REPORT

NOWCASTING

Progress Report: WP2

A General Overview of the Progress Made during Work Package 2 of the NIA Project “Solar PV Nowcasting”.

OPEN CLIMATE FIX

Prepared by	Open Climate Fix Limited
Reference	NIA21-WP2-3
Version	v1.0.0
Part of	Network Innovation Allowance Project “Solar PV Nowcasting”
Date of Issue	June 30, 2021
Project Number	NIA2_NGESO002

Acronyms

API	Application Programming Interface
CNN	Convolutional neural network
EUMETSAT	European Organisation for the Exploitation of Meteorological Satellites
GPU	Graphics Processing Unit
GSP	Grid Supply Point
HRV	The High Resolution Visible channel of EUMETSAT satellite imagery
MAE	Mean Absolute Error
ML	Machine Learning
MW	Megawatt
NG-ESO	National Grid Electricity System Operator
NMAE	Normalised Mean Absolute Error
NWP	Numerical Weather Prediction
OCF	Open Climate Fix
PV	Photovoltaic
PEF	Platform for Energy Forecasting
REST	Representational State Transfer
TB	Terabyte
UI	User Interface
UX	User Experience
WP {1, 2, 3}	Work Package 1, 2, or 3 (e.g. "WP2" denotes "Work Package 2")

Table 1. Table of common acronyms used in this document.

Executive Summary

Work Package 2 (WP2) spanned six months (from 1 January to 30 June 2022) and was a critical phase of the delivery of The Network Innovation Allowance Project "Solar PV Nowcasting". Over that period, Open Climate Fix has pulled together work on machine learning research, UX & UI development, open source community engagement and infrastructure to deliver a first version of what we believe will be a truly transformative PV forecast service for the UK.

WP1 allowed us to research the fundamental ML models, gather the basic data we required and to create a template for the deployment infrastructure. In WP2, we have continued the work on the ML modelling, resulting in a further almost 20% accuracy improvement from WP1. New areas such as the creation of the UI & UX commenced in WP2, culminating in the first release of the UI to NG-ESO ready for the start of July. The open-source community engagement accelerated in WP2, with Open Climate Fix's involvement in a machine learning competition being a highlight.

This report summarises the work conducted through the work package. This report is primarily summarising the progress made. The sister report WP2-1: Feasibility of Running ML-powered PV Nowcasts in Real-Time covers the forward looking outlook for a production service.

Key Deliverables and Objectives of WP2

WP2 had three key deliverables:

- WP2-1: Feasibility of Running ML-powered PV Nowcasts in Real-Time
  A forward-looking report considering challenges and opportunities surrounding running of real-time ML-powered PV Nowcasts
- WP2-2 Access to a Nowcasting Service Prototype web UI & PV Nowcasting service running in real-time.
- WP2-3: Progress Report: WP2 (this report)
  A report providing a summary of key achievements of WP2

The main objective of WP2 was to draw on the ML research and architecture design foundations laid in WP1 to productionize a high-performing ML model at the time and develop an operational prototype PV Nowcasting Service for the NG-ESO control room.

Another objective was to continue the ML research to improve the skill of the solar PV Nowcasting, as described in the first section below.

Machine Learning Research Progress

Introduction

At OCF, we are highly motivated to build the most skillful solar forecasting system we can (and, hopefully, to build the *best* solar nowcasting system in the world!) The more skillful the forecasts are, the greater the chances of reducing emissions of carbon dioxide and reducing costs for end-users.

As per the Future Work Section in [WP1-1: Research Report: PV Nowcasting Using Deep Learning Research], during WP2 we wanted to experiment with many ideas to improve the skill of the solar PV Nowcast. These ideas are explained briefly in the sections below.

We will continue to improve our forecasting models throughout WP3. We are excited about many ideas we have for how to improve performance. And we will adapt our research priorities in response to feedback from NG-ESO.

During WP2, we worked on two models: a relatively low-risk model named "PVNet" which is an evolution of the CNN we developed in WP1. And a high-ambition but more experimental model we named "Power Perceiver". The Power Perceiver model shows higher performance at the end of WP2, and has further potential for improvement. These two models are briefly described below.

The PVNet model is running in production now. The Power Perceiver will be put into production in early WP3.

In total, we conducted over 1,000 ML experiments over WP2. One common feature of PVNet and Power Perceiver is that both predict 8 hours into the future (the models we built in WP1 only predicted 2 hours into the future).

Data Used to Train and Evaluate ML Models

We used the same training and evaluation data as the data we used in WP1, to allow us to compare the WP2 models with the WP1 models. The models are trained on data from 2020, and evaluated on data from 2021. Every example is centred spatially on the "centroid" of a grid supply point region. We have written a new data processing engine which allows it to sample more freely from the training data, which reduces overfitting, and allows for more rapid experimentation.

PVNet Model

This model takes both satellite and NWP video data and puts them through separate 3D convolutional neural networks. These are then connected with a few fully connected layers, joined with some simple input data like historic PV data (see Figure

10). In addition, datetime features are added, and the position of the Sun is also used ('elevation' and 'azimuth' angles).

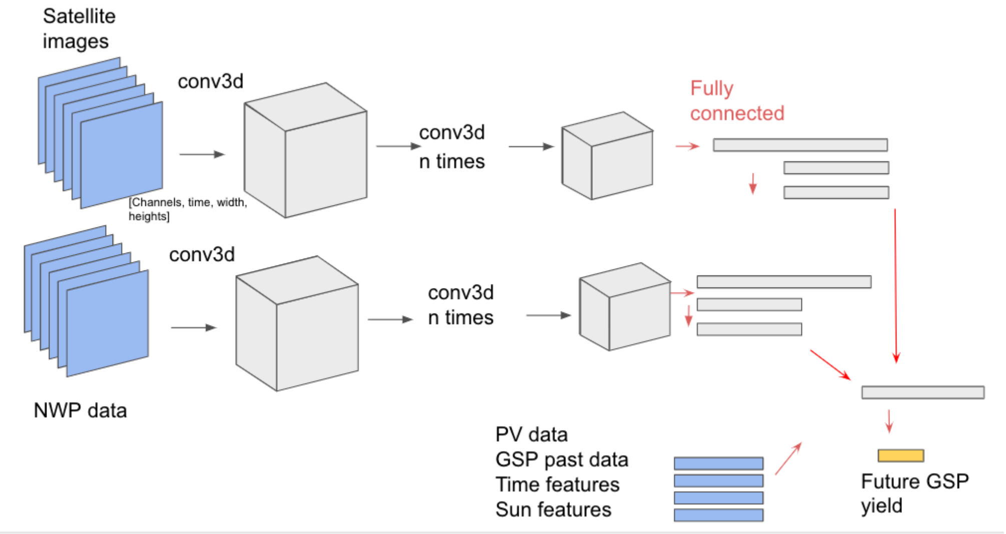

Figure 1. Illustration of convolution 3D network. Satellite and NWP are passed through several convolution neural network layers and then connected with some fully connected layers. Additional inputs like the PV past data are then input into the model to finally predict the GSP-level solar power generation.

New Power Perceiver Model

The "Power Perceiver" model was developed from scratch for WP2 and uses some of the most recent developments in machine learning, including the outputs of the ML competition addressing OCF's challenge. It consists of three main components:

1. The "Satellite Predictor": This predicts the next two hours of high-resolution visible (HRV) satellite imagery from the last hour of HRV satellite imagery, all at five minute intervals. This U-Net was developed by Jacob Levine, Ajay Arasanipalai and Jatin Mathur from University of Illinois Urbana-Champaign - the winning team of the ClimateHack.AI machine learning competition (more here). Their code is on GitHub[1]. OCF has slightly adapted their model to better fit with the subsequent two components of OCF's Power Perceiver model.
2. The "Satellite Transformer": This is a "Transformer encoder" (Vaswani *et al.* 2017[2]) which takes inspiration from the Vision Transformer (ViT) (Dosovitskiy *et al.* 2021[3]). Our "satellite transformer" operates on a single five-minutely timestep at a time. It operates across both the history and the "future" timesteps. It receives:
    a. A timestep of HRV satellite imagery (each "patch" is 4×4 pixels).
    b. Information for up to 16 PV systems. If this is a "history" timestep then this includes observed PV power for that PV system.

[1] https://github.com/jmather625/climatehack
[2] "Attention is all you need" by Vaswani *et al.*: https://arxiv.org/abs/1706.03762
[3] "An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale": https://arxiv.org/abs/2010.11929

   c. Information about the grid supply point region for this example (as with PV, if this is a historical timestep then this element includes estimated total GSP power from Sheffield Solar PV_Live).
   d. Every input element is concatenated with the Fourier encoding of the relative position in time and space, and the Sun's azimuth and elevation.
3. The "Time Transformer": This operates across time. It receives the PV and GSP elements from the "satellite transformer". It also receives numerical weather predictions and a set of half-hourly queries for GSP. There is a small recurrent neural network between the "Satellite Transformer" and the "Time Transformer". The final output goes through a mixture density network to produce probabilistic predictions for PV & GSP power.

Example Results from Power Perceiver

Below is an example GSP PV power forecast from our Power Perceiver model (in the upper-left sub-plot). The black "smudge" shows the predicted probability density. Note that the model has identified that this is a cloudy day (see the satellite images in the bottom two sub-plots), and so the prediction is fairly uncertain.

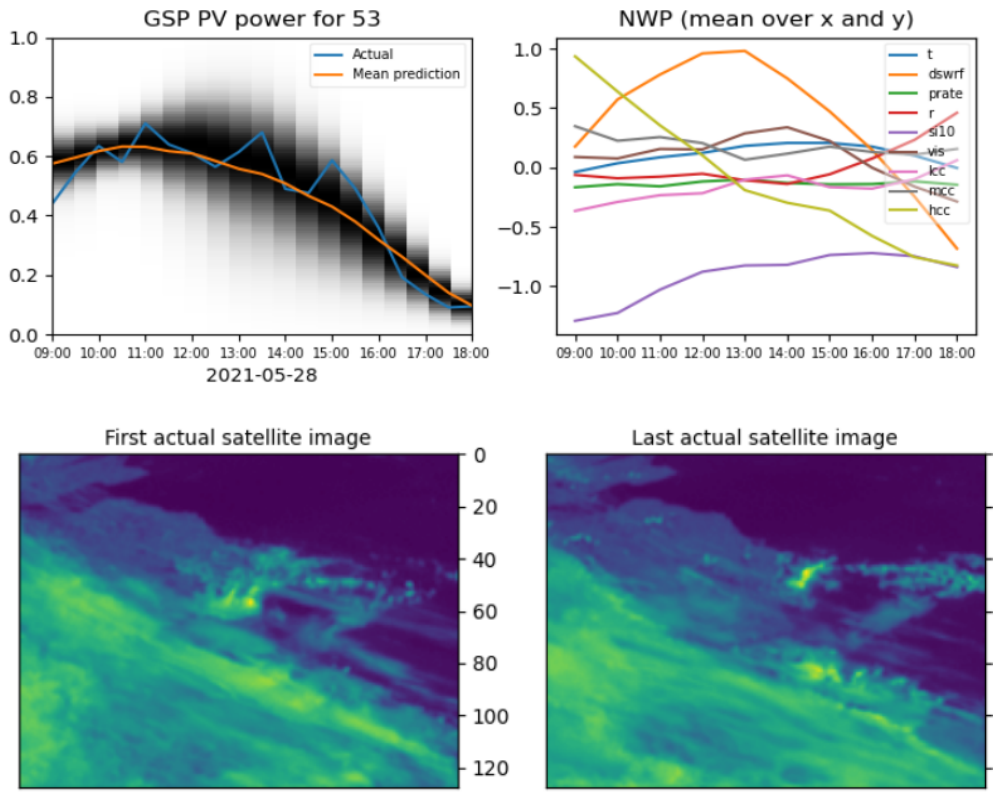

Figure 2. Probabilistic predictions of GSP PV yield from the Power Perceiver model on a cloudy day (28th May 2021). Top left: the actual GSP PV power in blue and the mean prediction in orange. The probability density is in black. Top right: The numerical weather predictions fed into the model. Bottom row: actual satellite images for the forecast period.

In contrast, here is a GSP PV power prediction for a clear day. Note that the probability density is far "sharper" than for the cloudy day. (Please ignore the first hour of the GSP timeseries in these plots. The first hour is the "recent history" which is not included in the objective function.)

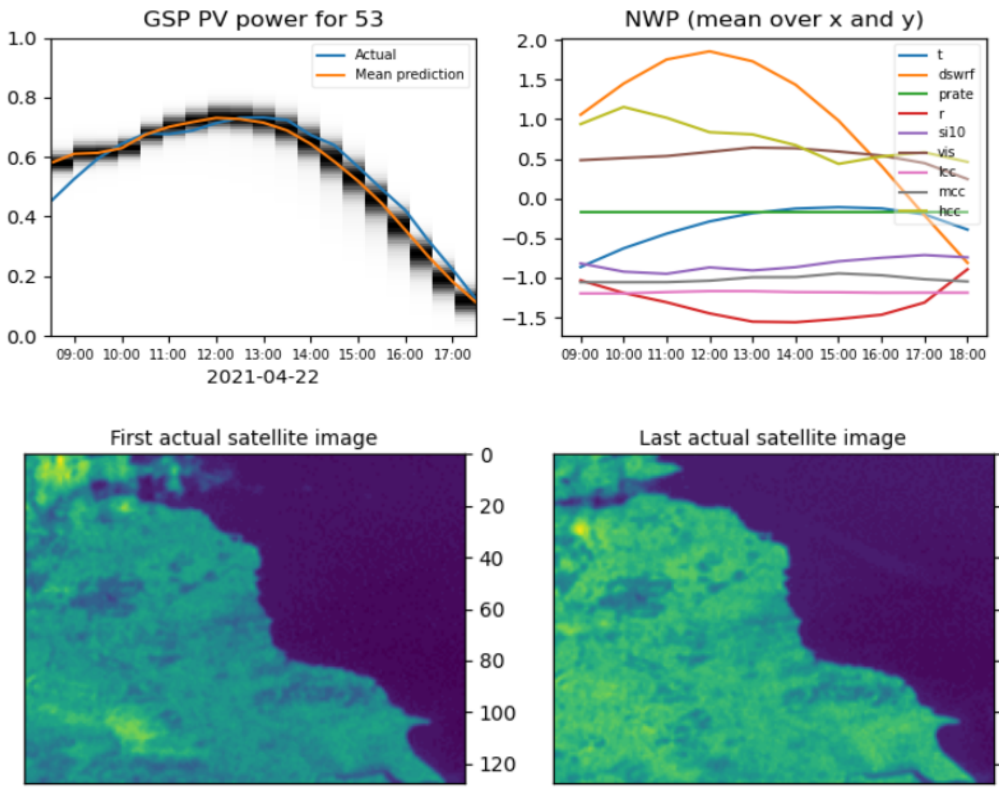

Figure 3. Probabilistic PV predictions for a single GSP on a clear day (22nd April 2021).

Comparison with NG-ESO's PV Forecasts

Table 2 shows the updated performance metrics for OCF's solar forecasting models.

We trained all our ML models on data from 2020 and tested it on data from 2021. We ignore nighttime during validation. Power Perceiver is validated in a slightly different way to previous models because we found that the performance varies a lot for different times, so Power Perceiver is validated across many more times (from 2021) than the previous models. In WP3, we plan to do a more detailed comparison of the models.

	Model Name	National Mean Absolute Error [MW]	GSP Normalised Mean Absolute Error [%]
Baselines	Baseline (use yesterday as today's forecast)	1,135.5	14.82
	NG-ESO	649.8	9.96
OCF WP1	OCF PerceiverIO	308.2	7.07
	OCF Perceiver	296.3	7.01
	OCF CNN (WP1)	276.4	7.03
	OCF OptiFlow	232.6	6.34
OCF WP2	OCF PVNet	To be validated in WP3	To be validated in WP3
	OCF Power Perceiver	**190.0**	**3.13**

Table 2. The national and GSP-level summary metrics for a 2-hour forecast horizon. Both MAE and NMAE are shown. Note the baseline results are made by using the solar generation outturn from one day before. Regarding PVNet: We expect the performance of PVNet to be similar to the performance of the OCF CNN from WP1.

Known Limitations & Planned Solutions

We are aware of several limitations of the existing models that we will aim to solve in early WP3. We are passionate about continuing to improve the performance of our solar PV forecasting models during WP3. Some limitations and proposed solutions are described below.

Explicitly Predict National Solar PV

Limitation of WP2's models: Table 2 shows something curious: Our best model at the end of WP2 *halves* the error for GSP-level forecasts compared to OCF's best model in WP1, but "only" reduces the *national* forecast error to about 80% of the error our best model from WP1 (from 233 MW to 190 MW). If we've halved the error for GSP-level forecasts, why haven't we *also* halved the *national* PV forecast error?

At the moment, our ML models predict PV power for each GSP individually. Our code adds up these predictions of GSP PV power to get a national forecast. The model predicts one GSP per "forward pass" through the model.

We had hoped that predicting individual GSPs and summing them would result in a skillful national PV forecast. Our reasoning was that, if the GSP PV forecast errors are normally distributed, then the errors should cancel out when summed to generate a national forecast. But, after seeing the results, we now realise that the errors are *not* normally distributed! Instead, the model appears to systematically over-predict or under-predict GSP PV power in particular weather regimes. Weather regimes tend to span many GSP regions at a time. Which can result in multiple GSP forecasts all being a little bit wrong *in the same direction*. These errors add up and produce a relatively large error in the *national* PV forecast.

The model has no concept of "national PV" during training, and so it is unaware that systematically under-predicting or over-predicting GSP PV power is bad.

That said, our existing models are already significantly better than ESO's old national PV forecast. But we think there's headroom to do even better in WP3 by explicitly predicting national PV (as well as GSP-level PV).

Proposed solution for WP3: We plan to train an ML model by adding up the GSP PV power in the objective function, and explicitly reward the model for producing a good *national* PV forecast whilst ensuring that GSP PV power adds up to national PV power (more detail here and here.) There are some tricky engineering challenges to address because the resulting model will need to see a large quantity of data to predict national PV in one go. But we are optimistic that we can overcome these challenges.

Give the Models More Recent History at Inference Time

Limitation of WP2's models: Our existing models only receive up to two hours of history at inference time. This seems to limit the models' ability to calibrate to recent history.

Proposed solution for WP3: We will experiment with giving the models at least 24 hours of recent history. This will require a small amount of re-engineering of the models because the current models cannot accept more than two hours of history. If we have time, we may also experiment with enabling the model to find "similar" weather regimes from the entire history, and use these selections to help inform the forecast (inspired by the way the control room use a few "similar" days to help forecast demand!). (More info here)

The Power Perceiver Model Takes a Long Time to Train

Limitation of WP2's models: The Power Perceiver model takes multiple days to train on two fast GPUs. This slows down model development. This is not a significant impediment to running a production system as retraining will only need to be performed every few months.

Proposed solution for WP3: Improve the speed of the data pipeline, and reduce the complexity of the models.

Try Predicting Abstract Features of Satellite Images Instead of Pixels

Limitation of WP2's models: Power Perceiver explicitly predicts satellite imagery for the next two hours, at five minute intervals, at full spatial resolution. This is probably far too onerous! It is actually really hard to predict *exactly* where clouds will move over the course of a few hours! And, crucially, it probably isn't *necessary*, and may be hurting the model's performance because we're asking it to do something that physically isn't possible! So the model throws up its hands and blurs the predicted satellite images.

Proposed solution for WP3: Use contrastive predictive coding (Hénaff *et al.* 2020) to encourage the model to learn encoders which are most informative for predicting the future *latent* state of the encoder, and most informative for predicting PV, rather than forcing the model to predict *every pixel* of satellite imagery.

Communicate Uncertainty

Limitation of WP2's UI: Even though our ML models create a probabilistic forecast, the probability densities are not communicated to the user. This was a deliberate decision: We wanted to start with a minimal viable UI, and iterate from there.

Proposed solution for WP3: Explore ways to communicate uncertainty to the user.

Forecasts Produced At Night Are Not Very Skillful

Limitation of WP2's models: When our current version of PVNet runs at night, the forecasts are not very skillful, because the model was not trained on night time forecast initialization times.

Proposed solution for WP3: For the alpha release in July we will filter out forecasts made when the sun is below the horizon. In the medium-term, we will retrain PVNet with night time initialization times which should improve the skill significantly. Longer-term we plan on replacing PVNet with the Power Perceiver, which will handle nighttime better.

Improve NWP-based PV Forecasts

Limitation of WP2's models: Beyond a few hours into the future, our models have to use numerical weather predictions. At the moment, our models use the "cloud cover" and "irradiance" diagnostics present in most NWPs. But these are well known to not be very accurate.

Proposed solution for WP3: Attempt an encoder of NWP humidity & temperature at cloud altitude that is maximally informative for the latent state of the "satellite image encoder", and maximally informative for predicting PV power. We hope this will allow us to predict many hours into the future, more skillfully than existing models.

Numerical Weather Predictions Take a Few Hours to Compute

Limitation of WP2's models: NWPs take a few hours to compute on huge super-computers. This means the forecasts are already "stale", even as soon as the model has finished running.

Proposed solution in WP3: This is highly experimental, and may not be possible if we do not have time, but we would love to further explore a "graph neural networks" approach (based on Ryan Keisler's 2022 "Forecasting Global Weather with Graph Neural Networks" paper[4]) to predict weather days ahead.

[4] https://arxiv.org/abs/2202.07575

Architecture of the Production System

We have implemented the following architecture:

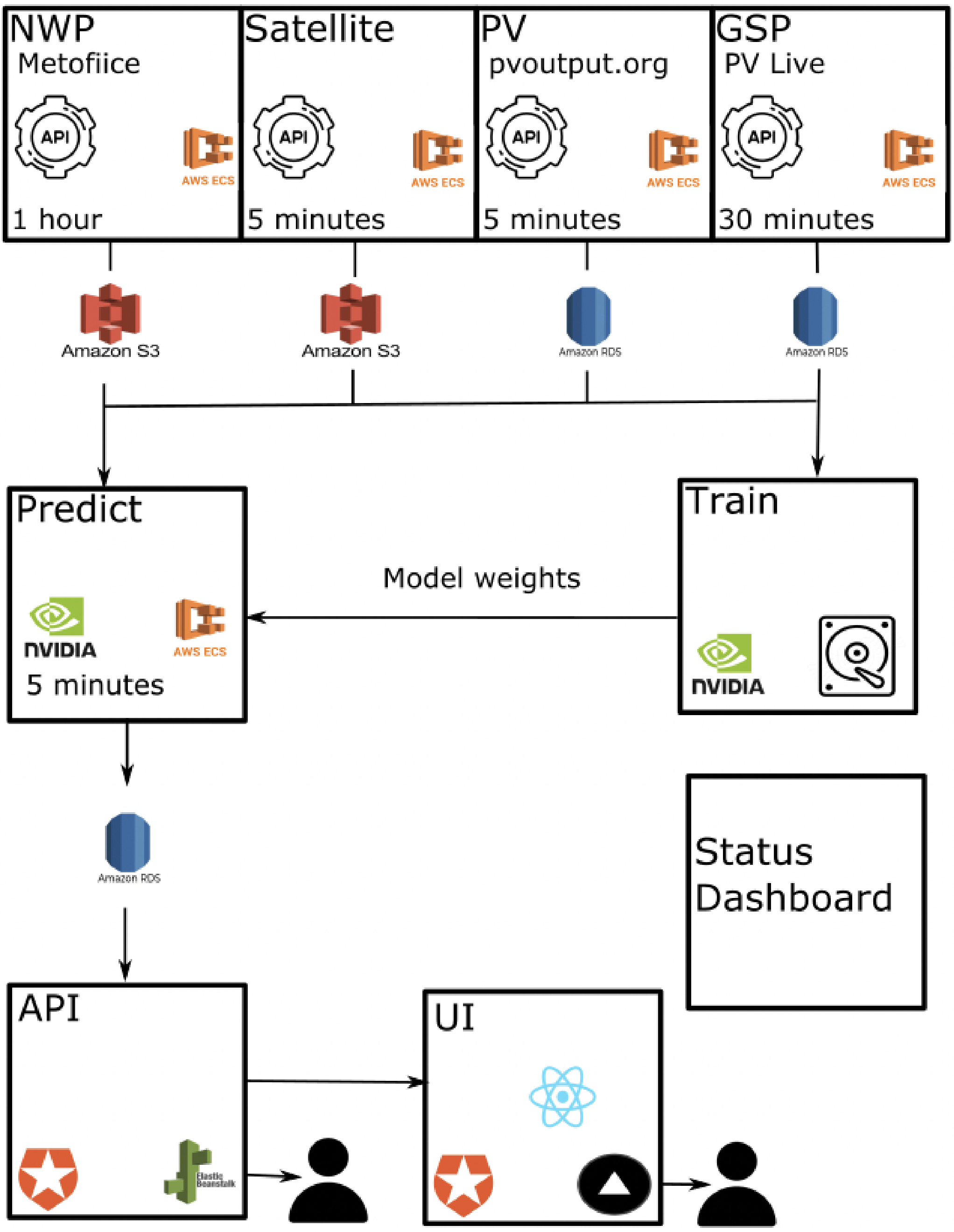

Figure 4. System architecture diagram of the OCF Nowcasting service.

Back-end and Nowcasting API

Data Consumer

We have built the following four data consumers:

- **Satellite**: This downloads satellite images from EUMETSAT and stores them in AWS s3. This occurs every five minutes and the images are approximately 15 minutes delayed from real time, although images can be up to an hour delayed.
- **NWP:** We download the latest NWP data using the new Met Office Weather Datahub API. This is also saved to s3.
- **PV:** We collect live PV data from pvoutput.org We are collecting data from approximately 1,000 PV systems from around the UK at a five minute granularity. We are also collecting live PV data from about 750 PV systems from Solar Sheffield from Passiv systems.
- **GSP:** Recent GSP Solar generation for Solar Shefield is collected every 30 minutes. These provide the training target for our ML models as well as displaying the information in our UI.

Production Process

The models we chose to take to production are run in the forecasting service which runs every five minutes. The service:

1. Loads the input data;
2. Converts it into the relevant data for each GSP location;
3. Validates the data;
4. Runs ML models;
5. Forecasts are saved in a database;
6. Post-processes results: small changes, for example making sure the forecasts are zero during the night.

As mentioned in the section covering machine learning research above, we decided to first put the PVNet model into production and are also in the process of preparing the Power Perceiver model for production.

API

We have built a REST API that serves the forecasts. This will allow users to access the numerical forecast. Currently, only live forecasts are exposed in the API. The documentation for the API is automatically produced using the OpenAPI format. The following routes are exposed:

API Endpoint	Description

/docs/	Automatic documentation where you can also try out the different routes
/v0/GB/solar/gsp/forecast/one_gsp/{gsp_id}	Get the latest forecast for yesterday and today, for one GSP
/v0/GB/solar/gsp/forecast/latest/{gsp_id}	Get the latest forecast for yesterday and today, for one GSP
/v0/GB/solar/gsp/forecast/all	Get the forecast for all GSPs
/v0/GB/solar/gsp/forecast/national	Get the national forecast
/v0/GB/solar/gsp/truth/one_gsp/{gsp_id}	Get the PV_Live values for a specified GSP
/v0/GB/solar/gsp/gsp_boundaries	Get the GSP boundaries
/v0/GB/solar/gsp/gsp_systems	Get all the GSP metadata
/v0/GB/solar/status	Get the status of the system

Table 3. Documentation for the API.

A key task was to speed up the API and underlying database. This was done using efficient SQL queries and optimisation of the data tables. We also save relevant data at a top level, rather than using complex SQL queries, as this takes slightly longer to save data, but speeds up loading times.

API Security

As part of WP2, OCF started engaging with NG-ESO Digital Risk and Security (DR&S) to understand the security requirements for a possible future business as usual version of Nowcasting. OCF will need to comply with both general security requirements as well as specific SaaS security requirements imposed by NG-ESO. Based on these early conversations, it was concluded that OCF already satisfies some of these requirements and more work in this area will be performed during WP3.

UI Development

It was clear from the initial project scope proposed to NG-ESO that the immediate benefit of the forecast would be realised in being able to deliver a real-time forecast into the control room through a graphical web UI. OCF will deliver a web UI accessible on the public internet through appropriate security controls.

From OCF's experience with user interface design, we appreciate the importance of the UI displaying key information clearly, allowing understanding below the headline figures, and delivering a user-friendly experience. Given these goals, we engaged a User Experience (UX) designer with experience in designing web user interfaces for global web applications.

Requirements Gathering & Refinement

Early in WP2, OCF embarked upon user interviews with the key stakeholders and users of the forecast: Operational Manager at ESO Control Room - Alex Carter, Lead Data Scientist - Lyndon Ruff and Senior Modelling Specialist - Dan Drew. OCF drew up design ideas and conducted a workshop onsite at NG-ESO with Alex and Lyndon in early April, which was used to drive the key features required for WP2.

OCF has developed the UI in an agile manner with feedback internally and with one further review session with NG-ESO conducted in June.

UI Features

In WP2 Alpha Delivery

The following features were incorporated in the alpha delivery stage:

- Headline national forecast figures for PV_Live and current forecasts (displayed without needing to pick up a mouse);
- GSP level information displayed graphically to allow easy inspection;
- Drilldown to GSP actual results and forecasts;
- Today and yesterday as a minimum display period - user chooses time to view;
- Security: Authentication is performed via Auth0. This allows users to log in safely with their email address and their password. OCF controls which users are allowed to log on.

Proposed WP3 Beta Delivery

We have the following features in our backlog, but will continue to work with users to determine which features should be scheduled:

- Time slider to display different periods of history;
- "Delta view" where user can see the difference between the last used forecast (usually 4 hours before) and the most current forecast;
- Various smaller graphical tweaks.

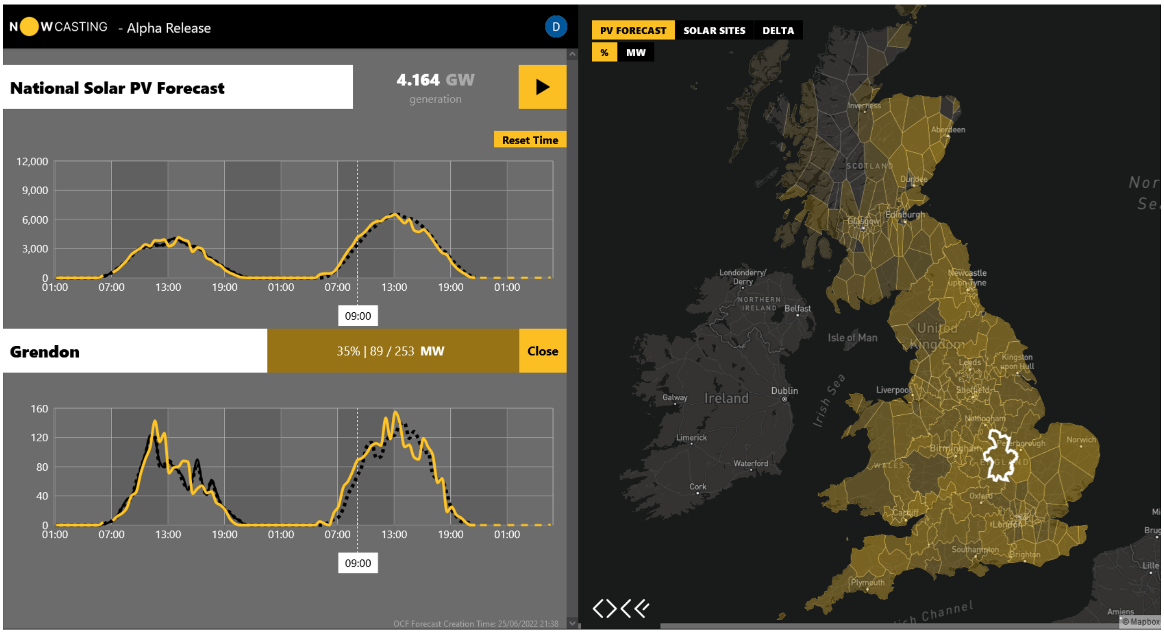

Figure 5. UI of the OCF Nowcasting service in its current stage of development.

Open Source

We have engaged with the open source community on various levels, including through the [ClimateHack.AI] competition, releasing data and models on the [Hugging Face Hub], engaging with public contributors on GitHub, and collaborating with academic researchers.

ClimateHack.AI was a two-month hackathon involving 132 participants from 25 top universities in the UK, US and Canada including Harvard, MIT, Stanford, Princeton, Oxford and Cambridge, who were tasked with a challenge to better predict future satellite imagery. The best performing model, from a team at the University of Illinois Urbana-Champaign, outperformed our current satellite image prediction model, and has been adapted into our current best performing PV forecasting model, Power Perceiver. This primarily involved making the model slightly smaller so that it would run quicker and fit better on the GPUs we use for training.

We have so far had over 50 external contributors to our codebase, helping with aspects from documentation to tests and bug fixes, and expanding functionality.

To help spur further open development and research, we've released one of the largest ever sets of PV training data on the Hugging Face platform, where anyone can access the individual PV site readings, corresponding satellite imagery and weather forecasts. We have also released the trained models that we have in current use for others to finetune or use as desired. We have released other related datasets in formats ready for machine learning practitioners to use, such as numerical weather predictions, and others that do or can help us continue to refine and improve our forecasting models. These datasets have already been downloaded tens to hundreds of times, and our models used for other forecasting problems by researchers in Sweden, Taiwan, and other countries around the world. In total, we have made over 8TB of weather, satellite, and PV data publicly available in formats that are easy for ourselves and others to use in training machine learning models. No dataset of comparable size has been released before for training PV forecast models.

Governance

In order to ensure we are getting feedback from National Grid ESO and maximising the potential for the project to hit its goals, we established a Working Steering Committee and conducted the first meeting of this group in May 2022. We aim to have quarterly meetings for the remainder of the project. Meetings are minuted.

National Grid ESO	Open Climate Fix
Carolina Tortora (Head of Innovation at NG-ESO)	Kasia Krasucka (Programme Manager)
Lyndon Ruff (Lead Data Scientist)	Dan Travers (OCF co-founder & Energy Markets Specialist)
Alex Carter (Operational Manager)	Jack Kelly (OCF co-founder & ML Research Engineer)
Charlotte Horne (Innovation Business Partner)	Peter Dudfield (ML & Fullstack Engineer)
Sumit Gumber (Product Manager for Forecasting for NG-ESO)	
Dan Drew (Senior Modelling Specialist)	
Oliwia Milek (Energy Forecasting Manager)	
Roya Ahmadi (Energy Balancing Transformation Lead)	

Table 4. Participants of the first Working Seeting Committee meeting that took place on 26 May 2022.

Lessons Learned

This section of the report considers key lessons identified during WP2. Some of them informed changes adopted already in WP2 and some will inform activity planning for WP3. They may also prove useful for future projects.

- **Having infrastructure as code allows the main production service to run uninterrupted**
  Having code to easily instantiate infrastructure is very useful for the efficient management of environments to ensure we can bring the algorithm into productive use. We used the Terraform software tool which makes spinning up (and down) environments very easy and repeatable. Being able to spin up new environments allows us to test new features in development environments while allowing the main production to keep on running uninterrupted.

- **Using microservices to “start simple and iterate” accelerates development**
  Using a microservice architecture allows us to upgrade individual components as we see benefit in improving them, independently of changing other components’ behaviour. This has been very useful when building out the prototype service, as it has allowed us to start with a simple architecture - even a trivial forecast model - and iteratively improve the functionality in the components. For example, first starting out with one PV provider of data has allowed us to get the prototype working, and in WP3 we will expand to onboard an additional PV provider.

- **Data processing may take longer than expected**
  While we initially planned to extend our dataset back to 2016 for all data sources during WP2, it turned out that data processing takes much longer than expected. This does not have a direct impact on project deliverables but is something to consider in further ML research.

- **Data validation is important**
  For both ML training and inference, using clear and simple data validation to build trust in the data we are using. This helps build a reliable production system and keeps software bugs at a minimum.

- **Engaging specialist UX/UI skills is important**
  By acknowledging that UX and UI design is a specialised area and incorporating those skills into the team, we believe we have developed a UI which will be easier to use and convey information effectively. We will aim to validate this over WP3 through working with the end users.

- **Building our own hardware demonstrates value for money but may pose other challenges for a small team**
  We have built two computers with a total of six GPUs and we estimate that using on-premises hardware instead of the cloud for data-heavy & GPU heavy machine learning R&D can significantly reduce the direct costs (savings of up to 80% for a two-year lifecycle even when including labour costs). However, we have underestimated the time it would require for our team to put together all the components (about 25 days for one person in total). While the total costs would still be much lower, should we want to upgrade our hardware in the future, appropriate

resource planning should be taken into account e.g. it might be a good idea to hire a contractor to help set up hardware.