cognizant WeatherIQ

Personalized Weather Forecast Video Generator using GenCast and AWS

April to September 2025

GenCast Forecasting Prototype

Internship Project Jacques Meyer– Cognizant UK

Project Overview

This project explores and demonstrates the capabilities of DeepMind’s GenCast weather prediction model.

→ **The goal:** to develop a prototype application that generates personalized, smartphone-format weather forecast videos based on user-selected dates and locations.

Key Objectives

- Evaluate GenCast model variants for real-time suitability
- Explore methods to increase temporal resolution (from 12h to hourly)
- Identify best sources of real-time weather data
- Deliver a client-facing prototype showcasing GenCast’s adaptability for weather-critical industries

Deliverables

- GitHub code depo (video generation + model endpoint + UI)
- Competitor Analysis.
- Evaluation report on GenCast model performance.
- Final presentation (This slide deck)
- Project School report
- Demo video(s) for client showcase.
- Video presentation of the project and its context made by Cognizant's Marketing team

User Case Definition Based on Model Capabilities

Based on the available variables predicted by GenCast and their respective forecasting performance, we designed a user case tailored to the strengths of the model and the needs of wind energy operations.

Selected user case :

Omar
Renewable Energy Operator
Yorkshire

Goals
- Predict power output to align with commitments
- Schedule maintenance during low-wind periods

Pain Points
- Forecasts are too generic for turbine-level planning
- Missed production estimates cause grid penalties

How GenCast Helps
- Provides micro-local precision at high granularity
- Offers clear visuals to aid decision-making

Explored user cases

David
Emergency Response Coordinator
Greater London Authority

Goals
- Pre-position teams and supplies for severe weather
- Communicate timely warnings to the public

Pain Points
- Uncertainty in forecasts leads to reactive decisions
- Weather data is difficult to translate into public guidance

How GenCast Helps
- Scenario-based forecast videos support briefings
- Personalised data enables earlier planning and mobilisation

Anita
Rail Logistics Coordinator
UK Midlands

Goals
- Minimise weather delays
- Optimise cargo handoffs

Pain Points
- Corridor forecasts are not specific enough
- Updates are too slow to support re-routing

How GenCast Helps
- Enables predictive routing across affected rail corridors
- Real-time updates support agile logistics planning

Sarah
Port Operations Manager
Port of Felixstowe, UK

Goals
- Optimise container loading/ unloading around weather windows
- Reduce delays from fog, wind, or storms

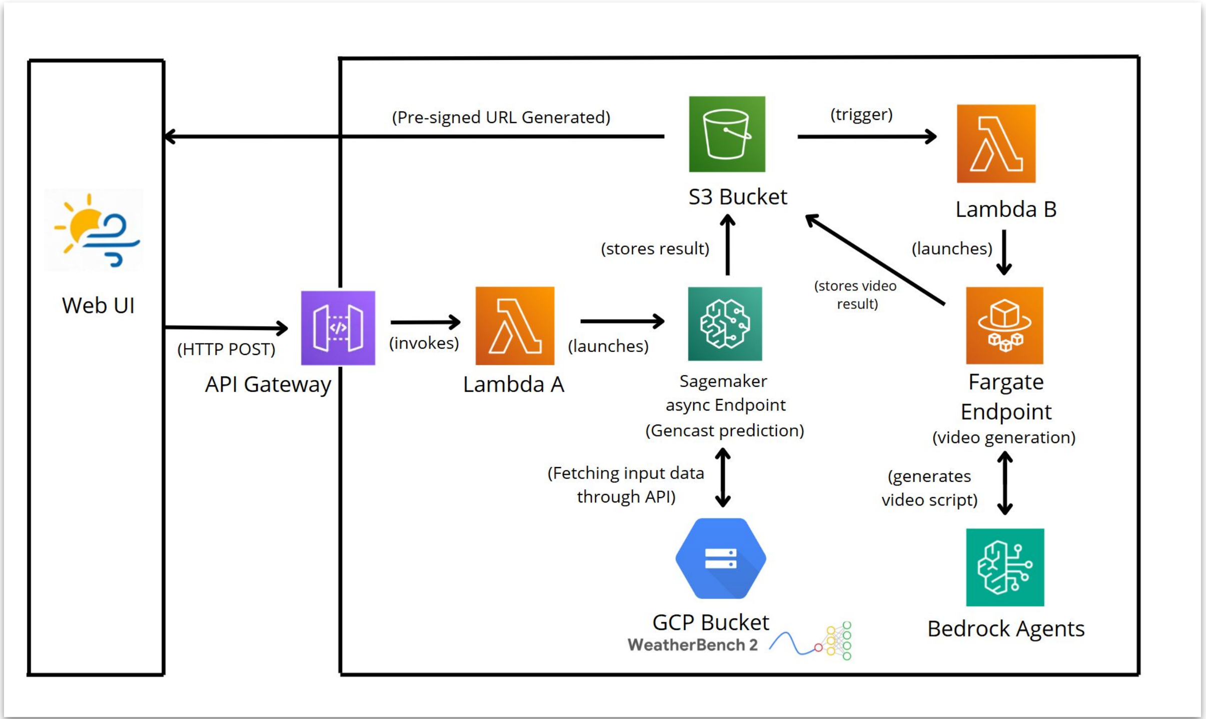
Pain Points
- Forecasts are too coarse and infrequent
- Difficult to visualise data and integrate it into operations

How GenCast Helps
- Provides hourly, location-specific forecasts for port operations
- Delivers clear visuals to support scheduling and shift planning

WeatherIQ global AWS service Architecture :

The entire project was designed and implemented on AWS

Global AWS service Architecture :



User Interface

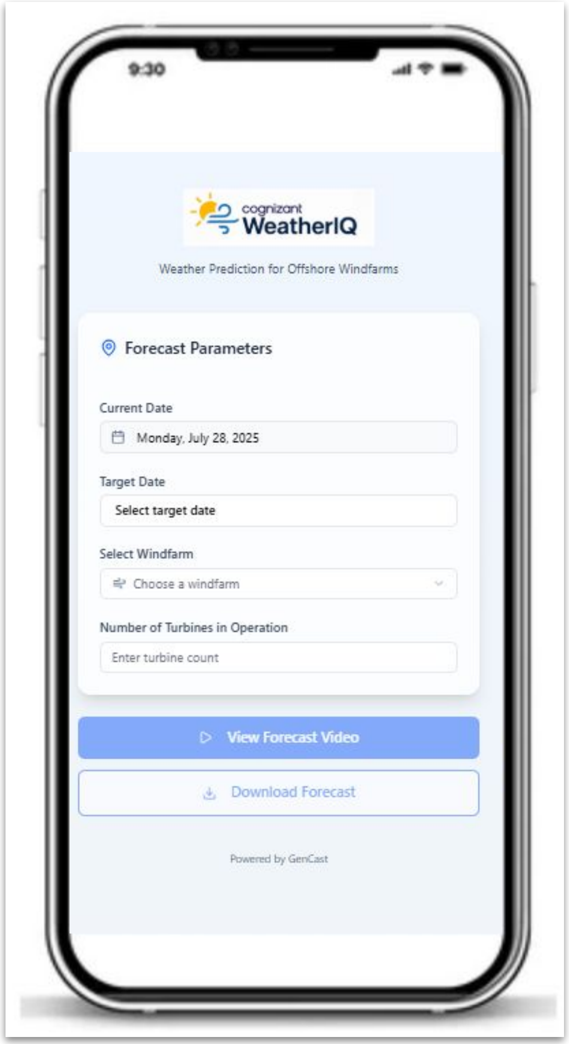
The user interface was designed using HTML and CSS, ensuring a clean and responsive layout.

Generated using Figma's AI-powered design tool, which accelerated the development.

User input :

- Current date (past or real time)
- Target date
- Selected windfarm
- Number of operating turbines

Weather IQ Mobile Interface.zip



9:30

cognizant WeatherIQ

Weather Prediction for Offshore Windfarms

Forecast Parameters

Current Date

Monday, July 28, 2025

Target Date

Select target date

Select Windfarm

Choose a windfarm

Number of Turbines in Operation

Enter turbine count

View Forecast Video

Download Forecast

Powered by GenCast

Gencast Sagemaker Endpoint

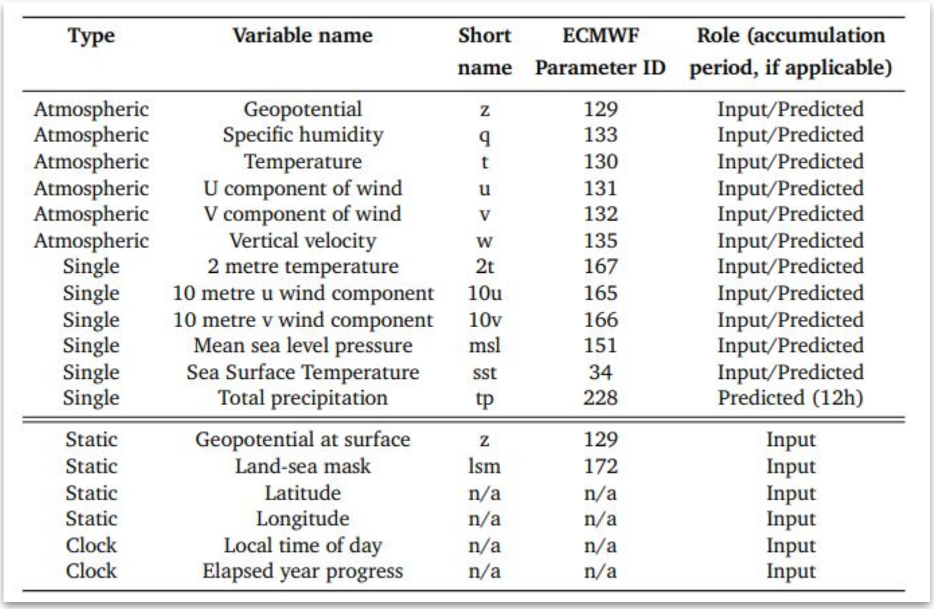
Model Overview

GenCast is a probabilistic generative AI model for weather forecasting, developed by DeepMind.

GitHub Repository https://github.com/google-deepmind/graphcast

- **Key Features:**
  - Leverages machine learning for fast, accurate forecasts
  - Outperforms the European Centre for Medium-Range Forecasts (ECMWF)'s ensemble forecast, ENS on 97.4% of 1320 targets evaluated, and better predicts extreme weather, tropical cyclones, and wind power production
- **Model Architecture**
  - Autoregressive Design:
    Predicts future atmospheric states using past snapshots (e.g., from t−12h and t, forecast t+12h)
  - Forecast Horizon:
    Iterative predictions up to 15 days ahead
- **Training Data**
  - Dataset: ERA5 (via ECMWF & Copernicus Climate Change Service)
  - Period:
    - Training: 1979–2018
    - Evaluation: 2019–present
  - Resolution: 0.25° or 1° grid, 12-hour intervals

17 input variables, 12 predicted variables

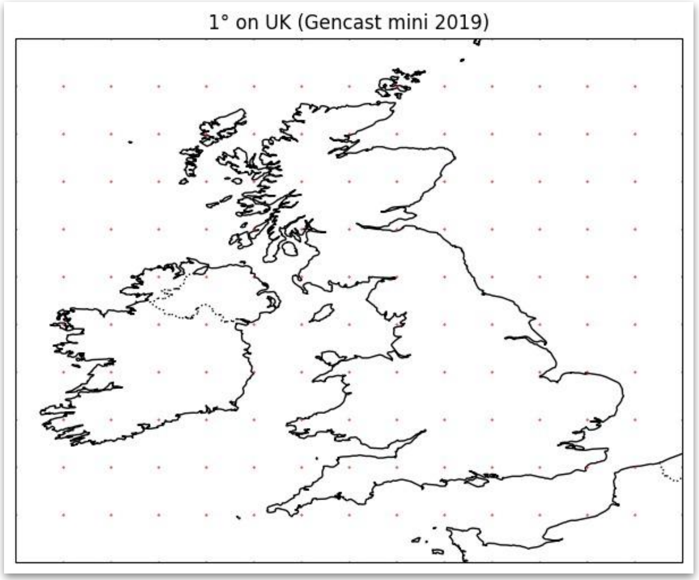


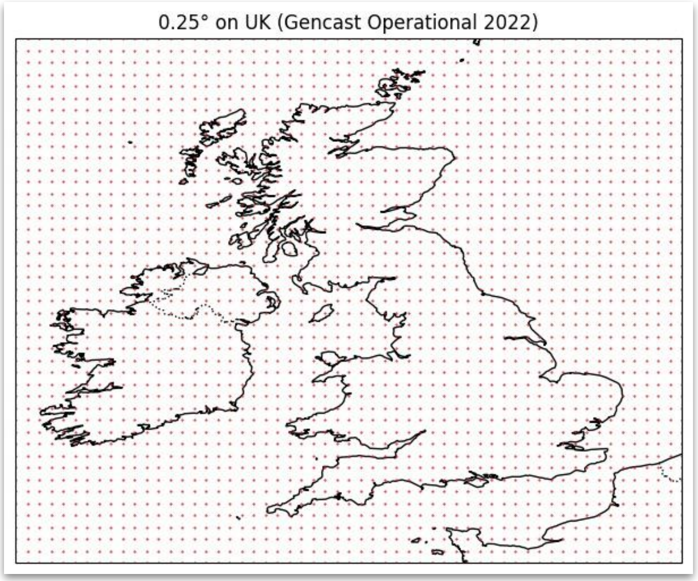
Type	Variable name	Short name	ECMWF Parameter ID	Role (accumulation period, if applicable)
Atmospheric	Geopotential	z	129	Input/Predicted
Atmospheric	Specific humidity	q	133	Input/Predicted
Atmospheric	Temperature	t	130	Input/Predicted
Atmospheric	U component of wind	u	131	Input/Predicted
Atmospheric	V component of wind	v	132	Input/Predicted
Atmospheric	Vertical velocity	w	135	Input/Predicted
Single	2 metre temperature	2t	167	Input/Predicted
Single	10 metre u wind component	10u	165	Input/Predicted
Single	10 metre v wind component	10v	166	Input/Predicted
Single	Mean sea level pressure	msl	151	Input/Predicted
Single	Sea Surface Temperature	sst	34	Input/Predicted
Single	Total precipitation	tp	228	Predicted (12h)
Static	Geopotential at surface	z	129	Input
Static	Land-sea mask	lsm	172	Input
Static	Latitude	n/a	n/a	Input
Static	Longitude	n/a	n/a	Input
Clock	Local time of day	n/a	n/a	Input
Clock	Elapsed year progress	n/a	n/a	Input

Gencast Sagemaker Endpoint

Resolution difference between models :

[GenCast 1p0deg <2019.npz]() [GenCast 0p25deg <2019.npz]()





Gencast Sagemaker Endpoint

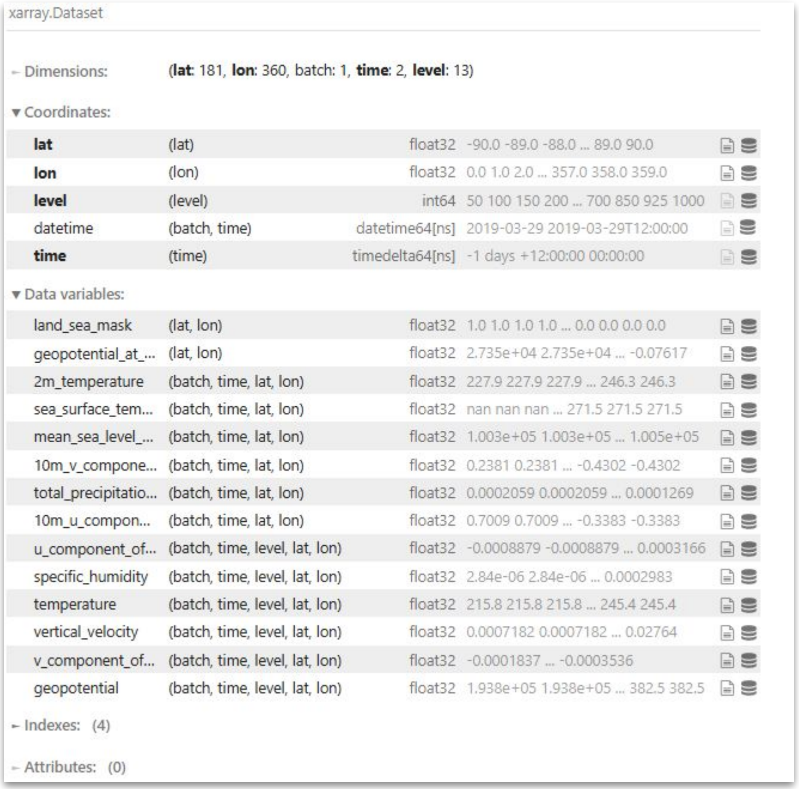
Data Architecture following Weather Bench2 :

WeatherBench 2 is an open-source benchmark developed by Google Research and ECMWF to evaluate weather forecasting models—both machine learning and traditional.

It provides:

- Standardized datasets (e.g., ERA5 in cloud-optimized format)
- Evaluation tools for comparing model accuracy and realism
- Support for probabilistic and ensemble forecasts

→ It helps researchers track progress and compare models fairly across different forecasting tasks.

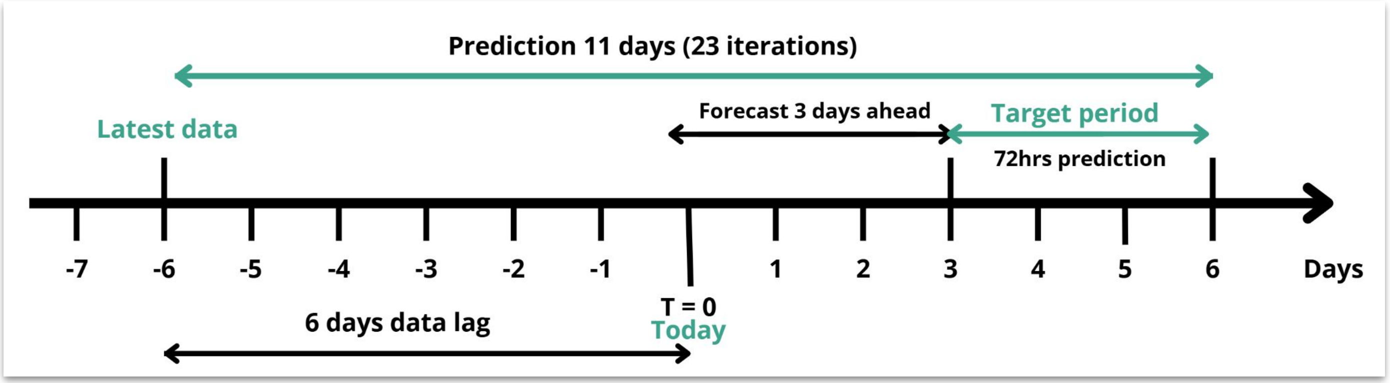


```
xarray.Dataset

► Dimensions:        (lat: 181, lon: 360, batch: 1, time: 2, level: 13)

▼ Coordinates:
   lat               (lat)                          float32  -90.0 -89.0 -88.0 ... 89.0 90.0
   lon               (lon)                          float32  0.0 1.0 2.0 ... 357.0 358.0 359.0
   level             (level)                          int64  50 100 150 200 ... 700 850 925 1000
   datetime          (batch, time)           datetime64[ns]  2019-03-29 2019-03-29T12:00:00
   time              (time)                 timedelta64[ns]  -1 days +12:00:00 00:00:00

▼ Data variables:
   land_sea_mask     (lat, lon)                     float32  1.0 1.0 1.0 1.0 ... 0.0 0.0 0.0 0.0
   geopotential_at_… (lat, lon)                     float32  2.735e+04 2.735e+04 ... -0.07617
   2m_temperature    (batch, time, lat, lon)        float32  227.9 227.9 227.9 ... 246.3 246.3
   sea_surface_tem…  (batch, time, lat, lon)        float32  nan nan nan ... 271.5 271.5 271.5
   mean_sea_level_…  (batch, time, lat, lon)        float32  1.003e+05 1.003e+05 ... 1.005e+05
   10m_v_compone…    (batch, time, lat, lon)        float32  0.2381 0.2381 ... -0.4302 -0.4302
   total_precipitatio… (batch, time, lat, lon)      float32  0.0002059 0.0002059 ... 0.0001269
   10m_u_compon…     (batch, time, lat, lon)        float32  0.7009 0.7009 ... -0.3383 -0.3383
   u_component_of…   (batch, time, level, lat, lon) float32  -0.0008879 -0.0008879 ... 0.0003166
   specific_humidity (batch, time, level, lat, lon) float32  2.84e-06 2.84e-06 ... 0.0002983
   temperature       (batch, time, level, lat, lon) float32  215.8 215.8 215.8 ... 245.4 245.4
   vertical_velocity (batch, time, level, lat, lon) float32  0.0007182 0.0007182 ... 0.02764
   v_component_of…   (batch, time, level, lat, lon) float32  -0.0001837 ... -0.0003536
   geopotential      (batch, time, level, lat, lon) float32  1.938e+05 1.938e+05 ... 382.5 382.5

► Indexes:    (4)

► Attributes:    (0)
```

Gencast Sagemaker Endpoint

Input Data for real time prediction :

- Source: Google Research GCP Bucket via API
  https://github.com/google-research/arco-era5?tab=readme-ov-file#analysis-ready-data
    - Dataset: google-research/arco-era5
    - Format: NetCDF (.nc) in WeatherBench2 structure
    - Description: Recipes for generating Analysis-Ready & Cloud-Optimized (ARCO) ERA5 datasets
- Data Type: ERA5 and preliminary ERA5T
  → ERA5T is available with ~1 week delay (5–6 days due to ECMWF processing)



- Example :

Current date	Target date	Last available data	Prediction	AWS Inference	Prediction duration
2025-08-06	2025-08-07	2025-07-31	9 days, 19 times	ml.g5.12xlarge	26.6 minutes

Gencast Sagemaker Endpoint

Model Inputs & Preprocessing

Required Datasets for Inference

1. `eval_inputs`
   - NetCDF file with two input time steps (12 hours apart)
   - Contains 18 variables (12 predicted)
   - Structured by latitude, longitude, pressure levels, and time
2. `eval_targets`
   - Empty NetCDF file with same dimensions as eval_inputs
   - Time coordinates represent future steps to predict
   - Filled with NaN values as placeholders for model output
3. `eval_forcings`
   - Stable, non-predicted variables derived from time and location:
     - `year_progress_sin`, `year_progress_cos`
     - `day_progress_sin`, `day_progress_cos`
   - Encodes temporal context for the model

Forcing Calculation Logic

- Compute actual datetimes by:
  - Extracting the last timestamp from eval_inputs
  - Adding time offsets from eval_targets
- Use these to calculate sinusoidal encodings for year/day progress

Integration with GenCast Demo Code

- Uses DeepMind's `extract_inputs_targets_forcings(...)` function to:
  - Split full dataset into `eval_inputs`, `eval_targets`, and `eval_forcings`
  - Normalize time (last input time = 0)
  - Validate structure and consistency

Adapting for Real-World Inference

- Only input data is known; targets are unknown
- Approach:
  1. Fetch ERA5 snapshots at t−12h and t
  2. Create placeholder eval_targets with future time steps
  3. Concatenate inputs and targets into a synthetic dataset
  4. Run `extract_inputs_targets_forcings(...)` to:
     - Extract inputs and targets
     - Auto-compute forcings
     - Ensure alignment and integrity

→ This enables robust, inference-only workflows using DeepMind's preprocessing logic.

Gencast Sagemaker Endpoint

AWS SageMaker instances :

Here's the comparison table for AWS SageMaker instances running the GenCast 1x0 model with 12-step prediction (6 days):

Instance Type	Cost per Hour (USD)	Inference Time (seconds)
ml.g5.8xlarge	2.448	1875.89
ml.g5.12xlarge → **Selected option**	5.672	517.64
ml.g5.24xlarge	8.144	516.96
ml.p3.2xlarge	3.060	1990.43
ml.p3.8xlarge	12.240	Too expensive
ml.p3.16xlarge	24.480	Too expensive

Notes

- Despite all tested instances offering more than enough performance to run the GenCast 1x0 model, inference times were slower than expected based on DeepMind's published benchmarks for GPUs.
- The 0.25° model could not be launched on any available AWS SageMaker instance due to memory limitations.
- As a result, we stayed with the 1x0 model, which remains compatible for our use case.

Gencast Sagemaker Endpoint

Hardware Requirements & Inference Performance :

https://github.com/google-deepmind/graphcast/blob/main/docs/cloud_vm_setup.md

- **Designed for GCP**

GenCast is optimized for Google Cloud Platform (GCP) using TPUs and the JAX library

JAX: A high-performance numerical computing library, similar to NumPy but optimized for hardware acceleration (TPUs/GPUs)

- **Attention Mechanisms**

GenCast uses two attention mechanisms:

- splash_attention → optimized for TPUs, faster
- triblockdiag_mha → compatible with GPUs, but slower and more memory-intensive

Inference Time Comparison (30-step rollout, 0.25° resolution)

Model Resolution	System Memory	GPU vRAM
0.25° GenCast	~300 GB	~60 GB
1° GenCast	~24 GB	~16 GB

→ TPUs are only available on GCP, making it the preferred platform for efficient GenCast inference.

- **Memory Requirements**

→ On AWS, use a SageMaker instance that meets these specs for GPU inference.

Configuration	Inference Time	Notes
TPU + splash_attention	~8 min	Fastest setup
TPU + triblockdiag_mha	~15 min	Slower due to attention type
GPU + triblockdiag_mha (AWS)	~25 min	Slowest; only option on AWS

Recommendation

For next Project iteration :

- Move to GCP to leverage TPUs and splash_attention and use 0x25 model
- Code is compatible GCP, making migration straightforward

Gencast Sagemaker Endpoint

Sagemaker Asynchronous Endpoints :

Due to the non-negligible cost of running GenCast inferences—especially on high-performance GPU instances—we chose to host the model using SageMaker Asynchronous Endpoints.

Why It’s Cost-Efficient :

- Pay-as-you-go: Instances run only during inference, avoiding 24/7 GPU costs.
- Ideal for prototyping or low-traffic setups (e.g., one user at a time).
- Avoids the high cost of keeping powerful GPU instances constantly active.

Downsides :

- Startup Latency:
  Each inference has a cold start delay of ~1–2 minutes, which can negatively impact user experience.
- Not Real-Time:
  Unsuitable for applications needing instant feedback.
- Extra Complexity:
  Requires managing job status, result retrieval, and S3 storage.

While asynchronous endpoints introduce a latency trade-off that affects responsiveness, they are cost-effective for prototypes. For GenCast, this setup is acceptable given the single-user context and the high cost of continuous GPU usage.

Gencast Sagemaker Endpoint

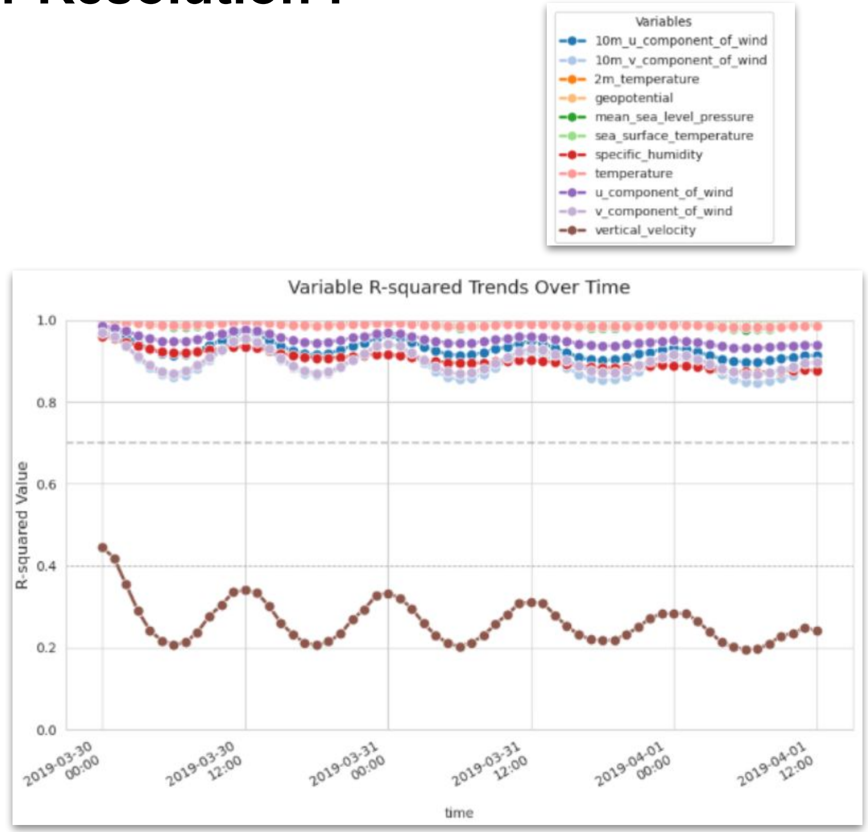
Reduce GenCast’s 12-Hour Resolution :

Explored Options:

1. <u>Multiple Offset Predictions</u> :
   Launching several predictions with time offsets and merging results.
   → Rejected due to high resource usage and poor UX/pricing scalability.
2. <u>Classic ML Model</u> (ERA5-based) :
   Train a model using ERA5 data, focusing on key variables and pressure levels.
   → Promising but requires location-specific fine-tuning. Not pursued due to time constraints.

Selected Option:

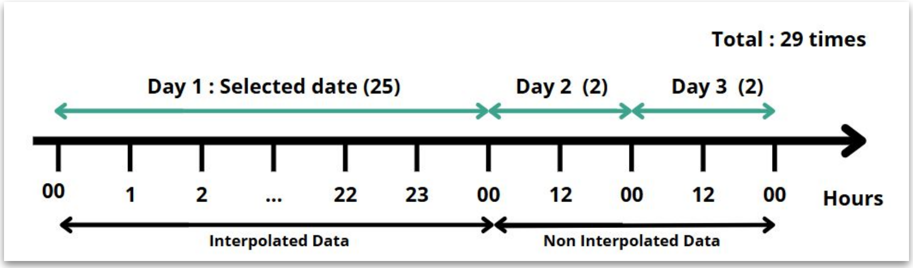
- <u>Simple Linear Interpolation</u> :
  → Easy to implement, low computational cost, and delivers good performance.

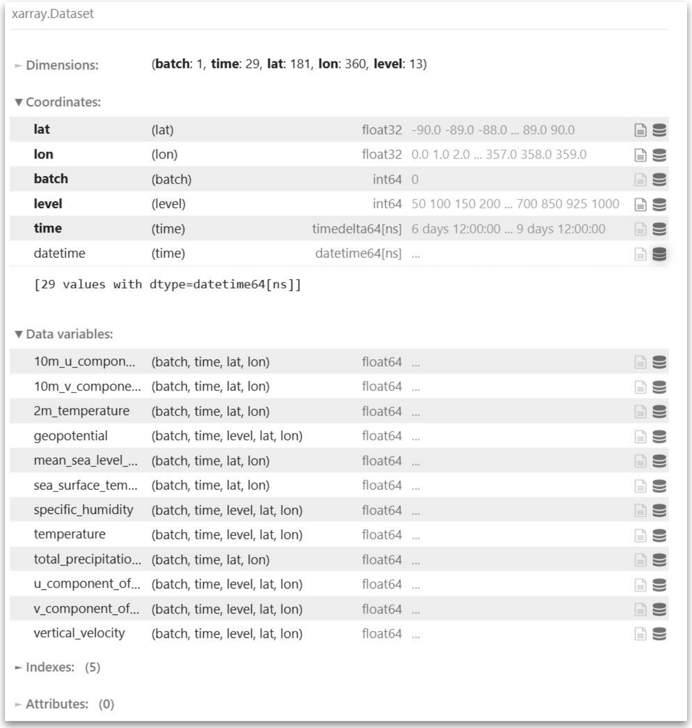


Gencast Sagemaker Endpoint

Output Format :

- **File Type**: NetCDF (.nc)
- **Forecast Duration :** 72 hours starting from the user-selected date (29 time)
  - Day 1: 25 interpolated time steps
  - Days 2–3: 3 time steps per day
- **Spatial Coverage :** Global (1° resolution)
  → Data is not filtered to the UK or specific windfarm to maintain flexibility and reusability across different user cases.





```
xarray.Dataset

► Dimensions:      (batch: 1, time: 29, lat: 181, lon: 360, level: 13)

▼ Coordinates:
   lat              (lat)        float32   -90.0 -89.0 -88.0 ... 89.0 90.0
   lon              (lon)        float32   0.0 1.0 2.0 ... 357.0 358.0 359.0
   batch            (batch)      int64     0
   level            (level)      int64     50 100 150 200 ... 700 850 925 1000
   time             (time)       timedelta64[ns]  6 days 12:00:00 ... 9 days 12:00:00
   datetime         (time)       datetime64[ns]   ...
   [29 values with dtype=datetime64[ns]]

▼ Data variables:
   10m_u_compon...    (batch, time, lat, lon)          float64  ...
   10m_v_compone...   (batch, time, lat, lon)          float64  ...
   2m_temperature     (batch, time, lat, lon)          float64  ...
   geopotential       (batch, time, level, lat, lon)   float64  ...
   mean_sea_level_... (batch, time, lat, lon)          float64  ...
   sea_surface_tem... (batch, time, lat, lon)          float64  ...
   specific_humidity  (batch, time, level, lat, lon)   float64  ...
   temperature        (batch, time, level, lat, lon)   float64  ...
   total_precipitatio... (batch, time, lat, lon)       float64  ...
   u_component_of...  (batch, time, level, lat, lon)   float64  ...
   v_component_of...  (batch, time, level, lat, lon)   float64  ...
   vertical_velocity  (batch, time, level, lat, lon)   float64  ...

► Indexes: (5)

► Attributes: (0)
```

Gencast Sagemaker Endpoint

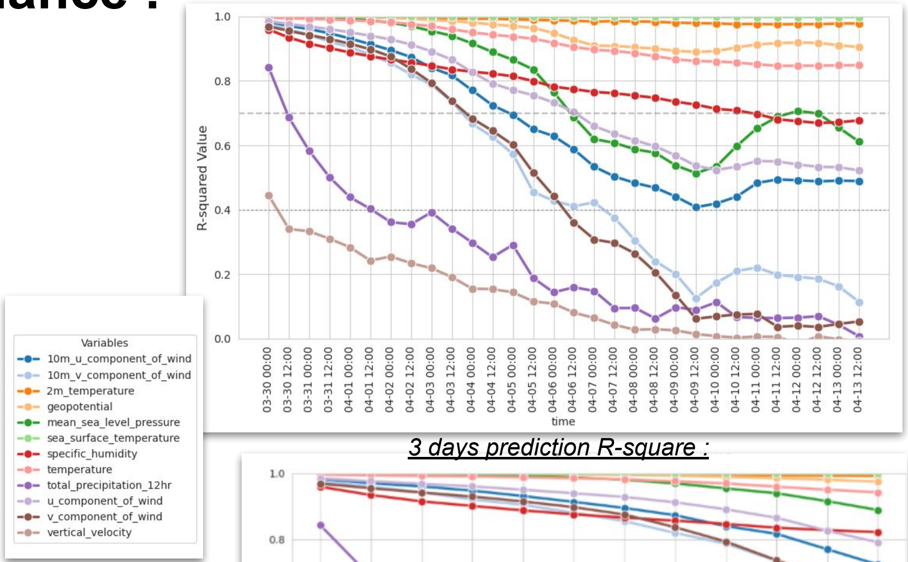
Variable-Specific Performance :

- GenCast's accuracy varies by atmospheric variable:
  - Stable variables (e.g., temperature, pressure) → easier to forecast
  - Volatile variables (e.g., wind speed, precipitation) → more challenging

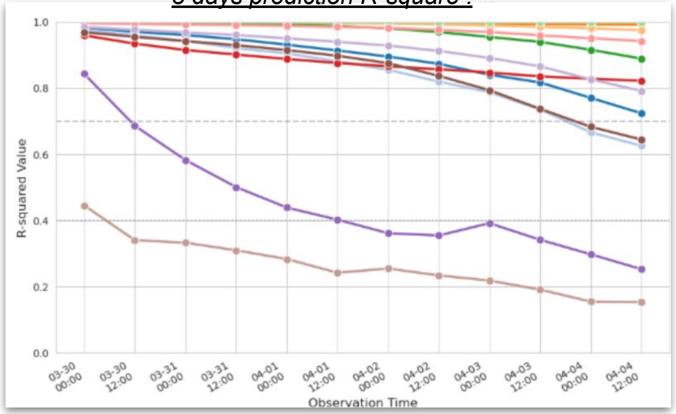
- Despite these differences, GenCast consistently delivers strong results across most variables—especially when using high-quality, well-formatted input data.

→ This highlights the model's robustness and reliability for medium-range forecasting.

15 days prediction R-square :



3 days prediction R-square :



GenCast Model: Strengths & Weaknesses

Strengths

- **User-Friendly**: Simple to use, for technical users and good documentation.
- **Fast Inference**: Quicker predictions compared to traditional weather forecasting methods.
- **Scalable Architecture**: Can be adapted for broader use with optimization.
- **Modular Design**: Easy to integrate with other services (e.g., AWS, DeepMind APIs).
- **Single-User Optimization**: Efficient for personalized forecasts.

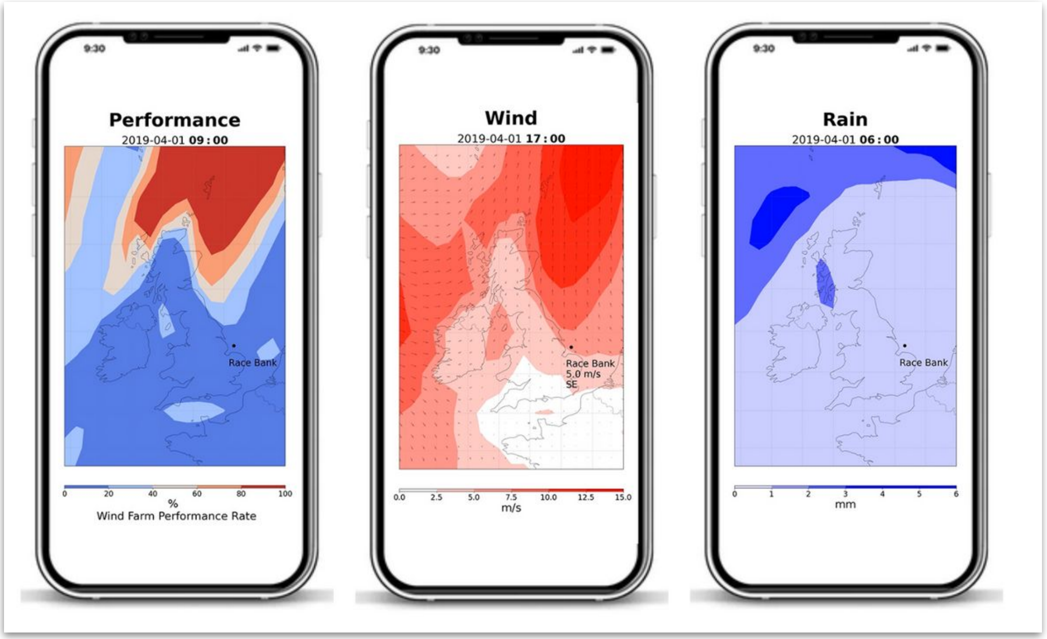
Weaknesses

- **High Computational Cost**: Requires significant GPU power for high precision.
- **Limited Geographical Precision**: Accuracy drops in less-covered regions; linked to computational constraints.
- **Dependence on Real-Time Data**: Predictions degrade with outdated inputs.
- **Missing Variables**: Lacks key features like cloud coverage.
- **Latency in Real-Time Mode**: lag in input data

Video generation Faregate Endpoint

Forecast Video – Visual & Functional Design

Visualizations : Python + Cartopy lib for maps



Structure & Style :

- The video presents a detailed weather forecast, inspired by traditional TV broadcasts.
- The narration walks the viewer through the day in two parts: Morning (AM) and Afternoon (PM).
- Each segment includes dynamic visualizations of:
  - Rainfall
  - Wind patterns
  - Turbine performance over time

Temperature :

- Temperature has minimal impact on turbine performance.
- Instead, we prioritized rainfall, which supports maintenance planning.

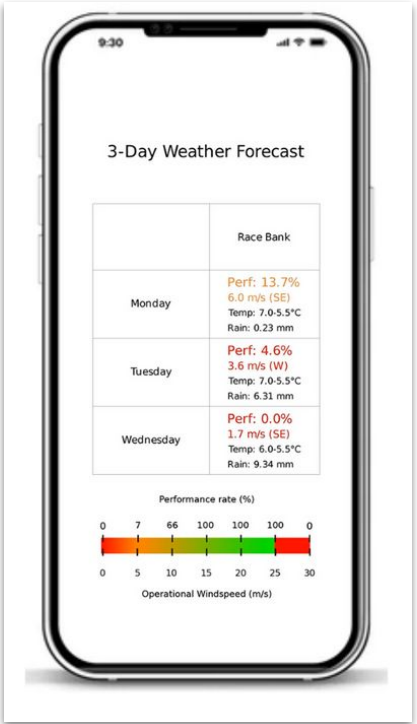
Performance Map :

- Shows geographical zones around the wind farm.
- Displays estimated turbine performance in each area.
- Calculated using:
  - Local wind conditions
  - The turbine's power curve
- Helps visualize how efficiently turbines would operate in different locations.

Video generation Faregate Endpoint

Extended Forecast Overview (3 Days)

Visualizations : Python + Manim lib for animated table



9:30

3-Day Weather Forecast

	Race Bank
Monday	Perf: 13.7% 6.0 m/s (SE) Temp: 7.0-5.5°C Rain: 0.23 mm
Tuesday	Perf: 4.6% 3.6 m/s (W) Temp: 7.0-5.5°C Rain: 6.31 mm
Wednesday	Perf: 0.0% 1.7 m/s (SE) Temp: 6.0-5.5°C Rain: 9.34 mm

Format & Flow

Transitions to a broader overview of the next two days.

Presented in a concise tabular layout, summarizing:

- Rainfall
- Wind
- Temperature
- Turbine performance

Video generation Faregate Endpoint

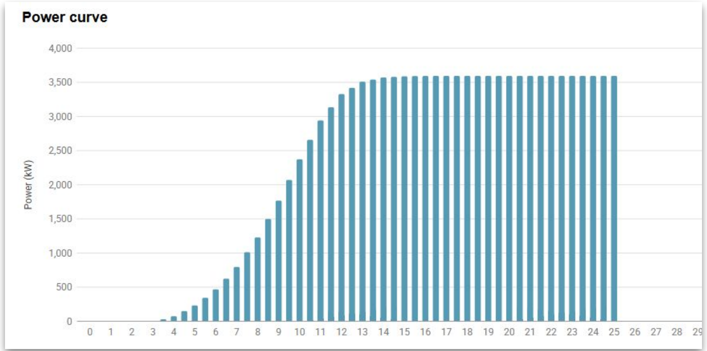
Energy output :

Turbine documentation:
https://www.thewindpower.net/turbine_en_20_siemens_swt-3.6-107.php

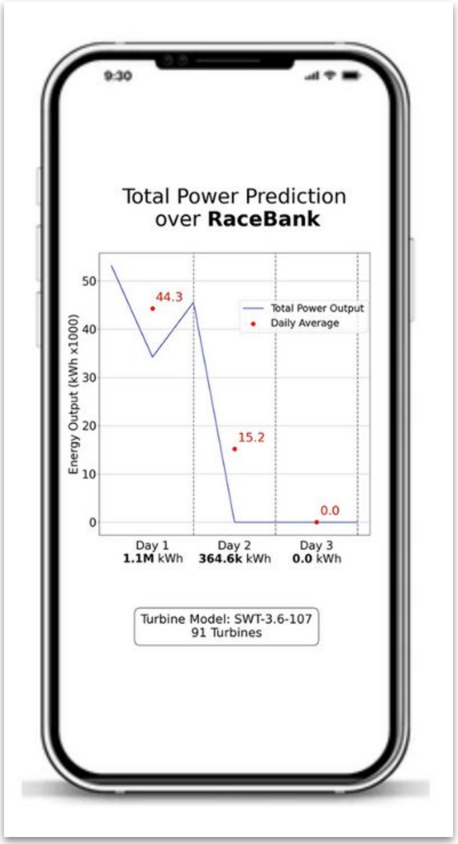
Wind speed and direction are the only variables that directly affect performance.

Incorporated the turbine's power curve to:

- Map wind speed to expected energy output for each forecasted time step
- Calculate total energy production per wind farm, based on:
    - Predicted wind conditions
    - Number of operational turbines entered by the user



Power Curve of a SWT-3.6 -107 Wind Turbine depending on the Weed Spend



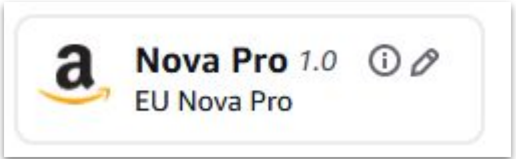
Visualizations : Python + matplotlib

Video generation Faregate Endpoint

Agentic Usage :

Script Generation

- Bedrock agents, each responsible for a specific video section:
- All scripts were generated using Amazon's Nova Pro model, for:
    - High-quality, context-aware text generation
    - Consistent tone and structure across segments


Nova Pro 1.0
EU Nova Pro

Audio Narration

- Used Amazon Polly with the "Amy" voice model to convert scripts into natural-sounding speech.
- Enabled seamless integration between generated content and visual storytelling

Link to the GitHub code depo :

https://github.com/JacquesMeyerCognizant/WeatherIQ.git
(send mail to Jacques Meyer to get access)

Link to final video demo :

https://drive.google.com/file/d/1J8Ec_V2eM0IDEsaUFekqFCfu3hzz0jCk/view?usp=drive_link

Key Objectives :

- Evaluate GenCast model variants for real-time suitability ✅
- Explore methods to increase temporal resolution (from 12h to hourly) ✅
- Real-time prediction and Identify best sources of weather data ✅
- Deliver a client-facing prototype showcasing GenCast's adaptability for weather-critical industries ✅

Next steps and improvement :

- Move the GenCast endpoint to GCP (For pricing + 0x25 model), no adaptation to make in the code
- Finish the implementation of the global project (connecting UI + Lambdas)
- Developing new user cases

References:

- Copernicus API :
  https://cds.climate.copernicus.eu/datasets/reanalysis-era5-pressure-levels?tab=download
- Deepmind forum :
  forecasting in real time with GenCast_mini_demo · Issue #131 · google-deepmind/graphcast
- Deepmind Era5 Bucket (recommended by Deepmind searcher) :
  google-research/arco-era5: Recipes for reproducing Analysis-Ready & Cloud Optimized (ARCO) ERA5 datasets.

  WeatherBench 2 Data Guide — WeatherBench 2 documentation
- Deepmind’s public GenCast model :
  https://github.com/google-deepmind/graphcast?tab=readme-ov-file
- Deepmind’s GenCast research paper :
  https://www.science.org/doi/10.1126/science.adi2336
- Wind Turbine information :
  https://www.thewindpower.net/turbine_en_20_siemens_swt-3.6-107.php