



Afbeelding 1: opstelling van het weerstation.

Meteoroloog met behulp van informatica

Door Konrad Strehlke V6 & Berend Lugtenaar V6

OSG West-Friesland

Vak: Informatica

Begeleider: F.P.Vonck

Datum: 11-02-2026

Inhoudsopgave

Voorwoord	2
Inleiding.....	3
Onderzoeksvraag en deelvragen	7
Hypothese.....	8
Materialen en methode.....	8
Materiaallijst.....	8
Hardware.....	8
Software	10
Resultaten.....	18
Discussie.....	21
Analyse van resultaten.....	21
Vergelijking met onze verwachting	22
Conclusie.....	22
Terugblik	22
Reflectie.....	23
Vervolgonderzoek.....	23
Literatuurlijst.....	23
Bijlagen	26
Logboek	30

Voorwoord

Voor ons profielwerkstuk wilde we onze eerdere kennis verbreden die we hebben opgedaan in de eerdere informatica lessen. Hierbij hebben we al eerder een weerwebsite gemaakt. Dit was erg leerzaam op het gebied van micro controllers en het gebruik van draadloze connecties, maar een echt weersstation konden we het niet noemen. Het nam metingen van de omgeving en liet deze zien op een website, alleen maakte het nog geen eigen voorspellingen. Bij dit profielwerkstuk willen we dit wel gaan doen. We gebruiken veel informatie afkomstig van het internet, verkregen met behulp van artificiële intelligentie zoals ChatGPT en Copilot, waarna het daarna gecontroleerd is door onszelf.

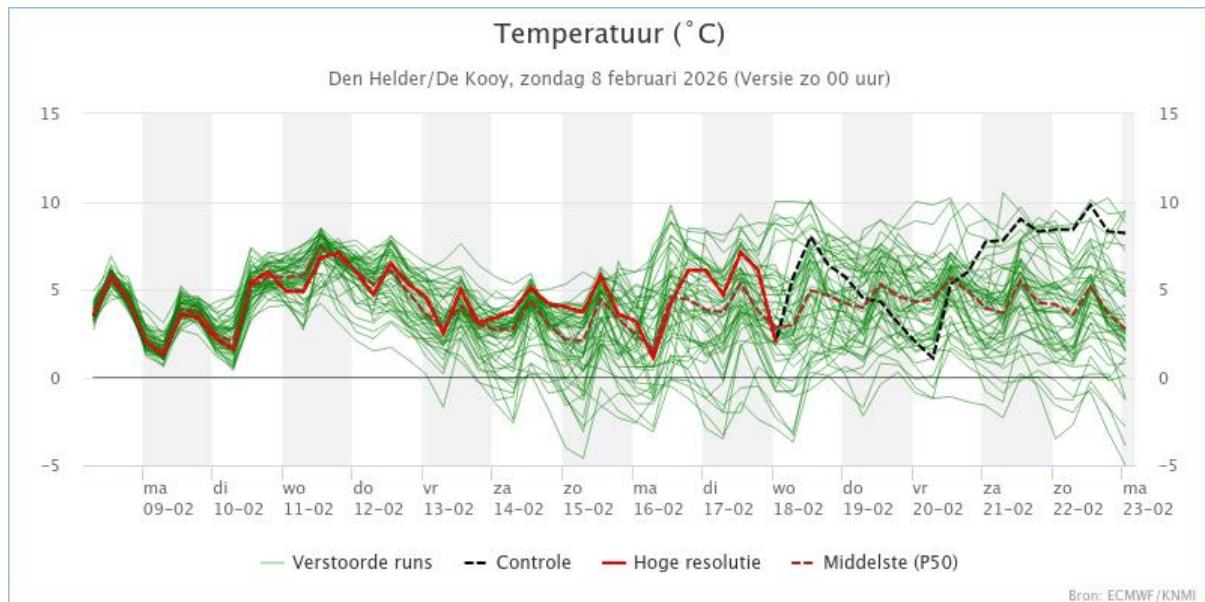
Inleiding

Weersvoorspellingen werden al van oudsher gedaan. Als de wolken donker worden gaat het misschien regenen, om een simpel voorbeeld te nemen. Op deze manier, door het herkennen van patronen, proberen instituten en organisaties steeds beter te worden om het weer te voorspellen. Deze voorspellingen zijn namelijk van groot belang voor veel bedrijven. Transport zoals luchtvaart en scheepvaart gebruiken het om gevaar en verlies te voorkomen en agro-industrieën hebben het nodig om juist hun winst te vergroten door op tactische momenten te planten of te oogsten.

Het weer verandert omdat het op aarde niet een constante temperatuur geldt. De aarde draait om haar as en wordt aan de kant van de zon warmer, terwijl de andere kant afkoelt. Floor (2001) legt uit in een online kennisnet dat wind ontstaat door luchtdruk verschillen, die op hun plek weer ontstaan door verschil in temperatuur. Wind is de verplaatsing van lucht en heeft als doel de plekken van lage druk en hoge druk gelijk te maken. Dit betekent dus dat als er lokaal een laag drukgebied is, ook wel een depressie genoemd, de lucht richting die positie zal verplaatsen (KNMI - Depressie, z.d.). Dit kan voor veel weer en wind zorgen. Wanneer twee verschillende luchtmassa's elkaar tegenkomen noem je die gebieden fronten. Een koud front en een warm front lopen daar in elkaar over en veroorzaken wolken of zelfs regenwolken wanneer warme lucht opstijgt en afkoelt. Waterdeeltjes zullen hierdoor condenseren tot wolken.

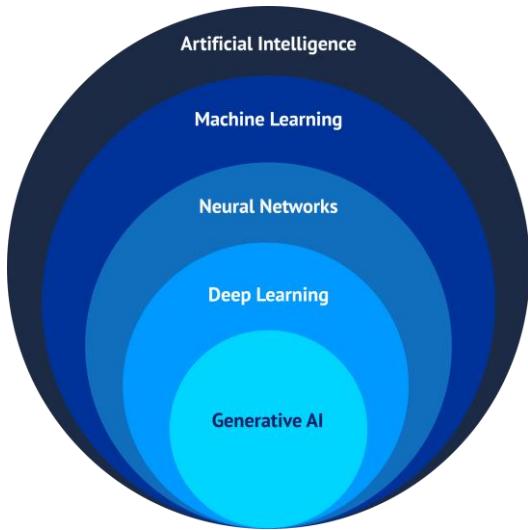
Het maken van weersvoorspellingen begint allemaal bij het verzamelen van data (Shetty, 2023). Zonder gegevens kun je niets. Dit gaat via weerstations op het land, boeien op het water en ballonnen en vliegtuigen in de atmosfeer. Al deze data moeten in modellen worden gestopt en geanalyseerd om te begrijpen wat het weer de komende tijd doet. Omdat de hoeveelheid data zo groot is gebruikt men hiervoor supercomputers die veel werkgeheugen hebben om alle wekenmodellen op tijd af

kunnen maken. Volgens NOSop3 volgt uit de supercomputer een pluimverwachting, een grafiek karakteristiek om de vorm van een pluim dat uitwaaiert naarmate de tijd verder wordt uitgedrukt in de afbeelding (KNMI - Weer- en klimaatpluim en Expertpluim, 2026). Dit is omdat verschillende modellen op basis van hun eigenschappen verschillende voorspellingen genereren. Doordat er zo veel factoren meespelen in het voorspellen van het weer en de natuur erg onvoorspelbaar van haarzelf is, ontstaat deze pluim.



Afbeelding 2: Pluimvoorspelling van de temperatuur op zondag 8 februari 2026. Bron: KNMI - Weer- en klimaatpluim en Expertpluim, 2026

Voor ons onderzoek gebruiken we daarentegen geen supercomputers en terabytes aan data om het weer te voorspellen, maar maken we gebruik van kunstmatige intelligentie (AI). AI is een benaming voor zelf denkende computer die menselijk werk zouden moeten kunnen verrichten, wat niet allemaal hetzelfde is. Tegenwoordig is de vorm Generatieve AI erg populair om neppe video's te genereren om op sociaal media veel likes te werven. Dit is een nieuwere, complexere vorm van AI, alleen gebruiken wij die niet voor ons onderzoek. Wij gebruiken namelijk Machine learning. Volgens Chen (2024) is het een techniek die voorheen onbekende relaties in gegevens ontdekte door potentieel zeer grote gegevenssets te doorzoeken om patronen en trends te ontdekken die verder gaan dan eenvoudige statistische analyse.



Afbeelding 3: Understanding the Technical Foundations of Generative AI. Bron: InfoWorks, 2025

Voor het maken van het model gebruiken we River, een online Machine learning programma die wordt integreert in vorm van een python library. Hierover zal meer worden uitgelegd in het hoofdstuk Software, pagina 10.

River werkt als een online regressie-architectuur die direct in de datastream wordt geplaatst, waardoor het model zich continu aanpast aan nieuwe sensorwaarden zonder ooit een aparte trainingsfase nodig te hebben. In tegenstelling tot klassiek *batch* learning, wat inhoudt dat het model van vaste datasets leert waarna het daarna statisch blijft, werkt River volgens het online learning-principe. Dit betekent dat het incrementeel leren uit één datastream. Elk nieuw data fungeert tegelijk als trainingsmateriaal en als update voor de interne modelparameters.

De integratie begint bij het opzetten van een pipeline (verwerkingsketen) waarin meerdere modelcomponenten sequentieel worden gekoppeld. In dit geval bestaat die keten uit een StandardScaler gevolgd door LinearRegression. De scaler verandert de schaalverdeling van de ingevoerde variabelen, wat essentieel is omdat lineaire regressie gevoelig is voor schaalverschillen tussen variabelen met verschillende grootheden, bijvoorbeeld bij luchtdruk rond 1000 hPa versus de temperatuur rond 20 °C. River doet dit via running statistics (doorlopende statistische berekening), waarbij gemiddelde en standaardafwijking voortdurend worden bijgewerkt zonder opslag van historische data. Dat is een vorm van constant-memory processing, wat betekent dat het werkt met een vaste geheugencapaciteit ongeacht de datagrootte.

Na normalisatie bereikt de data het regressiemodel, dat in River meestal gebaseerd is op Stochastic Gradient Descent, afgekort SGD, een iteratieve optimalisatiemethode. Hierbij wordt na elk datapunt een kleine correctie toegepast

op de modelgewichten volgens de afgeleide van de verliesfunctie (loss function, maat voor voorspellingsfout). Het model probeert dus continu de fout tussen voorspelde temperatuur en gemeten temperatuur te minimaliseren. In plaats van één grote parameterupdate zoals bij batch learning, voert River duizenden mini-updates uit, wat leidt tot een adaptief model dat zich kan aanpassen aan *concept drift* (verandering in onderliggende datapatronen, bv. seizoensverloop of weersomslag).

Dit gebeurt in ons script op het moment dat een JSON-regel van de ESP32 wordt ingelezen. Zodra een nieuw datapunt wordt ontleed, wordt een feature vector (gestructureerde invoer) opgebouwd met temperatuur, vochtigheid en druk. Deze vector gaat via `model.learn_one(x, y)` het systeem in. Dat commando voert drie dingen uit: normaliseren van features, berekenen van voorspelfout, en bijstellen van gewichten. Het model “onthoudt” dus geen ruwe data, maar alleen de geoptimaliseerde parameters en de lopende statistieken — dit heet een parametrische representatie (kennis opgeslagen in gewichten in plaats van voorbeelden).

Wanneer later een voorspelling wordt gedaan via `model.predict_one(x)`, wordt dezelfde transformatieketen toegepast, maar zonder gewichtsupdate. Het model produceert dan een regressiewaarde op basis van de huidige parameters. In ons 48-uurs voorspelling wordt dit iteratief gedaan in een recursive forecasting loop (voorspelling wordt de input voor de volgende stap), wat typisch is voor tijdreeksmodellen zonder expliciet tijdscomponent.

Samengevat functioneert River hier als een real-time adaptief leersysteem dat opereert op streaming sensorinput, gebruikmaakt van online normalisatie, gewichtsoptimalisatie via SGD, en geheugen-efficiënte parameteropslag.

Het resultaat is een zelflerend voorspellingssysteem dat zonder her training of datasetbeheer autonoom meegroeit met nieuwe meetgegevens en structurele veranderingen in het systeemgedrag kan volgen.

Om gegevens te meten is de gecombineerde sensormodule zoals de BMP28+AHT20 een handig component. Deze meet namelijk de temperatuur, luchtdruk en luchtvochtigheid.

De AHT20 is de sensor voor de temperatuur en de relatieve luchtvochtigheid. Aan boord heeft het een interne capacitieve vochtigheidssensor, wat door middel van verandering van elektrische capaciteit vocht meet, en een halfgeleider-temperatuursensor. Deze sensor heeft een interne fabriekskalibratie en een I²C-interface, wat betekent dat er stabiele digitale waarden worden verstuurd zonder analoge ruis zoals elektrische storingen op spanningssignalen. De AHT20 reageert

relatief snel op veranderingen in luchtvochtigheid, wat erg goed is voor nauwkeurigheid, maar hij is gevoelig voor thermische beïnvloeding door zowel de eigen elektronica als de zoninstraling. Daarom hebben we het in een witte behuizing in de schaduw geplaatst zodat de metingen representatief blijven voor de omgevingsslucht en niet voor de behuizingstemperatuur.

De BMP280 vult dit aan met luchtdrukmetingen via een piëzoresistieve drucksensor. Deze meet de luchtdruk door middel van een verandering in elektrische weerstand naarmate er meer vervorming op een membraan wordt uitgeoefend door de lucht. Deze sensor heeft een hoge resolutie en lage ruis, waardoor we met ons weerstation zelfs kleine drukveranderingen kunnen detecteren. Dat is meteorologisch interessant, want luchtdruktrend (stijgend of dalend) is vaak een sterkere indicator voor naderende weersverandering dan temperatuur alleen. Omdat temperatuur de drucksensor fysiek beïnvloedt, bevat de BMP280 ook een interne temperatuurmeting voor compensatie (temperatuurcorrectie van drukwaarden), wat de drukmetingen consistenteer maakt.

Door beide sensoren samen te gebruiken, meet ons weerstation drie kernparameters van de atmosfeer: temperatuur, relatieve vochtigheid en luchtdruk. Fysisch gezien kunnen we hieruit ook afgeleide grootheden berekenen, zoals het dauwpunt (temperatuur waarbij condensatie optreedt) en absolute vochtigheid (werkelijke hoeveelheid waterdamp per volume lucht). Dat betekent dat ons systeem niet alleen ruwe sensordata verzamelt, maar een volledige thermodynamische toestand van de lokale lucht zou kunnen beschrijven, maar omdat dat geen essentiële informatie voor de voorspellingen zijn hebben we ervoor gekozen om dit niet te doen. We wisten niet hoelang dit project ging duren, maar we waren wel van plan om ooit ook een online broadcast van actuele data en voorspellingen te maken. In dat geval zouden deze berekeningen wel handig zijn geweest.

Kort gezegd, in ons weerstation fungeert de AHT20 als thermisch-hygrometrische meeteenheid (temperatuur + vocht), terwijl de BMP280 de barometrische component levert (luchtdruk). Samen vormen ze een compacte, energieuze en meteorologisch zinvolle sensormodule die perfect past bij een data-streaming en online-ML systeem zoals wij gebruiken.

Onderzoeksraag en deelvragen

1. **In hoeverre is het mogelijk om een voorspellend weerstation te maken?**
2. Hoe betrouwbaar zijn de geproduceerde voorspellingen
3. Hoe vaak kloppen de voorspellingen in vergelijking met andere weer websites
4. Is het in staat houden en onderhouden voor particulieren te doen?

Hypothese

Het is mogelijk om het weer net zo accuraat te voorspellen als het landelijke meteorologisch instituut zonder hun technologie.

Wij denken dat onze voorspellingen door onze AI te trainen met genoeg data en de code te kalibreren we in ieder geval enkele juiste voorspellingen kunnen maken. Ook al hebben we niet dezelfde hoeveelheid data of dezelfde rekenmodellen zou het bij benadering mogelijk moeten zijn doordat het voorspellen een kwestie is van logica. Hoe nauwkeurig het zal zijn zullen we via het onderzoek achter komen.

Materiaal en methode

Materiaallijst

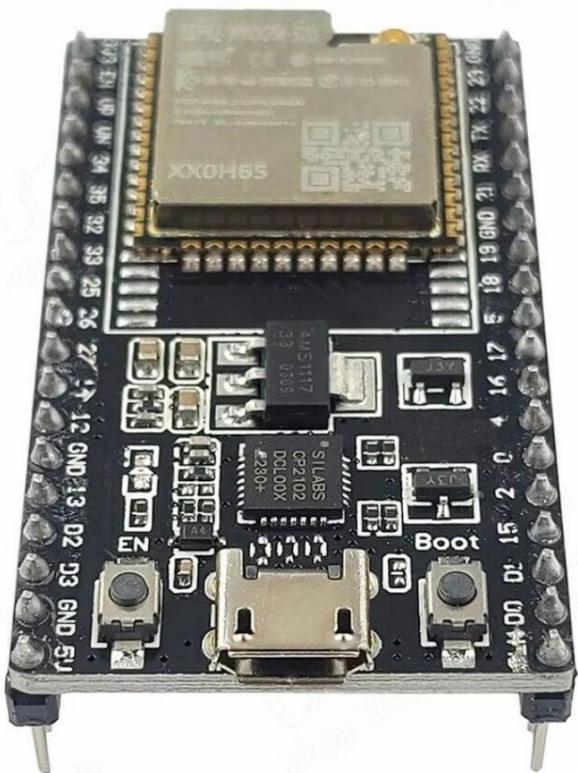
- ESP32-WROOM-32U
- AHT20+BMP280 sensor
- Jumpercables (female to female)
- Pins (soldeert op de AHT20+BMP280)
- Voedingskabel (USB-A naar micro USB)
- Wifi modem
- Laptop met software

Hardware

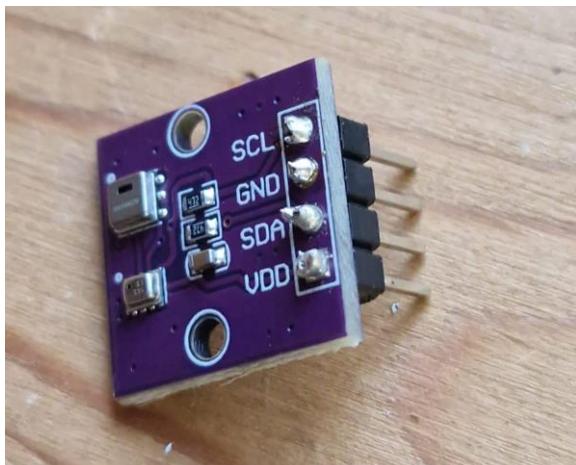
We hebben voor een ESP32 gekozen om een eenvoudige reden: we wilden de weerstation relatief autonoom maken, zodat je hem bijna overal kon neerzetten, zolang hij maar van stroom werd voorzien. De ESP32 heeft de mogelijkheid om gegevens via bluetooth en wifi te verzenden en ook te ontvangen. Dat is ons ook gelukt, maar niet op een stabiele manier, maar daarover meer op pagina 25 onder conclusie. Daarom hadden we ook zonder problemen met vergelijkbare microcontrollers kunnen werken, ook al hebben deze geen ingebouwde draadloze communicatiemogelijkheden, zoals bijvoorbeeld een Arduino Nano of een Raspberry Pi Pico.

Het verzamelen van informatie doen wij door middel van een sensor verbonden aan een ESP32. Met deze gegevens maken we de weersvoorspellingen met onze ML. In ons weerstation gebruiken we een sensorcombinatie van de AHT20 en de BMP280. Door beide sensoren samen te gebruiken, meet ons weerstation drie kernparameters

van de atmosfeer: temperatuur, relatieve vochtigheid en luchtdruk. Fysisch gezien kunnen we hieruit ook afgeleide grootheden berekenen, zoals het dauwpunt (temperatuur waarbij condensatie optreedt) en absolute vochtigheid (werkelijke hoeveelheid waterdamp per volume lucht). Dat betekent dat ons systeem niet alleen ruwe sensordata verzamelt, maar een volledige thermodynamische toestand van de lokale lucht zou kunnen beschrijven, maar omdat dat geen essentiële informatie voor de voorspellingen zijn hebben we ervoor gekozen om dit niet te doen. We wisten niet hoelang dit project ging duren, maar we waren wel van plan om ooit ook een online broadcast van actuele data en voorspellingen te maken. In dat geval zouden deze berekeningen wel handig zijn geweest.



Afbeelding 4: Close up van de ESP32 micro controller. Bron: DirenC, 2025



Afbeelding 5: Close up van de BMP280+AHT20. Bron: Eigen werk

Software

Hier zijn de meest recente en, vanwege problemen met mijn router, ook de eenvoudigste versies van de software die we hebben gebruikt. Met deze software zijn de meest recente metingen uitgevoerd. Als u geïnteresseerd bent, kunt u de oude versies van de code vinden op onze GitHub: <https://github.com/D-A-R-N-O-K/PWS-Weerstation/tree/main/code>.

```
// Inclusie van benodigde bibliotheken
#include <Arduino.h> // Standaard Arduino
bibliotheek
#include <Wire.h> // Voor I2C communicatie met
sensoren
#include <Adafruit_BMP280.h> // Bibliotheek voor BMP280
luchtdruk- en temperatuursensor
#include <Adafruit_AHTX0.h> // Bibliotheek voor AHT20
temperatuur- en luchtvochtigheidssensor

// Initialiseer sensor objecten
Adafruit_BMP280 bmp; // Object voor BMP280 sensor
Adafruit_AHTX0 aht; // Object voor AHT20 sensor

void setup() {
    // Start seriële communicatie met 115200 baud
    Serial.begin(115200);

    // Initialiseer sensoren en controleer of ze gevonden
    zijn
    if (!bmp.begin()) Serial.println("BMP280 niet
    gevonden!");
    if (!aht.begin()) Serial.println("AHT niet gevonden!");
}
```

```

void loop() {
    // Variabelen voor sensorgegevens
    sensors_event_t humidity, temp;

    // Lees temperatuur en luchtvochtigheid van AHT20 sensor
    aht.getEvent(&humidity, &temp);

    // Lees luchtdruk van BMP280 sensor en converteer naar
    hPa
    float pressure = bmp.readPressure() / 100.0F; // hPa
    float temperature = temp.temperature;           // Temperatuur in °C
    float hum = humidity.relative_humidity;         // Luchtvochtigheid in %

    // Verzend sensorgegevens als JSON via seriële poort
    Serial.print("{\"temperature\":");
    Serial.print(temperature);
    Serial.print(", \"humidity\":");
    Serial.print(hum);
    Serial.print(", \"pressure\":");
    Serial.print(pressure);
    Serial.println("}");

    // Wacht 1 minuut voor de volgende meting (60 seconden *
    1000 milliseconden)
    delay(60 * 1000);
}

```

Afbeelding 6: Dit is de code die op de ESP32 zit. De link voor de github van deze code: <https://github.com/D-A-R-N-O-K/PWS-Weerstation/blob/main/code/weerespV2.ino> en hier de link met de uitleg: <https://github.com/D-A-R-N-O-K/PWS-Weerstation/blob/main/code/uitlegWeerespV2.ino>. Bron: Eigen werk.

Dit programma draait op de ESP32 en fungeert als een eenvoudige methode voor omgevingsmetingen. Na het laden van de benodigde bibliotheken voor I2C-communicatie en de twee sensoren, de BMP280 voor luchtdruk en de AHT20 voor temperatuur en relatieve luchtvochtigheid, worden in de globale scope sensorobjecten aangemaakt die de hardware abstraheren.

Tijdens de setupfase wordt eerst de seriële communicatie geïnitialiseerd op 115200 baud, zodat de microcontroller data kan uitsturen naar de server/laptop. Daarna probeert de code beide sensoren via de I2C-bus te initialiseren. Als een sensor niet reageert op zijn adres, wordt dit direct gemeld via de seriële monitor. Dit is een eenvoudige hardware-check (device detection) om te verifiëren dat de meetketen operationeel is.

In de ‘loop’, het herhalende deel, doorloopt het programma steeds hetzelfde meetproces. Eerst worden datastructuren klaargezet om sensorwaarden te ontvangen. De AHT20 levert temperatuur en luchtvuchtigheid als zogenaamde sensor events. Vervolgens wordt de luchtdruk uit de BMP280 uitgelezen. Deze waarde wordt omgerekend van Pascal naar hectopascal (hPa), wat gebruikelijk is in meteorologische toepassingen. De temperatuurwaarde wordt uit het AHT-event gehaald (in graden Celsius) en de relatieve luchtvuchtigheid in procenten.

Daarna worden alle meetwaarden samengevoegd tot één JSON-object en via de seriële poort verstuurd. Dit betekent dat de ESP32 zich gedraagt als een dataprovider die gestructureerde, machine-leesbare output levert voor ons Python-script op de server die de seriële data inleest. Tot slot wacht het systeem exact één minuut voordat de volgende meetcyclus start, waardoor er een vaste sample-interval van 60 seconden ontstaat. Het geheel vormt dus een periodiek meetsysteem dat sensordata verzamelt, structureert en extern beschikbaar maakt.

```
import serial
import json
from flask import Flask, render_template_string
from river import linear_model, preprocessing
from datetime import datetime, timedelta
import math

app = Flask(__name__)
data = []

model = preprocessing.StandardScaler() |
linear_model.LinearRegression()

ser = serial.Serial('/dev/ttyACM0', 115200, timeout=1)

def read_from_esp():
    while True:
        try:
            line = ser.readline().decode('utf-8',
errors='ignore').strip()
            if not line:
                continue
            if line.startswith("{") and
line.endswith("}"):
                try:
                    d = json.loads(line)
                    d["timestamp"] = datetime.now()
                    data.append(d)

                    x = {"temp": d["temperature"],
"humidity": d["humidity"], "pressure": d["pressure"]}

```

```

        model.learn_one(x, d["temperature"])
        print(f"[{datetime.now()}] Received:
{d}")
    except json.JSONDecodeError:
        print(f"Ungültiges JSON: {line}")
    except Exception as e:
        print(f" Fehler in Serial-Thread: {e}")

import threading
esp_thread = threading.Thread(target=read_from_esp,
daemon=True)
esp_thread.start()

@app.route("/")
def index():
    if not data:
        return "Nog geen data"

    last = data[-1]
    predictions = []
    x = {"temp": last["temperature"], "humidity": last["humidity"],
"pressure": last["pressure"]}
    now = datetime.now()
    for i in range(48):
        pred_temp = model.predict_one(x)
        predictions.append((now + timedelta(hours=i+1),
pred_temp))
        x["temp"] = pred_temp

    html = """
<!DOCTYPE html>
<html lang="nl">
<head>
<meta charset="UTF-8">
<title>Weerstation Zwaag</title>

<style>
:root {
    --bg: #0f172a;
    --card: #111827;
    --text: #e5e7eb;
    --muted: #9ca3af;
    --accent: #38bdf8;
    --border: #1f2933;
}
* { box-sizing: border-box; }

```

```
body {
    margin: 0;
    padding: 24px;
    font-family: system-ui, -apple-system,
BlinkMacSystemFont, "Segoe UI", sans-serif;
    background: var(--bg);
    color: var(--text);
}

h1 { margin-bottom: 4px; }
h2 { margin-top: 0; }

.subtitle {
    color: var(--muted);
    margin-bottom: 24px;
}

.card {
    background: var(--card);
    border: 1px solid var(--border);
    border-radius: 12px;
    padding: 16px 20px;
    margin-bottom: 24px;
}

.metrics {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(160px,
1fr));
    gap: 16px;
}

.metric {
    background: #020617;
    border-radius: 10px;
    padding: 12px;
}

.metric span {
    display: block;
    font-size: 0.85rem;
    color: var(--muted);
}

.metric strong {
    font-size: 1.4rem;
}

table {
```

```

        width: 100%;
        border-collapse: collapse;
        margin-top: 12px;
    }

th, td {
    padding: 10px 12px;
    text-align: left;
}

th {
    font-weight: 600;
    color: var(--muted);
    border-bottom: 1px solid var(--border);
}

td {
    border-bottom: 1px solid var(--border);
}

tr:hover {
    background: rgba(56, 189, 248, 0.05);
}

.temp {
    color: var(--accent);
    font-weight: 600;
}
</style>
</head>

<body>

<h1>Weerstation Zwaag</h1>
<div class="subtitle">Stand: {{ now }}</div>

<div class="card">
    <h2>Actuele Meetingen</h2>
    <div class="metrics">
        <div class="metric">
            <span>Temperatuur</span>
            <strong>{{ temp }} °C</strong>
        </div>
        <div class="metric">
            <span>Luchtvochtigheid</span>
            <strong>{{ hum }} %</strong>
        </div>
        <div class="metric">
            <span>Luchtdruk</span>

```

```

        <strong>{{ pres }} hPa</strong>
    </div>
</div>

<div class="card">
    <h2>Voorprognose (48uur)</h2>
    <table>
        <tr>
            <th>Tijd</th>
            <th>Temperatuur</th>
        </tr>
        {% for t, p in pred %}
        <tr>
            <td>{{ t }}</td>
            <td class="temp">{{ p|round(2) }} °C</td>
        </tr>
        {% endfor %}
    </table>
</div>

</body>
</html>
"""
    return render_template_string(html,
temp=last["temperature"],
                                         hum=last["humidity"],
                                         pres=last["pressure"],
                                         pred=predictions,
                                         now=now.strftime("%d-%m-
%Y %H:%M"))
if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)

```

Afbeelding 7: Dit is de code voor de server. # worden herkend als een titel, maar in een .py bestand een commentaar, heb ik hier de versie zonder uitleg neergezet. Hier de link naar de code: <https://github.com/D-A-R-N-O-K/PWS-Weerstation/blob/main/code/mlserverV3.py> en hier de link naar de code met de uitleg: <https://github.com/D-A-R-N-O-K/PWS-Weerstation/blob/main/code/uitlegMLServerV3.py>. Bron: Eigen werk.

Dit programma vormt de softwarelaag boven op het ESP32 en functioneert als een combinatie van datacollector, online machine-learning pipeline en web dashboard.

Na het importeren van bibliotheken voor seriële communicatie, JSON-verwerking, tijdsfuncties en math wordt een Flask-webserver geïnitialiseerd. Daarnaast wordt een lege lijst data aangemaakt die dient als tijdelijke opslag (in-memory buffer) van alle

binnenkomende meetwaarden. Vervolgens wordt een online learning model opgebouwd met de River-bibliotheek: een StandardScaler voor feature-normalisatie gekoppeld aan een LinearRegression model. Dit model leert continu bij terwijl er nieuwe sensordata binnenkomt (incremental learning).

Daarna opent het programma een seriële verbinding met de ESP32 via /dev/ttyACM0 op 115200 baud (dus via een van de USB-A ports van mijn laptop/server. De functie `read_from_esp()` draait in een aparte achtergrondthread en vormt de datastream-ingang van het systeem. Deze thread leest continu regels van de seriële poort, filtert op geldige JSON-berichten en zet ze om naar Python-dicts. Aan elk datapunt wordt een timestamp toegevoegd en het geheel wordt opgeslagen in de data-lijst.

Tegelijkertijd wordt het machine-learning model direct getraind met de nieuwe meting: temperatuur, luchtvochtigheid en luchtdruk zijn de inputfeatures, terwijl temperatuur ook de doelvariabele is. Het model past zich dus realtime aan veranderende omstandigheden aan.

De Flask-route vormt de presentatielaag. Wanneer een gebruiker de webpagina opent, haalt de server de meest recente meting op. Op basis daarvan wordt een 48-uurs temperatuurvoorspelling gegenereerd met een uurlijkse resolutie. Dit gebeurt iteratief: de voorspelde temperatuur van elk uur wordt opnieuw als invoer gebruikt voor de volgende stap, wat neerkomt op een eenvoudige autoregressieve forecast.

Ten slotte wordt een HTML-pagina dynamisch gerenderd met de actuele sensorgegevens en een tabel met voorspellingen. De web interface fungeert dus als een realtime monitoring dashboard voor het weerstation, terwijl op de achtergrond dataverzameling, modeltraining en voorspelling continu doorgaan. Het totale systeem is daarmee een end-to-end pipeline:

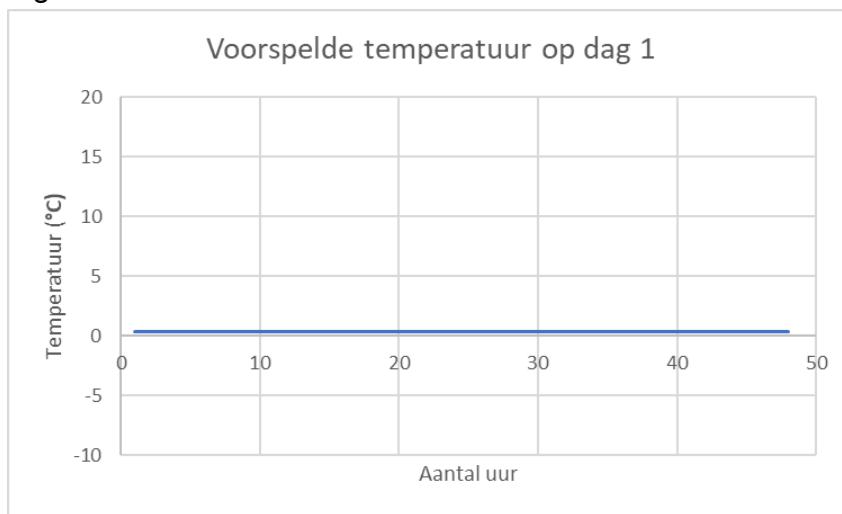
1. Sensor
2. Seriële data-ingestie
3. Online modeltraining
4. Voorspelling
5. Web visualisatie

Het is ook belangrijk dat men de Python code in een map plaatst die een actieve virtual environment heeft en dat men natuurlijk Python (-3) en de benodigde bibliotheken heeft gedownload voordat men het programma start. In de code zelf moet men soms de seriële poort en natuurlijk de locatie veranderen.

Resultaten

Niet alle pogingen waren succesvol dus hebben we niet veel resultaten, maar ons weerstation heeft wel enkele uren buiten doorgebracht om data te verzamelen en heeft daarmee ook de AI beïnvloed. Dat is te zien door de eerste data te vergelijken met de laatste data. Op de eerst gemeten dag, drie februari, was het 16,56°C, een luchtvochtigheid van 60,3% en een luchtdruk van 999,93 hectopascal (hPa of ook wel millibar genoemd). Ook voorspelde het dat de komende 48 uur een constante temperatuur zou houden van 0,33°C. Deze lijn is weergeven in figuur 1.

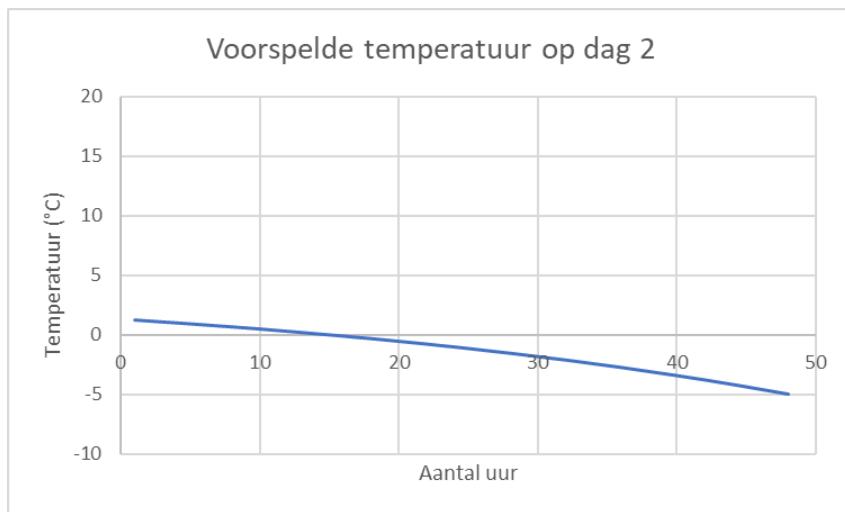
Figuur 1



Figuur 1: Voorspelde temperatuur op dag een (03-02-2026 17:29) van de volgende 48 uur

24 uur later, op vier Februari, meet het station een temperatuur van 1,29°C en voorspelt het voor de komende 48 uur een toenemend dalende temperatuur, totdat het stopt bij –5 graden Celsius. Dit is al meer beweging in het model dan de vorige dag. Verder is de lucht vochtigheid 82% gemeten en luchtdruk van 1001,0 millibar.

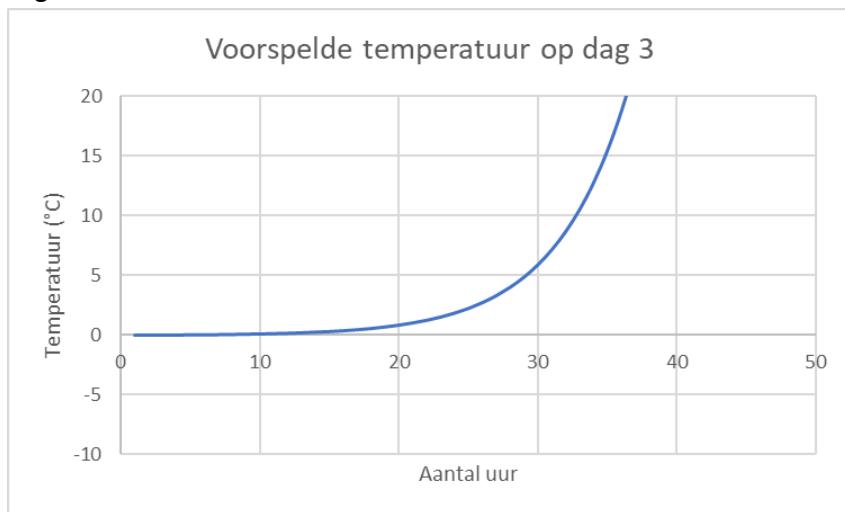
Figuur 2



Figuur 2: Voorspelde temperatuur op dag twee (04-02-2026 17:30) van de volgende 48 uur

Nu hoopte we een verbetering te gaan zien en dat het steeds een beter beeld geeft van hoe het weer kan gaan lopen, alleen is dat op dag drie nog steeds niet aan de hand. Een temperatuur van 0°C geeft het aan dat de komende 48 de temperatuur vanaf de 0°C een duidelijke curve naar boven, zoals te zien in grafiek 3. De grafiek geeft alleen niet aan waar deze eindigt omdat de assen in elke grafiek dezelfde grootte zijn. De curve stopt echter bij een temperatuur van 188°C wat ver boven de normale temperatuur ligt in Nederland.

Figuur 3

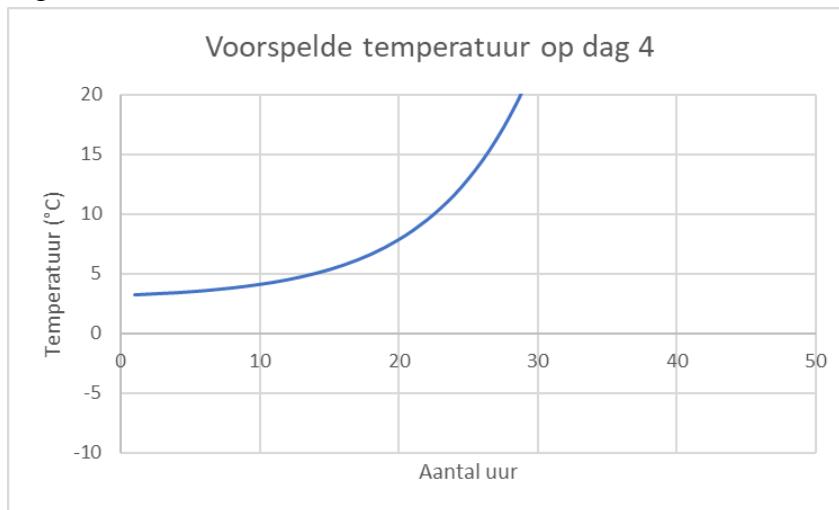


Figuur 3: Voorspelde temperatuur op dag drie (05-02-2026 17:30) van de volgende 48 uur

De curve loopt zo omdat het model waarschijnlijk in de war raakt en nog steeds geen goede correlatie heeft kunnen vinden tussen de eerdere temperatuur. Dit probleem is op dag vier nog steeds niet opgelost. Hier is wederom een curve die de lucht in

schiet vanaf de gemeten 3,3°C tot wel 260°C. De curve start dus wel op de juiste temperatuur, maar eindigt daar verre van.

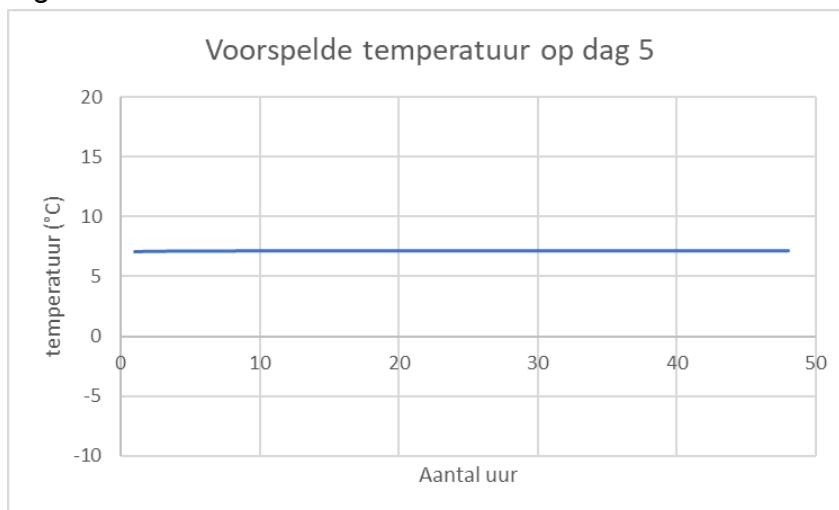
Figuur 4



Figuur 4: Voorspelde temperatuur op dag vier (06-02-2026 17:30) van de volgende 48 uur.

Op dag vijf leek dit probleem over. De voorspelde temperatuur schoot niet meer tot de honderden graden Celsius boven het vriespunt uit. Wel bleef de temperatuur zoals in het begin van onze testen rond een constante waarde. Deze bewoog voor de eerste 3 voorspelde uren nog in de eerste decimaal, maar dit is amper te merken uit grafiek 5. Dit heeft wederom te maken met de schaalverdeling, maar zou wanneer het een goede voorspelling weergeeft wel merkbaar moeten zijn.

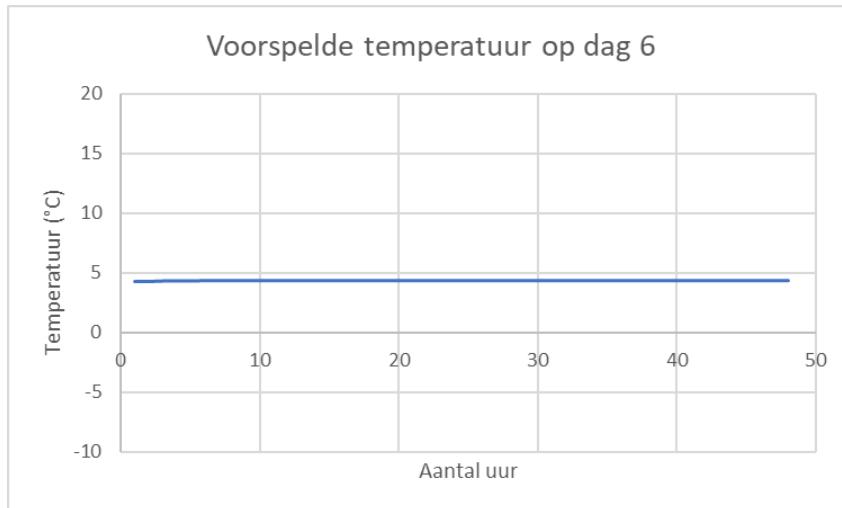
Figuur 5



Figuur 5: Voorspelde temperatuur op dag vijf (07-02-2026 17:31) van de volgende 48 uur.

Als laatste geeft dag zes alweer hetzelfde als de vorige dag, een nauwelijks veranderende lijn. Net zoals op dag vijf beweegt de lijn in de eerste voorspelde uren wat, maar weer niet merkbaar. Zo lijkt grafiek 6 ongeveer hetzelfde als grafiek 1 van dag een.

Figuur 6



Figuur 6: Voorspelde temperatuur op dag zes (08-02-2026 17:31) van de volgende 48 uur.

Om de resultaten af te sluiten zijn in tabel 1 de gemeten waarden te zien van de dagen uit de grafieken. De startwaarden van de voorspellingen en het eerst volgende uur is dan wel deels correct, alleen is dit natuurlijk geen representatief model voor betrouwbare voorspellingen.

Tabel 1

Datum	Temperatuur	Luchtvochtigheid	Luchtdruk
03-02-2026 17:29	16,56°C	60,29%	999.9 hPa
04-02-2026 17:30	1,29°C	82.58 %	1001.0 hPa
05-02-2026 17:30	-0,06°C	78.57 %	994.5 hPa
06-02-2026 17:30	3,32°C	80.46 %	995.6 hPa
07-02-2026 17:31	7,15°C	79.03 %	1006.5 hPa
08-02-2026 17:31	4,43°C	82.26 %	1012.4 hPa

Tabel 1: meetgegevens van ons weerstation, elke dag om half 6

Discussie

Analyse van resultaten

De resultaten waren iets minder dan we hadden gehoopt maar daar is nu niets meer aan te veranderen. Ten eerste is de gemeten temperatuur van dag een veel te hoog

voor die dag, wel 16,56 graden Celsius. Dat was veel hoger dan het landelijke gemiddelde van 0,1°C (KNMI - Daggegevens Van Het Weer in Nederland, z.d.). Dit zou een technische fout kunnen zijn, zoals dat het een verkeerd doorgegeven waarde was. Ook zou de sensor nog warm kunnen zijn van het coderen in huis, of veroorzaakt zijn doordat op dat moment de zon geconcentreerd gericht stond op de behuizing, wat ervoor zorgde dat de temperatuur hoger was. Verder is er weinig te zeggen over de luchtvochtigheid en luchtdruk, want ook al worden deze gegevens wel meegenomen in de voorspelling, voorspellen we alleen de temperatuur van de komende 48 uur.

Vergelijking met onze verwachting

Onze verwachtingen waren een stuk hoger dan wat we uiteindelijk hebben kunnen aantonen. Hoewel het bouwen van het weerstation wel gelukt is, kloppen de voorspellingen niet en zien ze er ook niet reëel uit, in tegenstelling tot onze verwachting van een vloeiende, golvende lijn zoals de pluimverwachting van het KNMI. Hoewel het model nog niet heel lang actief is hadden we wel verwacht dat het aan de hand van de gemeten gegevens steeds meer zou gaan lijken op een dag- en nachtcyclus, wat het ook niet heeft bereikt.

Conclusie

Op basis van het onderzoek is dus de vraag of een voorspellend weerstation mogelijk is om te maken negatief, ook wel, niet mogelijk. Alleen denken wij wel dat het onderzoek wat we verricht hebben niet voldoende is om de vraag te beantwoorden. Uit het onderzoek bleek wel dat het model zich aanpast naarmate het meer data tot zijn beschikking had. Hierdoor blijft het idee dat er een tijd zal zijn waar het model wel genoeg data heeft gekregen dat het wel op de pluimverwachtingen begint te lijken. De eerste deelvraag: is het betrouwbaar, word dan ook meteen beantwoord, namelijk ook negatief. De temperatuur van de komende tijd lijkt niet op dat van ons model en is dus ook niet betrouwbaar. Men zou kunnen zeggen dat de tweede deelvraag wel een deel gelukt is, aangezien de voorspelling wel klopt voor de eerste paar minuten na de voorspelling alleen zal het toch echt kouder worden in de nacht en overdag meestal weer opwarmen. Dit is niet terug te zien en komt dus ook niet overeen met de weersvoorspelling van dezelfde dag, maar, misschien komt dat wel ook door een gebrek aan data. Dit is terug te leidden tot deelvraag drie, het onderhouden van deze weerstation was relatief moeilijk, tenminste voor ons. Het duurde relatief lang om ze überhaupt te bouwen, en door het automatische veranderen van de IP-adres in de router kon de server geen data meer ontvangen. Dit gebeurde toen we op vakantie waren (kerstvakantie) en dat is ook de reden hoezo we alleen maar de resultaten van een relatief korte tijd konden onderzoeken.

Terugblik

Bij het evalueren van de hypothese komen we op het antwoord: nee het is niet mogelijk om zonder dezelfde technologie als het KNMI of met alleen maar één weerstation een correct weersvoorspelling te maken. Dit zou kunnen komen door het gebrek aan data of kunnen komen omdat het in het algemeen praktisch nog niet mogelijk is. In theorie zou het mogelijk moeten zijn, alleen zullen we nooit kunnen zeggen dat het mogelijk is zonder dat iemand het heeft bewezen. Met ons onderzoek is dit ons niet gelukt.

Reflectie

Om terug te reflecteren naar hoe we het onderzoek starten, waren toen nog wat enthousiaster. Een plan en een idee wat we samen zouden gaan aanpakken en waar we al een beetje kennis in hadden, wat alleen maar zou groeien. We hebben veel geleerd op het gebied van machine learning, het verbinden van de micro controller via draadloze verbinding over het Wi-Fi netwerk, en over het onderzoeken in het algemeen. Niet alle onderzochte bladen en websites waren namelijk even handig. Het ontwerpen en bouwen van het weerstation ging wel met enige veranderingen en tegenslagen. In het begin waren we van plan een complexer weerstation te maken, met een windvaan en een anemometer om de windsnelheid te meten. Alleen hebben we deze weggelaten om het simpeler te houden. Dit was misschien niet handig geweest aangezien het wel meer data is wat het model positief zou kunnen beïnvloeden, maar daarmee ook meer werk en ook mogelijk moeilijker voor het model om een correlatie te vinden die past bij de realiteit. Wel was ons onderzoek zo een begin naar een nieuwe manier van voorspellen, namelijk niet op actuele data zoals meteorologische instituten rond de wereld dat doen, maar met AI. Dit idee kan nog controversieel worden opgevat aangezien niet iedereen de kunstmatige intelligentie accepteert om de morele reden dat het de aarde belast en veel datacenters kost. Links om of rechtsom zullen beide manieren veel computerkracht nodig hebben, dus uiteindelijk zullen beide manieren hetzelfde kosten, maar mogelijk andere uitkomsten leveren.

Vervolgonderzoek

Omdat we niet genoeg tijd hadden om meer dan alleen de winter te meten zal een vervolgonderzoek zijn om ook de andere seizoenen mee te nemen. Zo kan het model ook de seizoenen beter voorspellen als deze ooit geweest zijn, om toe te voegen aan de data. Zo krijgt men een beter beeld bij het gebruik van een online machine learning model dat misschien in de toekomst een toepassing vindt, mogelijkerwijs niet als weerstation maar voor een andere functie.

Literatuurlijst

Chen, M. (2024, 25 november). *What Is Machine Learning?*

Geraadpleegd op: 10-12-2025. Geraadpleegd van:

<https://www.oracle.com/nl/artificial-intelligence/machine-learning/what-is-machine-learning/>

Direnc.net®. (2025, 1 maart). [Close ESP32-WROOM32U Direnc.net]. Geraadpleegd op:

12-02-2026. Geraadpleegd van: <https://www.direnc.net/esp32-wroom-32u-wifi-bluetooth-gelistirme-modulu-en>

The Fellowship of Online Machine Learning. (z.d.). *river - River*. <https://riverml.xyz/latest/>

Floor, K. (2004, 1 januari). *Weerkunde, Meteorologie voor iedereen: Luchtdruk en wind*.

NEMOKennislink. Geraadpleegd op: 12-12-2025, Geraadpleegd van:

<https://www.nemokennislink.nl/publicaties/weerkunde-meteorologie-voor-iedereen-luchtdruk-en-wind/>

Hartman, P. (2025, 3 januari). InfoWorks. [Understanding the Technical Foundations of

Generative AI] *InfoWorks*. Geraadpleegd op: 10-02-2026. Geraadpleegd van

<https://www.infoworks-tn.com/resources/understanding-the-technical-foundations-of-generative-ai/>

Interfacing BMP280 with ESP32 on I2C with errors and solutions – Circuit Schools. (2020,

28 september). <https://www.circuitschools.com/interfacing-bmp280-with-esp-32-on-i2c-with-errors-and-solutions/>

KNMI - Daggegevens van het weer in Nederland. (2026, 8 februari). Geraadpleegd op: 08-

02-2026. Geraagpleegd van: <https://www.knmi.nl/nederland-nu/klimatologie/daggegevens>

KNMI - lagedrukgebied. (z.d.). Geraadpleegd op: 12-12-2025. Geraadpleegd van:

<https://www.knmi.nl/kennis-en-datacentrum/uitleg/lagedrukgebied>

KNMI - luchtdruk. (z.d.). Geraadpleegd op 12-12-2025. Geraadpleegd van:

<https://www.knmi.nl/kennis-en-datacentrum/uitleg/luchtdruk>

KNMI - Weer- en klimaatpluim en Expertpluim. (2026, 8 februari). [Pluimvoorspelling van de temperatuur op zondag 8 februari 2026]. Geraadpleegd op: (08-02-2026).

Geraadpleegd van: <https://www.knmi.nl/nederland-nu/weer/waarschuwingen-en-verwachtingen/weer-en-klimaatpluim>

Leon Renner. (2023, oktober 20). *MS Word Create style for code snippet box*  2

MINUTES [Video]. YouTube. Geraadpleegd op: 08-02-2026. Geraagpleegd van:

<https://www.youtube.com/watch?v=busnzKKSOxU>

NOSop3. Hoe voorspelt een meteoroloog het weer? (2019, 18 februari). [Video].

Geraadpleegd op: 09-12-2025. Geraagpleegd van: <https://schooltv.nl/video-item/hoe-voorspelt-een-meteoroloog-het-weer-metingen-berekeningen-en-de-pluimverwachting>

Online-Ml. (z.d.). *GitHub - online-ml/river*:  *Online machine learning in Python*. GitHub.

<https://github.com/online-ml/river>

Shetty, V. (2023, 19 oktober). *How does weather forecasting work?* ScienceABC.

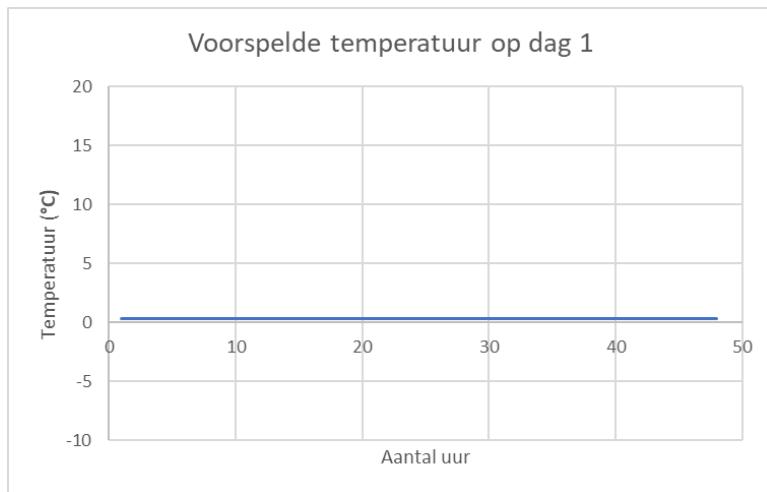
Geraadpleegd op: 07-02-2026. Geraagpleegd van:

https://www.scienceabc.com/innovation/how-does-weather-forecasting-work.html#google_vignette

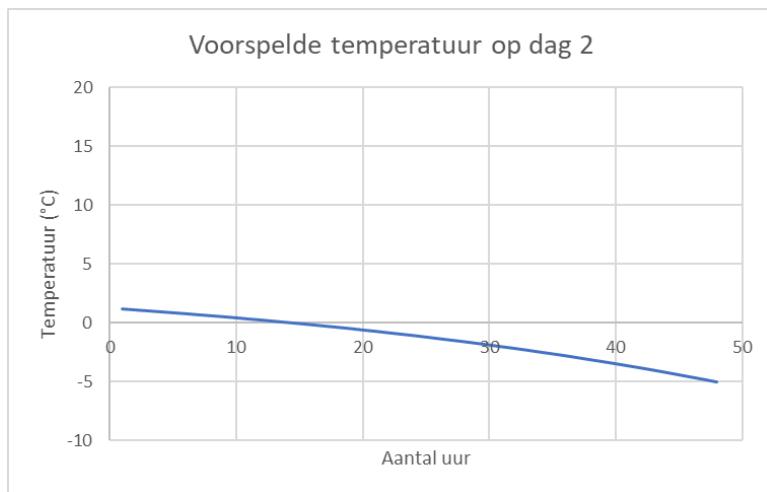
W3Schools.com. (z.d.). <https://www.w3schools.com/cpp/default.asp>

W3Schools.com. (z.d.-b). <https://www.w3schools.com/python/default.asp>

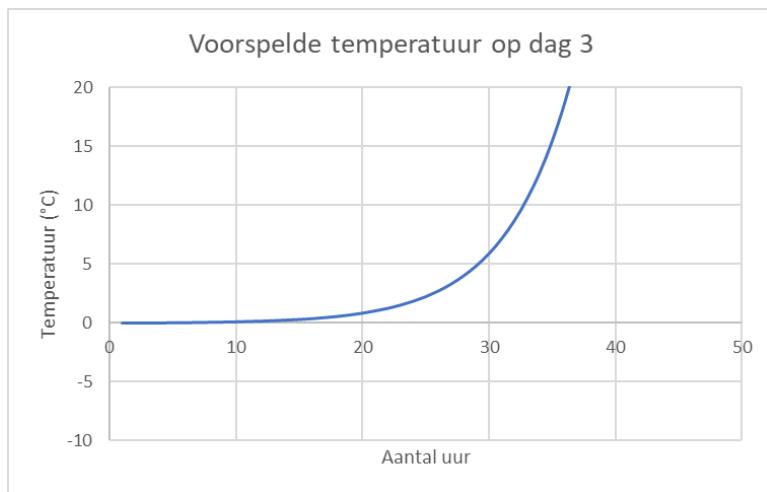
Bijlagen



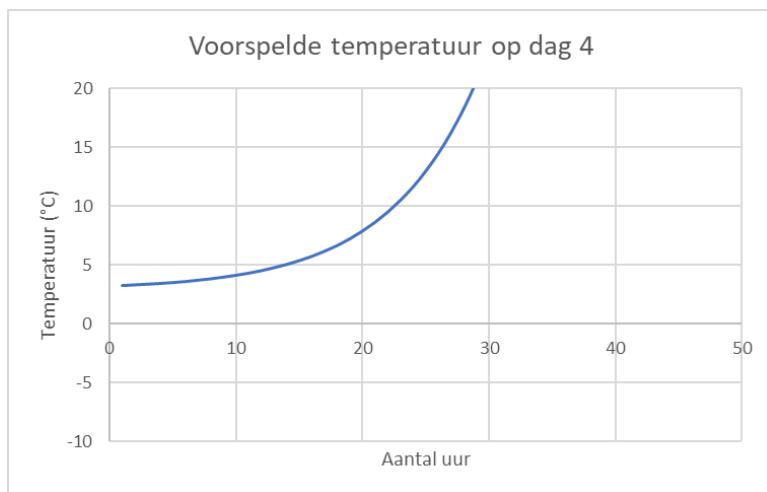
Figuur 1: Voorspelde temperatuur op dag een (03-02-2026 17:29) van de volgende 48 uur



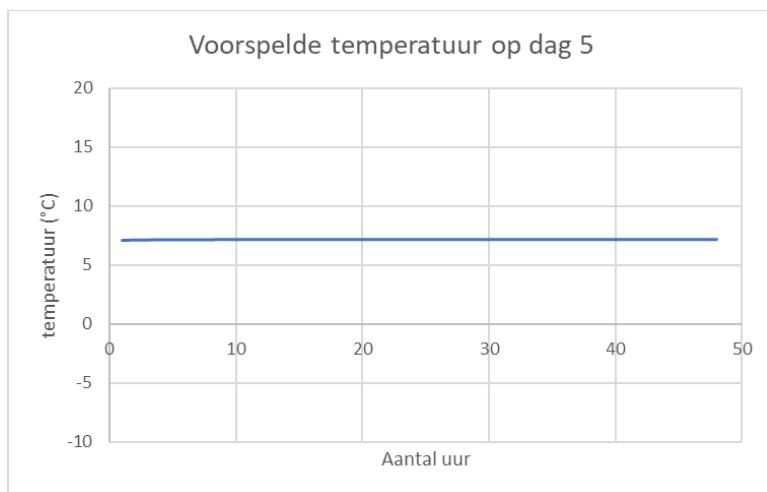
Figuur 2: Voorspelde temperatuur op dag twee (04-02-2026 17:30) van de volgende 48 uur



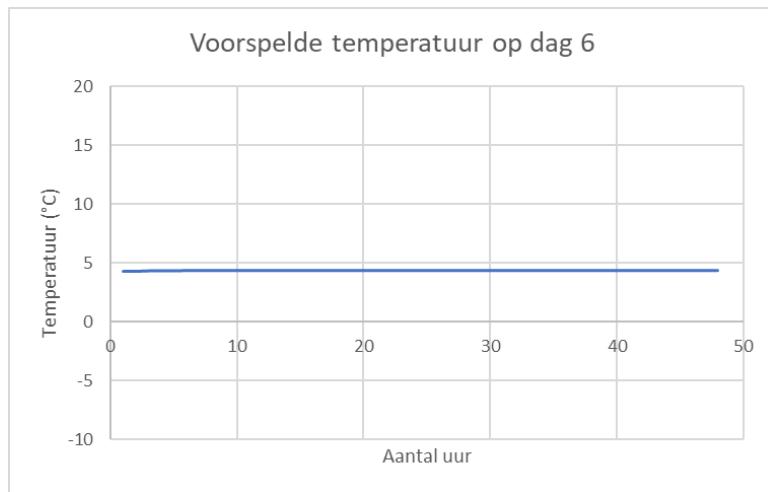
Figuur 3: Voorspelde temperatuur op dag drie (05-02-2026 17:30) van de volgende 48 uur



Figuur 4: Voorspelde temperatuur op dag vier (06-02-2026 17:30) van de volgende 48 uur.



Figuur 5: Voorspelde temperatuur op dag vijf (07-02-2026 17:31) van de volgende 48 uur.



Figuur 6: Voorspelde temperatuur op dag zes (08-02-2026 17:31) van de volgende 48 uur.

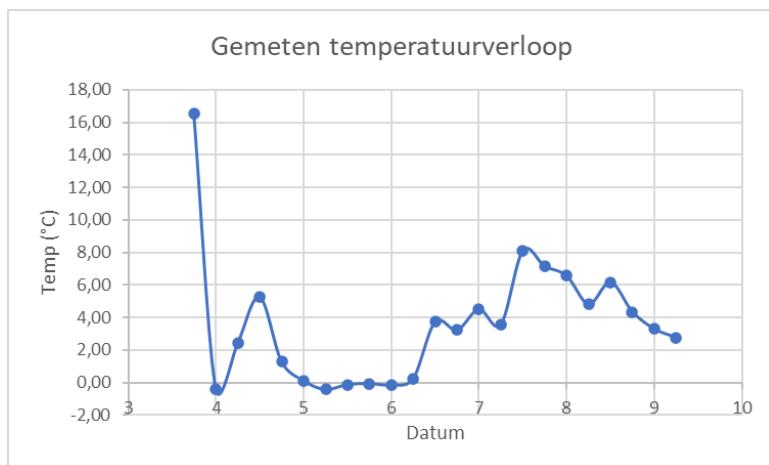
Datum	Temperatuur	Luchtvochtigheid	Luchtdruk
03-02-2026 17:29	16,56°C	60,29%	999.9 hPa
04-02-2026 17:30	1,29°C	82.58 %	1001.0 hPa
05-02-2026 17:30	-0,06°C	78.57 %	994.5 hPa
06-02-2026 17:30	3,32°C	80.46 %	995.6 hPa
07-02-2026 17:31	7,15°C	79.03 %	1006.5 hPa
08-02-2026 17:31	4,43°C	82.26 %	1012.4 hPa

Tabel 1: meetgegevens van ons weerstation per dag (half 6) de temperatuur, luchtvochtigheid en luchtdruk weergeven.

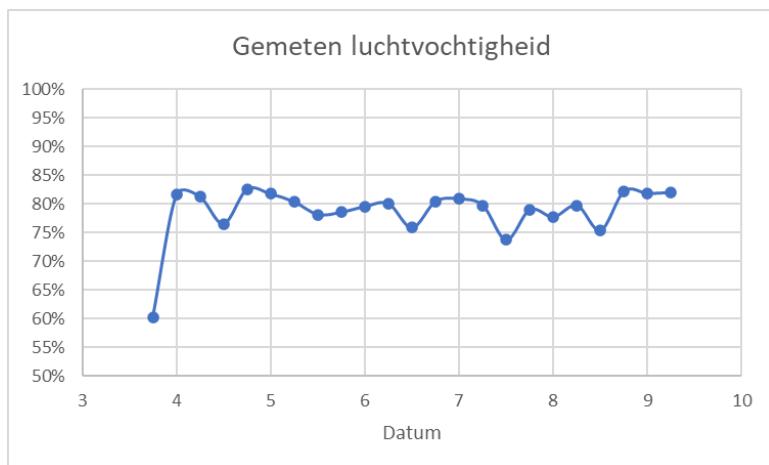
Datum	Temperatuur	Luchtvochtigheid	Luchtdruk
3-2-2026 17:29	16,56	60.29 %	999.93 hPa
3-2-2026 23:29	-0,37	81.62 %	1001.31 hPa
4-2-2026 05:29	2,41	81.28 %	1000.76 hPa
4-2-2026 11:29	5,27	76.56 %	1001.95 hPa
4-2-2026 17:30	1,29	82.58 %	1001.01 hPa
4-2-2026 23:30	0,12	81.76 %	999.92 hPa
5-2-2026 05:30	-0,43	80.39 %	998.0 hPa
5-2-2026 11:30	-0,12	78.18 %	996.75 hPa
5-2-2026 17:30	-0,06	78.57 %	994.57 hPa
5-2-2026 23:30	-0,15	79.57 %	993.31 hPa
6-2-2026 05:30	0,24	80.08 %	993.36 hPa
6-2-2026 11:30	3,73	75.87 %	994.35 hPa

6-2-2026 17:30	3,23	80.46 %	995.6 hPa
6-2-2026 23:30	4,50	80.9 %	998.75 hPa
7-2-2026 05:30	3,57	79.78 %	1001.83 hPa
7-2-2026 11:30	8,10	73.73 %	1005.4 hPa
7-2-2026 17:31	7,15	79.03 %	1006.5 hPa
7-2-2026 23:31	6,56	77.76 %	1008.84 hPa
8-2-2026 05:31	4,82	79.76 %	1010.01 hPa
8-2-2026 11:31	6,17	75.41 %	1012.31 hPa
8-2-2026 17:31	4,34	82.26 %	1012.42 hPa
8-2-2026 23:31	3,30	81.86 %	1013.08 hPa
9-2-2026 05:31	2,77	82.04 %	1011.03 hPa

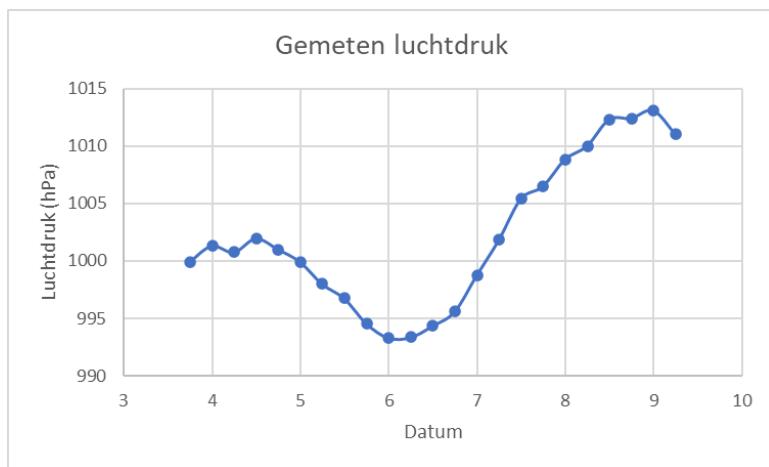
Tabel 2: Uitbereiding van tabel 1 met alle gemeten gegevens weergeven over de dagen



Figuur 7: Aanvulling van de gegevens uit tabel 1. Elk dag deel een temperatuurmeting gedaan, uitgezet tegen de bijbehorende datum



Figuur 8: Gemeten luchtvochtigheid uitgezet tegen de momenten uit tabel 2



Figuur 9: Gemeten luchtdruk uitgezet tegen de momenten uit tabel 2

Logboek

Het logboek is hier te vinden: <https://github.com/D-A-R-N-O-K/PWS-Weerstation/blob/main/docs/logboek.md>