

# Evaluierung der BEMS-Fähigkeiten für KI-basierte Strombedarfsprognose

Machbarkeitsstudie im Auftrag der SWM im Rahmen des unIT-e<sup>2</sup> Projekts



DLR Projektträger



Stadtwerke München

Förderkennzeichen SWM: 01MV21UN19

Gefördert durch:



Bundesministerium  
für Wirtschaft  
und Klimaschutz

aufgrund eines Beschlusses  
des Deutschen Bundestages

# Ausgangslage

## Das BEMS - Eine wesentliche Komponente zukünftiger KI-basierter Lösungen

- Im Cluster Cit-E-Life spielt das BEMS (Gebäudeenergiemanagementsystem) eine entscheidende Rolle bei der Integration dezentralisierter Systeme.
- Das BEMS wird eine erhebliche Menge an Daten verarbeiten, wodurch es zu einem entscheidenden Bestandteil für KI-basierte Lösungen wird. Besonders interessant ist es für die Entwicklung der Strombedarfsprognose (Power Demand Forecast, PODF).
- Aufgrund der begrenzten Rechenkapazität des BEMS und der Notwendigkeit einer hohen Sicherheit in Energiesystemen werden sowohl Cloud-basierte als auch Edge-basierte Implementierungen mit einzigartigen Herausforderungen einhergehen.

# Inhalt

1. Systembeschreibung
2. Power Demand Forecasting: Use Case Beschreibung
3. KI-Prognosen: Was ist KI-basierte Zeitreihenprognose?
4. KI-Prognosen im Kontext PODF
5. Cloud-Lösung vs Edge-Devices
6. Take Aways

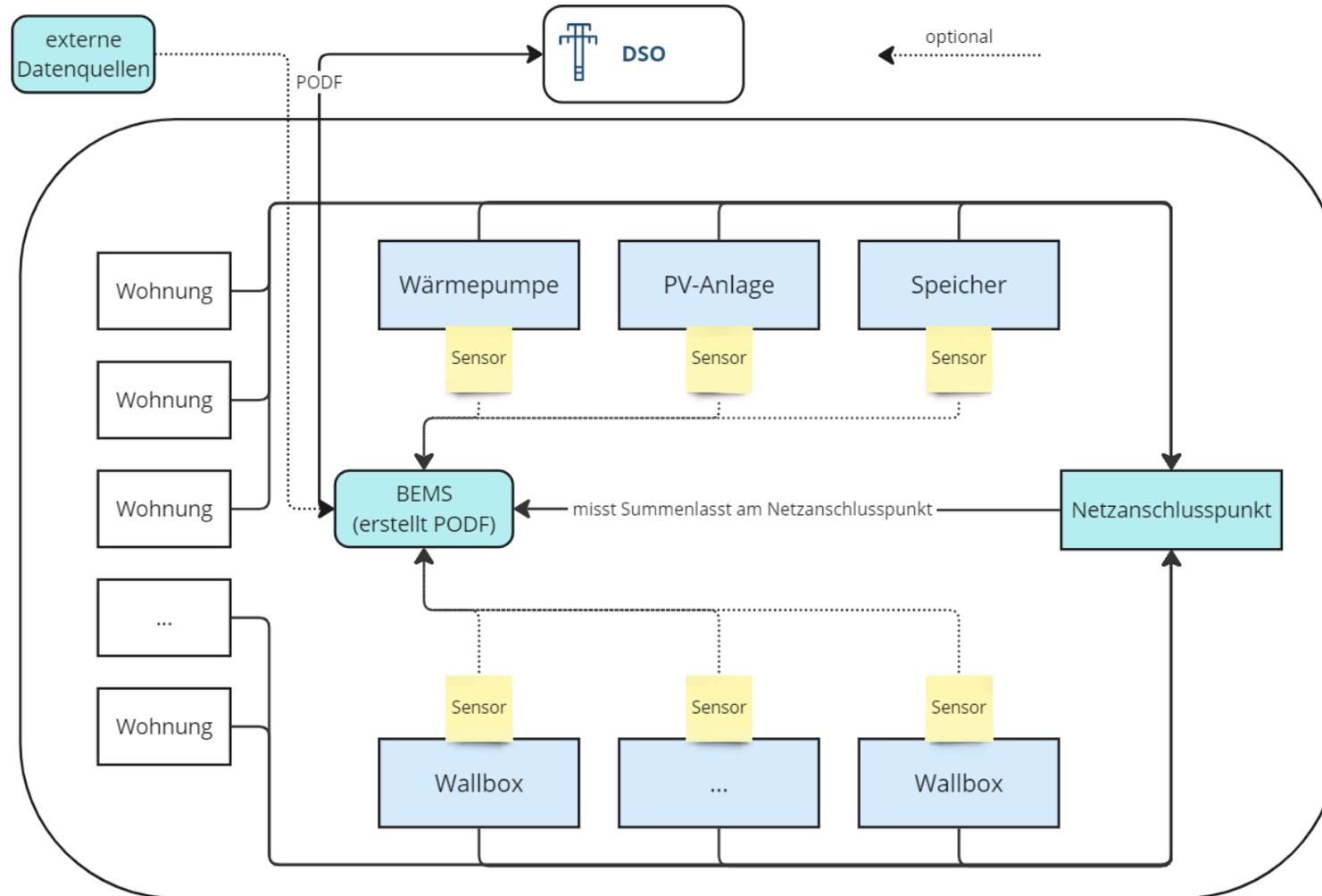




# Systembeschreibung

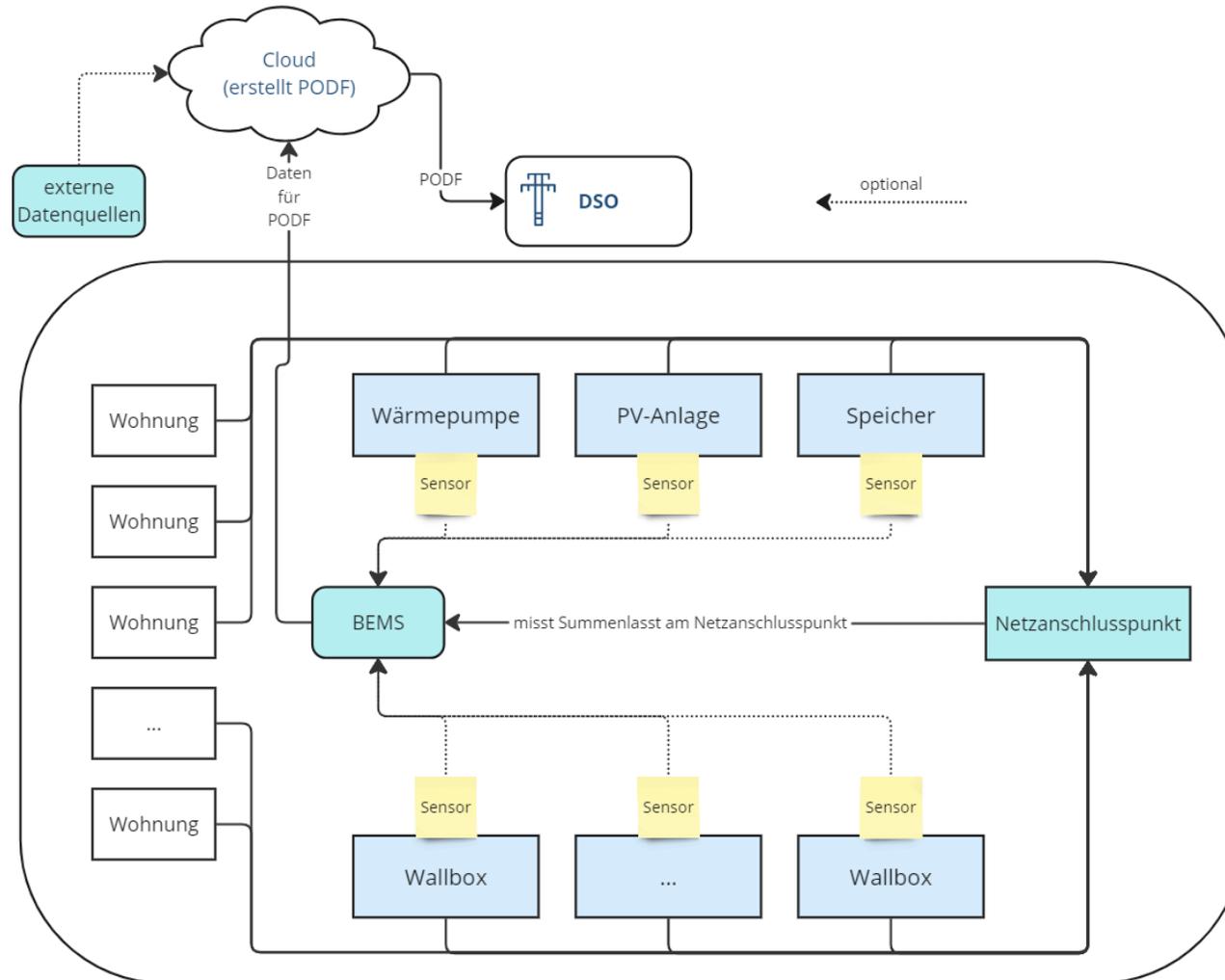
# Systembeschreibung

## PODF wird auf BEMS erstellt



# Systembeschreibung

## PODF wird in der Cloud erstellt





# Power Demand Forecasting

Use Case Beschreibung

# Power Demand Forecasting (PODF)

## Allgemeine Use Case Beschreibung

### Allgemeine Use Case Beschreibung

**Building Energy Management Systeme (BEMS) werden verwendet, um den Stromverbrauch des Gebäudes zu prognostizieren.** Das BEMS ermittelt über eine Differenzmessung am Netzanschlusspunkt den aktuellen Anteil am Netzbezug der nicht steuerbaren Verbraucher (Wohnungen, Anlagen, etc.). Diese zentrale Information kann über dezentrale Sensorik an diesen Systemen weiter ergänzt werden. In Kombination mit einem KI-Modell werden die Daten genutzt, um die Last des Gebäudes zu prognostizieren.

### Beteiligte Akteure

- Verteilnetzbetreiber (DSO)
- Gebäudeeigentümer

### Mehrwert

- DSO nutzt PODF zur Verbesserung der Netzführung
- Anreizmöglichkeiten für die Gebäudeeigentümer:
  - weniger Netzentgelte
  - vergünstigter Strompreis (wenn DSO = EVU)
  - Prämie

### Erforderliche Daten

- (historische) Summenlast des Gebäudes
- (historische) Wetterprognose (Temperatur, Sonneneinstrahlung)
- (optional) Last-/Erzeugungsgänge einzelner Geräte innerhalb des Gebäudes

# Relevante Datenquellen für den PODF

Die Daten werden in drei Kategorien zusammengefasst



1

**Last und Erzeugungsgänge** – Hierzu zählt die Summenlast des Gebäudes sowie optional weitere Messungen innerhalb des Gebäudes

2

**Stammdaten** – Informationen, die das Gebäude und dessen Ausstattung beschreiben

3

**Externe Daten** – nicht gebäudespezifische Daten

# Relevante Datenquellen für den PODF

## Last- und Erzeugungsgänge

### Summenlast des Gebäudes

- Messung am Netzanschlusspunkt und daher die wichtigste Größe für den DSO
- Hierzu zählt der gesamte Verbrauch / Erzeugung aller Wohneinheiten / Systeme innerhalb des Gebäudes
- Die Summenlast ist die Zielgröße des PODF – diese Daten sollen also prognostiziert werden
- Daher sind sie von zentraler Bedeutung (vor allem hinsichtlich des Trainings von KI-Modellen)

### (optional) weitere Messungen

- Mit entsprechender Sensorik können alle oder einige der einzelnen Bestandteile der Summenlast erfasst werden. Dazu zählen PV-Anlage, Speicher, Wallboxen, etc.
- BEMS können über den Einbau passender Sensoren Betriebsmessungen oder Zustandsinformationen für Bestandsanlagen erfassen
- Diese Messungen bieten gewisse Mehrinformation, welche ggf. zu einer besseren Qualität beim PODF führen kann
- Sie sind nicht zwingend notwendig

# Relevante Datenquellen für den PODF

## Stammdaten

### Wofür können Stammdaten genutzt werden?

- Diese Daten beschreiben das Gebäude und dessen Ausstattung
- Ähnlich wie die Last- und Erzeugungsgänge der einzelnen Systeme innerhalb des Gebäudes, ergänzen diese Informationen den PODF-Datensatz
- Sie bieten gewisse Mehrinformation, welche ggf. zu einer besseren Qualität beim PODF führen kann

### Welche Stammdaten können genutzt werden?

- Anzahl der Wohneinheiten
- PV-Anlage inkl. Spezifikation (Leistung, etc.)
- Stromspeicher inkl. Spezifikation
- Wärmepumpe inkl. Spezifikation
- Anzahl Wallboxen inkl. Spezifikation
- Anzahl der Elektrofahrzeuge im Besitz der Mieter
- Wallboxen öffentlich zugänglich?
- Klimaanlage
- ...

# Relevante Datenquellen für den PODF

## Externe Daten

### Wofür können externe Datenquellen genutzt werden?

- Als externe Daten werden alle Daten bezeichnet, die nicht gebäudespezifisch sind
- Vor allem, wenn wenig Informationen zum Gebäude vorhanden sind (z.B. nur Summenlast), können durch das Einbeziehen externer Daten Zusammenhänge durch ML / KI besser erkannt und gelernt werden

### Welche externen Daten können genutzt werden?

- Wetterprognosen
  - Außentemperatur (vor allem relevant bei elektrischer Heizung / Kühlung des Gebäudes)
  - Sonneneinstrahlung (vor allem relevant bei Nutzung einer PV-Anlage)
- Strompreisprognose (relevant bei Nutzung dynamischer Tarife z.B. an den Wallboxen oder in den Wohneinheiten)

# Power Demand Forecasting (PODF)

## 4 Fragestellung im Rahmen der Studie

**1**

Wo kann der PODF berechnet werden, Lokal (BEMS) oder auf der Cloud?

Eignen sich KI-Ansätze für das PODF?

**2****3**

Für welchen zeitl. Horizont wird der PODF gemacht?

Welche Daten sind notwendig, um ein gutes Ergebnis zu erzielen?

**4**

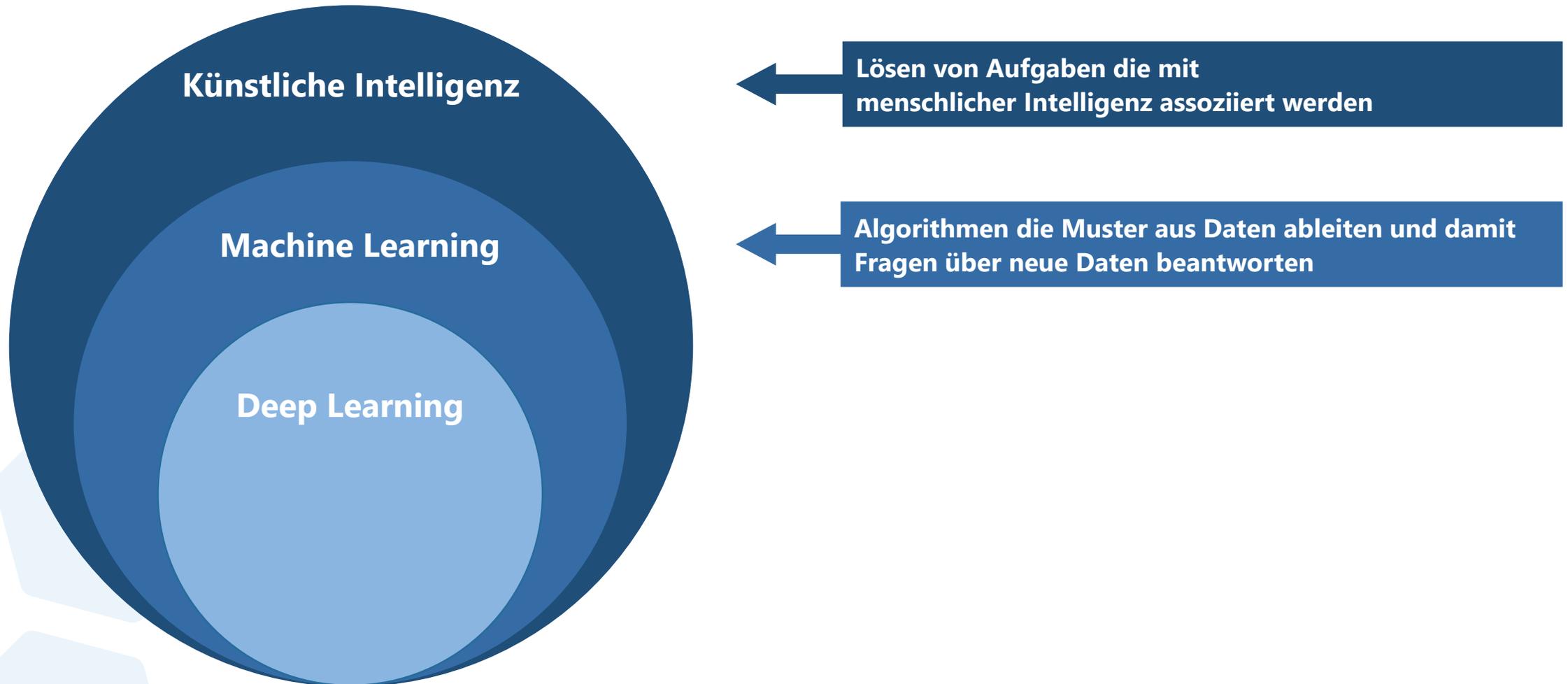


# KI-Prognosen

Was ist KI-basierte Zeitreihenprognose?

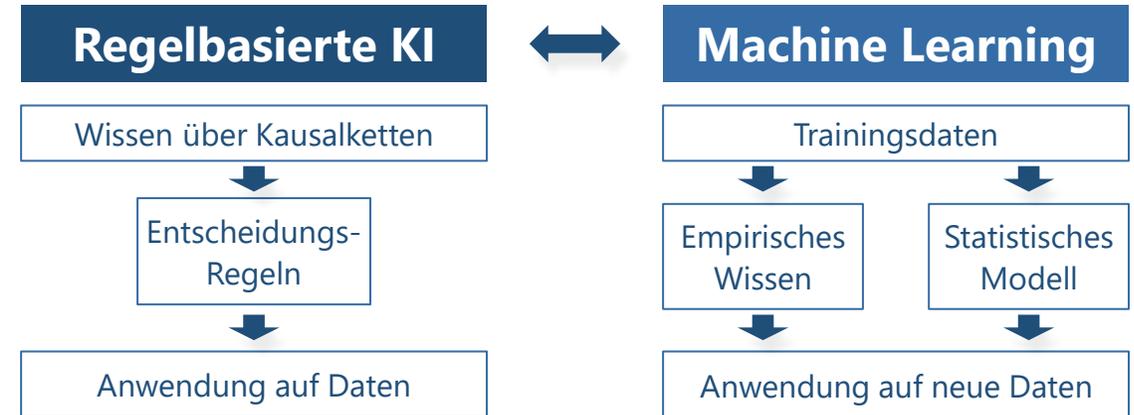
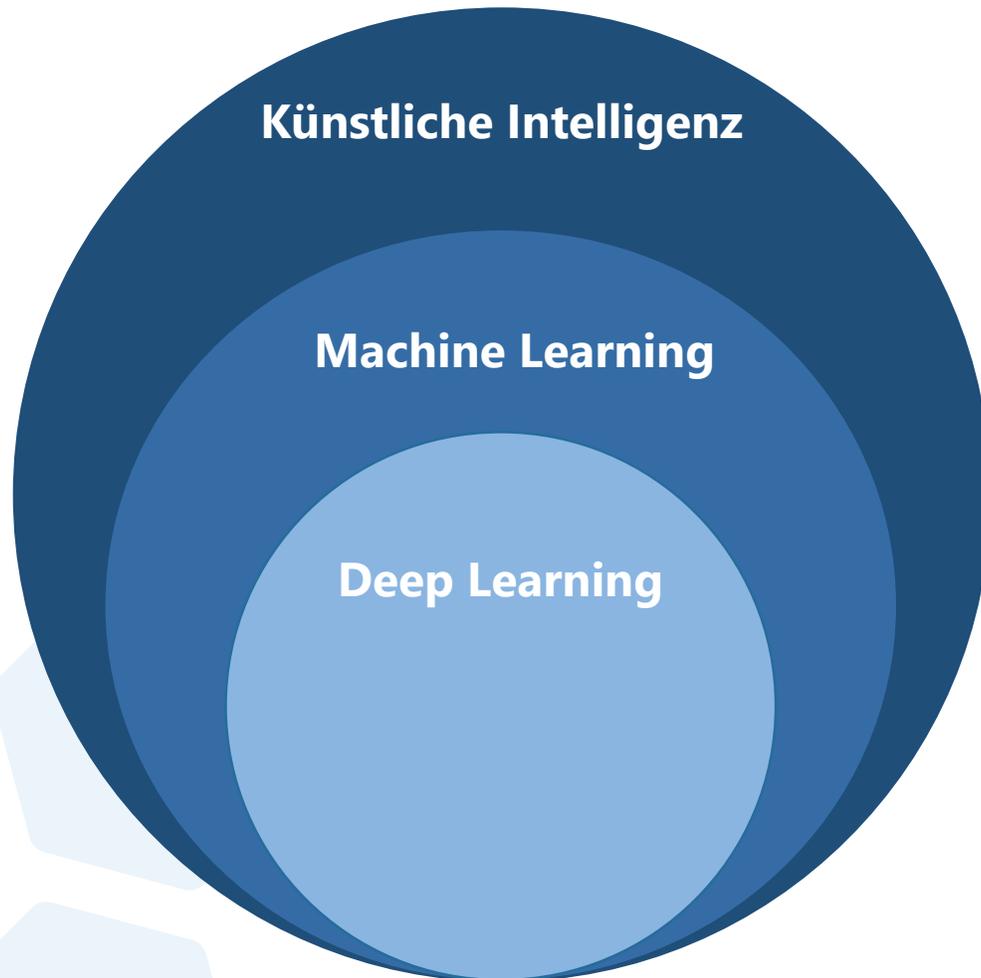
# KI, Machine Learning, Deep Learning

## Was ist was?



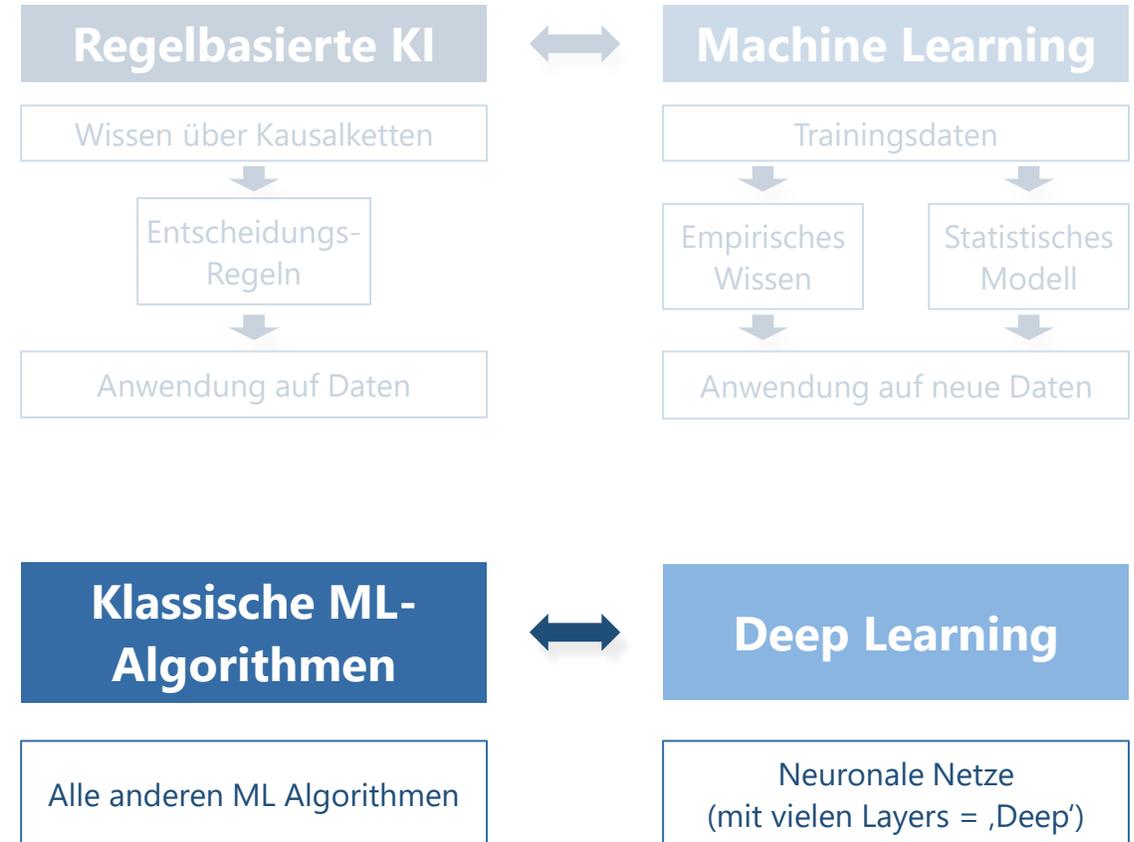
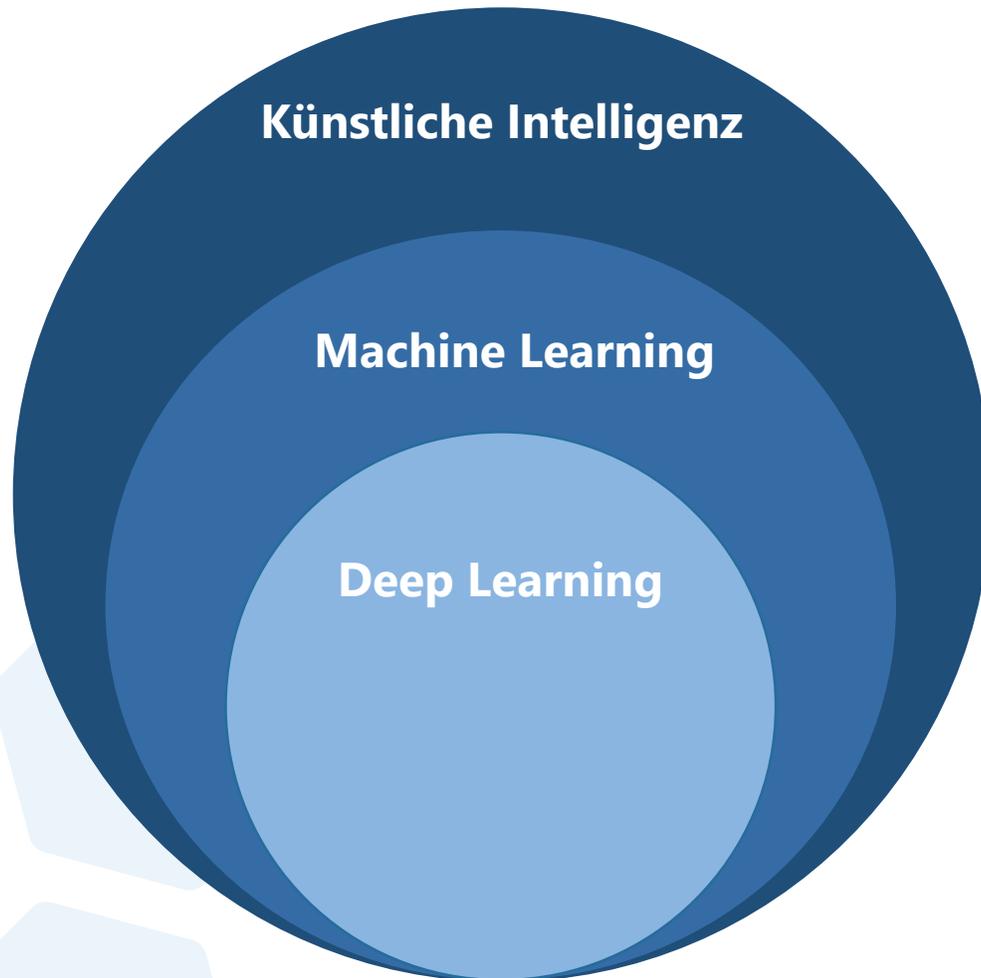
# KI, Machine Learning, Deep Learning

Was ist was?



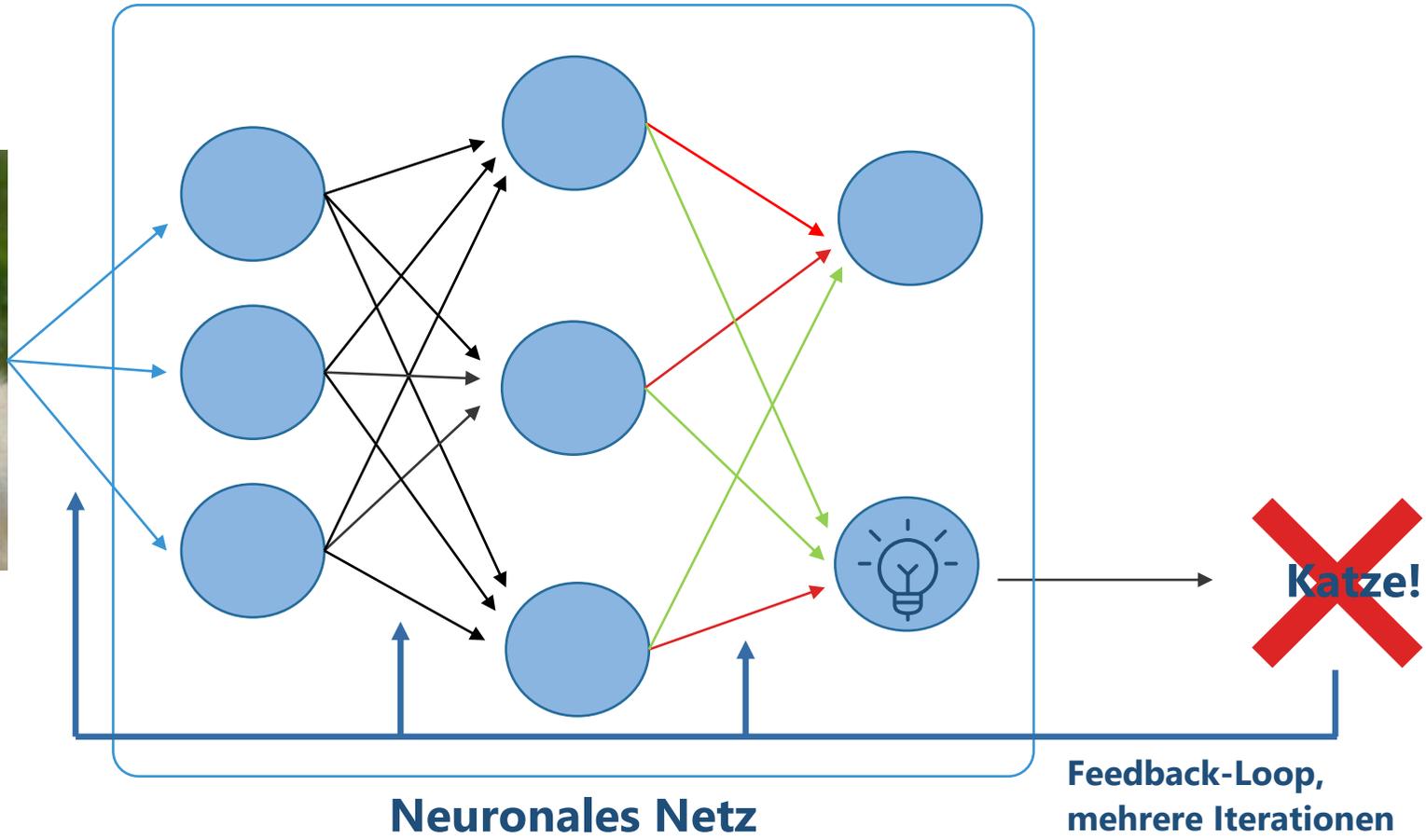
# KI, Machine Learning, Deep Learning

Was ist was?



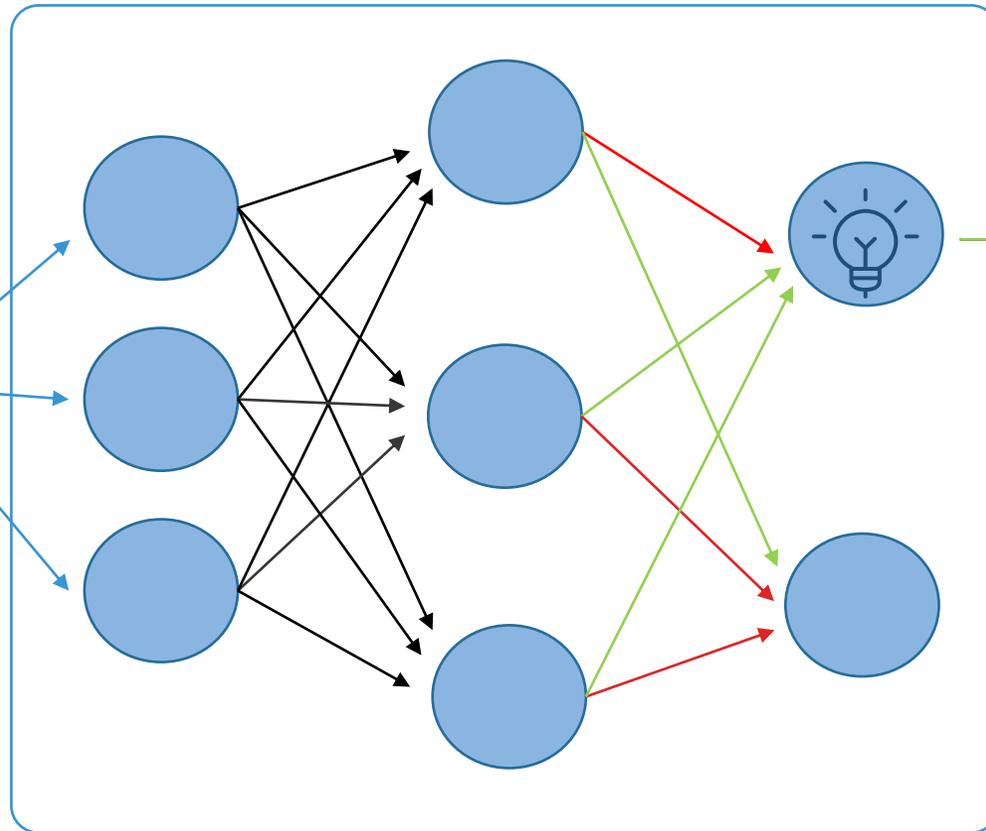
# Exkurs

## Künstliches neuronales Netz



# Exkurs

## Künstliches neuronales Netz

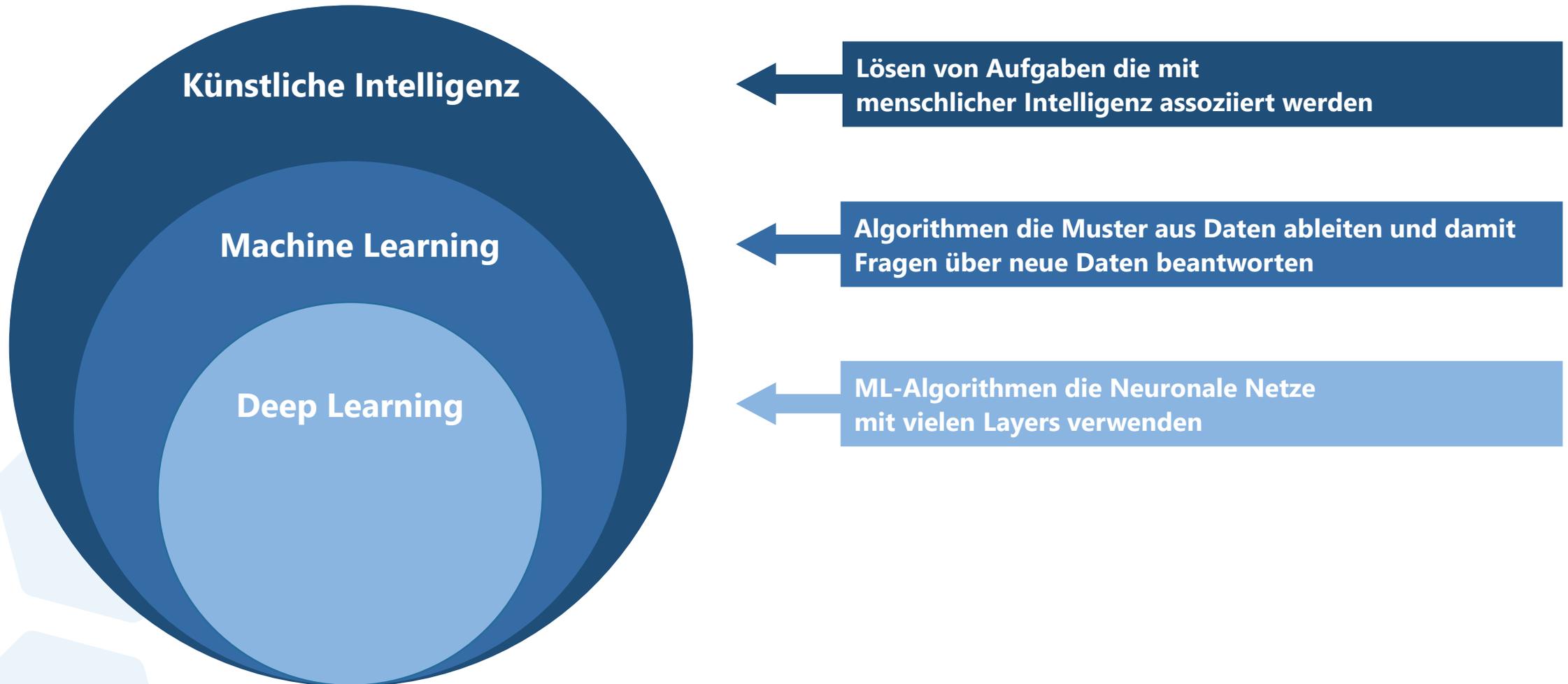


**Hund!**

Neuronales Netz

# KI, Machine Learning, Deep Learning

## Was ist was?



# Time Series Forecasting

## Definitionen und zentrale Begriffe

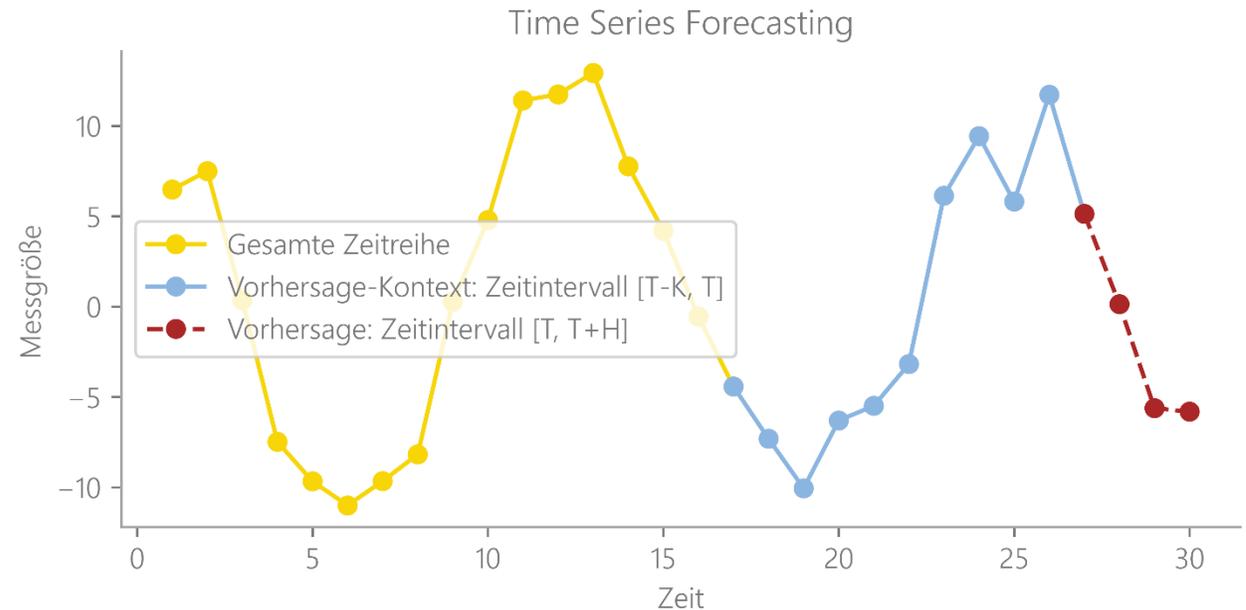
### Zeitreihe

Messgröße (einzelne oder mehrere Variablen), die in gleicher Form zu mehreren Zeitpunkten erfasst wird.

### Time Series Forecasting

Auf Basis der letzten Datenpunkte im Zeitraum  $[T-K, T]$  sollen die Zeitreihenwerte im Zeitintervall  $[T, T+H]$  vorhergesagt werden.

Der **Prognosehorizont H** und die **Kontextlänge K** sind hierbei wichtige Parameter von Vorhersagemodellen.



Die **zeitliche Auflösung der Zeitreihe** ist definiert durch die Dauer zwischen zwei Messungen (bei Zeitreihen mit regelmäßigen Abständen).

# Time Series Forecasting

## Machine Learning Ansatz

### Training eines Machine Learning (ML) Modells

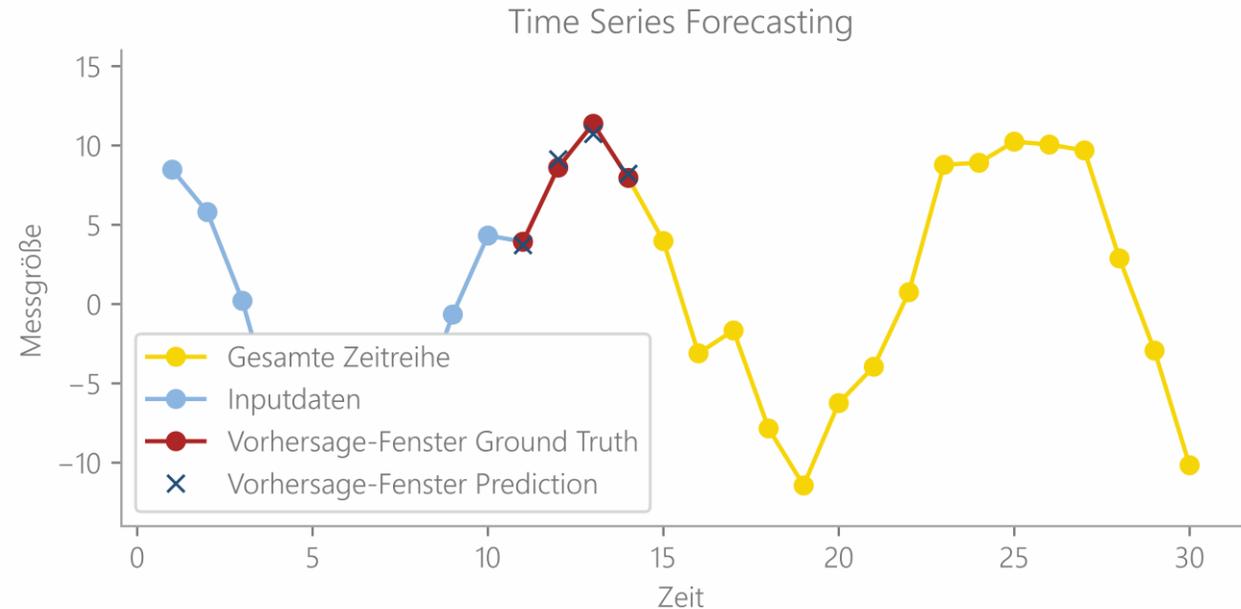
ML-Modelle werden trainiert, indem sie eine große und repräsentative Menge an Beispielen gezeigt bekommen für die die tatsächliche Vorhersagegröße (= ‚**Ground Truth**‘) bekannt ist.

Das Modell wird durch Vergleich der vorhergesagten und der tatsächlichen Werte schrittweise angepasst und verbessert.

### Evaluierung eines ML Modells

Die Abweichung von vorhergesagten und tatsächlichen Werten wird auf einem Datensatz evaluiert, der nicht im Training verwendet wurde.

Typische Metriken für TS Forecasting: Mean Squared Error (**MSE**), Root Mean Squared Error (**RMSE**), Mean Absolute Error (**MAE**), Mean Absolute Percentage Error (**MAPE**).



# Time Series Forecasting

## Machine Learning Ansatz

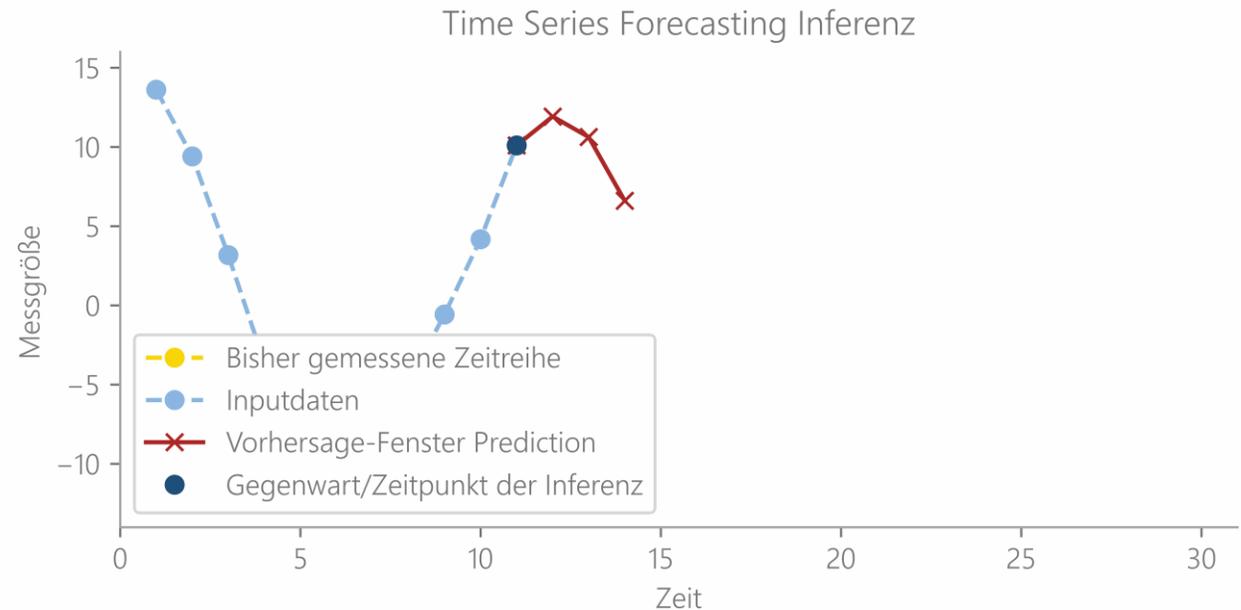
### Inferenz eines ML Modells

Das fertig trainierte ML Modell wird zum Einsatz gebracht, indem die zum Vorhersagezeitpunkt verfügbare Daten als Input verwendet werden.

Das Modell sagt die in der Zukunft liegenden Datenpunkte auf Basis seiner ‚Erfahrungen‘ aus den historischen Daten voraus.

### Weitere Inputs

Im Allgemeinen treffen ML-Modelle Vorhersagen nicht nur auf Grundlage der vergangenen Datenpunkte der Vorhersage-Zeitreihe, sondern verwenden noch weitere Input-Features (z.B. wären für Wetterprognosen der Lufttemperatur in den nächsten 3 Tagen nicht nur die Temperatur der letzten 3 Tage, sondern auch bspw. die Windrichtung und -stärke relevant)



# “Klassische” Machine Learning Modelle

## Wichtigste Ansätze

### **ARIMA / SARIMA / Exponential Smoothing (ES)**

- Relativ simple statistische Modelle
- Können bereits mit geringen Datenmengen und sehr geringem Rechenaufwand trainiert werden

### **Gradient Boosting (z.B. LightGBM/XGBoost) / Random Forests (RF) / Support Vector Machines (SVM)**

- Weit verbreitet in vielen ML Einsatzbereichen
- Zeitreihen-spezifische Herausforderungen müssen bei Training nur wenig berücksichtigt werden, daher schnelles Training und gute Interpretierbarkeit
- Langfristige Trends und Abhängigkeiten in Datenreihen weniger gut abbildbar

# Deep Learning Modelle

## Wichtigste Ansätze

### Recurrent Neural Networks (RNNs):

- Erster Durchbruch für Time Series Forecasting:  
v.a. **LSTMs** (Long Short-Term Memory, Hochreiter et al. 1997), bis heute verwendet
- Relativ **effizientes Training** bzgl. benötigter Datenmengen und Rechenleistung

### Transformer Modelle:

- Neuere Modell-Generation (Vaswani et al. 2017), verwendet in verschiedensten Disziplinen (z.B. auch Chatbots, Chat**GPT** = **G**enerative **P**retrained **T**ransformer)
- Bei **hoher Datenverfügbarkeit und** heutiger GPU-**Rechenleistung** für viele Use Cases die überlegene Modellklasse
- Adaptionen speziell für Zeitreihen: z.B. **Temporal Fusion Transformer (TFT)**



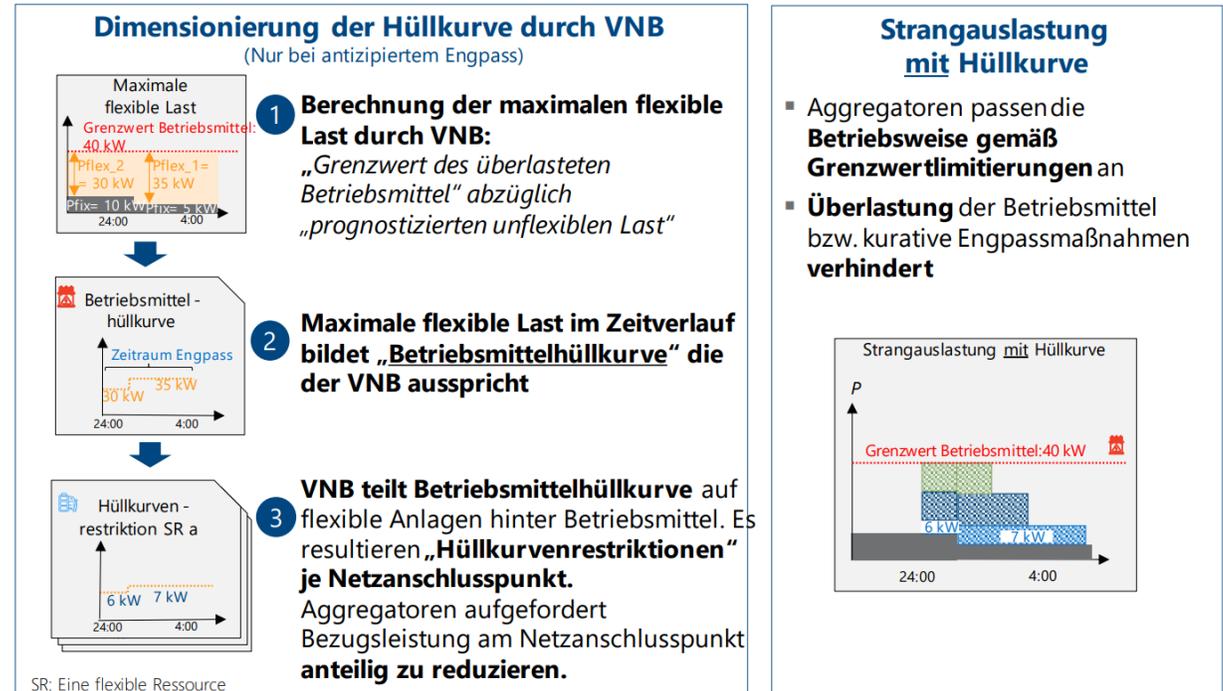
# KI-Prognosen im Kontext PODF

Aktuelle Landschaft

# Überblick

## Hüllkurvenkonzept als Anwendungsfall

- Die Prognose für das PODF kann verschiedene Formen annehmen. Sie hängt stark von den verfügbaren Daten, dem zu prognostizierenden Zeithorizont und der für die Einrichtung verwendeten Hardware ab.
- Um Prognoseansätze herauszufiltern und zu identifizieren, die im Rahmen unseres Projekts liegen, werden wir das Hüllkurvenkonzept als unseren Endanwendungsfall heranziehen.



Das Hüllkurvenkonzept [1]

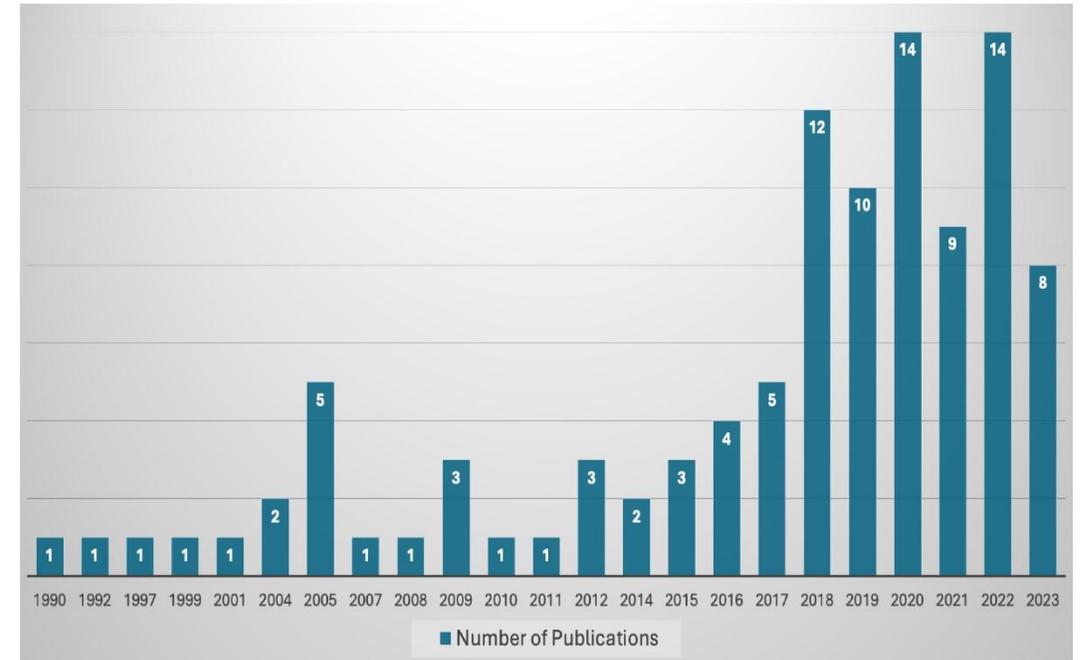
# Überblick

## Verschiedene Arten von Prognosen

**Neue Prognosemethoden werden zunehmend veröffentlicht und basieren nicht ausschließlich auf KI.**

**In der Literatur lassen sich drei Kategorien unterscheiden:**

- **Physikbasiert:** Schätzt den Energieverbrauch anhand thermodynamischer Regeln. (White Box)
- **Datengetrieben (KI-basiert):** Beruht auf Zeitreihenstatistiken und maschinellen Lernmodellen. (Black Box)
- **Hybride Ansätze:** Kombiniert physikalische und datengetriebene Methoden. (Grey Box)



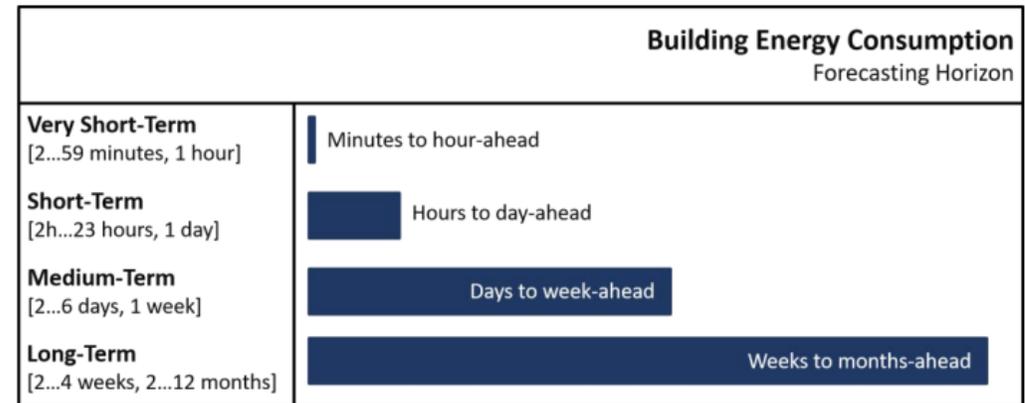
Wachstum der Energy Forecast-bezogenen Forschungspublikationen von 1990 bis 2023 [1]

# Überblick

## Prognosehorizont

In der Literatur unterscheiden wir hauptsächlich vier Prognosekategorien:

- Very short-term load forecasting (VSTLF)
- Short-term load forecasting (STLF)
- Medium-term load forecasting (MTLF)
- Long-term load forecasting (LTLF)



Prognosehorizont des Energieverbrauchs von Gebäuden. [1]

Der Prognosehorizont ist ein wichtiger Aspekt bei der Identifikation geeigneten Prognoseansätze für den Anwendungsfall. Im Kontext des Hüllkurvenkonzepts befinden wir uns im STLF/VSTLF-Bereich.

[1] A REVIEW OF ENERGY CONSUMPTION FORECASTING IN SMART BUILDINGS: METHODS, INPUT VARIABLES, FORECASTING HORIZON AND METRICS

# Implementierung der KI-Prognose

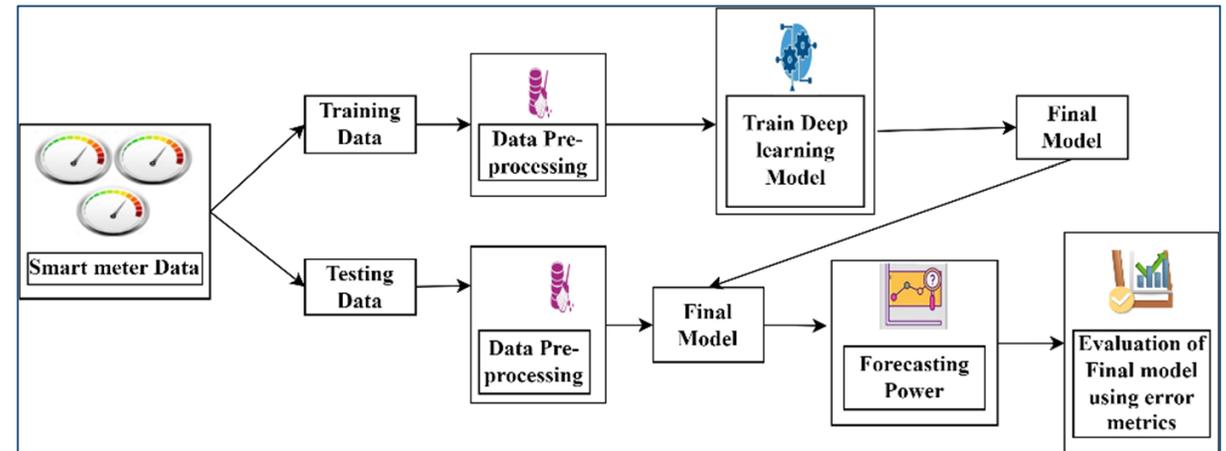
## Die Trainingspipeline

Die Besonderheit von KI-basierten Ansätzen ist, dass zunächst ein Trainingsprozess implementiert werden muss, der Folgendes erfordert:

**Trainingsdaten:** Alle relevanten Eingabedaten mit den zugehörigen Ausgaben, aus denen das Modell Muster lernen soll.

**Testdaten:** Gleiche Struktur wie die Trainingsdaten, aber zur Bewertung der Modellleistung verwendet.

**Fehlermetriken:** Formel zur Schätzung der Modellfehler.



Trainingspipeline-Umsetzung: Beispiele im Kontext von Energieprognosen. [1]

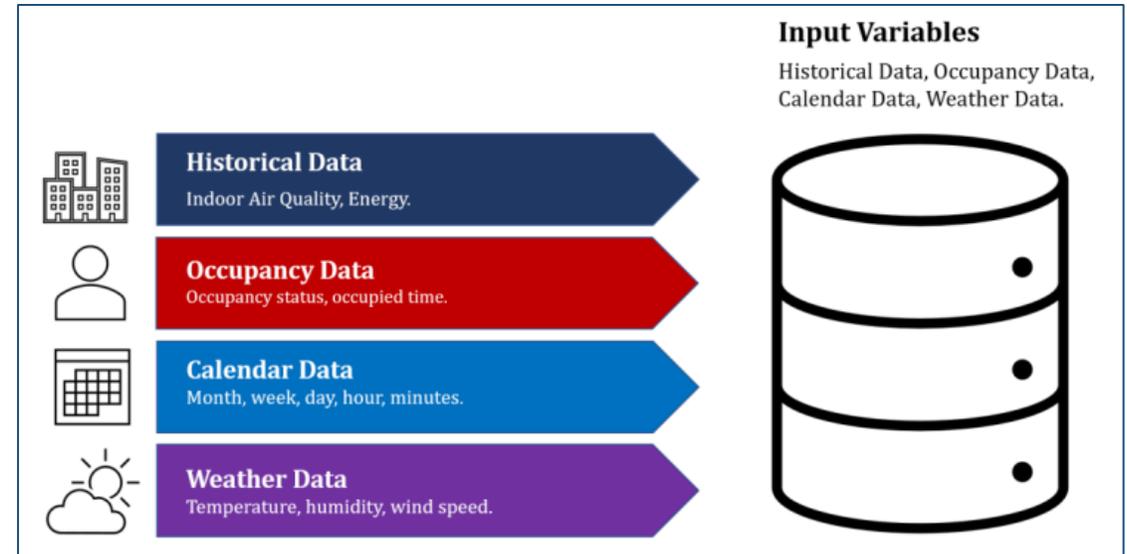
**Wie viele Daten werden benötigt?** Dies hängt stark von mehreren Faktoren ab, wie der Datenkomplexität, dem Rauschpegel und der Saisonalität.

# Implementierung der KI-Prognose

## Input Data

In KI-basierten Prognosen für PODF-Zwecke finden wir oft ähnliche Eingabedaten:

- **Historische Daten:** Energiedaten, alle spezifischen Sensordaten (PV, WP, Speicher usw.)
- **Wetterdaten**
- **Belegungsdaten:** Bezieht sich auf menschliches Verhalten und Belegungszeiten. (Wird oft für „non-residential“ Gebäude verwendet)



Typische Eingabedaten im Kontext von Energieprognosen. [1]

**In der Literatur konzentrieren sich die meisten umgesetzten Ansätze auf „non-residential“-Gebäude, daher sind viele der hier präsentierten Informationen möglicherweise nicht vollständig auf den Kontext von Mehrfamilienhäusern anwendbar**

[1] A REVIEW OF ENERGY CONSUMPTION FORECASTING IN SMART BUILDINGS: METHODS, INPUT VARIABLES, FORECASTING HORIZON AND METRICS

# Implementierung der KI-Prognose

## Metriken

Welche Metriken verwendet werden, ist entscheidend für die korrekte Bewertung von Prognoseansätzen und die Benchmark-Leistung. Im Bereich PODF sind die folgenden Metriken am häufigsten:

- **Mean Absolute Error (MAE):** Verwendet das gleiche Gewicht für alle Fehler.
- **Root Mean Squared Error (RMSE):** Verwendet ein höheres Gewicht für größere Fehler.
- **Mean Absolute Percentage Error (MAPE):** Stellt den Fehler in Prozent des jeweiligen tatsächlichen Variablenwerts dar.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - f_i|$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - f_i)^2}$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - f_i|}{|y_i|}$$

# Überblick KI-Prognoseansätze

## Landschaft der Nutzung von KI-Ansätzen

- Im Bereich der KI-Prognosen gibt es viele verschiedene Modelle und Implementierungen. **Es ist schwer zu sagen, welches für einen bestimmten Anwendungsfall am besten geeignet ist, daher bewerten wir normalerweise mehrere Ansätze in einem Benchmark.** Die Tabelle rechts zeigt bestehende Ansätze, die für Aufgaben der Energiebedarfsprognose in Gebäuden verwendet werden.
- Innerhalb derselben Kategorie (Deep Learning/Machine Learning) sind die Kosten und die Rechenanforderungen im Allgemeinen ähnlich.**

**Table 1** Taxonomy of related literature findings in building-related ELF

#	Model	Input resolution	Exogenous Inputs	Forecast horizon	Indicative metrics
1	BPN <sup>1</sup>	15 min	No	2 h	MAPE: 6.21%
2	SVR	15 min	No	2 h	MAPE: 5.2%
3	MLP	1 h	Yes	24 h	MAPE: 1.7, R <sup>2</sup> : 0.98
4	MLP	15 min	No	1 h	MAPE: 12.96%, RMSE(kWh):0.651
5	LSTM	15 min	No	1 h	MAPE: 14.09%, RMSE(kWh):0.668
6	XGBoost	15 min	No	1 h	MAPE: 15.37%, RMSE(kWh):0.699
7	LGBM	30 min	Yes	1 h	MAPE: 21.5%
8	LSTM	30 min	No	1 h	MAPE: 9.1%
9	XGBoost	1 h	Yes	1 h	MAPE: 12%
10	XGBoost	1 h	No	1 h	MAPE: 3.50%
11	EBT <sup>2</sup>	15 min	Yes	1 h	MAPE: 4.62%
12	XGBoost	1 h	Yes	1 h	R <sup>2</sup> : 0.977
13	RF	1 h	Yes	1 h	R <sup>2</sup> : 0.972
14	MLP	1 h	Yes	1 h	R <sup>2</sup> : 0.98
15	LSTM	1 h	Yes	1 h	R <sup>2</sup> : 0.977
16	FF-DNN <sup>3</sup>	1 h	Yes	1 h	MAPE: 1.42%
17	GBDT <sup>4</sup>	1 h	No	24 h	MAPE: 1.32%
18	RF	1 h	No	24 h	MAPE: 1.97%
19	MLP	15 min	No	24 h	MAPE: 14.53%
20	ELM <sup>5</sup>	15 min	No	24 h	MAPE: 14.54%
21	Model Ensemble	15 min	Yes	24 h	MAPE:2.32%
22	ANN	1 h	Yes	1 h	MAPE: 5%

# Überblick KI-Prognoseansätze

## Machine Learning-based vs Deep Learning-based

Methods		Long-term load forecasting	Short-term load forecasting
DL-based	Advantages	Good features extraction ability	1. High robustness against input uncertainty  2. Flexible and support ensemble methods, for example, popular hybrid models using DL to capture features and ML to improve predictive accuracy
	Limitations	1. Learning results may not be particularly relevant or helpful;  2. Requires more training data	Requires more training data
ML-based	Advantages	High robustness against overfitting, especially XGBoost and ANN	Flexible and support ensemble methods, for example, popular hybrid models using DL to capture features and ML to improve predictive accuracy
	Limitations	Additional feature selection processes are required to ensure high accuracy	
Statisticsbased	Advantages	Simple, small training dataset, computationally efficient	
	Limitations	Generally low predictive accuracy	

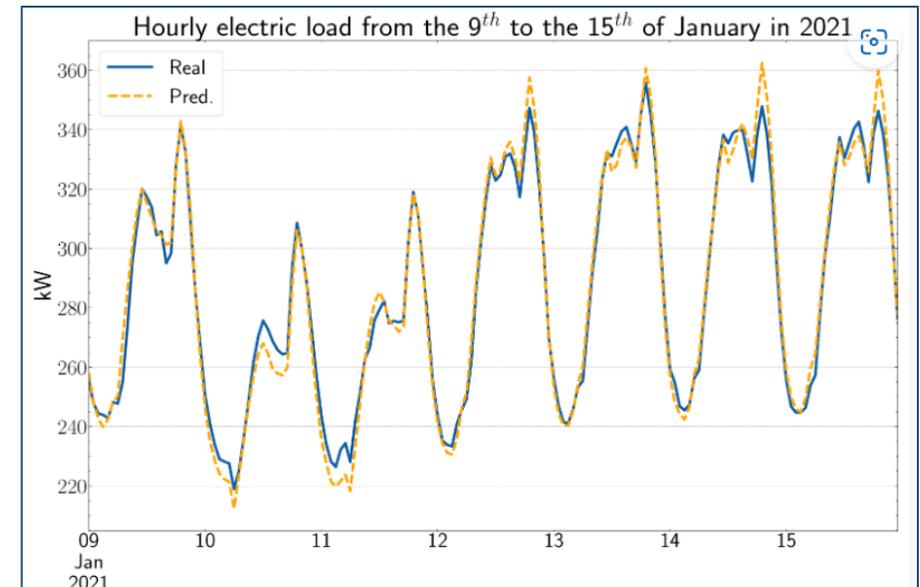
**Ein weiterer relevanter Aspekt ist, dass ML-Ansätze in der Regel deutlich recheneffizienter sind als DL-Ansätze. Dies kann ein Vorteil sein, falls in Betracht gezogen wird, das PODF auf dem BEMS auszuführen**

# Beispiel KI-Prognoseansätze

## Deep Learning: LSTM

### LSTM (Long short-term memory):

- Ist eine Art von Recurrent Neural Network, das darauf ausgelegt sind, sich an relevante vergangene Muster gut zu „erinnern“ und unnötige Informationen zu vergessen.
- Einer der am häufigsten in der Literatur umgesetzten Ansätze für Energieverbrauchsprognosen in intelligenten Gebäuden, insbesondere für VSTLF-Aufgaben, aber auch relevant für alle Prognosehorizonte.
- Wird durch Transformer-basierte Ansätze herausgefordert, aber weiterhin in laufender Forschung verwendet



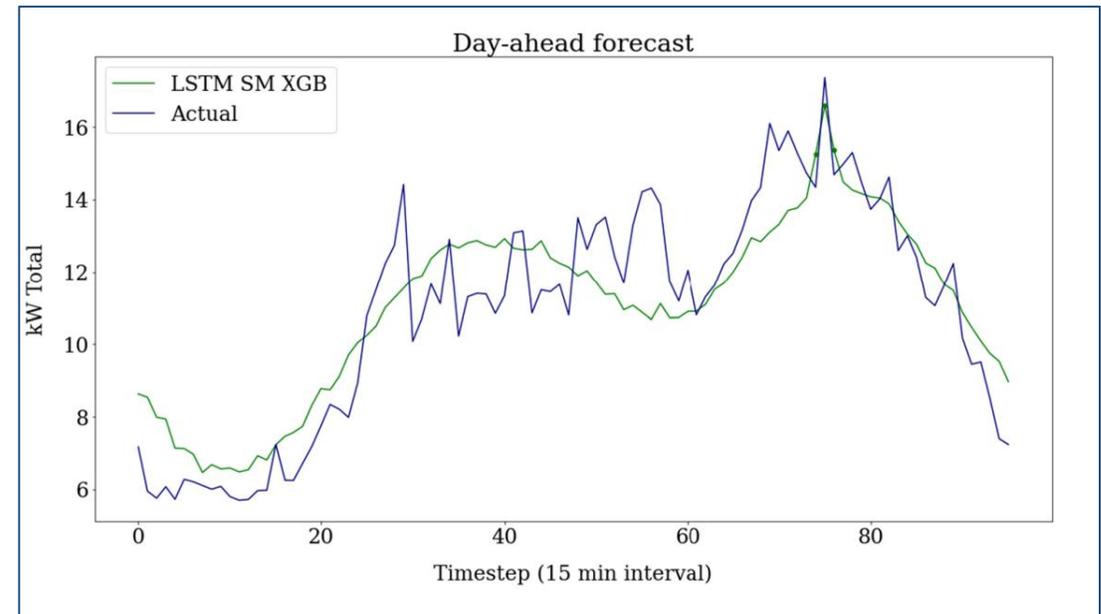
LSTM-basierte Stromlastprognose. [1]

# Beispiel KI-Prognoseansätze

## Machine Learning: XGBoost

### XGBoost (eXtreme Gradient Boosting):

- Ist eine hoch effiziente und skalierbare Implementierung des Gradient Boosting.
- Sehr gängiger Ansatz in der Literatur für Energieverbrauchsprognosen in intelligenten Gebäuden und für Prognoseaufgaben im Allgemeinen.
- Wird tendenziell von Deep-Learning-Ansätzen übertroffen, hat aber den Vorteil, mit geringen Ressourcen zu laufen, und wird oft als Baseline verwendet, um neue Ansätze zu benchmarken.



Hier werden XGBoost und LSTM für die Day-Ahead-Prognose kombiniert. [1]



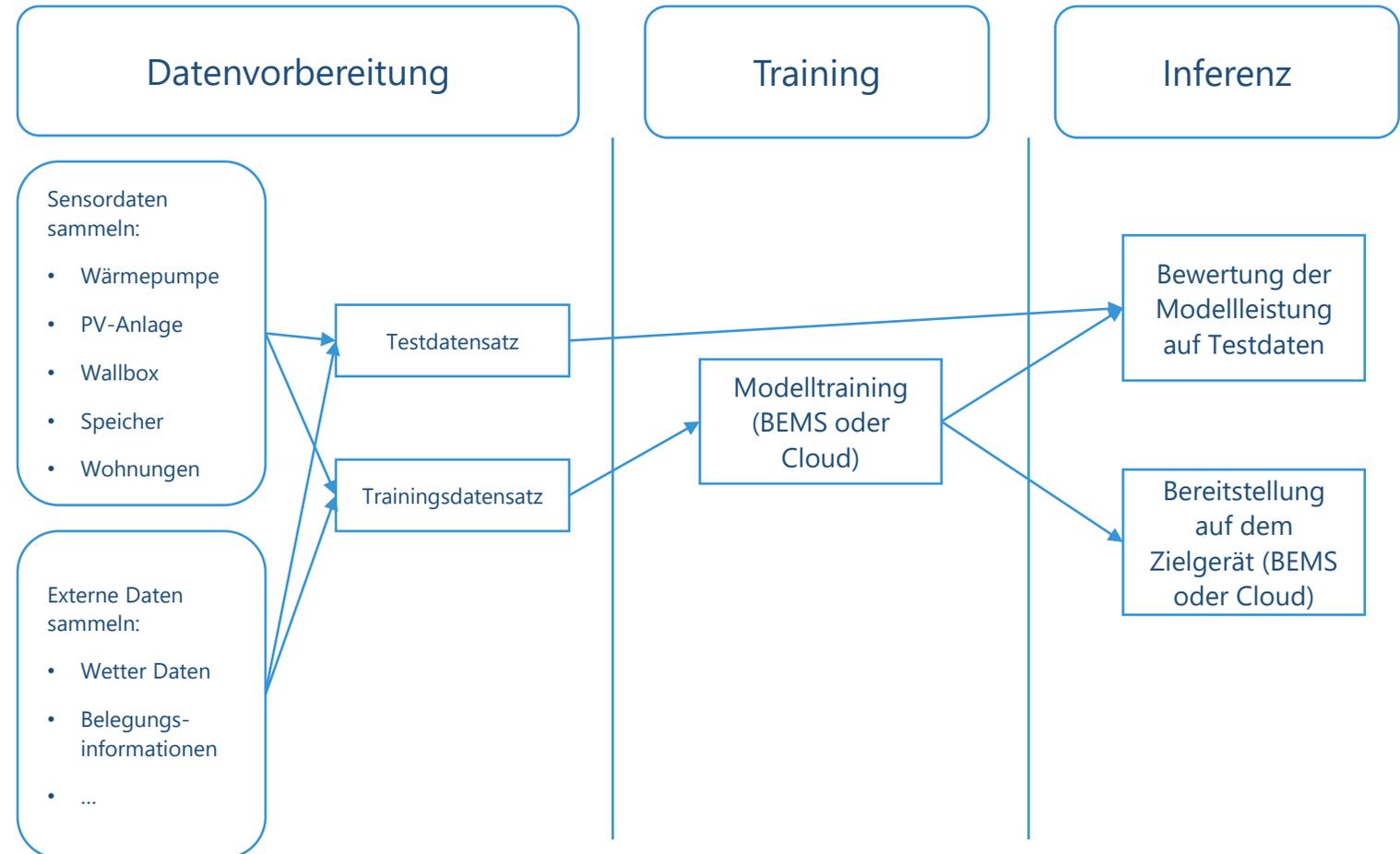
# KI-Prognosen im Kontext PODF

Qualitative Leistungsschätzung

# Implementierung eines PODF-Ansatzes

## Die Trainingspipeline im Kontext PODF

- Um die potenzielle Leistung von KI-basierten Prognosen zu bewerten, müssen wir zunächst die Hardware-Konfiguration (HEMS oder Cloud) nicht berücksichtigen. Dies wird im nächsten Teil untersucht.
- Daher konzentrieren wir uns hier auf die Abstraktion auf der Ebene der Trainingspipeline.



# Implementierung eines PODF-Ansatzes

## Wie kann die mögliche Leistung des trainierten Modells bewertet werden?

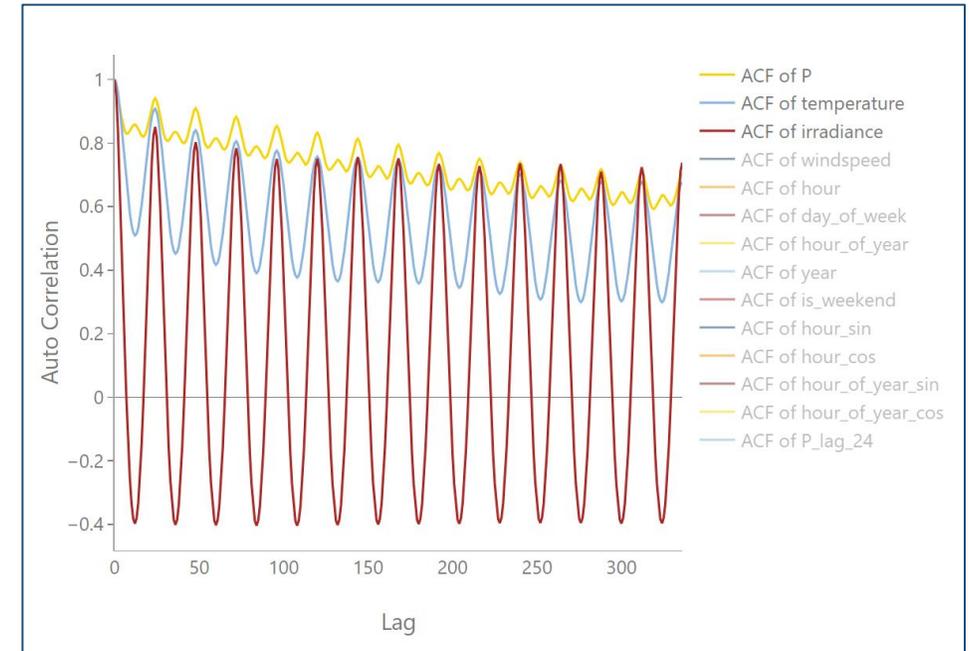
Die Leistung des Modells hängt direkt von der Komplexität der Daten ab, die wir vorhersagen wollen. Dies wird als „Data Forecastability“ bezeichnet.

Welche Informationen können helfen, diese Forecastability zu schätzen?

- Saisonalität der Daten (zeitbezogene Muster).
- Direkte Korrelation mit extern verfügbaren Daten (z.B. PV-Erzeugung und Wetter)
- Allgemeine regelmäßige Muster / Rauschpegel
- Plötzliche Peaks

### Nächste Schritte:

- Für jede Datenquelle: Erscheinen die Daten „forecastable“? Wie schneiden KI-Modelle bei ähnlichen Aufgaben in der Literatur ab?
- Im Falle von schlechter Forecastability: Welche Lösungen können ins PODF integriert werden?

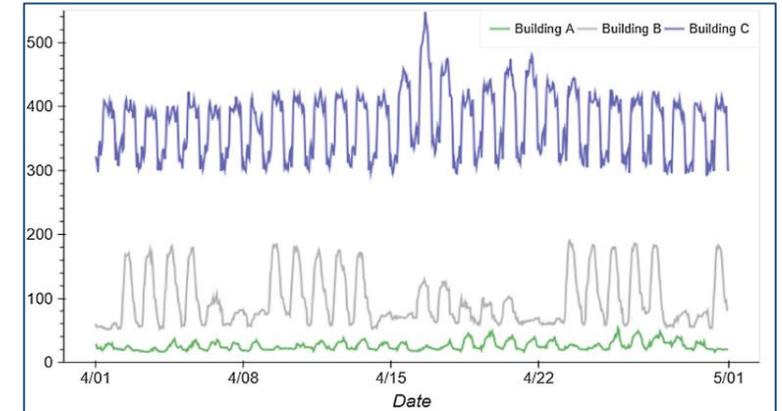


Autokorrelationsdiagramme werden häufig verwendet, um die „Forecastability“ einer Zeitreihe zu bewerten. [1]

# Bewertung „Forecastability“

## Aufgabe: STLF / VSTLF für Wohnhaushalte

<b>Verfügbare / Erforderliche Datenauflösung</b>	<p>Von Smart Metern erhalten wir 15-Minuten-Intervalle. 15 Minuten sind nicht ideal, da sie Energiespitzen verbergen können. Eine Auflösung von etwa 1 Minute oder Sekunden sollte verwendet werden, damit das Modell realistische Informationen lernen kann.</p>
<b>Forecastability</b>	<p>Rauschpegel im Zusammenhang mit menschlichem Verhalten sind natürlicherweise schwer vorherzusagen. Es gibt jedoch eine gute Saisonalität auf Tages-/Wochen-/Jahresbasis, die die Daten für Prognoseansätze geeignet macht</p>
<b>Geschätzte KI-Prognoseleistung</b>	<p>Für STLF zeigt die Literatur eine gute Leistung mit LSTM/XGBoost-Ansätzen, die Standardlastprofile übertreffen können. Für VSTLF hängt es von der gegebenen Datenauflösung ab (besser als 15 Minuten ist ein Muss).</p>
<b>Zusätzlich verwendete Daten</b>	<ul style="list-style-type: none"> <li>• Wetterdaten</li> <li>• Belegungsdaten (falls verfügbar)</li> </ul>



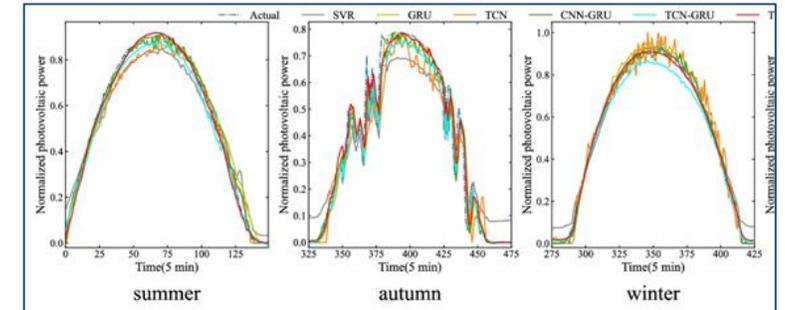
Lastprofile von verschiedenen Gebäuden. [1]

**Fazit: Gut geeignet für KI-basierte Ansätze**

# Bewertung „Forecastability“

## Aufgabe: STLF / VSTLF für PV-Anlage

<b>Verfügbare / Erforderliche Datenauflösung</b>	Ein modernes System sollte eine Auflösung von mindestens 1 Minute bieten, was ausreichend wäre.
<b>Forecastability</b>	Sehr gut: sichtbare Muster/Saisonalität und starke Korrelation zu Wetterdaten.
<b>Geschätzte KI-Prognoseleistung</b>	Sehr gut: Viele bestehende KI-Ansätze werden hier mit hohen Performance-Kennzahlen verwendet, insbesondere im Kontext von VSTLF und STLF.
<b>Zusätzlich verwendete Daten</b>	<ul style="list-style-type: none"> <li>• Wetterdaten</li> <li>• Beleuchtungsdaten</li> </ul>



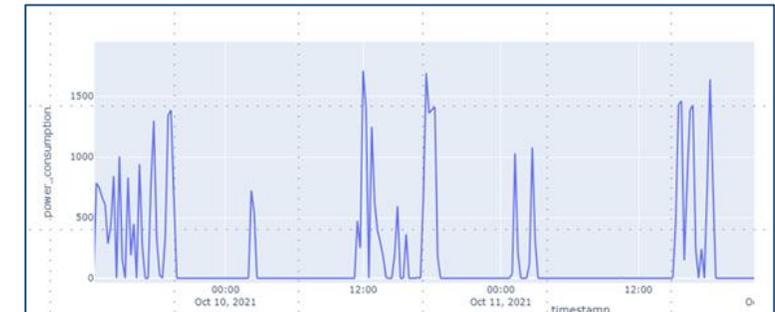
Erzeugungprofil eines PV-Systems. [1]

**Fazit: Sehr gut geeignet für KI-basierte Ansätze**

# Bewertung „Forecastability“

## Aufgabe: STLF / VSTLF für Wärmepumpe

<b>Verfügbare / Erforderliche Datenaufösung</b>	Wie bei PV-Systemen sind auch hier bei modernen Systemen Auflösungen von mindestens 1 Minute verfügbar.
<b>Forecastability</b>	<b>Komplex:</b> Aus unserer Erfahrung entstehen Probleme durch die Nutzung. Das Erhitzen von Brauchwasser erzeugt Spitzen, die schwer vorherzusagen sind, da sie direkt mit dem menschlichen Verhalten verbunden sind. Im Winter zeigt der Heizverbrauch charakteristische Muster und korreliert gut mit der Außentemperatur.
<b>Geschätzte KI-Prognoseleistung</b>	<b>Komplex:</b> In der Literatur liegt der Schwerpunkt meist auf der Lastprognose für mehrere Wärmepumpen. Aus unserer Erfahrung sehen wir Potenzial darin, sich auf die Luftheizung/-kühlung zu konzentrieren und die Brauchwassererwärmung auszuschließen. Dies hängt jedoch auch stark von den Einstellungen der Wärmepumpe ab
<b>Zusätzlich verwendete Daten</b>	<ul style="list-style-type: none"> <li>• Wetterdaten</li> </ul>



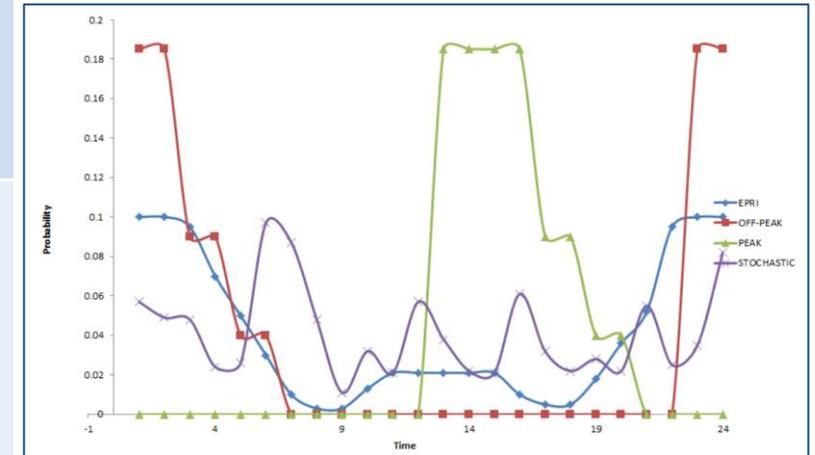
Lastprofil einer Wärmepumpe [1]

**Fazit: Nicht sehr gut geeignet für KI-basierte Ansätze**

# Bewertung „Forecastability“

## Aufgabe: STLF / VSTLF für Wallbox

<b>Verfügbare / Erforderliche Datenauflösung</b>	Moderne Wallbox-Systeme können mit einer Auflösung von mindestens 1 Minute kommunizieren.
<b>Forecastability</b>	<b>Sehr komplex:</b> Das Laden von Elektrofahrzeugen ist direkt mit dem menschlichen Verhalten verbunden und folgt bei Wohngebäuden keinen klaren saisonalen Mustern. In der Literatur beziehen sich die meisten Datensätze auf Nichtwohngebäude, wie Büros, wo das menschliche Verhalten viel vorhersehbarer ist.
<b>Geschätzte KI-Prognoseleistung</b>	<b>Sehr komplex:</b> In der Literatur wird die KI-basierte Vorhersage für ein einzelnes Fahrzeug kaum untersucht. Prognosen werden entweder für Gruppen von Wallboxen oder Ladestationen entwickelt, an denen mehrere Fahrzeuge laden können. Hier scheint es realistischer, direkte Wallbox-Informationen und Belegungsdaten zu verwenden.
<b>Zusätzlich verwendete Daten</b>	<ul style="list-style-type: none"> <li>Bei Nichtwohngebäuden: Belegungsdaten</li> </ul>



Verschiedene Arten von EV-Ladeverhalten. [1]

**Fazit: Nicht geeignet für KI-basierte Ansätze**

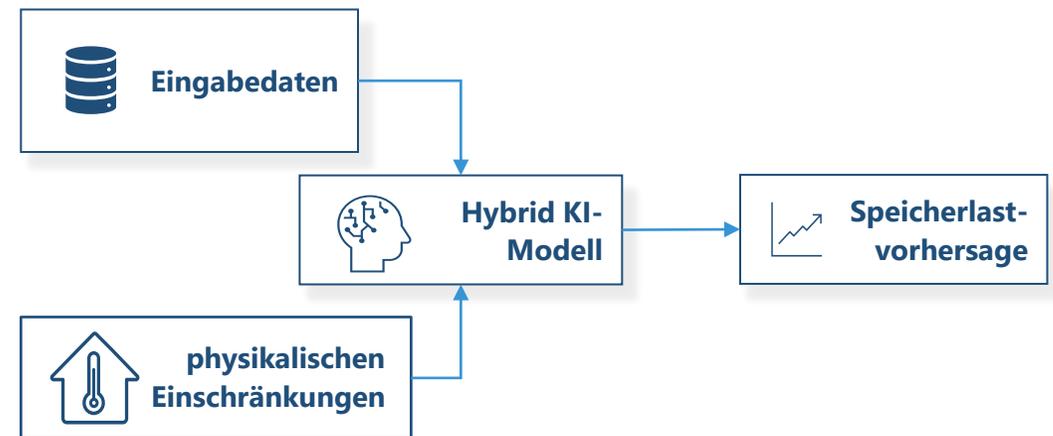
# Bewertung „Forecastability“

## Task: STLF / VSTLF für Speicher, Spezialfall

Die Vorhersage der Speicherlast ergibt wenig Sinn, da sie direkt vom BEMS und den Nutzerstrategien abhängt. Idealerweise sollten diese Informationen in das berechnete PODF integriert werden.

Falls diese Informationen nicht verfügbar sind (in der Literatur wird die Speicherlastvorhersage kaum untersucht), empfehlen wir zur Implementierung:

- **Inputdaten:** Prognosen von PV, WP, EV und Gebäudelast. Als externe Datenquellen können andere Faktoren wie Strompreiserwartungen einbezogen werden, die die Speicherladestrategie beeinflussen könnten.
- **Modell:** Hybride Modelle könnten aufgrund der klaren physikalischen Einschränkungen des Systems vorteilhaft sein.



**Fazit: Nicht sehr gut geeignet für KI-basierte Ansätze. Besser ist es, Zugang zur Speicherladestrategie zu erhalten und diese in das berechnete PODF zu integrieren**

# Integration schwer vorhersagbarer Daten

## Smoothing

Daten mit geringer Saisonalität und vielen Peaks sind schwer vorherzusagen. Zur Schätzung der Werte für die nächsten Stunden kann ein „Data-Smoothing“ hilfreich sein. Dabei werden die Zeitreihenwerte über ein bestimmtes Fenster gemittelt, um klare Trends zu erkennen und das Rauschen zu reduzieren.

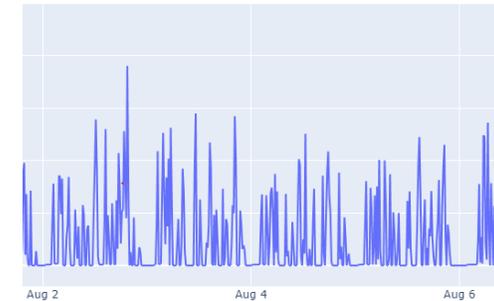
### Wärmepumpen:

Wie rechts gezeigt, profitieren Wärmepumpen von einem 2-Stunden-Glättungsfenster, das die Vorhersagbarkeit verbessert. Die Frage ist, ob die geglätteten Informationen im Kontext von PODF nützlich sind.

### EVs:

Bei EVs ist es komplizierter, da das Glättungsfenster wahrscheinlich 6-12 Stunden betragen müsste, um das Rauschen effektiv zu reduzieren, wodurch jedoch die meisten Informationen verloren gehen würden.

Smoothingeffekt auf ein 2-Stunden-Durchschnittsfenster bei einem Wärmepumpenlastprofil

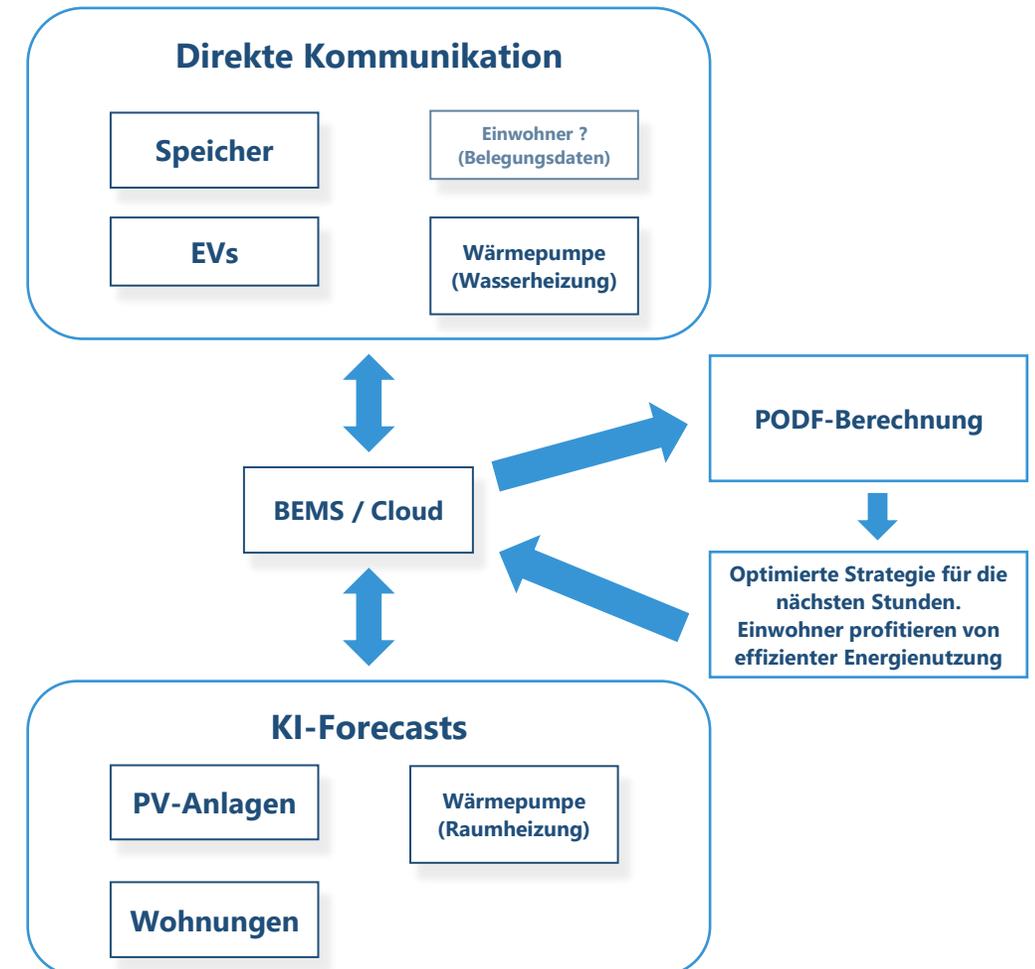


# Integration schwer vorhersagbarer Daten

## Direkte Kommunikation zwischen Geräten und dem BEMS

- Wie bereits beim Speicher erwähnt, scheint die beste Lösung für Wärmepumpen und EVs **die direkte Integration ihrer Programme in das BEMS zu sein**. Aufgrund der schlechten „Forecastability“ dieser Lasten stößt die KI-Vorhersage hier an ihre Grenzen.
- **Idealerweise weiß das BEMS, wann ein EV geladen wird, und hat alle und Informationen zum Ladezustand des angeschlossenen Elektrofahrzeugs** (State of Charge, Energiebedarf und Abfahrzeit).
- Für Wärmepumpen (insbesondere Wassererwärmung) **könnte das Programm der Wärmepumpe ebenfalls an das BEMS kommuniziert werden**. Moderne Wärmepumpen verfügen bereits über Sensoren und „Monitoring Software“, um solche Informationen zu kommunizieren und in HEMS / BEMS zu integrieren.

**Gibt es mögliche Anreize, dass Nutzer und Ersteller im Smart-Building-Kontext Gerätedaten (z. B. den Ladezustand von Elektrofahrzeugen), Belegungsdaten usw. teilen?**





# Cloud-Lösung vs Edge-Devices

# Cloud-Lösung vs Edge-Devices

## Unterscheidung und relevante Abwägungen

### Cloud-Lösung:

Alle Komponenten der Machine Learning Lösung befinden sich auf einem extern gehosteten Server, durch die BEMS-Geräte werden lediglich die Sensordaten zusammengeführt und gesendet.

### Edge-Devices:

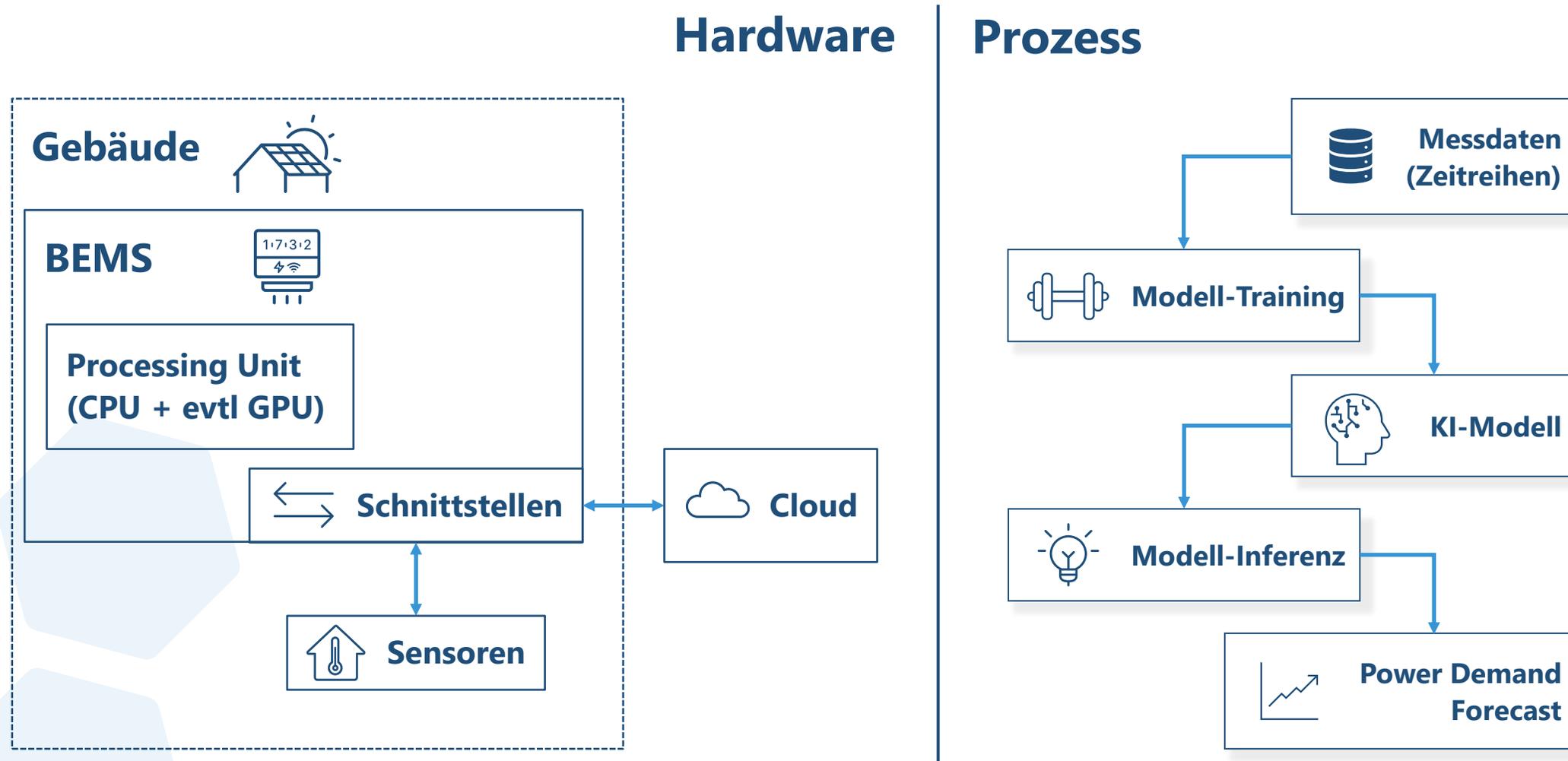
Teile der ML-Lösung werden direkt auf den BEMS-Geräten (den Edge-Devices) ausgeführt. Komponenten wie das Training des Basismodells<sup>1</sup>, die Daten aller BEMS-Geräte benötigen, finden jedoch auch hier weiterhin in der Cloud statt.

### Zu berücksichtigende Abwägungen (u.a.):

- Performance & Hardwareanforderungen, Kostenaspekte
- Komplexität der Modellentwicklung und -wartung, Skalierbarkeit und Robustheit
- Datenströme und Datensparsamkeitsaspekte

# Ausgangslage

## Involvierte Hardware & Prozess der ML-Lösung



# Mögliche technische Setups

## Welche Prozesse finden in der Cloud, welche auf dem BEMS statt?

### 1. Inferenz und Training in der Cloud

- BEMS nur als Kommunikations-Gateway für Rohdaten
- Modell wird in der Cloud trainiert mit historischen Daten aller Gebäude, optional spezialisierte Modelle pro Gebäude
- Inferenz findet in Echtzeit auf aktuellen Rohdaten der jeweiligen Gebäude statt

### 2. Modell-Training in der Cloud, Inferenz auf BEMS-Hardware

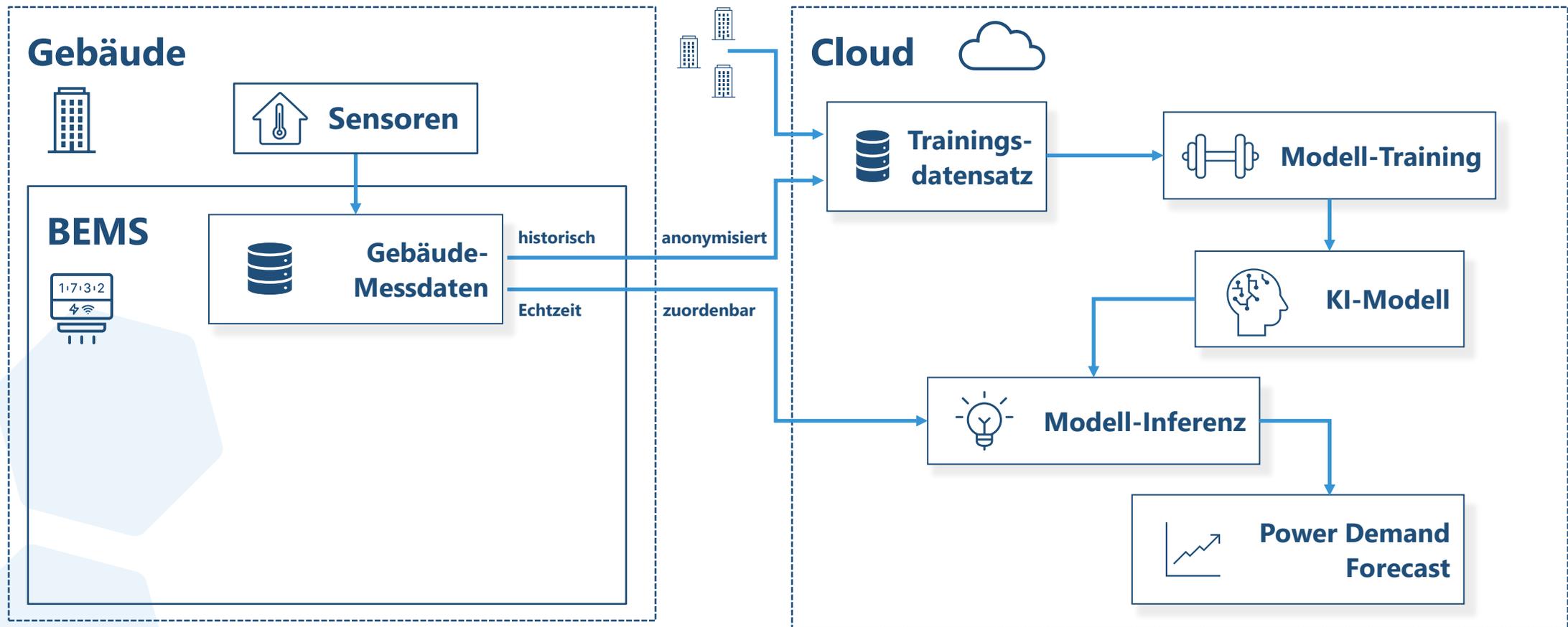
- Modell-Training in der Cloud mit historischen Daten aller Gebäude (müssen nicht zuordenbar sein)
- Inferenz auf BEMS-Hardware, nur fertiger Forecast wird nach außen kommuniziert
- ML-Modell ist nicht gebäudespezifisch angepasst, ein Modell für alle Gebäude

### 3. (Re-)Training/Finetuning des ML-Modells und Inferenz auf BEMS-Hardware

- Basis-Modell-Training in der Cloud mit historischen Daten aller Gebäude (müssen nicht zuordenbar sein)
- Spezifisches Modell wird auf BEMS mit Gebäude-eigenen Daten feingetuned, Inferenz auf BEMS

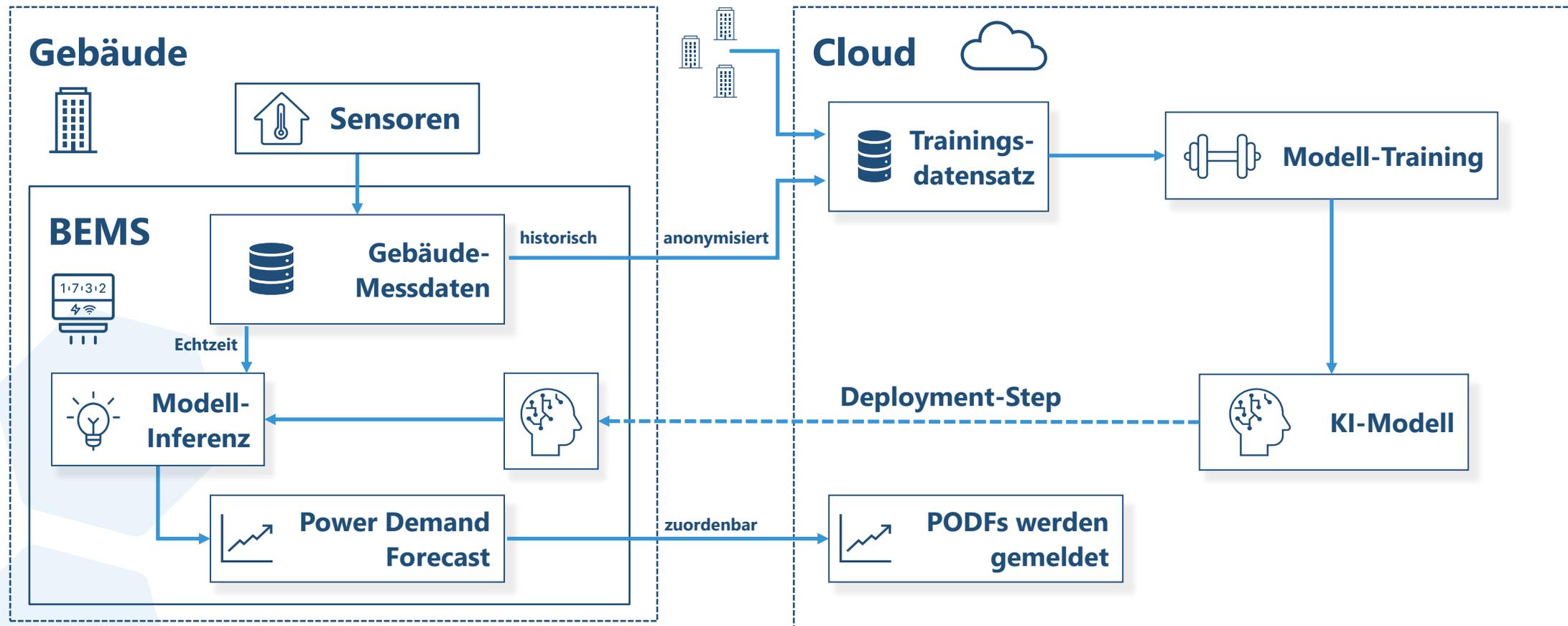
# Inferenz und Training in der Cloud

## Setup 1



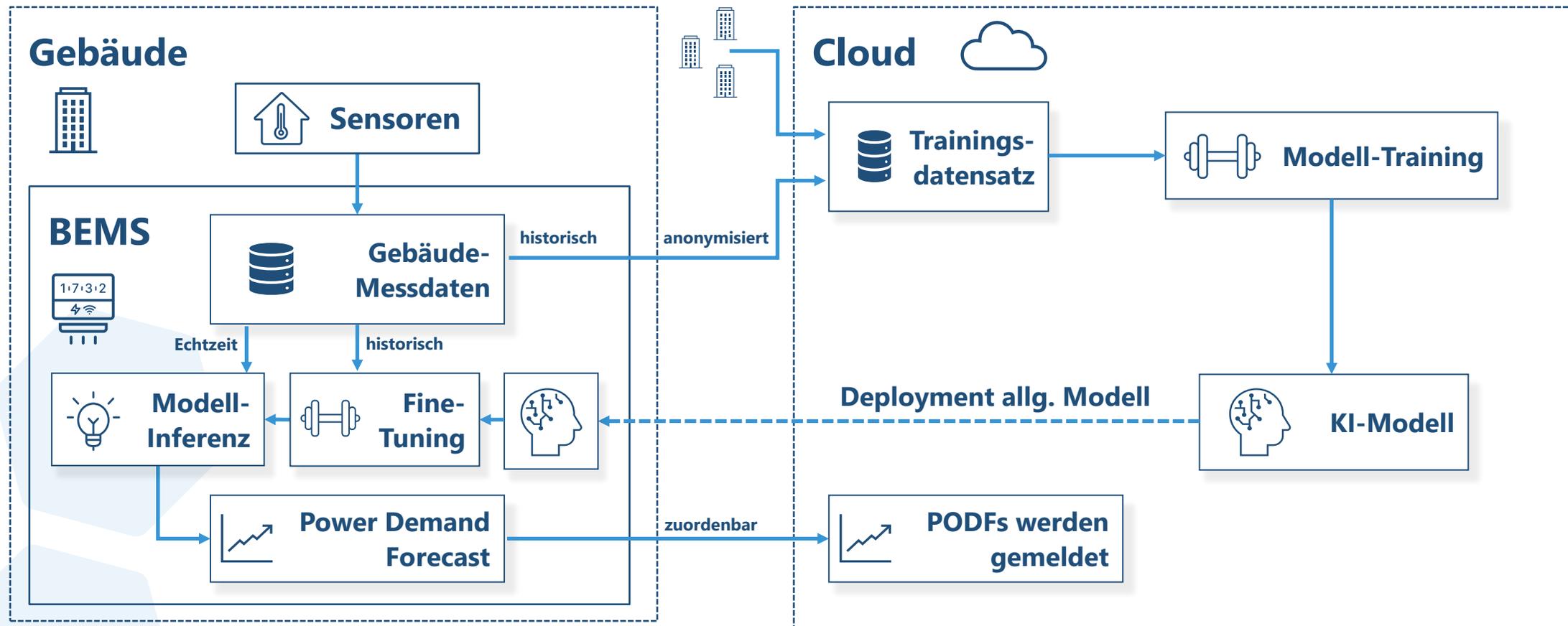
# Modell-Training in der Cloud, Inferenz auf BEMS

## Setup 2



# Inferenz und Modell-Finetuning auf BEMS-Hardware

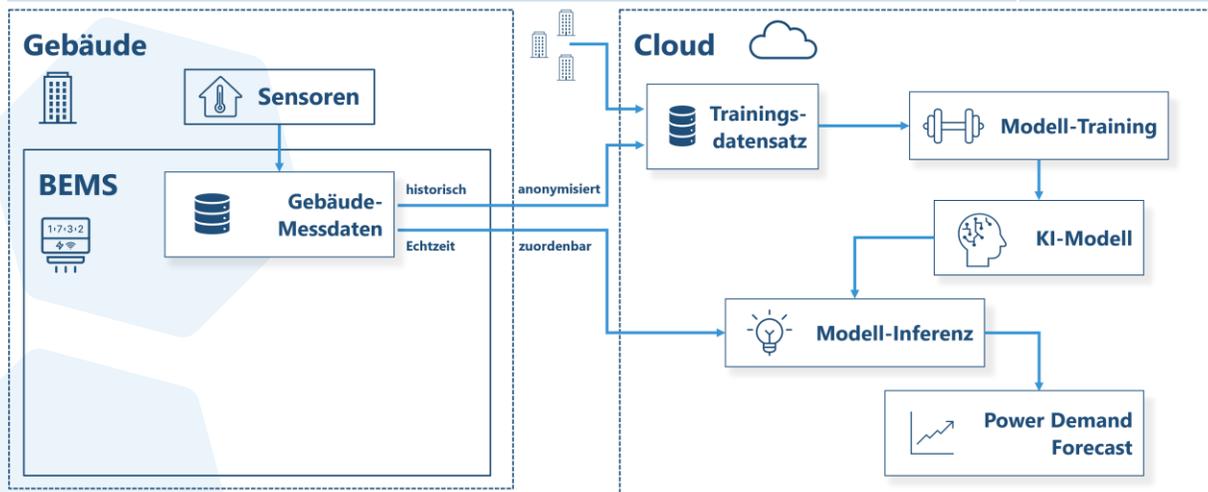
## Setup 3



# Inferenz und Training in der Cloud

## Setup 1

Pro	Contra
<ul style="list-style-type: none"> <li>• Niedrigste Anforderungen an BEMS-Hardware</li> <li>• Skalierbar, beliebige Hardware-Ressourcen für Training und Inferenz verfügbar</li> <li>• Vereinfachte Wartung und Updates durch Verlagerung der kritischsten Schritte in zentrale und gut zugängliche Cloud-Struktur</li> </ul>	<ul style="list-style-type: none"> <li>• Datensparsamkeit: Übertragene Messdaten haben höheren Personenbezug als PODF</li> <li>• Größere Mengen an Daten müssen fortlaufend übertragen werden</li> <li>• Durch Zentralisierung: Ausfall der Cloud-Lösung hätte Forecast-Lücken für alle Geräte zur Folge</li> </ul>



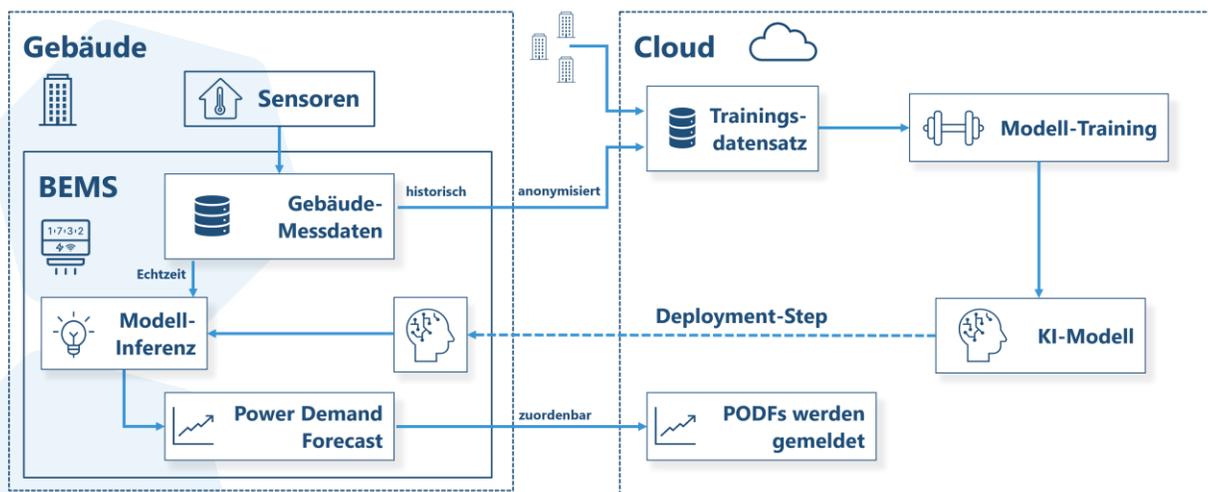
**Hardware-Anforderungen:**

- Übertragungs-Schnittstelle die alle Inputdaten in Echtzeit übertragen kann
- Geringe Anforderungen an Processing Unit

# Modell-Training in der Cloud, Inferenz auf BEMS

## Setup 2

Pro	Contra
<ul style="list-style-type: none"> <li>Nur der PODF selbst, aber keine weiteren personenbezogenen Daten verlassen das Gebäude</li> <li>Dezentral, potentiell resistenter gegen Ausfälle</li> </ul>	<ul style="list-style-type: none"> <li>Updates der ML-Lösung erfordern Deployment komplexer Software auf potentiell verschiedenen Hardwareversionen</li> <li>Anpassung des ML Modells auf Basis der eigenen historischen Daten nur durch Übertragung der Rohdaten möglich</li> </ul>



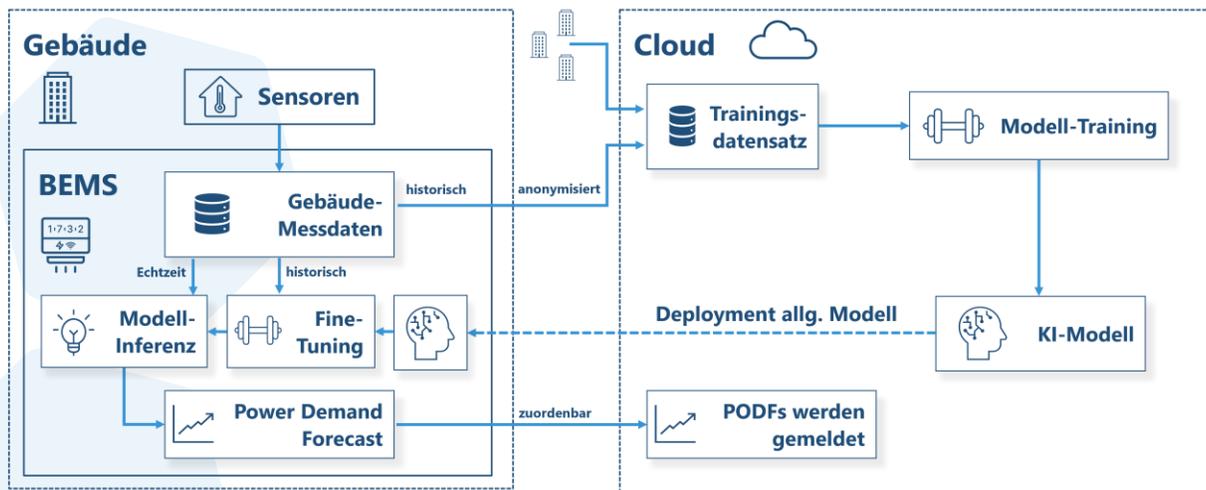
**Hardware-Anforderungen:**

- Inferenz auf BEMS-Hardware stellt gewisse Mindestanforderungen an Rechenleistung
- Hardware muss jeweils neueste Version des ML-Setups bei Updates unterstützen

# Inferenz und Modell-Finetuning auf BEMS-Hardware

## Setup 3

Pro	Contra
<ul style="list-style-type: none"> <li>Modell kann an spezifische Gegebenheiten angepasst und auf Grundlage eigener Daten weitertrainiert werden, ohne Rohdaten zu verschicken</li> </ul>	<ul style="list-style-type: none"> <li>Hohe Hardware-Anforderungen durch ML- Modell-Training auf BEMS</li> <li>Erschwerte Maintenance und Update durch inhomogene Hardware wie in Setup 2</li> </ul>



### Hardware-Anforderungen:

- Relativ starke Processing Unit, idealerweise on-board-GPU oder NPU ('KI-Chip')
- Hardware muss jeweils neueste Version des ML-Setups bei Updates unterstützen

# Hardware-Kriterien

## Vergleich der 3 Setups

Hardware-Beschränkung	Mindestanforderung Training	Mindestanforderung Inferenz	Typische HEMS-Hardware: Consolinno Leaflet HEMS 1U0022
<b>Rechenleistung (CPU)</b>	CPU mind. ‚Notebook-stark‘ (~8 Kerne, ~2-3GHz), integrierter Grafikprozessor	Etwas schwächer als für Inferenz, je nach verfügbarer Zeitspanne zwischen Inferenzschritten	NXP i.MX6ULL, <b>Single Core, 792 MHz</b>
<b>Arbeitsspeicher (RAM)</b>	Mind. <b>16 GB</b>	Mind. <b>4 GB</b> (klass. ML), mind. <b>8-16 GB</b> (Deep Learning)	<b>512 MB</b>

Grund für schwache Rechenleistung der HEMS-Hardware ist v.a. Optimierung auf **Energieeffizienz**:

Durchschnittlicher Stromverbrauch von HEMS-CPU: 41.15 mW, zum Vergleich Notebook-CPU: 15-45 W

→ Upgrade auf ‚Notebook-Hardware‘ würde 50-100€ höheren Stromkosten im Jahr entsprechen.

Weitere relevante Hardware-Kriterien: **Robustheit/Lebensdauer** und **Wartungsintensität**

→ Oft Tradeoff zwischen diesen Kriterien und hoher Rechenleistung (z.B. höhere thermische Belastung)

→ Einbau stärkerer CPUs in HEMS-Hardware kann zu Nachteilen in anderen Bereichen führen

**Fazit:** Mit heutiger HEMS-Hardware ist **nur Setup 1** (Training + Inferenz in der Cloud) **umsetzbar**.

# Kostenaspekte

## Vergleich der 3 Setups

Kostenaspekte	Gewichtung	Setup 1: Inferenz + Training in der Cloud	Setup 2: Training in der Cloud, Inferenz auf BEMS	Setup 3: Finetuning + Inferenz auf BEMS
Hardware	€€€€	Nur Schnittstellen-funktionalität	Erhöhte Hardwareanforderungen	Hohe Hardwareanforderungen
Rechenkosten Training	€	Basismodell-Training findet für alle Setups identisch in der Cloud statt		
Rechenkosten Inferenz	€€	Laufende Kosten für Inferenz in der Cloud	Erhöhte Stromkosten auf Anwenderseite für Inferenz auf BEMS, vmtl. niedriger als Cloudkosten	Erhöhte Stromkosten auf Anwenderseite für Inferenz + Finetuning auf BEMS, vmtl. niedriger als Cloudkosten
Software Engineering (Modellentwicklung, Deployment, Software-Infrastruktur, Wartung & Updates)	€€€€€	Alle Entwicklungs- und Updateschritte finden zentral in der Cloud statt und sind unabhängig von den BEMS-Hardware-Versionen	Alle Entwicklungs- und Updateschritte müssen auf jedem BEMS-Gerät deployed werden, Kompatibilitätsprobleme mit verschiedenen Hardware-Versionen sind nach einigen Jahren Betriebsdauer fast unvermeidlich, insbesondere wenn die Lösung mit den schnellen Fortschritten im KI-Bereich mithalten soll	

**Fazit:** Insbesondere in den kostenintensivsten Bereichen Hardware und Software Engineering weist **Setup 1** durch die geringere Abhängigkeit von den einzelnen Hardwareversionen signifikant **niedrigere Kosten** auf.

# Handlungsempfehlung

## Vergleich der 3 Setups

Die Cloud-Lösung (Setup 1) weist die folgenden wesentlichen Vorteile gegenüber einer Verlagerung von Teilen der Machine-Learning-Lösung auf die BEMS-Geräte auf:

- **Skalierbarkeit** der Modellkomplexität, vereinfachte **Wartung und Updates** des ML-Modells
- **Deployment neuer Modellversionen** auf verschiedenen Hardware-Generationen wird mit jedem Update komplexer und fehleranfälliger, **höhere Robustheit der Cloud-Lösung**
- Derzeitige typische **HEMS-Hardware ist nicht leistungsfähig genug für ML-Inferenz/Training**
- **Höhere Kosten** in kritischen Bereichen Hardware und Software-Engineering-Aufwand

**Auf Grundlage der vorangehenden Analysen zu Vor- und Nachteilen der verschiedenen Lösungen, Hardware-Restriktionen und Kostenaspekten empfehlen wir daher die Fokussierung auf eine Cloud-basierte Lösung (Setup 1).**

# Vorhandene Lösungen

## PODF & Forecasting-Algorithmen

### Forecasting-Algorithmen:

- Verschiedene Forecasting-Algorithmen wurden in den vorherigen Kapiteln beschrieben und verglichen
- Zu diesen existiert eine Vielzahl an **Open Source Implementierungen**, die kommerziell frei nutzbar sind
- Open Source Projekte werden oft von Freiwilligen erstellt und gewartet und stellen daher meist keinen produktionsreifen Code, sondern eher eine Basis für eigene Implementierungen dar
- ML-Tasks sind sehr inhomogen, jeder Datensatz hat individuelle Herausforderungen

### PODF-Lösungen:

- Bisher gibt es keine Dynamik im Markt für fertigen PODF-Lösungen als Produkt.
- Verschiedene Aspekte der PODF-Entwicklung sind jedoch Gegenstand von aktueller Forschung (Referenzen in den vorherigen Kapiteln)
- Durch verschiedene Anforderungen je nach Use Case ist eine individuelle Entwicklung sinnvoll

Für die meisten Problemstellungen ist keine ‚out-of-the-box-Lösung‘ vorhanden, dies macht die **Entwicklung eines spezifischen Forecast-Modells** notwendig



# Handlungsempfehlung

# Evaluierung der BEMS-Fähigkeiten für KI-basierte Strombedarfsprognose

## Fazit

**1**

**KI-basierte Prognosen zeigen großes Potenzial, um zur PODF-Berechnung beizutragen**, insbesondere für Erzeugungsanlagen und den Haushaltsverbrauch von Wohnungen. Mehrere KI-Ansätze in der Literatur übertreffen die Verwendung von Standardlastprofilen.

**2**

KI-Prognosen für Geräte wie Wärmepumpen und Elektrofahrzeuge scheinen komplex. **Wir empfehlen, die Schnittstelle zwischen BEMS und den Geräten zu verbessern (sowohl Sensorik als auch Software)**, um wertvolle Daten direkt von den Geräten (z.B. Heizprogramme, EV-Ladezustand) abzurufen.

**3**

Ein wesentlicher Aspekt, der präzise und zuverlässige PODF erschwert, ist die Unvorhersehbarkeit menschlichen Verhaltens. **Könnten neue Anreize direkte Nutzereingaben zur Wohnungsbelegung, zu Ladegewohnheiten von Elektrofahrzeugen und zu Heizgewohnheiten ermöglichen?**

**4**

**Wir empfehlen ein cloudbasiertes PODF**, wobei das BEMS als Aggregations- und Kommunikationspunkt dient. Die Implementierung eines PODF auf dem Gerät würde erheblichen Aufwand erfordern, ohne wesentliche Vorteile zu bieten.

**Die Empfehlungen basieren auf qualitativen Schätzungen. Wir empfehlen, einen Feldtest durchzuführen, um unsere Annahmen zu bestätigen und Erfahrungen zu sammeln, bevor mit der Implementierung der Lösung fortgefahren wird.**