



Solar Forecast ML - Benutzerhandbuch

Version 12.0.0 "Sarpeidon II "

Professionelle Dokumentation für die Solar Forecast ML Integration

Version 12.0.0 "Sarpeidon" | Erstellt: Dezember 2025 | Sprache: Deutsch



Inhaltsverzeichnis

1. [Einführung](#)
2. [Was ist Solar Forecast ML?](#)
3. [Wie funktioniert die Integration?](#)
4. [Systemvoraussetzungen](#)
5. [Benötigte Sensoren und Einheiten](#)
6. [Panel-Gruppen Konfiguration](#)
7. [Installation und Einrichtung](#)
8. [Funktionsübersicht](#)
9. [Sensoren und ihre Bedeutung](#)
10. [Dateistruktur und JSON-Dateien](#)
11. [Maschinelles Lernen - Technische Grundlagen](#)
12. [Services und Funktionen](#)
13. [Konfigurationsoptionen](#)
14. [Tägliches Solar-Briefing](#)
15. [Fehlersuche und Problemlösung](#)
16. [Häufig gestellte Fragen](#)

1. Einführung

Willkommen beim Solar Forecast ML Benutzerhandbuch. Dieses Dokument erklärt die Funktionsweise, Einrichtung und Nutzung der Integration im Detail. Es richtet sich an technisch interessierte Anwender, die verstehen möchten, wie die Integration arbeitet und wie sie optimal konfiguriert wird.

Über dieses Handbuch

Dieses Handbuch verzichtet bewusst auf Code-Beispiele und technische Symbole. Stattdessen erklärt es die Konzepte, Funktionen und technischen Hintergründe in verständlicher Sprache. Ziel ist es, Ihnen ein fundiertes Verständnis der Integration zu vermitteln, damit Sie diese optimal nutzen können.

2. Was ist Solar Forecast ML?

Solar Forecast ML ist eine Home Assistant Integration, die mithilfe von **maschinellern Lernen** präzise Vorhersagen über die zu erwartende Solarstromerzeugung Ihrer Photovoltaik-Anlage erstellt. Im Gegensatz zu einfachen Wettervorhersage-Diensten lernt die Integration kontinuierlich aus den tatsächlichen Produktionsdaten Ihrer spezifischen Anlage.

Grundprinzip

Die Integration kombiniert drei Informationsquellen:

- **Wetterdaten:** Temperatur, Bewölkung, Niederschlag und weitere meteorologische Parameter werden automatisch von Open-Meteo bezogen. Es ist keine separate Wetter-Integration erforderlich.
- **Astronomische Daten:** Sonnenstand, Sonnenauf- und -untergangszeiten, Sonnenhöhe und weitere astronomische Berechnungen werden präzise für Ihren Standort ermittelt.
- **Historische Produktionsdaten:** Die tatsächliche Stromerzeugung Ihrer Anlage wird kontinuierlich erfasst und mit den Vorhersagen verglichen.

Aus diesen drei Datenquellen erstellt ein maschinelles Lernmodell personalisierte Vorhersagen, die sich täglich verbessern.

Warum maschinelles Lernen?

Jede Solaranlage ist einzigartig. Dachausrichtung, Neigungswinkel, lokale Verschattungen, Modultyp und Wechselrichter-Effizienz beeinflussen die Stromerzeugung unterschiedlich. Ein maschinelles Lernmodell kann diese individuellen Faktoren berücksichtigen und lernt mit der Zeit, wie Ihre spezifische Anlage auf verschiedene Wetterbedingungen reagiert.

3. Wie funktioniert die Integration?

Die Integration arbeitet in einem mehrstufigen Prozess, der sich täglich wiederholt:

Datenerfassung

Jede Stunde erfasst die Integration verschiedene Datenpunkte:

- **Wetterdaten** werden automatisch abgerufen. Zusätzlich wird eine zweite Quelle für Multi-Weather-Blending verwendet.
- **Astronomische Berechnungen** werden präzise für Ihren Standort durchgeführt. Die Integration berechnet den aktuellen Sonnenstand, die Sonnenhöhe über dem Horizont, den Azimutwinkel und die theoretisch maximal mögliche Solarstrahlung unter wolkenlosem Himmel.
- **Produktionsdaten** Ihrer Anlage werden vom konfigurierten Leistungssensor ausgelesen.

Datenverarbeitung

Die gesammelten Daten werden in ein strukturiertes Format gebracht und gespeichert. Dabei werden verschiedene abgeleitete Werte berechnet, beispielsweise Interaktionen zwischen Temperatur und Sonnenhöhe oder jahreszeitliche Muster.

Vorhersageberechnung

Die Integration verwendet mehrere Vorhersagemethoden parallel:

1. **Regelbasierte Vorhersage:** Eine physikalisch fundierte Berechnung basierend auf Sonnenstand, Wetterdaten und der Anlagenkapazität. Diese Methode funktioniert ab dem ersten Tag.
2. **Maschinelles Lernmodell:** Ein selbstlernendes Modell, das aus historischen Daten trainiert wird.
3. **Adaptive Gewichtung:** Die Integration kombiniert beide Methoden intelligent. Zunächst dominiert die regelbasierte Vorhersage. Mit zunehmender Trainingsdatenmenge erhält das maschinelle Lernmodell mehr Gewicht, sofern es sich als zuverlässig erweist.
4. **KI-Learning:** Mittels KI (lokal) lernt die Integration Muster und Spezifikationen der Anlage kennen und greift in die Prognose ein. Hierfür ist je nach Situation und Gegebenheiten eine unterschiedliche Anzahl an Lerndaten notwendig.

Kontinuierliche Verbesserung

Nach jedem Tag vergleicht die Integration die Vorhersage mit der tatsächlichen Produktion. Diese Informationen fließen in das maschinelle Lernmodell ein, wodurch zukünftige Vorhersagen präziser werden.

4. Systemvoraussetzungen

Home Assistant Version

Die Integration erfordert **Home Assistant Core in Version 2023.1 oder neuer**. Empfohlen wird die jeweils aktuelle stabile Version. Bitte nicht auf Systemen mit SD-Karten oder Pi kleiner 4 verwenden. Bei HA in VM z.B. Proxmox ist die korrekte Konfiguration zwingend erforderlich. Empfohlen wird die native HA Installation! - Kein Support für Proxmox, bei Problemen bitte direkt an Proxmox wenden.

Recorder Integration

Die **Recorder-Integration** muss aktiviert sein, da historische Daten für das maschinelle Lernen benötigt werden. Die Standard-Einstellungen sind ausreichend.

Python-Pakete

Folgende Python-Bibliotheken werden automatisch installiert:

- **aiofiles**: Ermöglicht asynchrone Dateizugriffe für optimale Performance ohne Blockierung des Home Assistant Systems.
- **numpy**: Stellt mathematische Funktionen für die Datenverarbeitung und das maschinelle Lernen bereit.

Speicherplatz

Die Integration benötigt etwa **5 bis 20 Megabyte** Speicherplatz im Home Assistant Konfigurationsverzeichnis.

Rechenleistung

Die Integration ist ressourcenschonend konzipiert. Auf einem Raspberry Pi 4 verursacht sie eine durchschnittliche CPU-Last von unter einem Prozent.

Internetverbindung

Eine Internetverbindung ist für den Abruf von Wetterdaten erforderlich. Die Integration nutzt die kostenlose Open-Meteo API automatisch, ohne dass Sie sich registrieren oder einen API-Schlüssel konfigurieren müssen.

5. Benötigte Sensoren und Einheiten

Für die korrekte Funktion der Integration müssen bestimmte Sensoren in Home Assistant vorhanden sein.

Pflicht-Sensoren

Diese Sensoren müssen zwingend konfiguriert werden:

Leistungssensor (power_entity)

Eigenschaft	Details
Beschreibung	Ein Sensor, der die aktuelle Leistung Ihrer Solaranlage in Echtzeit misst.
Einheit	Watt (W)
Wichtig	Der Sensor muss die tatsächliche Erzeugungsleistung anzeigen, nicht die Netzeinspeisung oder den Hausverbrauch.

Tagesertragssensor (solar_yield_today)

Eigenschaft	Details
Beschreibung	Ein Sensor, der den gesamten erzeugten Solarstrom des aktuellen Tages anzeigt.
Einheit	Kilowattstunden (kWh)
Mitternachts-Reset	Der Sensor muss sich jeden Tag um Mitternacht automatisch auf Null zurücksetzen. Dies ist kritisch für die korrekte Funktion.
Quelle	Üblicherweise vom Wechselrichter bereitgestellt oder als Utility Meter in Home Assistant erstellt.

Warnung: Riemann-Integrations-Sensoren und Cloud-Dienste

WICHTIG: Viele Benutzer verwenden Riemann-Summen-Integrations-Sensoren (der "Integration - Riemann sum integral" Helper in Home Assistant) um aus einem Leistungssensor (W) einen Energiesensor (kWh) zu berechnen. Diese Methode kann jedoch **erhebliche Ungenauigkeiten von bis zu 25% pro Tag** verursachen.

Das Problem

Die Riemann-Integration berechnet die Energie durch Aufsummieren von Leistungswerten über die Zeit. Die Genauigkeit hängt direkt von der Aktualisierungsrate des Quellsensors ab:

- **Bei lokalen Integrationen** (Modbus, lokale API): Ungenauigkeit typischerweise **1–3%** (akzeptabel).
- **Bei Cloud-Diensten** (Tuya, eWeLink, Cloud-only Wechselrichter): Aktualisierung oft nur alle 1–5 Minuten. **Typische Ungenauigkeit: 10–25% pro Tag** (problematisch).

Achtung: Die Riemann-Summe nimmt an, dass die Leistung zwischen zwei Messungen konstant bleibt. Bei einem Cloud-Dienst, der nur alle 3 Minuten aktualisiert, summiert sich der Fehler bei schnell wechselnder Bewölkung stark.

Die Lösung: SQL-Sensoren

Für präzise Energiemessungen bei Cloud-Diensten empfehlen wir **SQL-Sensoren**, die direkt aus der Home Assistant Datenbank lesen.

Vorteile von SQL-Sensoren:

- Nutzen die tatsächlich aufgezeichneten Daten aus der Recorder-Datenbank.
- Berücksichtigen alle verfügbaren Datenpunkte, nicht nur die letzten.
- Können mit fehlenden Werten besser umgehen.

Empfehlungen nach Sensor-Quelle

Quelle	Riemann geeignet?	Empfehlung
Modbus (lokal)	Ja	Riemann oder Utility Meter
SolarEdge/Fronius/SMA (lokal)	Ja	Riemann oder Utility Meter
Tuya/eWeLink (Cloud)	Nein	SQL-Sensor empfohlen
Growatt/Solarman/Hoymiles (Cloud)	Bedingt	SQL-Sensor empfohlen

Optionale Sensoren

Diese Sensoren verbessern die Vorhersagegenauigkeit, sind aber nicht zwingend erforderlich:

Sensor	Einheit	Verwendung
Anlagenkapazität (solar_capacity)	kWp	Berechnet theoretische Maximalwerte.
Wechselrichter Max-Leistung (inverter_max_power)	kW	Schließt Clipping-Stunden vom ML-Training aus.
Tagesverbrauch (total_consumption_today)	kWh	Ermöglicht die Berechnung der Autarkie.
Temperatur-Sensor (temp_sensor)	°C	Erfasst lokale Temperaturabweichungen.
Solarstrahlung-Sensor (solar_radiation_sensor)	W/m ²	Direkteste Messung der verfügbaren Sonnenenergie.

6. Panel-Gruppen Konfiguration

Panel-Gruppen ermöglichen die Modellierung von Solaranlagen mit **unterschiedlich ausgerichteten Modulen** (z.B. Ost-West-Anlagen) für genauere Vorhersagen.

Warum Panel-Gruppen?

Ohne Panel-Gruppen würde die Integration bei komplexen Anlagen einen ungenauen Durchschnitt berechnen. Mit Panel-Gruppen wird **jede Teilanlage individuell berechnet**.

Format der Panel-Gruppen

Grundformat: Leistung(Wp)/Azimut(°)/Neigung(°)/sensor.entity_id (kWh/Tag)

Erweitertes Format mit Energy-Sensor: Leistung(Wp)/Azimut(°)/Neigung(°)/sensor.entity_id

Parameter erklärt

- **Leistung (Wp):** Die Spitzenleistung der Gruppe in Watt Peak.
- **Azimut (°):** Die Himmelsrichtung (0° Nord, 180° Süd, 270° West).
- **Neigung (°):** Der Winkel der Module zur Horizontalen (0° flach, 90° vertikal).
- **Energy-Sensor:** Ein Tages-kWh-Sensor für diese spezifische Gruppe. Ermöglicht gruppenspezifisches Lernen.

Beispiele

- **Ost-West-Anlage:**
1500/90/15/sensor.pv_gruppe1_energy_today,
1500/270/15/sensor.pv_gruppe2_energy_today

7. Installation und Einrichtung

Installation über HACS

Die Installation erfolgt über HACS (Benutzerdefinierte Repositories) durch Hinzufügen der GitHub-URL und Herunterladen der Integration "Solar Forecast ML". Anschließend ist ein Neustart von Home Assistant erforderlich.

Einrichtung der Integration

Der Einrichtungsassistent führt Sie durch die Auswahl der **Pflicht-Sensoren** (Leistung in W, Tagesertrag in kWh) und die Konfiguration der optionalen Sensoren und Panel-Gruppen.

Erste Schritte nach der Installation

In den ersten Tagen arbeitet die Integration ausschließlich mit **regelbasierten Vorhersagen**. Das maschinelle Lernmodell beginnt mit dem Training, sobald ausreichend Daten (typischerweise **4 bis 14 Tage**) gesammelt wurden.

8. ✨ Funktionsübersicht

Funktion	Beschreibung	Nutzen
Solarvorhersage	Prognose für Heute, Morgen und Übermorgen.	Basis für Planung und Automatisierung.
Multi-Weather-Blending	Intelligente Kombination von Open-Meteo und wtrr.in.	Verbessert die Prognosegenauigkeit bei wechselhafter Bewölkung.
Stündliche Vorhersagen	Detaillierte Prognosen pro Stunde.	Präzise Steuerung von Verbrauchern (E-Auto, Waschmaschine).
Spitzenlastzeit-Erkennung	Identifiziert die beste Stunde zur Aktivierung großer Verbraucher.	Optimierung des Eigenverbrauchs.
Genauigkeits-Tracking	Täglicher Vergleich von Prognose und IST-Wert.	Transparenz über die Zuverlässigkeit des ML-Modells.

9. Sensoren und ihre Bedeutung

Die Integration erstellt zahlreiche Sensoren (Entities) in Home Assistant.

Hauptvorhersage-Sensoren

Sensor	Beschreibung	Einheit
Prognose (heute)	Erwartete Tagesproduktion	kWh
Prognose (morgen)	Vorhersage für den kommenden Tag	kWh
Beste Stunde	Uhrzeit der höchsten erwarteten Produktion	Zeit

Leistungs- und Ertrags-Sensoren

Sensor	Beschreibung	Einheit
Max. Peak Heute	Höchste gemessene Leistung heute	W
Ø Ertrag 30 Tage	Durchschnittlicher Tagesertrag der letzten 30 Tage	kWh

Genauigkeits-Sensoren

Sensor	Beschreibung	Einheit
Prognosegenauigkeit Gestern	Wie genau war die gestrige Vorhersage	%
Ø Genauigkeit 30 Tage	Durchschnittliche Genauigkeit der letzten 30 Tage	%

ML-Status-Sensoren

Diese Sensoren zeigen den Zustand des maschinellen Lernsystems an:

Sensor	Beschreibung	Zustände/Werte
ML-Service Status	Aktueller Status des ML-Systems	Nicht trainiert, Training läuft, Bereit, Fehler
Aktives Vorhersagemodell	Welches Modell aktuell verwendet wird	ml, weather (regelbasiert)
Training Bereitschaft	Status der Trainingsbereitschaft	Sammele Daten, Frühes Training, Bereit, Exzellent

10. Dateistruktur und JSON-Dateien

Die Integration speichert alle Daten als JSON-Dateien im Verzeichnis `/config/solar_forecast_ml/`.

Verzeichnisstruktur

```
/config/solar_forecast_ml/
├── data/          # Laufzeitdaten und Caches
├── stats/         # Statistiken und Prognosen
├── ml/           # Maschinelles Lernmodell
├── physics/       # Physik-basiertes Lernen
├── logs/         # Log-Dateien
└── backups/      # Automatische Sicherungen
```

11. Maschinelles Lernen - Technische Grundlagen

Die Modelle

Die Integration verwendet ein **Ensemble von Modellen**:

1. **Regressionsmodell**: Lernt aus historischen Daten (Wetter, Astronomie, Produktion).
2. **Physik-basiertes Modell**: Eine regelbasierte Berechnung, die als **Safeguard** dient.

Feature Engineering

Das Modell lernt aus abgeleiteten, hochkorrelierten "Features" wie:

- **Relative Sonnenhöhe**
- **Temperatur-Effekt** (Leistungsminderung durch Modultemperatur)
- **Wolken-Interaktionen**

ML-Training (Täglich)

Das Training findet **täglich um 23:30 Uhr** statt. Dabei werden die gestrigen Daten gesammelt, die Genauigkeit geprüft und das ML-Modell mit der gesamten historischen Datenbank neu trainiert.

Adaptive Gewichtung

Die Integration kombiniert Physik- und ML-Modell. Mit zunehmender Zuverlässigkeit erhält das ML-Modell mehr Gewicht. Das Ziel ist eine **ML-Gewichtung von über 70%**.

Data Safeguard (Überprüfung der Plausibilität)

Jede ML-Prognose wird durch einen **Data Safeguard** auf Plausibilität geprüft, um Fehler zu vermeiden:

- **Wetter-Korrektur:** Bei extremer Bewölkung wird die Prognose automatisch auf den Mindestwert der Physik-Formel korrigiert.
- **Speicher-Bereinigung:** Extrem unplausible historische Datenpunkte (Sensorfehler) werden aus der Trainingsdatenbank entfernt.

12. Services und Funktionen (NUR FÜR ENTWICKLER)

Die Integration stellt mehrere Home Assistant Services zur Steuerung bereit:

Service	Beschreibung	Anwendungsfall
solar_forecast_ml.force_update	Erzwingt eine sofortige Aktualisierung aller Daten und Prognosen.	Manuelle Aktualisierung nach Konfigurationsänderung.
solar_forecast_ml.force_ml_retrain	Löst das tägliche ML-Training sofort aus.	Beschleunigtes Training in der frühen Lernphase.
solar_forecast_ml.reset_ml_model	Löscht die ML-Trainingsdatenbank und setzt das Training zurück.	Bei einem Wechsel des Wechselrichters oder einer Anlagen-Erweiterung.

13. Konfigurationsoptionen

Die Konfiguration erfolgt über die Home Assistant Benutzeroberfläche (UI).

Option	Beschreibung	Standardwert
Stündliche Prognosen aktivieren	Erstellt die Sensoren für die stündlichen Vorhersagen.	Aus
ML-Training Frequenz	Frequenz des ML-Trainings (Täglich/Wöchentlich).	Täglich
Plausibilitätsgrenze	Prozentsatz, um den die Prognose den Peak überschreiten darf (Safeguard).	10 %

14. Tägliches Solar-Briefing

Das Solar-Briefing ist ein dynamisch generierter Bericht (solar_forecast_ml.request_briefing), der die wichtigsten Informationen des Tages zusammenfasst (Hauptprognose, Beste Stunde, Lern-Status, Empfehlung).

15. Fehlersuche und Problemlösung

Problem	Mögliche Ursache(n)	Lösung
Prognose ist Null.	Tagesertragssensor setzt sich nicht um Mitternacht zurück.	Utility Meter Helper korrekt mit Zyklus Täglich konfigurieren.
Genauigkeit ist schlecht (>20% Abweichung).	Riemann-Summen-Sensor bei Cloud-Diensten.	Unbedingt auf SQL-Sensor umstellen (siehe Abschnitt 5).
ML-Service Status bleibt auf "Nicht trainiert".	Es wurden nicht genügend Daten (<14 Tage) gesammelt.	Warten Sie 1–2 Wochen oder den Service force_ml_retrain auslösen.
Atomic write Fehler	Dateisystem-Fehler, Snapshot aktiviert (Proxmox), Ressourcen Problem, i/o Performance zu gering	Auf Proxmox Konfiguration prüfen, SD-Karte durch SSD ersetzen
Prognose schlecht	Zuwenig unterschiedliche Wettertage (grau in grau)	System laufen lassen und verschiedene Wettersituationen abwarten
Prognose schlecht II	Falscher Power-Sensor, falscher kWh Sensor	Der Powersensor muss zwingend die DC-Leistung der Solarzellen in echtzeit liefern. Der kWh (yield) Sensor muss auf dem Powersensor aufbauen.
Viele Samples aber kein ML	Die Samples weisen keine ausreichenden Unterschied in Wetter / Leistung auf	Die Anzahl der Samples zeigt nur an das das System korrekt funktioniert. 100 grau in grau Tages-Samples entsprechen 1 Learning-Sample(!!)
Warnung keine Prognose / kein Produktionsfenster / kein Astro-Cache, kein Wttr-Cache / kein Weather-Cache	Neuinstallation / Backup eingespielt	Das System repariert sich selbst und baut die Caches selbstständig wieder neu auf.

16. ? Häufig gestellte Fragen

Wie lange dauert es, bis die Prognose genau ist?

Das ML-Modell benötigt mindestens **14 Tage** an historischen Daten, um ein erstes verwertbares Modell zu erstellen. Nach **30 Tagen** sollte die Genauigkeit deutlich besser sein.

Funktioniert die Integration ohne Internet?

Nein. Die Integration benötigt eine Internetverbindung, um die Wetterdaten abzurufen.

Wie erstelle ich einen Utility Meter?

Navigieren Sie in Home Assistant zu **Einstellungen -> Geräte und Dienste -> Helfer** und wählen Sie **Utility Meter**. Verwenden Sie Ihren kumulativen Energie-Sensor und wählen Sie als Zyklus **Täglich**.