Lightning Talk

# Why your Forecasting Transformer Model is not working

How to fix it in Python

PyTorch Conference 2026

# Introduction

## Rosheen Naeem

Software Engineer at Miro

Erasmus Mundus Master's
in
Software Engineering for
Green Deal (SE4GD)

Google Summer of Code

# Open climate Fix

Sustainability by IT

Renewable energy is unpredictable in terms of power generation

- Solar Forecasting
- Cloudcasting
- Wind Forecasting
- Mapping Solar Power
- Unlocking Energy Data

OPEN
CLIMATE
FIX

# Why Solar Forecasting is important

We can't control the weather, but we can predict it

☀

### Variable by nature

PV output can swing dramatically within minutes due to cloud cover — not controllable, only predictable.

⚡

### Grid stability

Grid operators need forecasts to balance supply and demand in real time — bad forecasts mean fossil fuel backup.

🏠

### Smart homes & trading

Accurate forecasts enable energy trading decisions and smart-home optimisation for solar households.

# Open Quartz Solar Forecast project

Case study

## The project

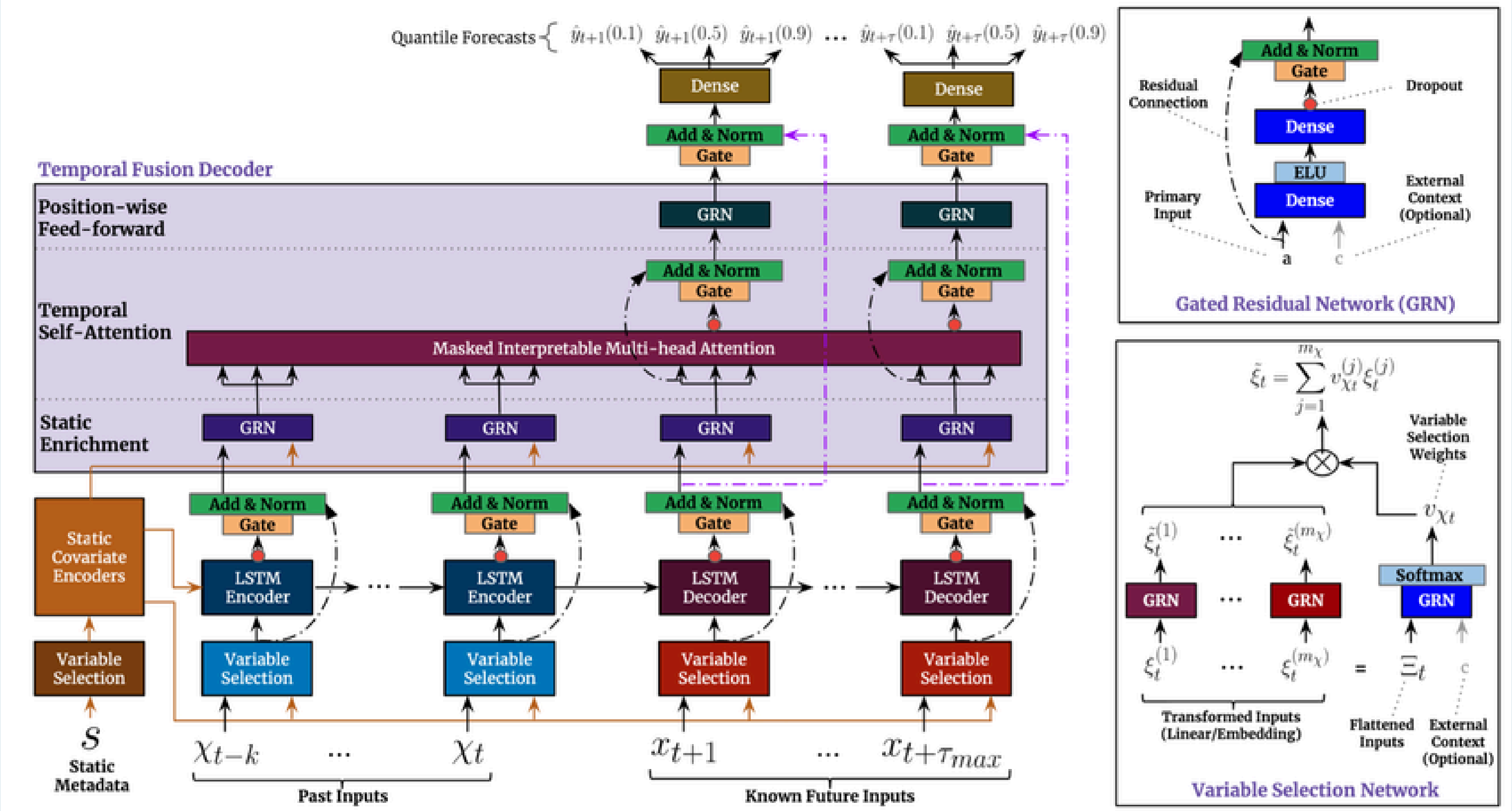
- Open-source PV regular ML forecasting model
- Uses public data
- Originally a single-model architecture

## What i built

- Added a multi-horizon, multi-source forecasting model
- Fused 3 data sources into one pipeline
- 36-hour continuous multi-horizon forecasts

# Temporal Fusion Transformer (TFT)

- Combines LSTM encoders + attention layer to model complex time series data
- An attention mechanism to dynamically weigh the importance of different variables
- Handles three feature types: static, time-varying known, and time-varying unknown
- Interpretability



# Building the Data Pipeline

**UK PV Data**

Hugging Face dataset
Hourly generation values
2018 – 2021

**Site Metadata**

Lat, lon, tilt
Orientation, capacity
Per installation

**NWP Weather**

ECMWF IFS model
14 weather variables
Init time + step

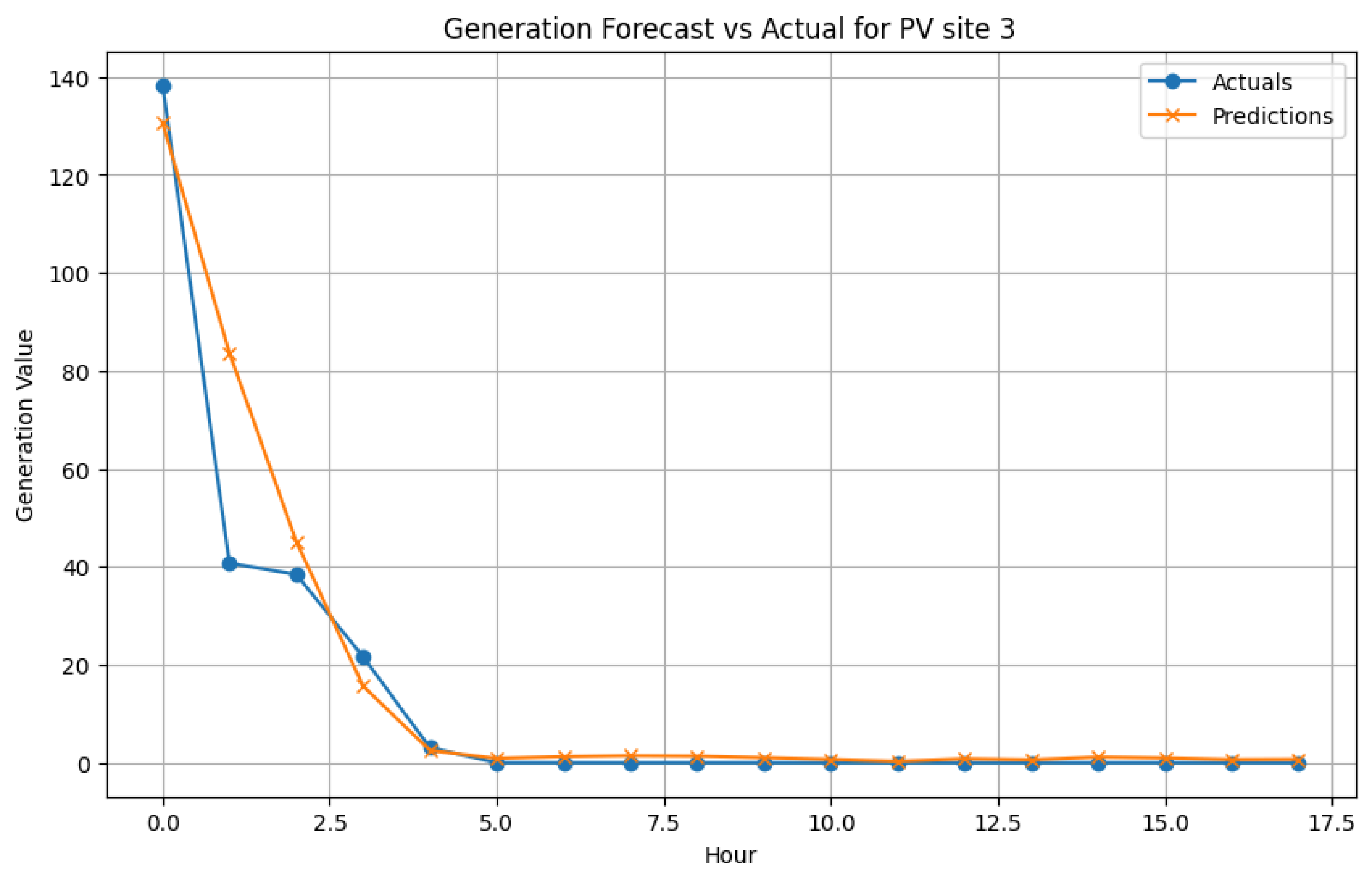
**Merged Dataset — 36-hour continuous batches, no nulls**

ss_id · pv_datetime · generation · horizon (1→36) · lat/lon/tilt/orientation/capacity · init_time · step · dlwrf · dswrf · duvrs · hcc · lcc · t2m · u10 · v10 · month · day_of_week · day_hour

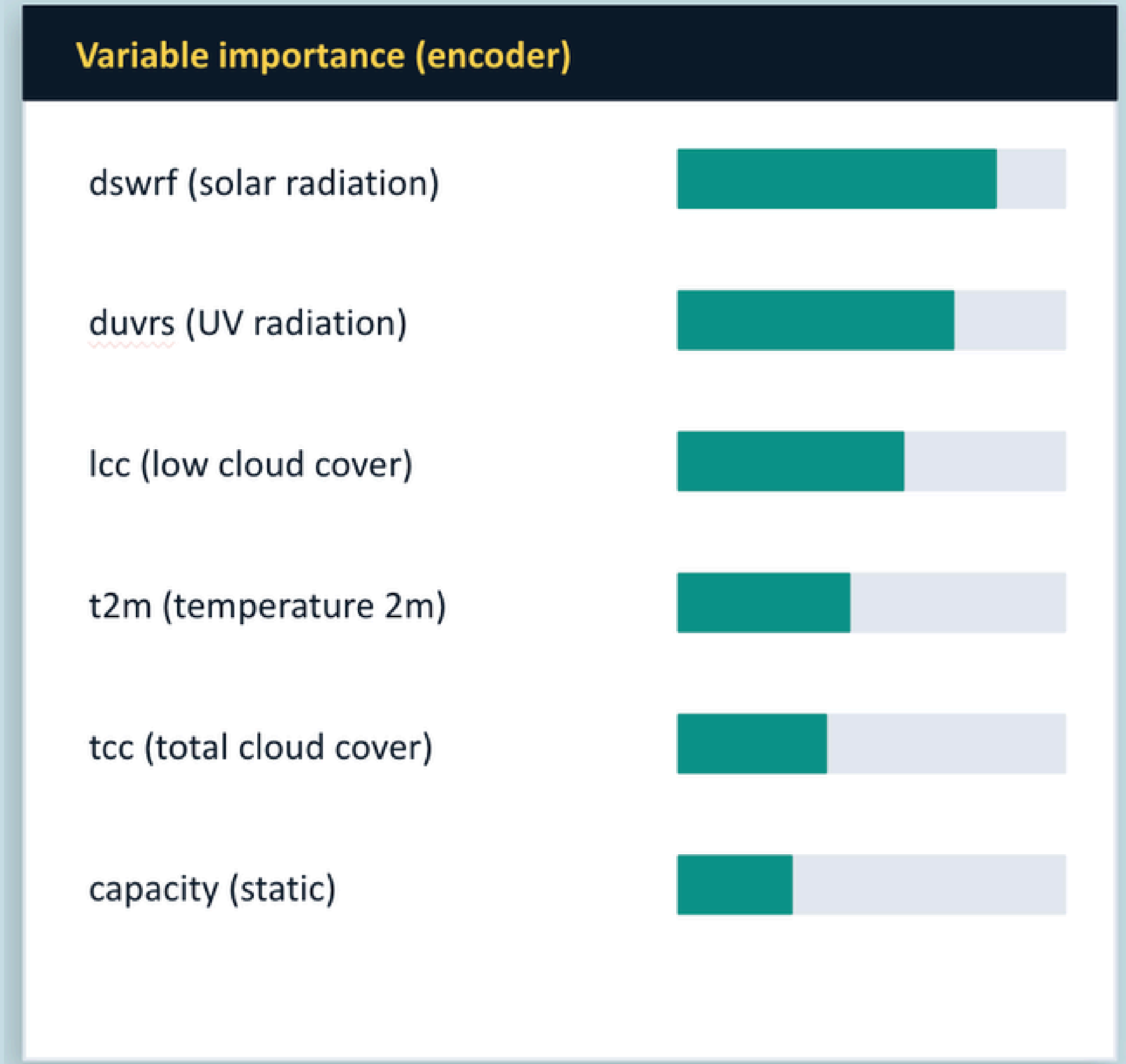
# Model Implementation

- The TFT model implementation used PyTorch Forecasting with PyTorch Lightning as the training framework.
- Learning rate was optimised using PyTorch Lightning Tuner's lr_find method
- Early stopping to prevent overfitting
- Gradient clipping set to 0.1

# Results



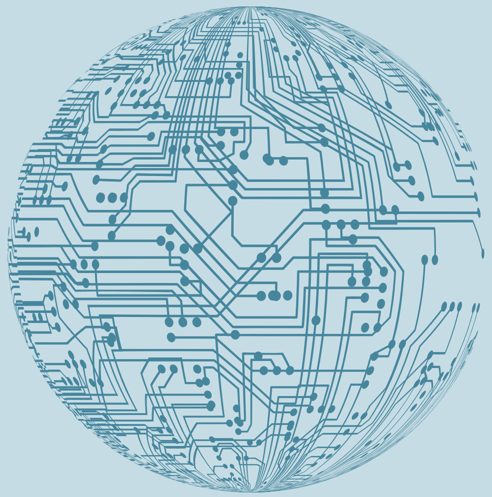
Generation Forecast vs Actual for PV site 3
Actuals
Predictions
Generation Value
Hour
0
20
40
60
80
100
120
140
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

Variable importance (encoder)
dswrf (solar radiation)
duvrs (UV radiation)
lcc (low cloud cover)
t2m (temperature 2m)
tcc (total cloud cover)
capacity (static)

# Key Takeaways

1. PyTorch Forecasting abstracts away a lot of TFT's complexity.
2. Data quality over model complexity — continuous windows, zero nulls, correct horizon column
3. Classify every feature — static, time-varying known, time-varying unknown — wrong classification silently kills performance
4. Always run lr_find and set gradient_clip_val=0.1



# Thank you

Project Code: github.com/openclimatefix/gsoc-open-quartz

Organization: openclimatefix.org

Open Quartz Solar Forecast: github.com/openclimatefix/Open-Source-Quartz-Solar-Forecast