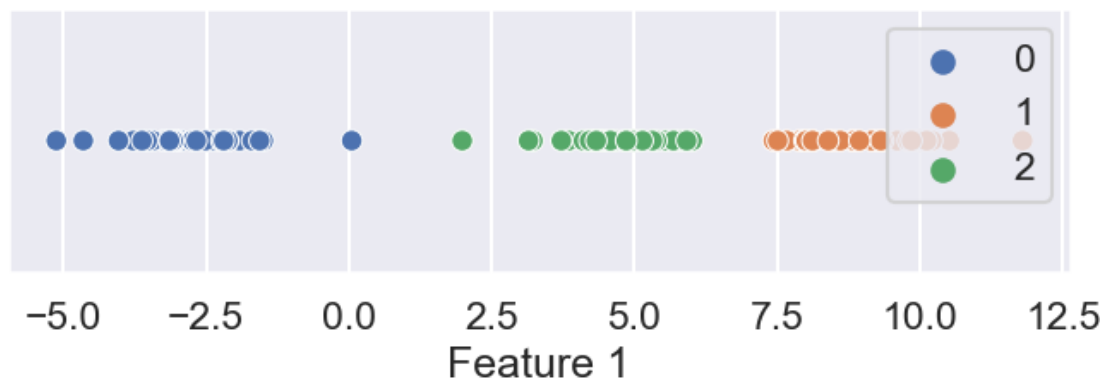
- Ein RNN besteht aus beliebig vielen verdeckten Schichten und einer Ausgabe-Schicht
- Rekurrente Verbindungen ermöglichen es, den vorherigen verdeckten Zustand zu benutzen
- Die Ausgabe-Schicht löst eine Aufgabe (Klassifikation oder Regression)
- Alle Verbindungen (auch rekurrente) werden mittels Backpropagation trainiert
- RNNs können einen zeitlichen Kontext berücksichtigen



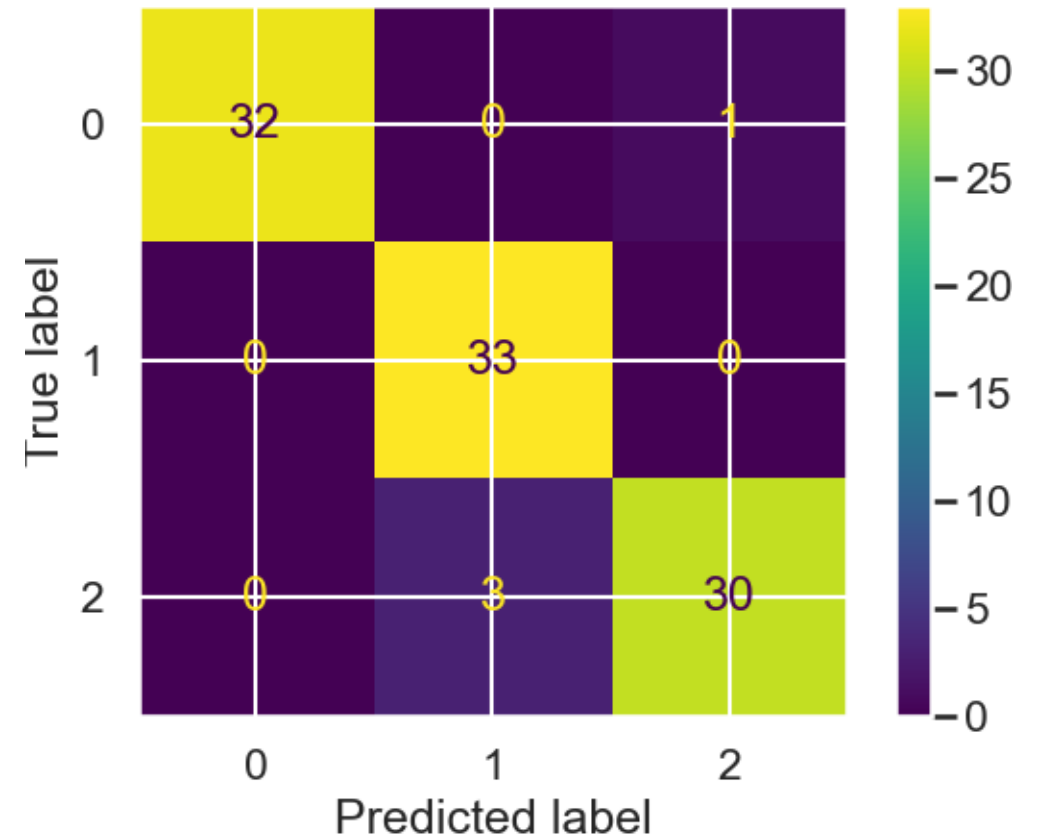
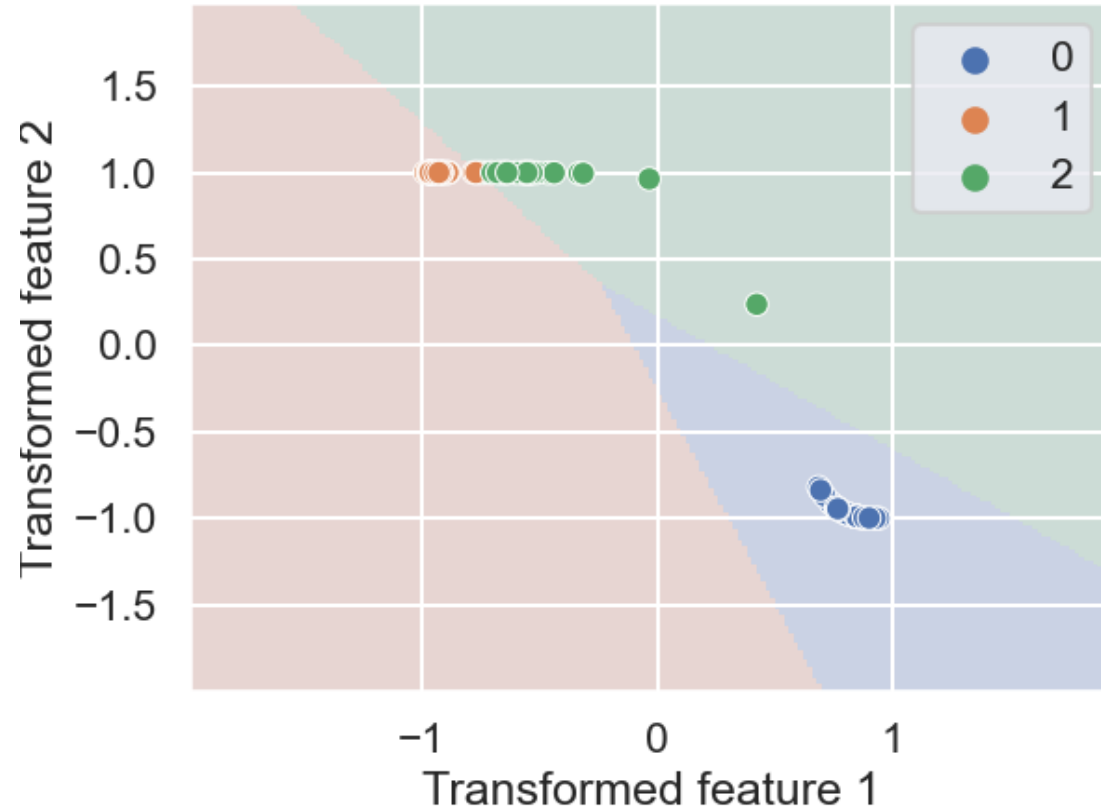
# Reservoir Computing Network (RCN)

- Ein RCN besteht aus einer Eingabe-, einer verdeckten rekurrenten, und aus einer Ausgabe-Schicht
- Rekurrente Verbindungen ermöglichen es, den vorherigen verdeckten Zustand zu benutzen
- Die Ausgabe-Schicht löst eine Aufgabe (Klassifikation oder Regression)
- Nur die Verbindungen von der verdeckten Schicht zur Ausgabeschicht werden trainiert
- RCNs können einen zeitlichen Kontext berücksichtigen









# Wie funktioniert Reservoir Computing generell?

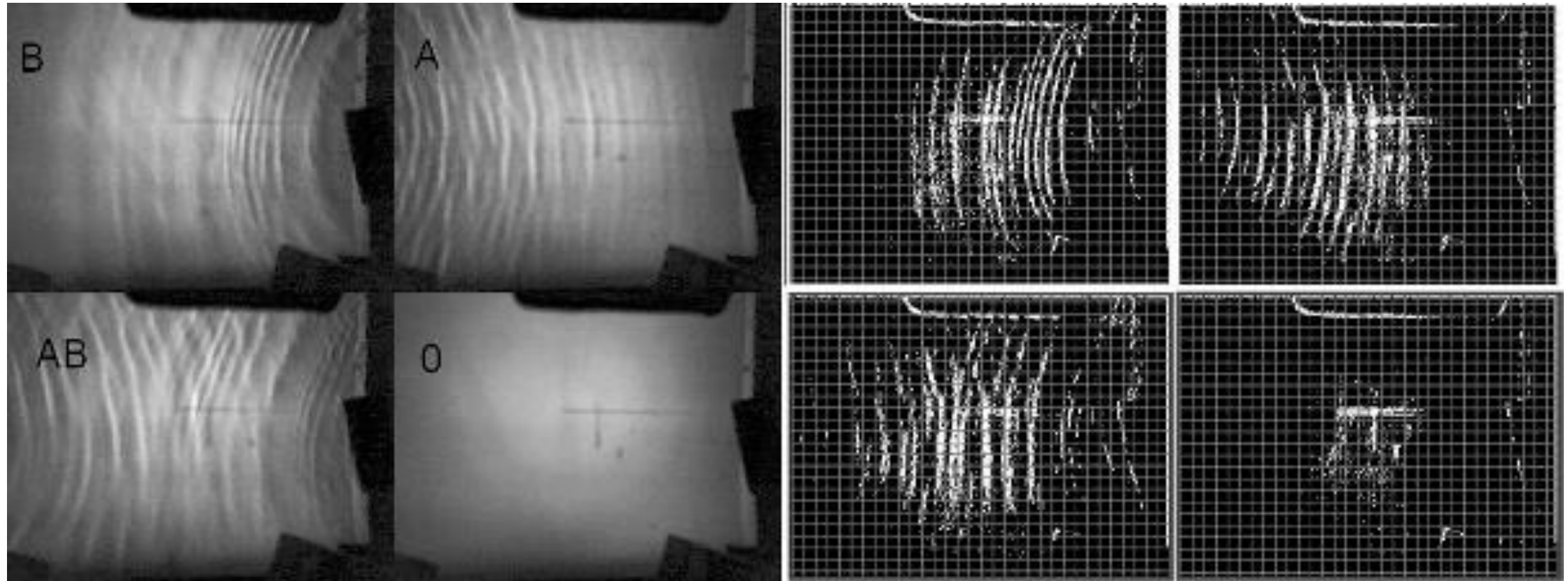
## Pattern recognition in a bucket



Fernando, C., & Sojakka, S. (2003, September). Pattern recognition in a bucket. In *European Conference on Artificial Life* (pp. 588-597). Springer, Berlin, Heidelberg.

# Wie funktioniert Reservoir Computing generell?

## Pattern recognition in a bucket



Fernando, C., & Sojakka, S. (2003, September). Pattern recognition in a bucket. In *European Conference on Artificial Life* (pp. 588-597). Springer, Berlin, Heidelberg.

# Reservoir Computing

- **Hauptsächlich umfasst Reservoir Computing zwei Arten von RNNs:**
  - Liquid State Machines (Maass 2002)
  - Echo State Networks (Jaeger 2001)
- **Zum breiteren Framework gehören Extreme Learning Machines (Huang 2006)**
- **All diese Ansätze haben Gemeinsamkeiten:**
  - Große verdeckte Schichten mit häufig zufällig initialisierten Eingangs- und Reservoirgewichten
  - Nur ein kleiner Teil des Netzwerkes, nämlich die Ausgangsgewichte, werden trainiert

Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11), 2531-2560.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report, 148(34), 13.

Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3), 489-501.

# Echo State Network (ESN)

- Ein ESN besteht aus Eingabe-, Bias-, rekurrenten, und aus einer Ausgabe-Schicht

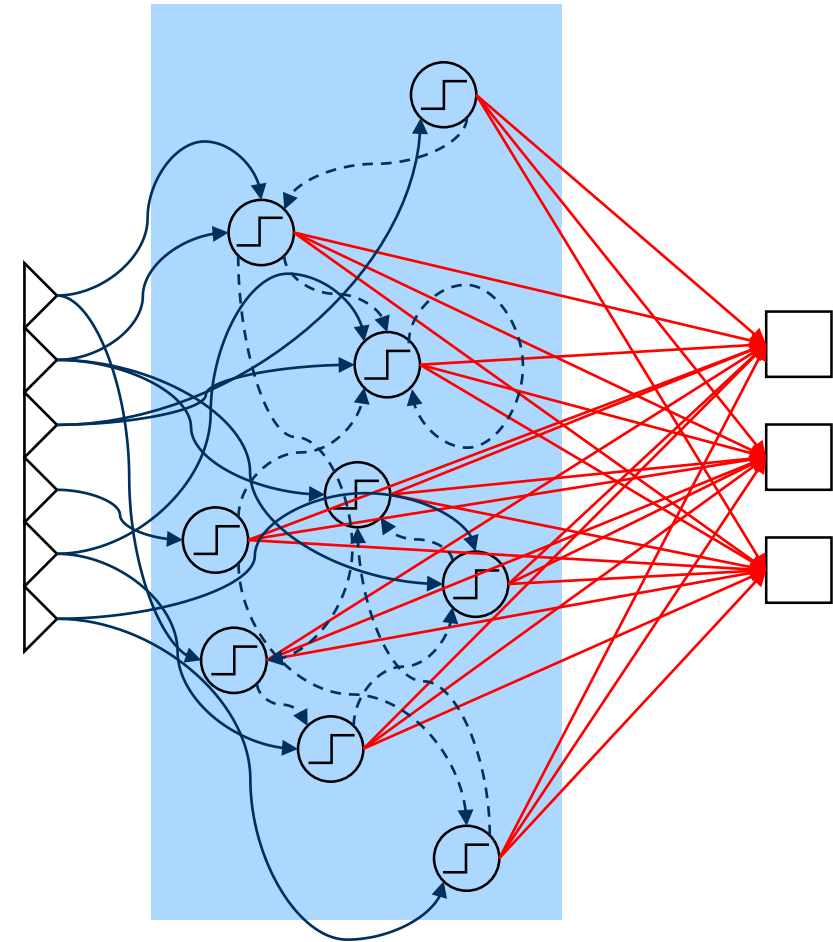
- Nichtlineare Transformation in einen hoch-dimensionalen rekurrenten Merkmalsraum

$$\mathbf{r}'[n] = f'(\mathbf{W}^{\text{in}}\mathbf{u}[n] + \mathbf{w}^{\text{bi}})$$

$$\mathbf{r}[n] = (1 - \lambda)\mathbf{r}[n - 1] + \lambda f(\mathbf{r}'[n] + \mathbf{W}^{\text{res}}\mathbf{r}[n - 1])$$

- Lineare Abbildung von diesem Merkmalsraum in den Zielraum, beispielsweise auf Klassen

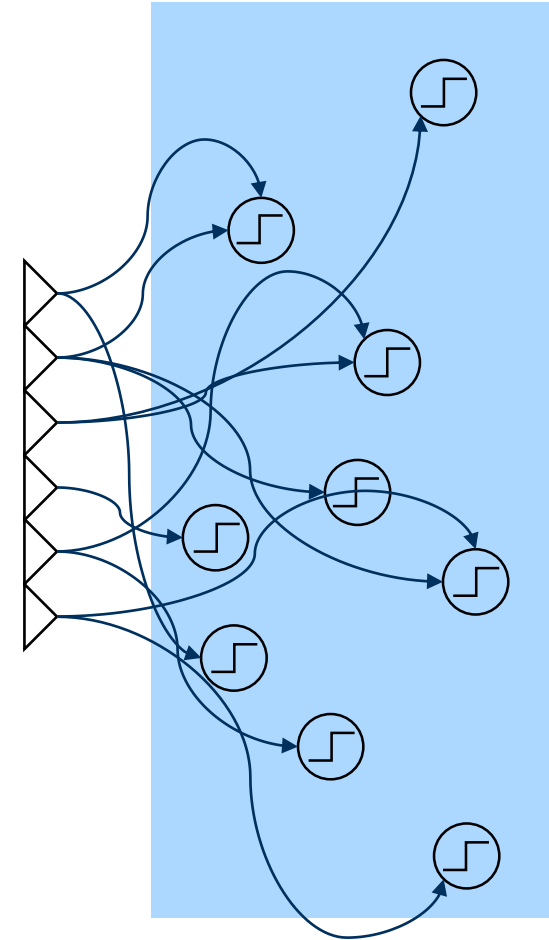
$$\mathbf{y}[n] = \mathbf{W}^{\text{out}}\mathbf{r}[n]$$



# Initialisierung und Training von ESNs

## Eingangs- und Bias-Gewichte

- Die Eingangs- und Bias-Gewichte werden aus einer Gleichverteilung zwischen  $\pm 1$  initialisiert
- Jedes Neuron empfängt nur eine begrenzte Anzahl an Eingangsmerkmalen – schmal besetzte Gewichtsmatrizen
- Jedes Neuron empfängt einen konstanten Bias-Eingang

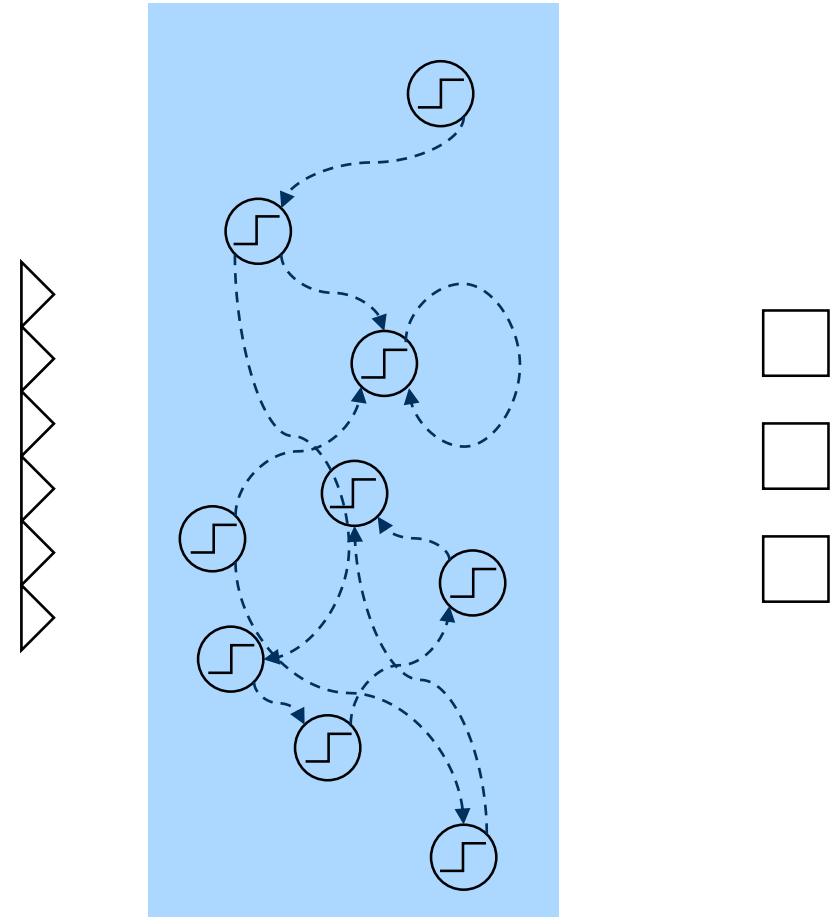




# Initialisierung und Training von ESNs

## Rekurrente Gewichte

- Die rekurrenten Gewichte werden aus einer Standardnormalverteilung initialisiert
- Jedes Neuron empfängt nur eine begrenzte Anzahl an Ausgängen von anderen Neuronen – schmal besetzte Gewichtsmatrizen
- Die rekurrenten Gewichte werden auf ihren maximalen absoluten Eigenwert normiert, um Stabilität zu gewährleisten



# Initialisierung und Training von ESNs

## Ausgangsgewichte

1. Initialisieren und fixieren der Eingangs-, Bias-, und rekurrenten Gewichte – diese sind nicht Teil des Trainings

2. Propagieren der Trainingsdaten durch das ESN, um die verdeckten Zustände zu berechnen

$$\mathbf{r}'[n] = f'(\mathbf{W}^{\text{in}}\mathbf{u}[n] + \mathbf{w}^{\text{bi}})$$

$$\mathbf{r}[n] = (1 - \lambda)\mathbf{r}[n - 1] + \lambda f(\mathbf{r}'[n] + \mathbf{W}^{\text{res}}\mathbf{r}[n - 1])$$

3. Sammeln aller verdeckten Zustände und der Zielausgaben

4. Lösung der linearen Regression

$$\mathbf{W}^{\text{out}} = (\mathbf{R}\mathbf{R}^T + \epsilon\mathbf{I})^{-1}(\mathbf{D}\mathbf{R}^T)$$

- Vergleichsweise einfaches Training, verglichen mit dem iterativen Backpropagation Prozess
- Inkrementelles Training und eine einfache Adaption an neue Daten möglich



# Zeitreihenvorhersage

## Tutorial

# Zeitreihenvorhersage

- **Dieses Tutorial dient als praktische Einführung in die Zeitreihenvorhersage**
- **Wir beschäftigen uns mit folgenden Themen:**
  - Grundsätzliche Methodik bei der Analyse eines Datensatzes
  - Vorverarbeitung eines Datensatzes (Konvertierung von Datentypen, Beschreibung und Skalierung)
  - Optimierung verschiedener linearer und nicht-linearer Modelle für die Zeitreihenvorhersage
  - Design komplexerer Merkmale
- **Materialien für dieses Tutorial stehen auf Github zur Verfügung und dürfen gerne verwertet werden**

# Zeitreihenvorhersage

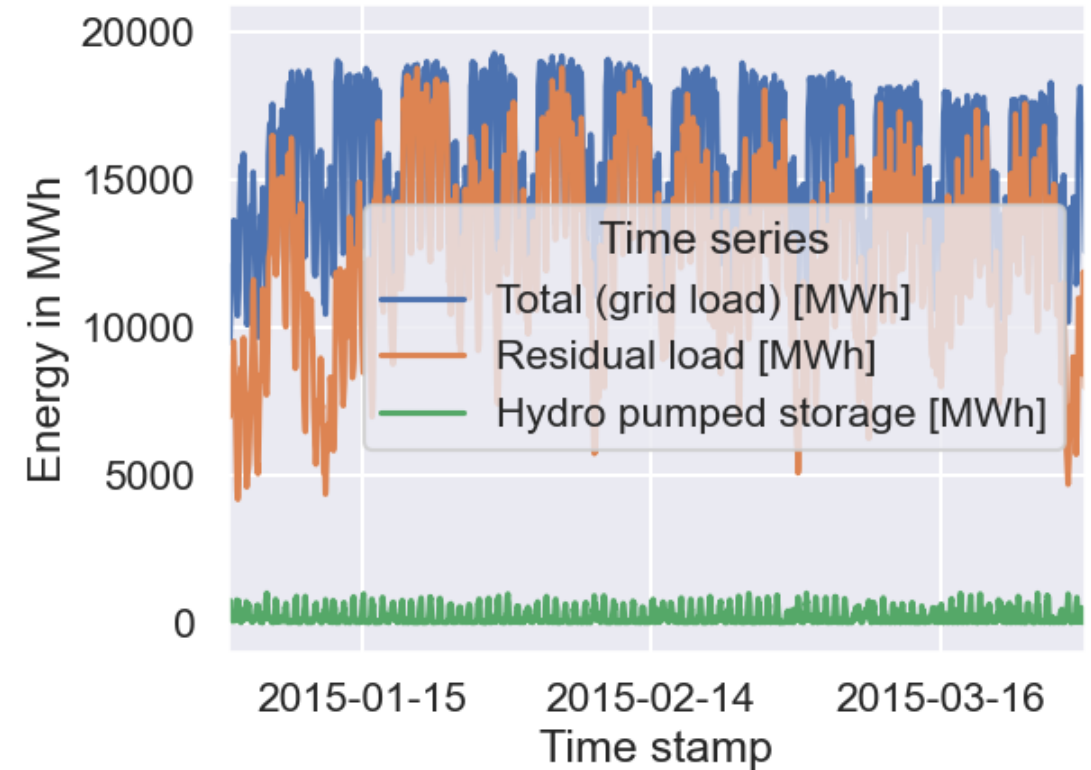
## Datenanalyse und -konvertierung

- **Der erste Schritt bei jeder Aufgabe ist eine Analyse der vorhandenen Daten:**
  - Welche Daten haben wir in welcher Form?
  - Mit welchen Datentypen arbeiten wir?
  - Was wollen wir machen?
- **Nach der Definition der Aufgabenstellung kümmern wir uns um die Aufbereitung der Daten:**
  - Konvertierung der Daten in ein lesbares Format (Zeitangaben, richtiges Zahlenformat)
  - Gibt es ungültige Daten? Dann verwerfen wir diese entweder, oder konvertieren sie systematisch
- **Anschließend visualisieren wir die Daten in geeigneter Form, beispielsweise als Zeitreihen, Histogramme, oder Boxplots.**
- **Hilfreich ist es gelegentlich auch, die Statistiken des Datensatzes deskriptiv zu analysieren.**

# Zeitreihenvorhersage

## Datenanalyse und -konvertierung

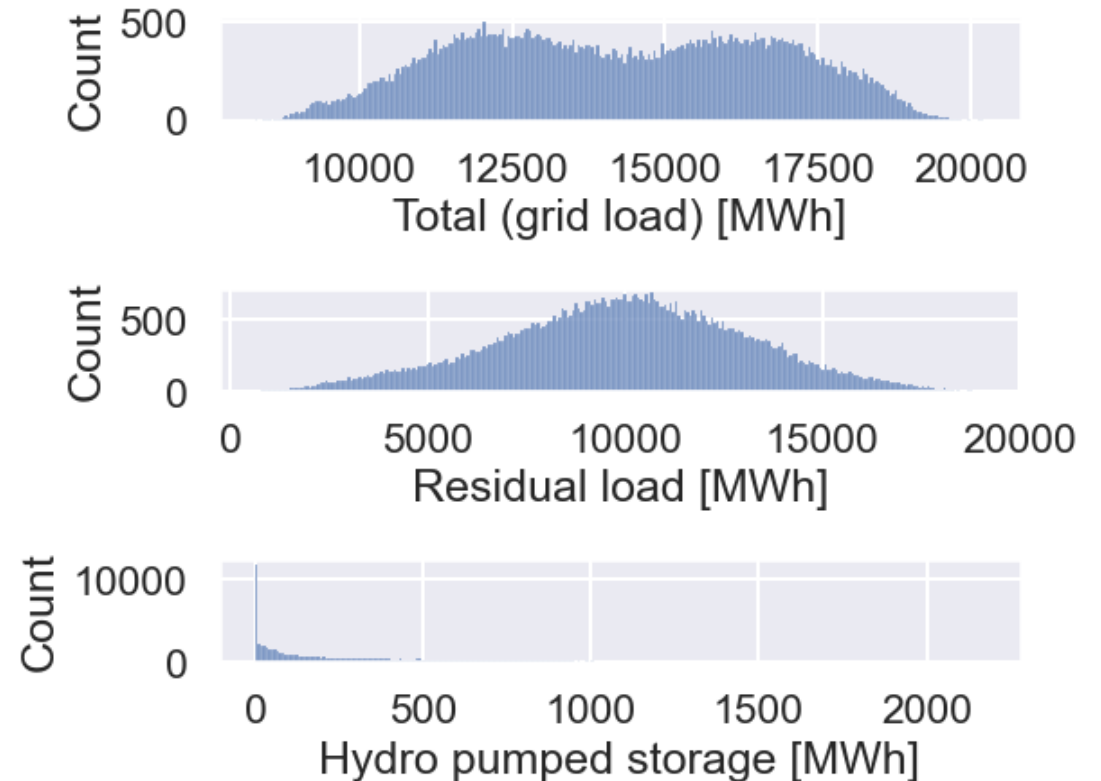
- **SMARD-Datensatz**
  - Realisierte Stromerzeugung ab dem 01.01.2015
- Die rohen Zeitreihen lassen auf ein quasiperiodisches Verhalten schließen



# Zeitreihenvorhersage

## Datenvorverarbeitung

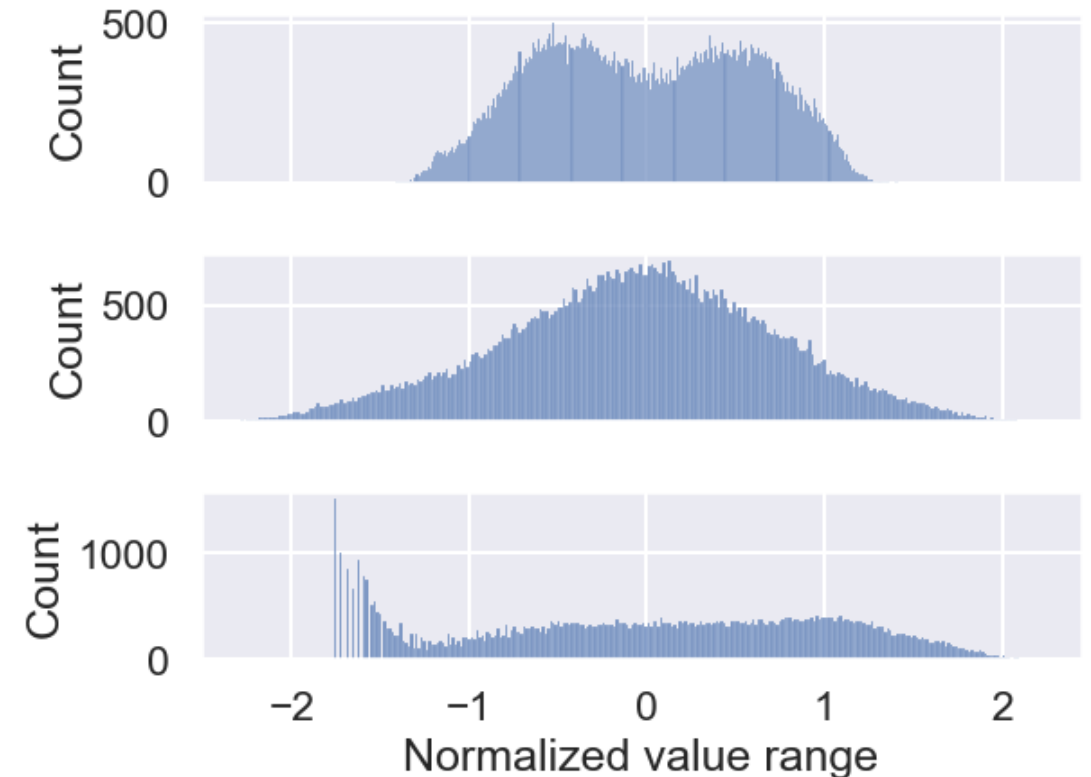
- **SMARD-Datensatz**
  - Realisierte Stromerzeugung ab dem 01.01.2015
- **Die rohen Zeitreihen lassen auf ein quasiperiodisches Verhalten schließen**
- **Es gibt große Unterschiede zwischen dem Pumpspeicher und den übrigen Größen**
  - Unterschiedliche Wertebereiche
  - Unterschiedliche Verteilungen
  - Passende Skalierung notwendig – DEMO



# Zeitreihenvorhersage

## Datenvorverarbeitung

- **SMARD-Datensatz**
  - Realisierte Stromerzeugung ab dem 01.01.2015
- **Die rohen Zeitreihen lassen auf ein quasiperiodisches Verhalten schließen**
- **Es gibt große Unterschiede zwischen dem Pumpspeicher und den übrigen Größen**
  - Unterschiedliche Wertebereiche
  - Unterschiedliche Verteilungen
  - Passende Skalierung notwendig – DEMO
- **Nach der Skalierung deutlich günstigere Verteilungen**



# Zeitreihenvorhersage

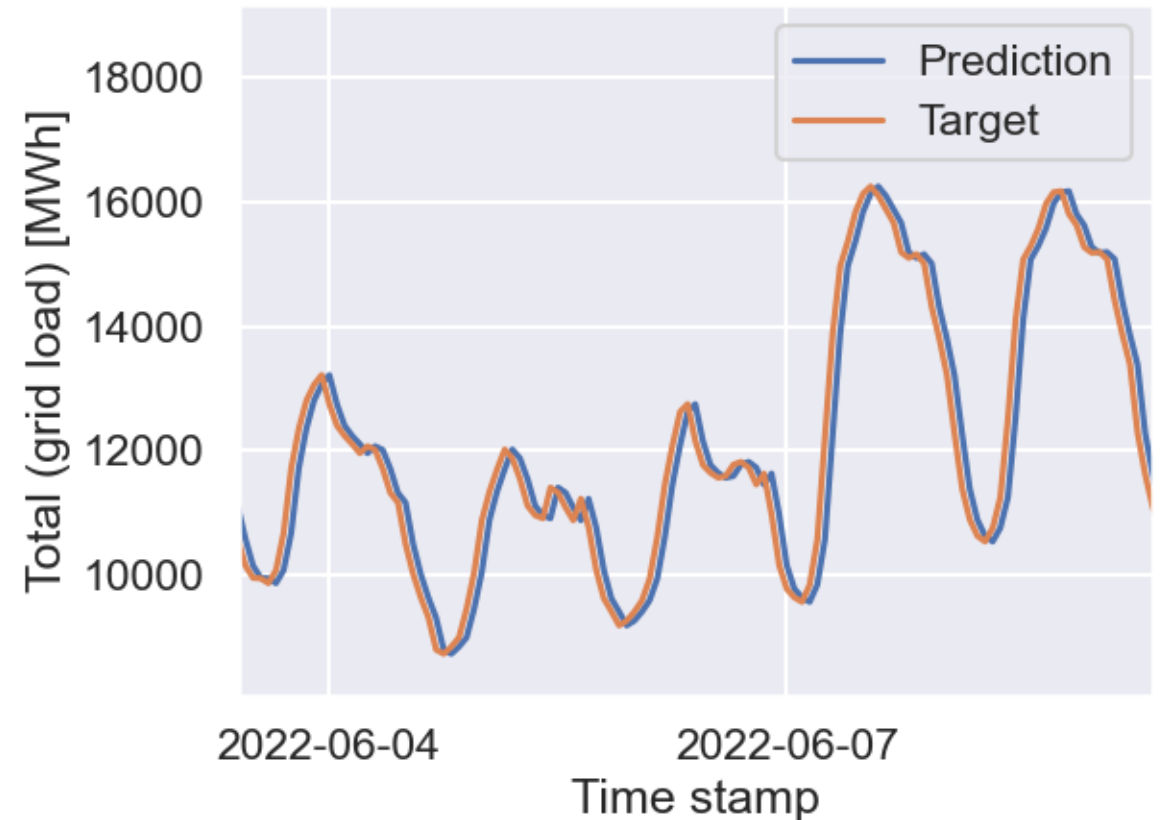
## Optimierung von Modellen zur Vorhersage

- **Der Datensatz ist analysiert und eine angemessene Methode zur Vorverarbeitung wurde gefunden**
- **Das weitere Ziel ist die Vorhersage der gesamten Netzlast in der nächsten Stunde**
- **Verschiedene Optionen:**
  - Vorhersage auf Basis eines oder mehrerer Merkmale (Netzlast oder auch Residuallast und Pumpspeicher)
  - Vorhersage auf Basis eines oder mehrerer Zeitschritte
  - Vorhersage auf Basis linearer oder nichtlinearer Modelle
- **In jedem Fall muss man zunächst eine Basislinie festlegen**
  - Komplexere Modelle sollten diese Basislinie übertreffen
  - Die Frage nach der Basislinie kommt häufig!

# Zeitreihenvorhersage

## Naives Basismodell

- Dies kann ein untrainiertes Modell sein
- **Wir haben eine quasiperiodische Zeitreihe:**
  - Die aktuelle Eingabe kann vorhergesagt werden
  - „Kopieren“ der Eingabe zur Ausgabe



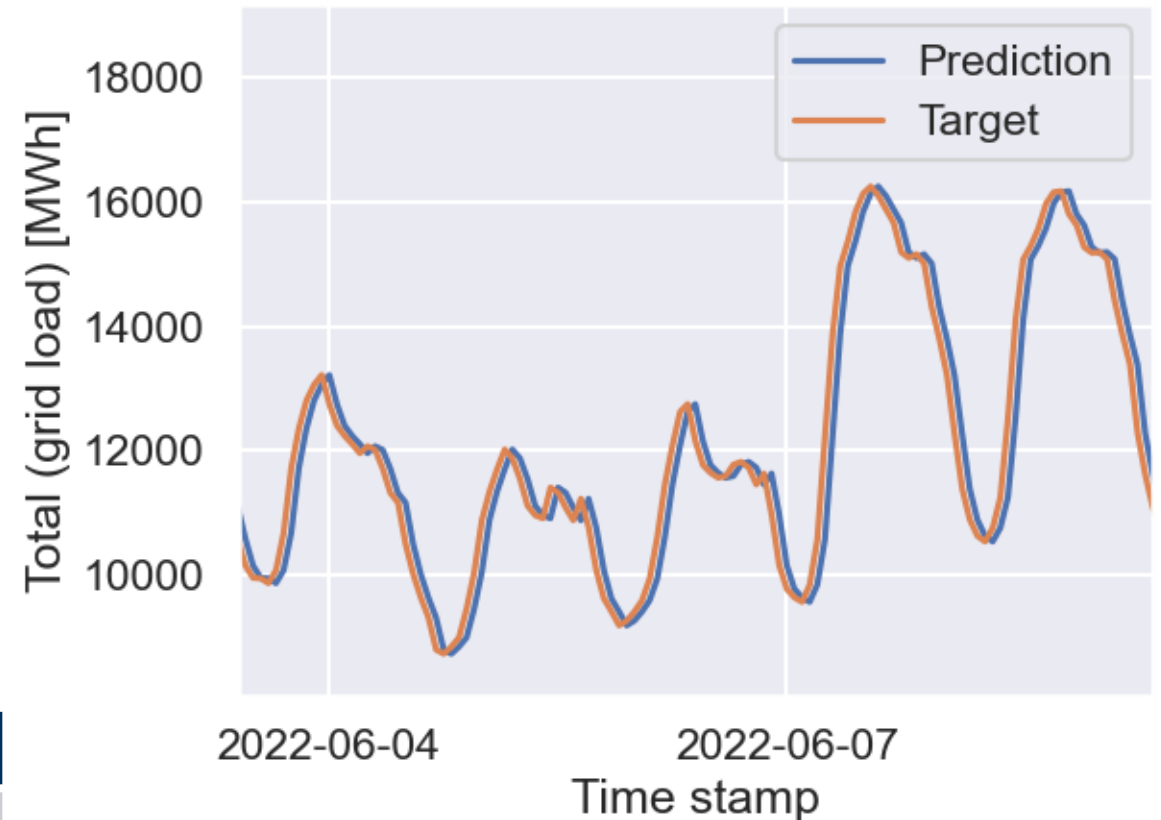


# Zeitreihenvorhersage

## Naives Basismodell

- Dies kann ein untrainiertes Modell sein
- Wir haben eine quasiperiodische Zeitreihe:
  - Die aktuelle Eingabe kann vorhergesagt werden
  - „Kopieren“ der Eingabe zur Ausgabe

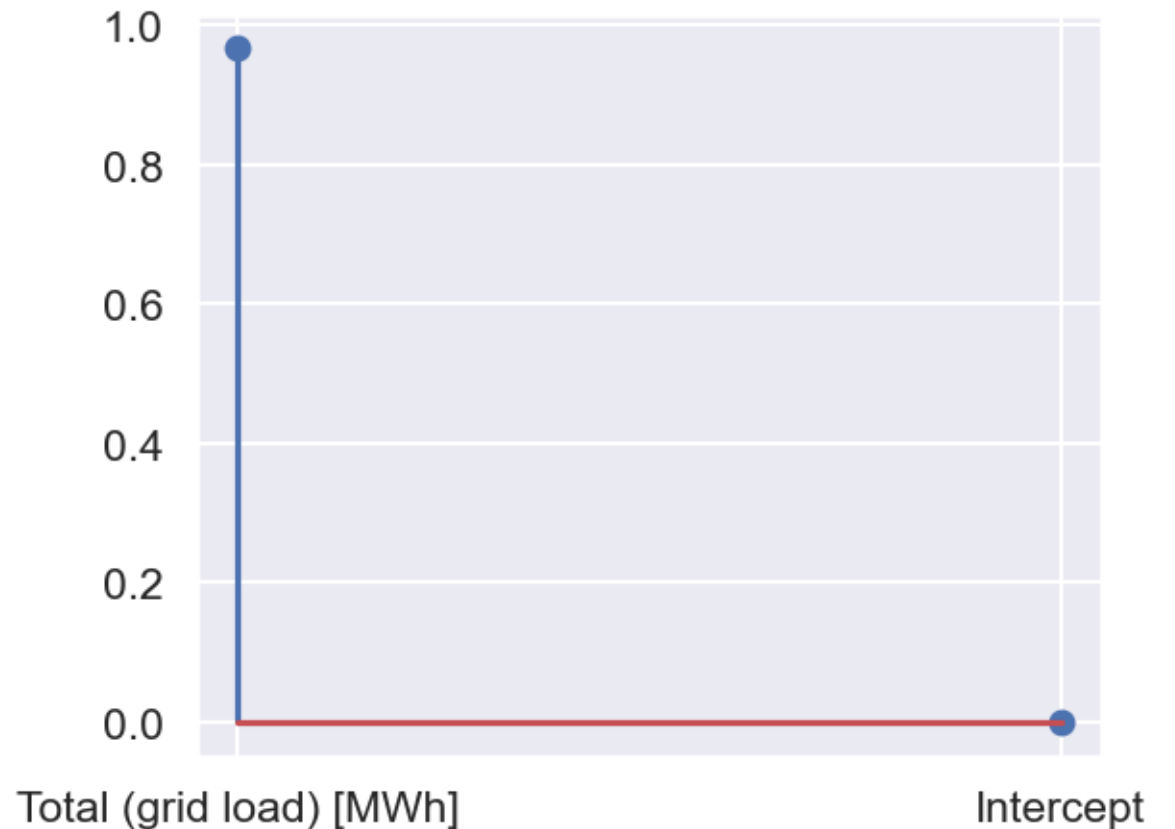
MSE	Korrelation
342181	0,9310



# Zeitreihenvorhersage

## Lineare Regression

- Hier ist das Ziel, eine lineare Abbildung zwischen der Eingabe und der Zielausgabe zu finden
- Die Berechnung der Ausgabe ist nur von der aktuellen Eingabe abhängig
  - In diesem Falle gibt es nur eine Eingabe
  - Die Eingabe wird höchstwahrscheinlich „kopiert“

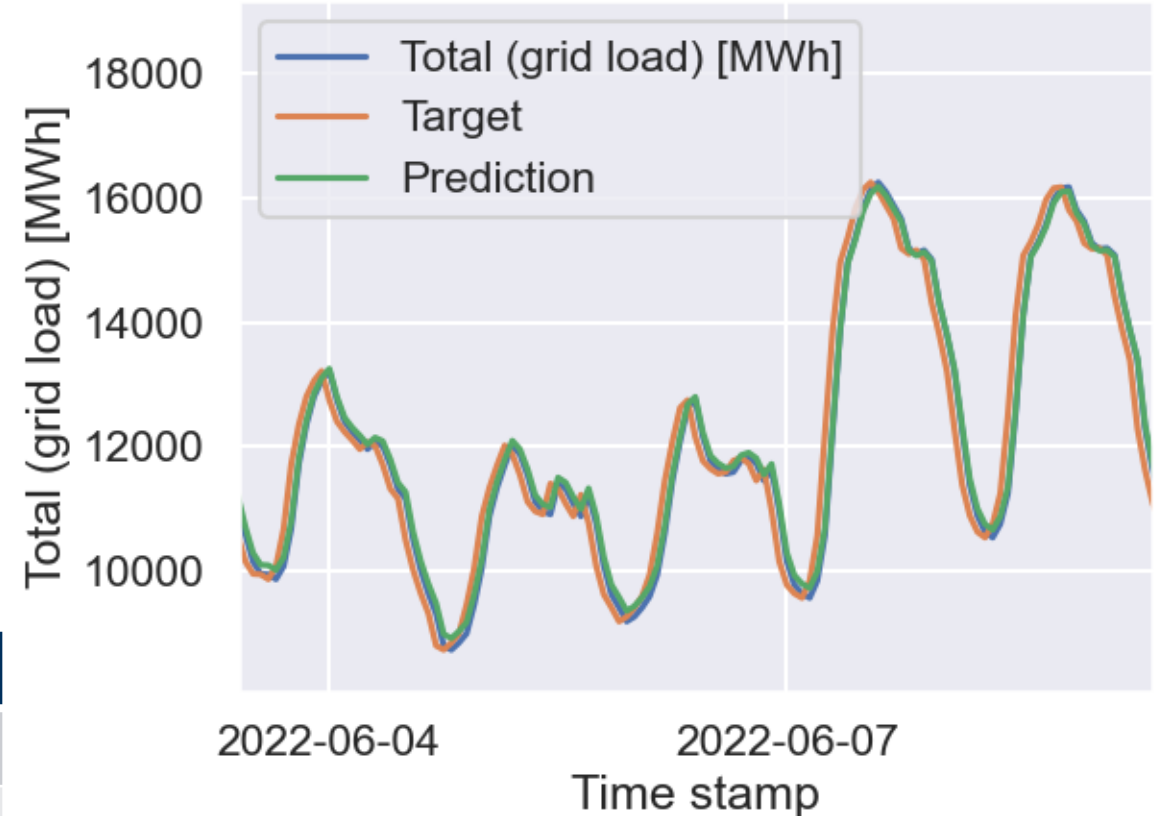


# Zeitreihenvorhersage

## Lineare Regression

- Hier ist das Ziel, eine lineare Abbildung zwischen der Eingabe und der Zielausgabe zu finden
- Die Berechnung der Ausgabe ist nur von der aktuellen Eingabe abhängig
  - In diesem Falle gibt es nur eine Eingabe
  - Die Eingabe wird höchstwahrscheinlich „kopiert“

MSE	Korrelation
341806	0,9316
337162	0,9320

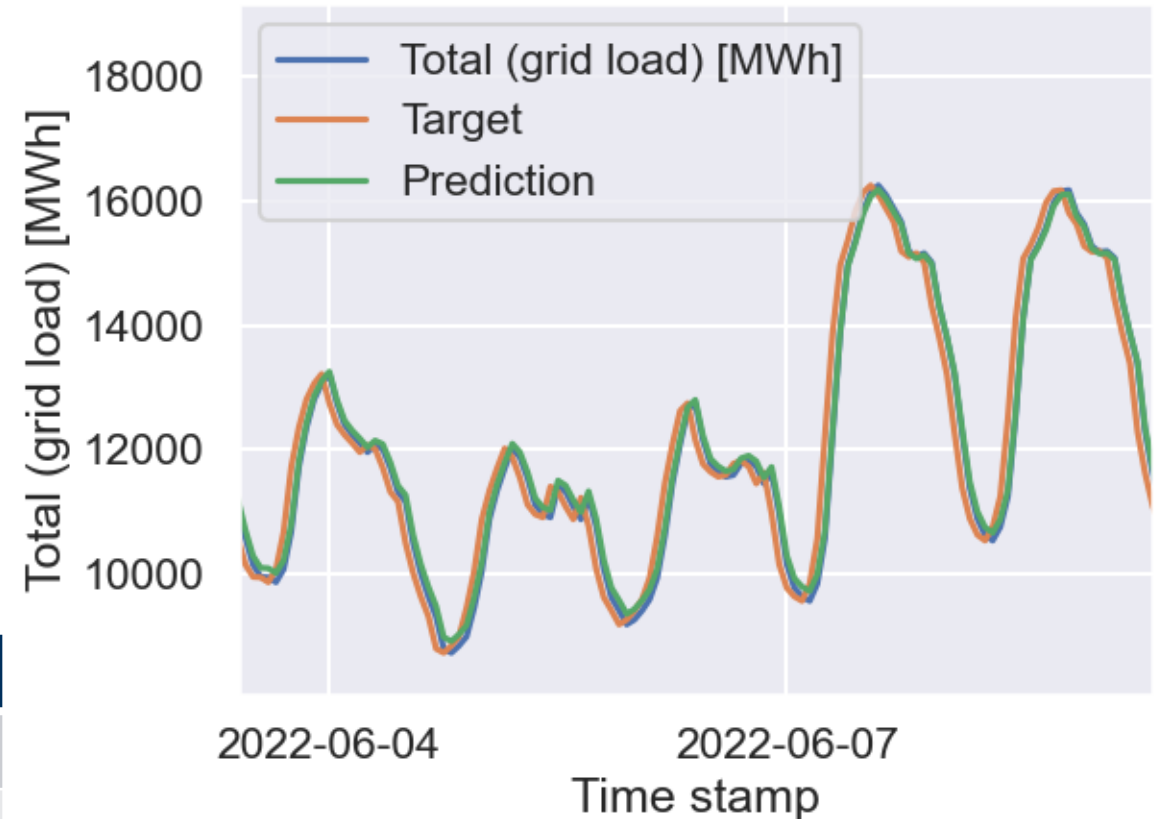


# Zeitreihenvorhersage

## Lineare Regression

- Hier ist das Ziel, eine lineare Abbildung zwischen der Eingabe und der Zielausgabe zu finden
- Die Berechnung der Ausgabe ist nur von der aktuellen Eingabe abhängig
  - In diesem Falle gibt es nur eine Eingabe
  - Die Eingabe wird höchstwahrscheinlich „kopiert“
- Dasselbe ist auch bei einem mehrschichtigen Perzeptron zu erwarten!

MSE	Korrelation
341806	0,9316
337162	0,9320



# Zeitreihenvorhersage

## Echo State Network

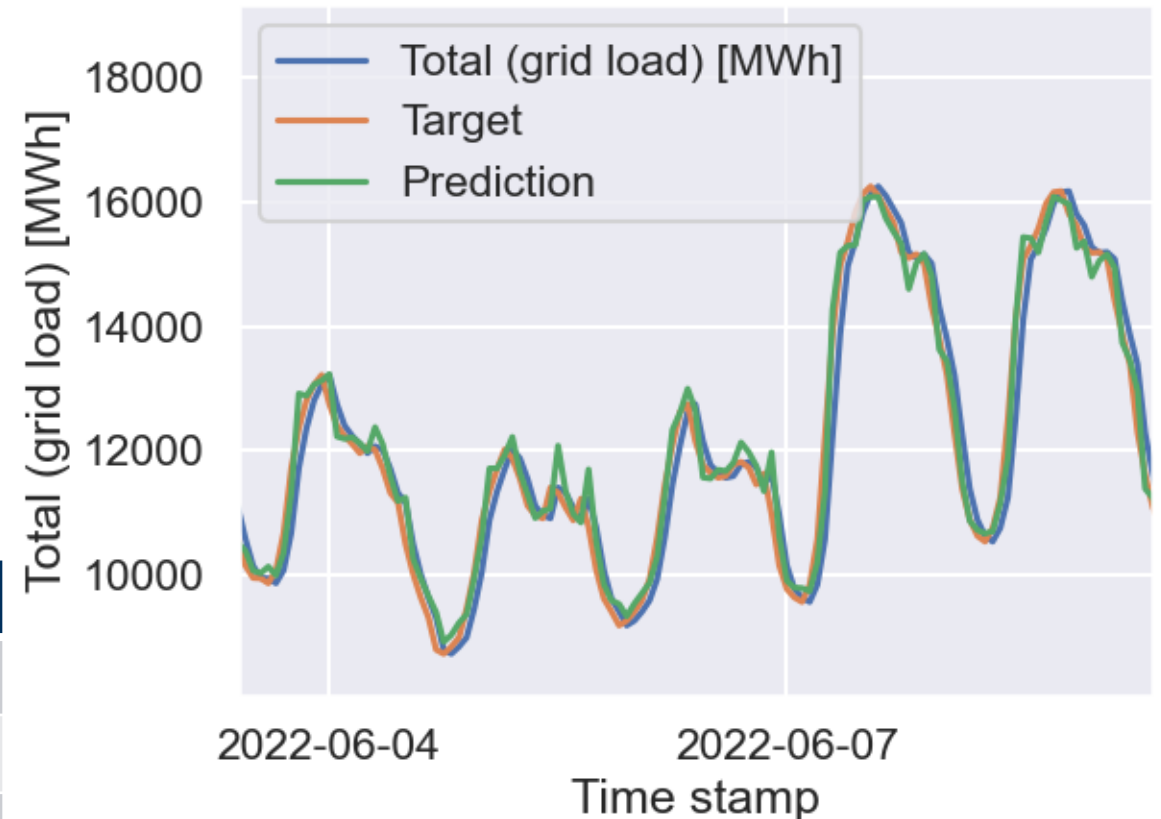
- **Wir können nun ein Echo State Network an dieselbe Stelle setzen**
- **Die Berechnung der Ausgabe ist nun von der aktuellen Eingabe und dem letzten verdeckten Zustand abhängig**
  - Intrinsisch gehen damit alle früheren Zustände in die Berechnung der aktuellen Ausgabe ein
  - Dies könnte eine mögliche Lösung sein

# Zeitreihenvorhersage

## Echo State Network

- Wir können nun ein Echo State Network an dieselbe Stelle setzen
- Die Berechnung der Ausgabe ist nun von der aktuellen Eingabe und dem letzten verdeckten Zustand abhängig
  - Intrinsisch gehen damit alle früheren Zustände in die Berechnung der aktuellen Ausgabe ein
  - Dies könnte eine mögliche Lösung sein

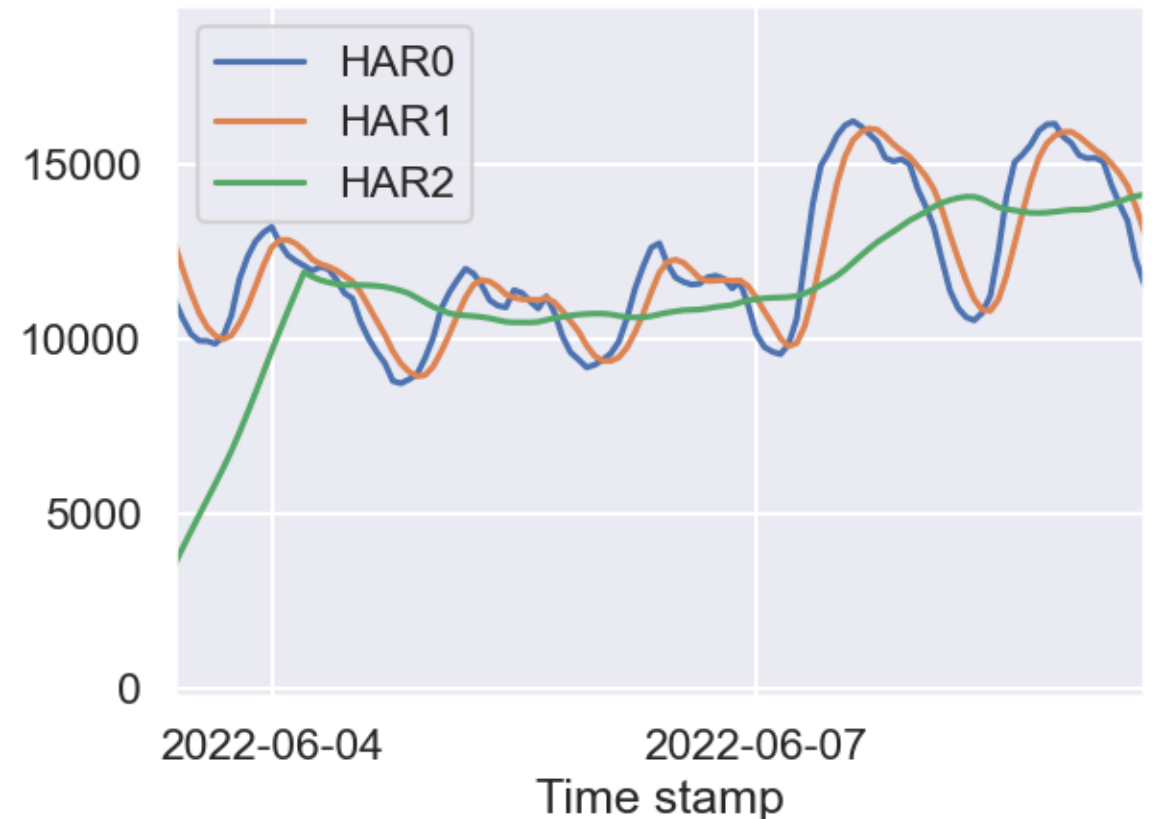
MSE	Korrelation
341806	0,9316
336821	0,9326
99211	0,98



# Zeitreihenvorhersage

## Komplexere Merkmalsextraktion

- **Heterogeneous Auto-Regressive Features:**
  - Ein Set aus verschiedenen Merkmalen
  - Originale Zeitreihe
  - Gefilterte Zeitreihen (gleitender Mittelwert über fünf bzw. 22 Zeitschritte hinweg)
- **Dieses Feature-Set hat Vorteile:**
  - Durch die Filterung können Langzeit-Trends in Zeitreihen berücksichtigt werden
  - Wir erhalten mehrere unkorrelierte Features



Corsi, Fulvio. "A simple approximate long-memory model of realized volatility." *Journal of Financial Econometrics* 7.2 (2009): 174-196.

# Zeitreihenvorhersage

## Komplexere Merkmalsextraktion

- **Wir können neben der vorherzusagenden Zeitreihe alle in einem Datensatz zur Verfügung stehenden Zeitreihen (Residuallast, Pumpspeicher, Zeitstempel) nutzen**
  - Jedes Merkmal muss immer richtig verarbeitet werden
  - Nicht alle Merkmale sind für jedes Modell geeignet – Feature-Engineering zwingend notwendig
- **Wir können auch die vorherzusagende Zeitreihe analysieren, beispielsweise eine FOURIER-Analyse**
  - Gibt es Perioden im Datensatz?
  - Können wir unsere Merkmale so vorverarbeiten, dass Perioden berücksichtigt werden?



# Zusammenfassung

- **Was sollten Sie heute gelernt haben:**
  - Die scheinbar einfache Zeitreihenvorhersage ist durchaus ein komplexes Gebiet
  - Eine gezielte Datenanalyse ist sehr wichtig für die Leistungsfähigkeit von Modellen
  - Eine geeignete Vorverarbeitung (Merkmalsextraktion) von Datensätzen erleichtert die Arbeit verschiedener Modelle signifikant
  - Für jede Aufgabe muss ein geeignetes Modell ausgewählt werden
- **Lineare bzw. kontextfreie Modelle in Kombination mit geeigneten Merkmalen sind mächtig:**
  - Die Leistungsfähigkeit eines MLPs kann an ein RNN heranreichen
  - Voraussetzung ist häufig, dass in den Merkmalen der zeitliche Kontext modelliert wird

# Zusammenfassung

Modell	MSE	Korrelation	Trainingszeit	Inferenzzeit
Basis	341806	0,9316	-	-
Lineare Regression (LR)	341806	0,9316	0,006s	0.002s
MLP	336821	0,9326	0,857s	0,003s
ESN	99211	0,98	1,195s	0,119s
HAR-LR	151334	0,9695	0,03s	0,01s
HAR-MLP	112004	0, 9774	1.459s	0,002s