Does explicit time series modelling improve short-term solar photovoltaic energy forecasts?

Exam candidate code
SNQJ3

MSc Computer Science

Internal supervisors
Dr Maria Perez Ortiz and Professor John Shawe-Taylor

External supervisor
Jacob Bieker, Open Climate Fix

Submission date: 12 September 2022

[1]**Disclaimer:** This report is submitted as part requirement for the MSc Computer Science at UCL. It is substantially the result of my own work except where explicitly indicated in the text.

Abstract

Anthropogenic climate change has caused more frequent and more intense weather events. Reaching levels of warming in the next few decades above 1.5C would increase multiple hazards and lead to significant humanitarian harms. Reducing emissions in line with climate goals requires the rapid integration of low carbon energy sources into the electricity grid. Solar photovolatics (PV) convert light to energy using solar panels. They are a low carbon energy source but the energy they produce can be uncertain due to clouds in the sky obscuring the sun's energy, making it difficult to integrate them into electricity grids. Solar PV forecasting between 30 minutes and 8 hours is used to help anticipate peaks and troughs in demand to support grid integration.

This dissertation explores different machine learning approaches to time series forecasting of sequences using a dataset of UK satellite imagery and solar PV energy readings over a 1- to 4-hour time range. Convolutional neural networks, or CNNs, were developed for image processing, but can also be used for dealing with sequences as a block. Recurrent neural networks, or RNNs, such as long short-term memory (LSTM) models, deal explicitly with sequences by carrying forward state information. This report assesses whether explicitly representing the task as a recurrent time series rather than using a three-dimensional convolution leads to greater predictive accuracy.

Weather forecasting includes both stochastic (e.g. clouds) and deterministic (e.g. brightness) features, which should work well in an LSTM architecture due to the combination of long- and short-term information. This report finds that convolutional and recurrent models tend to perform slightly better than purely convolutional models, most significantly when taking satellite images as input, perhaps because they contain relevant long-term information. This dissertation also explores the prediction errors, CNN activations, and estimates the carbon emissions generated by and averted by the use of different model architectures. Finally, it considers the implications for machine learning and solar PV forecasting more generally. The code for the project is made available open source in the appendix.

Video presentation

The project presentation is available at `https://www.youtube.com/watch?v=r7wUlhLoIRs`.

Acknowledgements

Thanks to my internal and external supervisors, Dr Maria Perez Ortiz and Professor John Shawe-Taylor at UCL, and Jacob Bieker at Open Climate Fix, for all of their helpful feedback, suggestions, and encouragement. I am grateful to Jack Kelly and the rest of the organisation at Open Climate Fix and their commitment to open source data and code, which made this research possible. I am also grateful to the open source community, for providing free software for students and researchers everywhere. This dissertation was written using LaTeX in Overleaf, and my experiments were conducted on Google Colaboratory [22].

Note

This dissertation was done as part of my MSc Computer Science course at UCL. Previous to this research, my only experience of machine learning was in COMP0142 Machine Learning for Domain Specialists, and I had not looked at neural networks, CNNs, RNNs, or LSTMs before.

Glossary

Term	Meaning
ANN	Artificial neural network - A function comprised of multiple nodes which learn different weights, see 2.1.
CNN	Convolutional neural networks - a class of neural network designed to work with image data by searching for particular patterns, see 2.1
Conv3D	A convolution over a three-dimensional tensor, in this case where the third dimension is time
ConvLSTM	A variant of LSTM which performs a convolution operation inside the cell, explained in detail in 2.1
GPU	Graphical processing unit - a specialised electronic circuit for processing instructions which uses a parallel structure to perform many operations simultaneously, often used in machine learning tasks
GHG	Greenhouse gas - the group of gases which increase global warming through the greenhouse effect, such as carbon dioxide and methane
LSTM	Long short-term memory - a class of neural network designed to work with sequence data and to carry information across multiple time steps, see 2.1
National Grid ESO	The energy service operator (ESO) for the UK, see 1.1
Nowcasting	Short-term forecasting, over a few hours in advance, used in this context to mean forecasting solar PV yield
NWP	Numerical weather prediction - a computationally intensive method for weather forecasting which uses multiple systems of differential equations based on physical principles
ML	Machine learning - an area of research in which computer models leverage data to improve performance on some set of tasks
OCF / Open Climate Fix	The external partner organisation for this project. One of OCF's main research areas is solar nowcasting. See 1.1
Persistence	A simple forecast made at time t, where the prediction for $t+1, t+2, ...$ is equal to the reading at $t-1$
PV	Photovoltaic - the conversion of light into energy
PV yield	The amount of energy produced through solar photovoltaics
RNN	Recurrent neural network - a class of neural network designed to work with sequence data which carries information from one time step to the next. See 2.1.
Solar altitude	The angle between the horizon and the line to the sun - zero degrees at sunset and sunrise, and at most 90 degrees at midday near the equator

Table 1: A glossary of terms used throughout the project

Contents

Chapter 1

Introduction

Chapter summary: This chapter explains why short-term solar photovoltaic forecasting is an important problem (1.1), and identifies open research questions (1.2). It defines project goals (1.3), and the project management approach taken (1.4). Finally it explain the structure of this report and how it answers the questions posed (1.5).

1.1 Problem statement: the importance and challenge of solar nowcasting

Climate change is a major global problem and humanity urgently needs to reduce emissions. As a result of anthropogenic greenhouse gas (GHG) emissions, global warming is expected to reach 1.5C in the next few decades and will cause unavoidable increases in multiple climate hazards and will present multiple risks to ecosystems and humans. But near-term actions that limit global warming to close to 1.5C would significantly reduce projected losses [25]. Global greenhouse gas emissions continue to rise, but the rate of global emissions growth each year has slowed.

Reducing emissions to safer levels of warming will require a transformation of our energy system. A range of future energy pathways have been modelled to project likely future emissions, which in turn have associated probabilities of levels of warming. All pathways associated with a greater than 50% chance of limiting warning to 1.5C, and all pathways associated with a greater than 67% chance of limiting warming to 2C necessitate rapid and deep GHG emission reductions in all sectors [26].

Providing energy from solar photovoltaics (PV) is a promising way to reduce emissions. Solar PV energy works through capturing energy from the sun's light, typically through solar panels, and transforming this into electricity to input to the grid. The environmental lifecycle costs of solar PV energy generation are far less than the environmental costs of energy generation from fossil fuels. A 2020 World Bank review [3] concluded that there was more than sufficient solar PV energy potential to meet global electricity demand. In the UK, there are currently approximately 13 terawatt-hours of solar PV supply in the UK, and this is expected to increase up to 82 terawatt-hours by 2050 [9].

However, the uncertainty in the energy obtained, or yield, from solar PV makes it difficult to integrate solar into the electricity grid. The yield of solar PV is inherently uncertain because clouds can obscure the sun, and clouds can become thicker, thinner, and their movement across the sky is hard to predict. On a sunny spring day in Great Britain, solar power can account for

30% of all the electricity produced [42]. If clouds suddenly form, in a few minutes, the power grid can lose much of the energy generated from solar panels.

The reason this uncertainty is a problem is that the grid must always be finely balanced. At each point in time, the demand and supply to the UK energy grid must be exactly matched at 50Hz [12]. If supply suddenly decreases, the energy grid operators must immediately provide backup power in order to avoid power outages. In order to guarantee stable energy supply, grid operators have to keep spinning-reserve online, because these generators take hours to spin up. These reserves are usually gas turbines. Keeping these turbines on standby causes gas to be burnt, releasing carbon dioxide into the atmosphere.

These factors create an incentive to forecast solar PV yield more accurately. If we knew what the energy coming into the grid would be, there would be less of a need to keep the reserves on standby, and every kilowatt could be used efficiently.

NationalGrid ESO is the electricity system operator for Great Britain, shown in Fig 1.1. Since 2015, 10 gigawatts of solar have come online, and NationalGrid ESO operates from the Electricity National Control Centre, where they manage over 1bn data points a day and make around 200 despatch instructions to the 300 - 400 generators that participate in the balancing market every hour [10]. Surprises have already happened. In 2019, an oversupply of PV yield on an unexpectedly sunny day led to such an excessive supply that energy prices were negative for six hours [48].

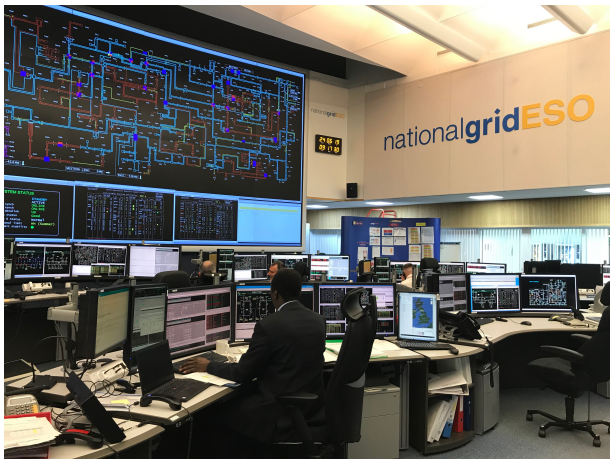


Figure 1.1: The UK's Energy National Control Centre, from [10]

Open Climate Fix have been working with National Grid ESO since 2019 to improve the accuracy of their predictions [11] over short-term horizons. This forecasting work is described as 'nowcasting' to emphasize the short-term nature of the forecasts. NationalGrid ESO anticipates that improvements in solar nowcasting could be used to optimise grid management globally.

In 2021, Carolina Tortora, head of innovation and digital transformation at National Grid ESO said "Work like this has real impact – reducing forecasting errors and the need to keep costly fossil fuel plants ticking over... Open Climate Fix's nowcasting research has potential to further improve the forecasting capabilities of electricity system operators around the world." [42].

1.2 Research opportunities

There is a need to accurately forecast PV yield with a few hours notice. Existing methods for solar nowcasting are often computationally intensive and have low accuracy, and may not incorporate

availiable information such as satellite images of the progression of clouds. Open Climate Fix have been working with ESO National Grid to develop better forecasts, incorporating the latest developments from machine learning [11].

Open Climate Fix experiments with different models to find which has the lowest mean absolute error, and then provides these models to National Grid ESO. Open Climate Fix experiment with a range of different approaches of using satellite images and other data for PV nowcasting. From 2019 to 2021, Open Climate Fix's basic production model was a convolutional neural network, or CNN. They are currently working on a power-perceiver using the the perceiver architecture [27] developed by Deepmind in 2021.

This dissertation aims to complement the PV nowcasting done by Open Climate Fix, by researching alternative methods for predicting PV yield, to examine and analyse the predictions and their errors.

1.3 Goals and aims

1.3.1 Project goals

The objective of this report is to evaluate approaches in solar PV nowcasting. This requires understanding existing work through a literature review, posing hypotheses, defining experiments to answer those hypotheses, and analysing the results.

- **Task definition**: Define a specific PV nowcasting task. Open Climate Fix have experimented with different model designs on a wide range of tasks, over different time frames and with different geographical extent, and both predicting PV values and predicting the next image frame. This research will review the main approaches used in computer vision and weather prediction and define a specific task which is feasible in the time available and provide useful information for solar PV nowcasting research. The dissertation will propose specific hypotheses to test open research questions.

- **Data collection**: Identify and validate appropriate data to assess these hypotheses. There are multiple potential datasets, such as satellite images, numerical weather predictions, and different datasets of PV readings in the UK. Select a dataset which can be used to perform the chosen task, and is representative of the real-world application of UK energy forecasting, and is computationally feasible with the time and resources available.

- **Pre-processing and data exploration**: Transform the data into a form suitable for analysis with machine learning approaches. Machine learning works best with scaled and normalised features, and there may be missing or unrepresentative data which needs to be dealt with appropriately. This section will also define how the dataset can be split into training, testing, and validation sections.

- **Model selection**: Select models to predict the PV yield over an appropriate timescale. Open Climate Fix have used a Conv3D CNN, this report reviews approaches in weather prediction, and identifies models to test experimental hypotheses.

- **Model evaluation and analysis**: Critically analyse and compare the results, and appraise this in relation to the work of Open Climate Fix and solar PV nowcasting generally. Use the models to derive predictions and compare these to true values, and see whether there are any systematic errors or patterns which indicate what the models are learning.

1.3.2 Personal aims

- Identify a valid and important computational problem, and narrow the scope such that I can address the problem in a dissertation
- Work with a combination of multiple real-world datasets
- Understand how neural networks and deep learning operate, going beyond what I have covered so far on my degree program
- Understand the main techniques and contemporary research in computer vision
- Deploy models with a high computational requirements, e.g. using GPUs and Google Colab
- Critically appraise the results and consider implications for solar PV nowcasting and machine learning applications more generally

1.4 Project management

This project and personal project goals required me to familarise myself with contemporary approaches in machine learning, computer vision, and find and complete a research project. To do this a highly interactive and experimental method was preferred.

One project management approach is a rational planning / waterfall based model, defined as taking separate stages of specification, development, validation, and testing [47]. However, a more Agile approach was chosen to create a shorter feedback loop and build up machine learning skills with simple examples.

As a result, the focus was on developing examples and implementing ideas from the literature as quickly as possible, and then checking back with the partner organisation on a regular basis. Regular updates were sent to OCF for feedback and general direction towards the first half of the project. The timing of different activities performed in the research are shown in the Table 1.1.

Project timeline

The project timeline is shown in the table below.

Description	June				July				August				Sep	
Task definition														
Initial literature review														
Scoping call with OCF														
Researched OCF models														
Decided to compare CNNs with ConvLSTMs														
Defined scientific hypotheses														
Data collection and pre-processing														
Explored climatehack dataset														
Explored satellite dataset														
Explored pv dataset														
Cropped PV and aligned datasets														
Experiments														
Same-period PV prediction with CNN														
LSTM model														
CNN model														
ConvLSTM model for single value														
CNN model for single value														
Developed persistence model														
ConvLSTM and CNN models for sequence														
Multi-input models with past PV yield														
Hyperparameter tuning														
Report writing														
Literature review														
Analysis of results														
Interpretability of CNN activations														
Comparison with existing literature														
Report writing														
Feedback from OCF														

Table 1.1: A timeline showing project progress: the top-level goals matching those in my project goals are shown in bold text with dark green shaded cells, and the specific activities are shown underneath each goal, with indented text and lighter green shaded cells. Feedback from OCF is added at the bottom.

1.5 Structure of this report

This report is divided into different sections dealing with each of these stages, as described below.

- Chapter 2 - Literature review: looks at different approaches to weather and PV nowcasting, and identify a research task that could be completed within the time available
- Chapter 3 - Datasets and preprocessing: details the sources and datasets, and the preprocessing to prepare them for analysis.
- Chapter 4 - Experimental design: explains how models from the literature are used to test the proposed hypotheses.
- Chapter 5 - Results and analysis: summarises numerical results, plots predictions against observed values, analyses accuracy, examines patterns of errors by time of day and year, visualises predictions, and interprets activations in a convolution.
- Chapter 6 - Conclusion and discussion: reviews project overall, assesses how well it as able to address the questions raised earlier, considers implications for research and solar PV nowcasting, and proposes future research directions.

The code for this project is also made available in the Appendix, through a link to the Google Drive folder.

Chapter 2

Literature review

Chapter Summary: This chapter briefly surveys the literature on time series forecasting with a focus on machine learning and computer vision (2.1). It examines how these techniques have been applied in the domain of solar PV nowcasting and related fields (2.2). Drawing on the related work, it outlines open research questions in (2.3) and develops two hypotheses to test these questions.

2.1 Time series forecasting approaches in machine learning

Polynomial regression

Forecasting problems can sometimes be dealt with as a polynomial regression. This section uses notation from Bishop [6], Chapter 3. For a simple linear regression of inputs $\mathbf{x}$ for which we minimise the weight vector $\mathbf{w}$, this would take the following form:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + ... + w_D x_D \tag{2.1}$$

If we maintain the constraint that the function is *linear in the weights* then we can extend the generality to a fixed nonlinear transformation of the inputs, and define this transformation as $\phi_j(\mathbf{x})$, which will output M transformed vectors of dimension *(1 X D)*, which we then multiply by $\sum_j^M w_j$, giving the following. We also simplify the representation of the bias term w_0 by defining a dummy basis function $\phi_0(\mathbf{x}) = 1$, which allows us to include $j = 0$ in the summation below.

$$\begin{aligned} y(\mathbf{x}, \mathbf{w}) &= \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) + w_0 \\ &= \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) \end{aligned} \tag{2.2}$$

The different possible values of $\mathbf{w}$ will give differing predictions, which we could compare to our training dataset using a loss function, such as mean squared error. However, polynomial regressions are limited in that they can only learn functions which are linear in the weights of the inputs, and they cannot learn more complex mappings of inputs to outputs.

Artificial Neural Networks

Artificial Neural Networks (ANNs) were developed to solve problems through multiple levels of abstraction, and were inspired by simplified models of biological neurons by neurophysiologist Warren McCulloch and the logician Walter Pitts [37] in 1943. An early application was the *Perceptron* in 1957 [43], and over time researchers stacked perceptrons on top of each other, with the outputs from one node forming the inputs into the next.

Each neuron works in a similar way to a polynomial regression, in that it is linear in the weights, but the difference with neural networks is that multiple neurons can be stacked on top of each other. This section follows notation used by Bishop [6], Chapter 5. The output from one neuron is transformed by an activation function, σ, which standardises the outputs to a range so they can be passed as inputs to the next layer.

The models begin by starting with a series $j = 1, ..., M$ of combinations of the input variables. If the input variables are $x_1, ..., x_D$, we can represent a neuron in the first layer as follows.

$$a_j = \sigma(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}) \tag{2.3}$$

The output of each of the j combinations is some activation, a_j. These values are then passed through an activation function, $h(.)$, such as the logistic sigmoid, σ_1, or the rectified linear function, σ_2, both shown below.

$$\begin{aligned} z_j &= \sigma_1(a_j) = \frac{1}{1 + \exp -a} \\ z_j &= \sigma_2(a_j) = \max(0, a) \end{aligned} \tag{2.4}$$

The output of the activation function is z_j, which is then input to the next layer of neurons, say $k = 1, ..., K$ neurons.

$$a_k = \sigma(\sum_{i=1}^{D} w_{ji}^{(2)} x_i + w_{j0}^{(2)}) \tag{2.5}$$

The output of the last year is $y(\mathbf{x}, \mathbf{w})$. By stacking these layers together, the model is able to more general version of the function shown in equation 2.2.

The crucial difference between ANNs and polynomial regressions is that ANNs exploit hierarchy, structure, and re-use of components. By stacking nodes on top of each other, nodes closer to the input data can learn low-level structures such as gradients of sequences or edges in images. The output of the lower nodes is then passed to the intermediate nodes. These nodes then combine intermediate-level features such as combinations of gradients in sequences, or basic shapes in images. In turn the outputs of these layers are fed into higher-level nodes, which learn features such patterns, faces, or generic images [7].

Neural networks are trained using backpropagation, developed in 1985 [44]. Backpropagation consists of repeated forward and backward passes through the data. The overall process is described below.

1. **Begin forward pass**: Take a batch of inputs from the training data and pass it through the neural network, which starts with randomly initialised weights.

2. **Calculate loss function**: At the output of the network, compare predicted outputs values and the actual output values. This comparison is captured in a *cost function*, and there are

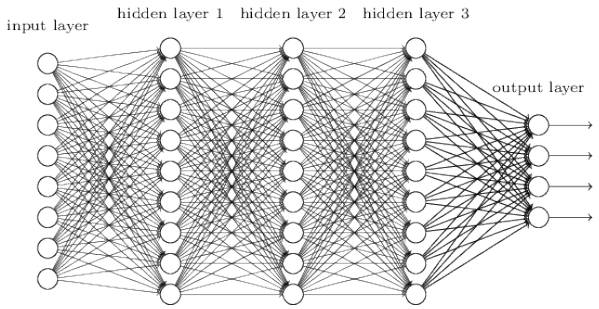


Figure 2.1: A neural network with three hidden layers, from Nielsen [38]

.

different choices on how to penalise divergences between the true value and expected values. Sum of the costs for this batch to get the *loss function*.

3. **Backward pass**: Start on the last layer. Adjust the weights in each neuron in the direction of the negative gradient, because the negative gradient is the direction of steepest descent on a surface. Adjust the weights of the previous layer in the same way. This leads to the neurons passing forward the information most helpful to the final layer. Continue until the input layer.

4. **Repeat from step 1**: As a result of passing through these steps, the weights within each node are progressively adjusted such that the predicted values are closer to the true values.

Theoretically the loss function should be calculated as the sum of the cost for the entire training set, but this would require passing through the entire training set each time, and so stochastic gradient descent only passes in a batch of inputs which can form an approximation of the loss function to use the in process described above.

Neural networks can learn a variety of general structures, but in the examples below these structures will need to be located in the same place within the input. A standard multi-layer neural network may learn patterns based on specific locations within the input. However, for images, the exact location of a feature can vary, and for this reason a different type of network is often used.

Convolutional neural networks

A convolutional neural network, or CNN is a specific form of neural network used to detect patterns and often used in image tasks, first developed in 1998 [30]. CNNs work in such a way that they are able to recognise features regardless of their locations within the input data.

CNNs have hidden layers called convolutional layers which apply a number of different filters to an input. Each filter begins as a randomly initialised matrix, which is passed over the input, where the size of the filter is specified as the kernel dimensions, such as (3x3). The filter will then pass over each area within the input which is the same size as the filter, and the dot product is computed between the value on the filter with the value of the input immediately below the filter.

A CNN begins with randomly initialised weights for each of its kernels, and it will pass over the image and generate a cost and loss function as described above in backpropagation. For a CNN, the values that can change are the weights for each filter. Over time through backpropagation this

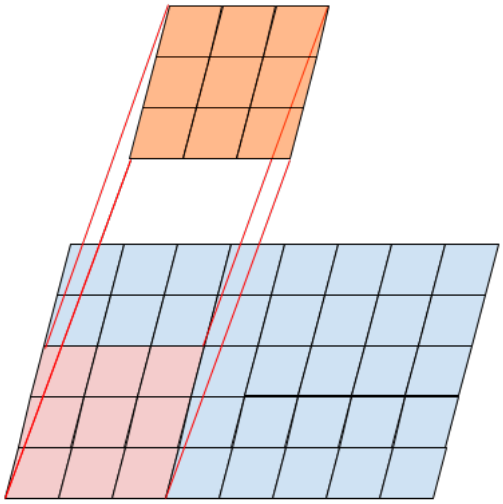


Figure 2.2: An example CNN with a (3x3) kernel part way through a convolution over an image, represented as a larger array.

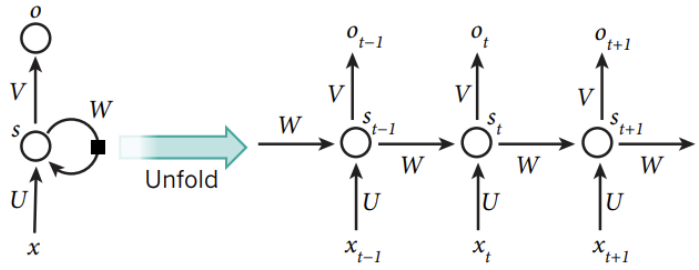


Figure 2.3: A representation a neuron in an RNN turning inputs to outputs, unfolded to show the same neuron at multiple points in time. The state is passed from left to right, and inputs are transformed to outputs moving from bottom to top. Taken from [31].

means the weights are tuned to give high activations for filters that recognise particular features in the input, such as lines, edges, corner, or bright or dark areas.

RNNs

Separately to CNNs, recurrent neural networks, or RNNs, have been developed to deal with sequence data. Feed-forward networks can only take a fixed-length input vector. Since sentences can be variable length, this motivated sequential processing in next word prediction.

RNNs have loops which allow for information to persist over time. This is done by passing state data between each prediction. For each prediction, an RNN takes input x_t and the past state h_{t-1}, and predicts output y_t and the next state h_t.

The internal state passed through each time step is based on a recurrence relation. The parameters in W are the same at each time step, and are learned during training.

$$h_t = f_W(h_{t-1}, x_t)$$
$$h_{t-1} = f_W(h_{t-2}, x_{t-1}) \tag{2.6}$$

The activation function is *tanh* and this is applied to the product of two different matrices, which together with the weights of the prediction matrix, give three matrices to be learned.

The prediction at time t is given by a weight matrix applied to output of an activation function, and that activation function includes both the input at t, which is x_t, and the *previous* hidden state h_{t-1}.

Feed-forward is trained by making a pass forward, then we back-propagate through the network and adjusting the parameters to minimise the loss. For RNNs, you need to work out the losses at each timestep in the sequence, and then finally across all timesteps back to the beginning of the sequence.

These means computing many gradients across time, and this can lead to exploding or vanishing gradients as the effect on *tanh* of increasing smaller or large inputs leads to increasingly large or small gradients. This makes it harder to capture long-term dependencies. An alternative activation function is the Rectified Linear, or ReLU function, which outputs the input directly if it is positive, or zero otherwise. This function does not lead to the same saturating behaviour, and is fast to calculate. This has led to ReLU becoming the 'default recommendation' for modern deep learning [21].

LSTMs

LSTMs [24] were introduced to address the vanishing gradients problem of remembering information from far back in the sequence. The difference between LSTMs and RNNs is that LSTMs have separate information flows for short-term and long-term information. This allows weights to be learned to understand how long-term information affects the output, and a separate set of weights to form the associations related to short-term information.

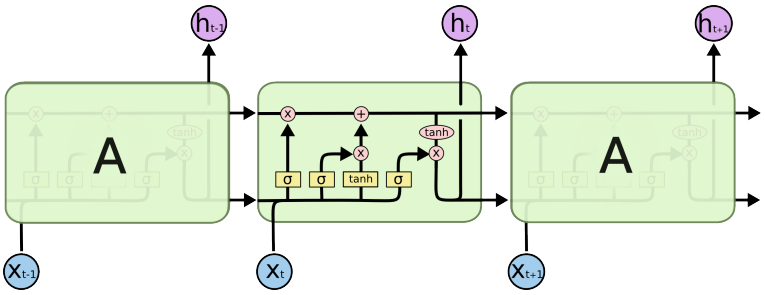


Figure 2.4: A visualisation of how an LSTM node is able to retain long-term information using gates, taken from Olah [39]. Circles represent gates.

LSTMs use gates to control the flow of information, based on an neural net layer, and pointwise multiplication, as shown in the LSTM figure. Within each LSTM node, there are three gates which Olah [39] interprets as being combined in four stages : forget, store, update, and output. I describe the mathematical operations in detail below. In the example below I use h_k to represent the output from the network at time $t = k$, x_t to represent the input at time t, and C_t to represent the context vector at time t, following the notation used by Olah [39] to match the diagram shown.

1. **The forget gate layer (forget)**: Take the current input x_t and previous output h_{t-1}, and learn a function which is input to the gate on the top left. We could interpret this as a function which tells us how much of the previous state we should keep.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.7}$$

2. **Determine information to update the cell state (store)**: A sigmoid layer determines which values we want to update, and the tanh layer creates a vector of candidate values, represented as $\widetilde{C}_t$.

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \widetilde{C}_t &= tanh(W_C \cdot [h_{t-}, x_t] + b_C) \end{aligned} \tag{2.8}$$

3. **Update the cell state (update)**: Use the outputs of the previous two functions, forgetting what we decided to forget, and updating what we decided to update, to update C_t. This uses the leftmost gate and the upper middle gate. Here, $*$ denotes a point-wise multiplication, because the output f_1 is a scalar between 0 and 1.

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{2.9}$$

4. **Generate a prediction (output)**: We now make a prediction using the current state C_t, the previous prediction h_{t-1} and the current input x_t.

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * tanh(C_t) \end{aligned} \tag{2.10}$$

LSTMs are designed to carry information across longer time frames than RNNs. They pass state information across time separately to the prediction, and use gates to determine how much of the state to update, and how much to forget. This means they are able to use both short- and long- term information in their predictions.

Peephole connections

A modification of LSTMs was introduced by Gers and Schmidhuber [19] and was formalised by Graves [23]. Gers and Schmidhuber observed that each gate does not have a direct interaction with the cell state, which meant the relationships learned between each step above was only indirectly linked. They added 'peephole' connections which meant each gate layer also took the previous step as input. A compact representation of the main equations is as follows. As before, $*$ denoting a point-wise multiplication, since the output of f_t is a scalar $[0, 1]$.

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t, C_{t-1}] + b_i) \\ f_t &= \sigma(W_f \cdot [h_{t-1}, x_t, C_{t-1}] + b_f) \\ C_t &= f_t * C_{t-1} + i_t tanh(W_C \cdot [x_t, h_{t-1}] + b_c \\ o_t &= \sigma(W_o \cdot [x_t, h_{t-1}, C_{t-1}] + b_o) \\ h_t &= o_t * tanh(C_t) \end{aligned} \tag{2.11}$$

ConvLSTMs

The ConvLSTM was introduced by Shi et al in 2015 [46] to predict precipitation using radar images, combining ideas from convolutional neural networks and long short-term networks. ConvLSTMs are essentially peephole LSTMs, but which use the convolution operator rather than the Hadamard operator, or point-wise product.

This section uses $\otimes$ to denote the convolution operator. To emphasise the difference between vectors x_t in LSTMs and the images in ConvLSTM, Shi et al [46] use χ_t to represent images as 3D tensors where the last two dimensions are row and columns. Similarly the hidden outputs and states are also now 3D tensors and so are denoted with $\mathcal{H}_t$ and $\mathcal{C}_t$ to distinguish them from their equivalents in LSTM models.

$$\begin{aligned} i_t &= \sigma(W_{xi,hi} \otimes [\mathcal{H}_{t-1}, x_t] + W_{ic} \cdot \mathcal{C}_{t-1} + b_i) \\ f_t &= \sigma(W_{xf,hf} \otimes [\mathcal{H}_{t-1}, x_t] + W_{fc} \cdot \mathcal{C}_{t-1} + b_f) \\ C_t &= f_t * C_{t-1} + i_t tanh(W_C \otimes [x_t, \mathcal{H}_{t-1}] + b_c \\ o_t &= \sigma(W_{xo,ho} \cdot [x_t, \mathcal{H}_{t-1}] + W_{oc} \cdot \mathcal{C}_{t-1}] + b_o) \\ h_t &= o_t \otimes tanh(C_t) \end{aligned} \tag{2.12}$$

By including the convolutional operator as an input, this allows the model to take image data as input, to learn features through updating the weights on the convolutional filters', and then use this in the same way as LSTM models. As shown in the equations above, this means there are more sets of weights to learn, and the different sets of filters should focus on what information is useful from both a long- and short-term perspective.

State-of-the-art research

Since around 2018, deep generative models have been used to learn the data distributions from training samples and use generators to produce novel samples which match the characteristics of the training data. One example is the stochastic adversarial video prediction (SAVP) model [32], which combines a generative adversarial network with a ConvLSTM. Generative adversial networks (GANs) use both a generative adversarial network to learn the distribution of the data and generate samples, and a discriminator model to evaluate their accuracy. Due to time and computational resource constraints, this dissertation will not include deep generative models but refers back to them as a potential future research direction in 6.4.

2.2 Approaches to solar PV nowcasting and related fields

Weather prediction has been an important area of computational research since the 1960s. Numerical weather prediction (NWP) models solve a series of equations describing the physical laws that govern processes in the atmosphere, such as the Navier-Stokes equation. Model resolution has increased from 10-20km cells to around 1km [40], allowing more accurate predictions, but this comes with an increasing computational cost.

Since 2010, there have been significant improvements in machine learning as a result of massively parallel processing, convolutional neural networks, and large benchmark datasets. Deep neural networks have achieved breakthrough results in computer vision tasks [29], leading to increased interest from other domains including weather prediction.

Deep learning models may be simpler to design and faster to run than NWP models, as a result of parallel computations and not having to solve prognostic equations. The computational requirements of NWP models has led to increased interest in deep learning approaches, but deep learning models often come under criticism from the weather and climate research community for not having principled approaches, being hard to interpret, and potentially performing poorly under extreme conditions [45].

In a systematic review of 65 related publications between 2011 and 2020, Martins et al [36] found that most solar PV nowcasting work uses 'classical' approaches based on identifying cloud velocity, using this to forecast the impact of shade above a particular area. However, the authors found that deep learning approaches are becoming more popular over time, with some years seeing more deep learning approaches than classical methods.

Several recent deep learning approaches have yielded results competitive with NWP. In 2015, Shi et al [46] develop a ConvLSTM model, combining of a CNN with an LSTM model, and demonstrate that it outperforms the state-of-the-art NWP optical flow algorithm in predicting precipitation over a 90 minute window in 6 minute increments.

Similarly, Bansal et al [4], who use data from 25 sites in the US over a year-long period. They find that using a combination of CNNs and LSTMs, they can develop point estimates that are highly accurate within 15 minutes compared with ground truth PV readings.

State-of-the-art work in weather forecasting uses techniques from deep generative networks [45]. An unpublished paper by Gong et al [20], uses an advanced approach called Stochastic Adversarial Video Prediction (SAVP), and compare their results with ConvLSTM. SAVP models combine both an adversarial generative approach with ConvLSTMs. The authors find that this approach outperforms ConvLSTM on several metrics. Due to time and resource constraints, this dissertation will not consider deep generative approaches, but refers back to them as potential directions for future research in 6.4.

2.3 Open questions

There is comparatively less literature using deep learning techniques to predict solar PV yield than there is on precipitation or temperature, likely in part due to less accurate and timely PV yield data, possibly due to commercial and regulatory reasons, or the relative recency of the renewable sector [49].

In addition, there is debate over whether the explicit time series modelling in LSTMs and ConvLSTMs provides benefits above CNNs. Theoretically, since LSTMs have the ability to carry information across a sequence, their design suggests that they would perform better than CNN. However, in an influential paper[1] in 2018 Bai et al [2] argue that simpler CNNs can outperform LSTMs. The authors argue that 'the common association between sequence modeling and recurrent networks should be reconsidered, and convolutional networks should be regarded as a natural starting point for sequence modeling tasks'.

To summarise, this leaves several gaps in the literature. How do different methods of solar PV nowcasting compare over timescales longer than 30 minutes? Are recurrent neural networks better than CNNs on sequence modelling tasks, or should this association be reconsidered as Bai et al suggest?

[1] By August 2022, the paper had reached over 2500 citations in four years.

Research focus and hypotheses

This dissertation aims to add to this discussion by explicitly comparing CNNs with LSTMs, and by extending the prediction period to four hours, and by examining the activations within the CNNs to identify which features are used to drive predictions. This led to the first hypothesis. Solar PV forecasting is also a relatively under-researched field, and so the differential effect of including satellite images in addition to past PV values has not been well studied. This led to the second research hypothesis, whether satellite images provided any additional information on top of simple PV inputs.

To summarise, there are two hypotheses which will be examined in this dissertation in relation to solar PV forecasting:

- **Hypothesis 1**: Explicitly modelling the data as a sequence improves performance.
- **Hypothesis 2**: Satellite images contain predictive information.

Chapter 3

Datasets and preprocessing

Chapter summary: This chapter begins by defining the two datasets used and how the measurements are captured (3.1). It explains how the datasets were cropped and aligned, how missing data was dealt with, then explores the data through visualisations (3.2). Finally the chapter summarises the overall pipeline (3.3), and explains how batches are generated for sequence modelling (3.3).

3.1 Data sources

Testing the hypotheses proposed in the previous chapter required finding open access data for both satellite images, and solar PV energy readings, then combining them in order to test how the images could be used to forecast PV readings.

Satellite images

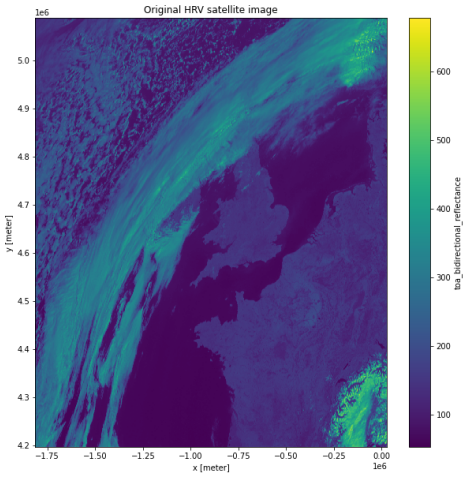


Figure 3.1: A satellite image of the UK, taken from the EUMETSAT dataset. Brighter sections indicate higher reflectivity.

The satellite images are provided by The European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT) from the Meteosat-10 satellite which is in geostationary orbit 36,000km above the equator [13].

The Spinning Enhanced Visible and Infrared Imager (SEVIRI) [14] is the satellite's primary instrument and observes the Earth in 12 spectral channels. The satellite spins, taking an image of the northern third of the Meteosat disc every five minutes. The original EUMETSAT dataset contains data from 2008 to the present day in 12 spectral channels, and for a wide geographical extent.

Open Climate Fix developed the `eumetsat_uk_hrv` dataset [15] from a small subset of the entire SEVIRI dataset: it takes a single channel, the 'high resolution visible' (HRV) channel; it only contains data from January 2020 to November 2021; and it only includes data over the United Kingdom and over North Western Europe.

Even this reduced dataset is fairly large - the `eumetsat_uk_hrv` satellite image dataset is 570.22 GB, with frames taken every five minutes, with 1843 x 891 images. The dataset was reduced to meet resource requirements. First, by taking 128 x 128 crop randomly chosen to focus on Devon, selecting images taken between 05:00 and 20:00 as these are typically daylight hours throughout the year, and as there would be minimal PV energy outside of this window for most of the year.

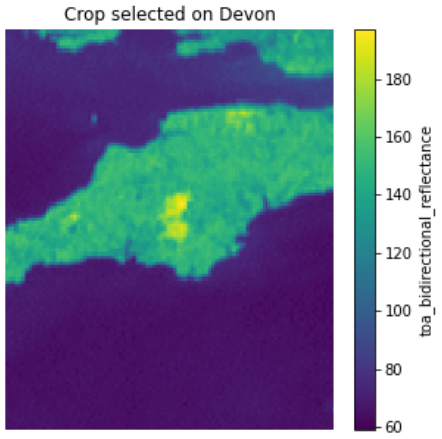


Figure 3.2: A crop of the EUMETSAT dataset described above, focusing on Devon

PV solar generation data

Solar generation data refers to the yield from specific solar panels. Open Climate Fix have a dataset of from 1311 PV systems from 2018-01-01 to 2021-10-27, originally provided by a PV operator. The time series of solar generation is in 5 minutes chunks.

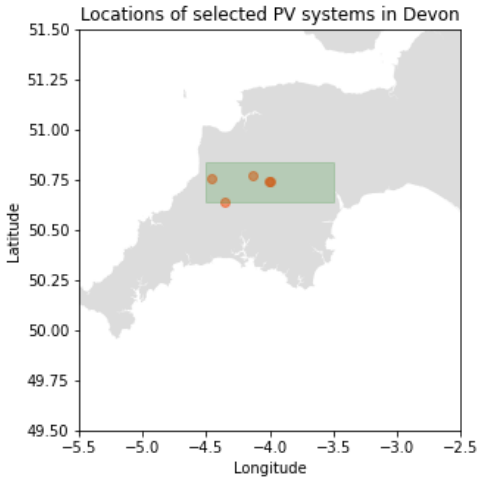


Figure 3.3: The locations of the selected solar photovoltaic stations in Devon, highlighted as orange dots. The green box shows the search area.

3.2 Preprocessing and visualising

Aligning datasets

The previous section described datasets relevant to the task. At this stage, the datasets were not aligned. The PV readings cover most of 2018 to 2021, and the satellite images are taken from 2020 and 2021. The raw datasets may have been missing data, for example from when it is too dark for the satellite to take accurate readings, and there also might be times when the satellite is offline.

To check that the datasets relate to the same area, they are visualised in Figure 3.4. The image on the left shows the locations of the PV stations from which the energy yield readings are drawn. The image on the right shows the satellite crop used.

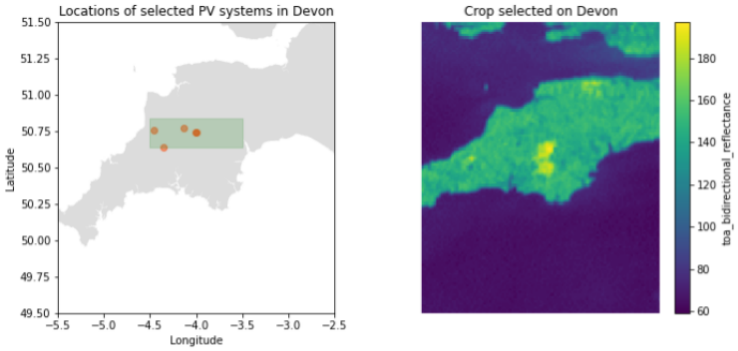


Figure 3.4: Side-by-side comparison of locations of satellite PV stations, and satellite image used in machine learning models.

At this stage in the process, the datasets only contained PV and satellite data readings present in both datasets, from 05:00 to 20:00. However, an inspection of the PV yield, shown on the left

hand of Fig 3.5 of the satellite images revealed that there was still a significant number of images which had a very low PV yield.

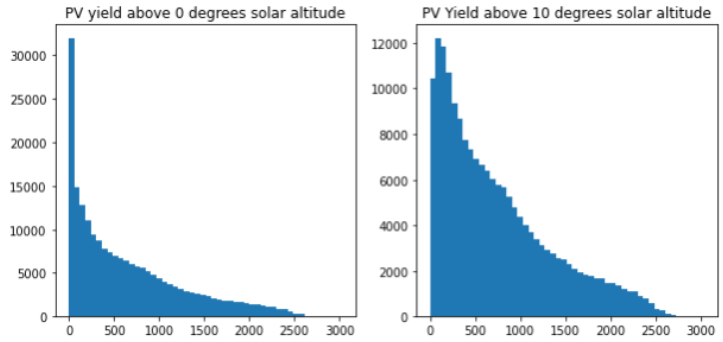


Figure 3.5: Histograms for the unscaled PV yield values for images taken in the spatially and temporally aligned datasets. The image on the left shows yield values for the dataset from 05:00 to 20:00 throughout the year. The image on the right shows the distribution after removing all satellite observations for solar altitudes below 10 degrees.

This required further investigation for two reasons. One reason is that machine learning algorithms tend to work better when data is normally distributed. This dataset had not yet been standardised to [0,1], but even once it was, the sigmoid activation function works best with normally distributed data, which this distribution clearly was not. The second reason was that during the day time, the value of PV yield during the day is probably very roughly normally distributed though with a skew towards lower values, and this extreme imbalance suggested an issue with the data source or preparation pipeline.

One method of exploration was to look at the corresponding satellite images for the timestamps with PV readings of zero. A plotting function shows a grid of the images at different intervals apart. Visualisation makes it easy to quickly identify any discrepancies or missing data. An example of the preprocessed satellite images with a two hours spacing between images is in Fig 3.6.

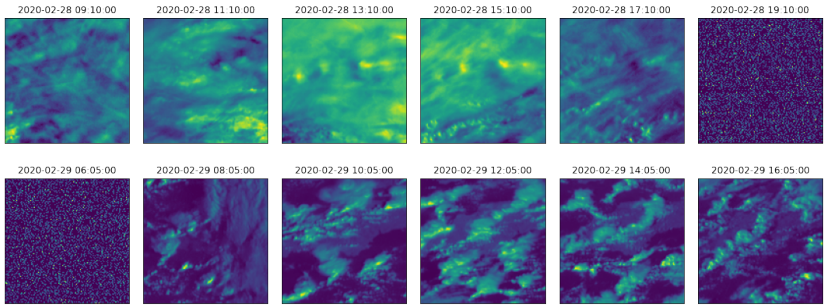


Figure 3.6: Preprocessed images, 2 hours apart

Manual inspection reveals many of the images were also effectively blanks, caused by the satellite trying to record an image while the particular section was dark. In the image above, this is visible on the top right image, taken at 19:10 on the 28th of February, and on the bottom left image, taken at 06:05 on 29th February. These images were unlikely to be informative and could be removed from the dataset.

In order to preserve as much data as possible, the preprocessing pipeline replicated an approach used by Open Climate Fix, and reduced the dataset to only those images taken when the solar altitude was above 10 degrees, using the Python package `suncalc` [5]. The solar altitude is a function of the date and time, and latitude and longitude, using the latitude and longitude to the mean of the co-ordinates of the solar PV stations used sample, giving values 50.73N, -4.19W. Removing those images led to a still skewed but more balanced distribution of mean pixel values, shown in the image on the right of Fig 3.5.

Initial examination suggested this variable might be a useful potential feature. Over short time scales such as one day, solar altitude appears to be highly correlated with solar PV yield, as shown in Fig 3.7.

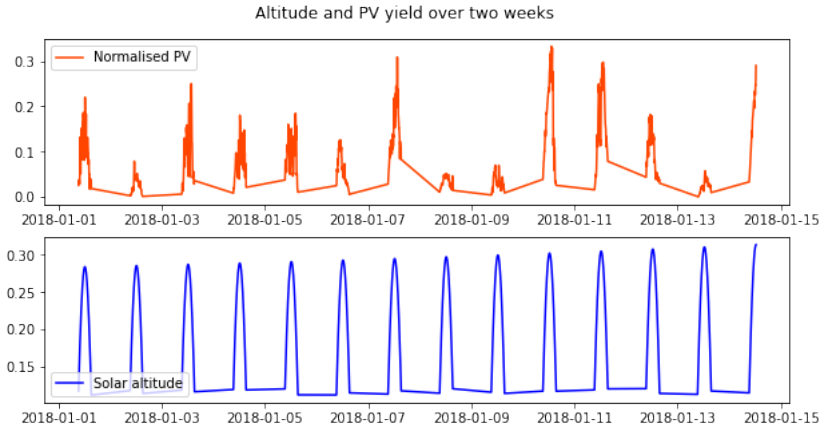


Figure 3.7: Solar altitude and PV yield over two weeks

However, over longer time periods the correlation does not appear to be as strong, as PV yield is likely to be a complex function of solar altitude, and highly dependent on weather conditions such as cloud cover. This is visible in 3.8. For simplicity it was not included in experimentation, though it could be a useful direction for future research.

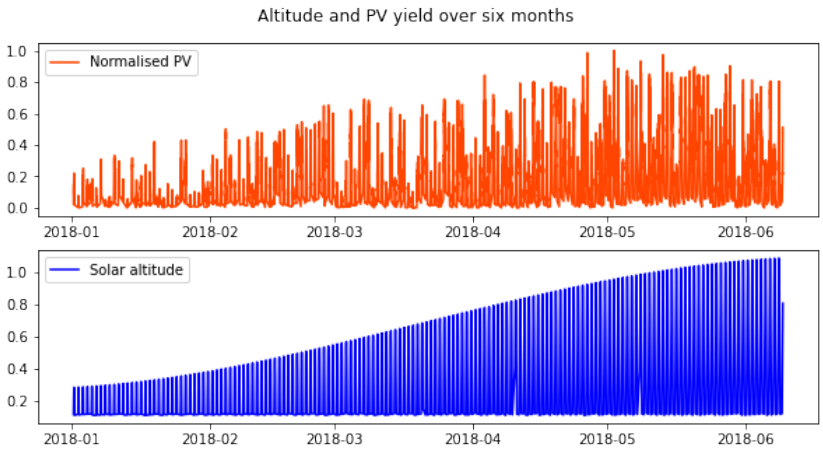


Figure 3.8: Solar altitude and PV yield over six months

The satellite images have values in the range [0,255], and were normalised to the range [0,1] by

dividing by the (1 + the maxmimum value recorded across the dataset). The PV readings were also normalised, this time by by dividing by (1 + the maximum PV reading).

The size of satellite image dataset was slightly reduced speed up training, from (128x128) to (64x64), by selecting a crop that still showed the relevant area, shown in Fig 3.9.

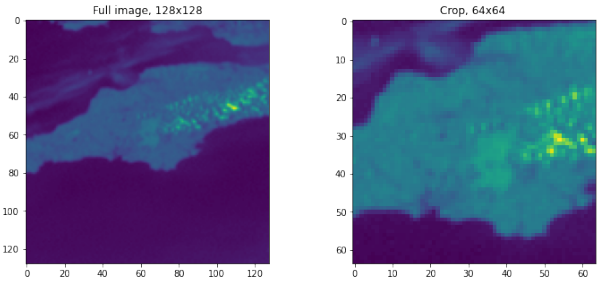


Figure 3.9: Smaller crop of satellite image, used to speed up training

3.3 Preprocessing pipeline

To summarise, the steps for preprocessing are: take data from two sources, crop both the relevant spatial and temporal area, clean the dataset using the solar altitude, create test/train/validation datasets using non-contiguous days from each month, then align the PV and readings and satellite images and save them to Google Drive.

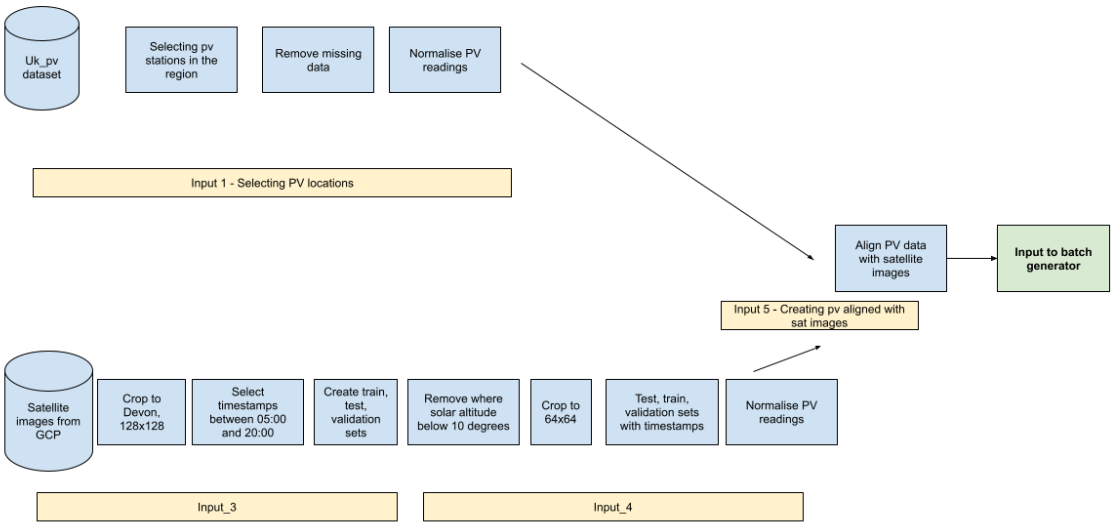


Figure 3.10: Preprocessing pipeline, taking satellite and PV readings, cleaning, aligning them, and inputting them to the batch generator.

Batch generation

Once the data is prepared, batches are prepared for each task. As the task can vary depending on the sequence length, this batching is done within each model file. The task is to take a sequence and to output a prediction sequence, and this required careful preparation of data, as results from one day should not be used to form predictions for the following day.

This is done using a custom dataloader, which takes the following steps:

1. Take the list of timestamps associated with a set of satellite readings, and return a dictionary of the days for each timestamp using `get_list_of_days`.
2. For each day, create a block which is the length of the number of readings for that day, using `get_blocks_of_timestamps_for_one_day`.
3. Concatenate all blocks together, using `get_full_block_list`.
4. Go through the block of timestamps, and make similar blocks to hold the satellite images, pv input reading, solar altitude inputs, and pv outputs, then populate the blocks with readings, using `get_sat_and_pv_blocks`.
5. Do this for each of the train, test, and validation datasets, then shuffle the blocks, expand the dimensions, then pad the input sequences so they match the output sequence length, using `dataloader`.

Chapter 4

Experimental design

Chapter summary: This chapter explains the models used, drawing on the literature review. It begins with a baseline model (4.1), then describes models only taking past PV as input, (4.1), then considers models based on those used by OCF taking both PV and satellite data as input (4.1). Time series forecasting with autocorrelated data requires careful train/test/split generation, described in 4.2. Hyperparameter turning and the loss function are examined in 4.3. Finally, the chapter explains how these experiments can be used to evaluate the hypotheses posed (4.5).

4.1 Selection of models

All models are compared against a baseline model, using a baseline forecast. This section details all the models used.

Baseline models

A simple baseline model used in time series forecasting is the persistence model. This model simply predicts the next future value as equal to the most recent recorded value.

$$\hat{y}_t = y_{t-1} \tag{4.1}$$

For our task, we aim to predict a sequence of values. For example, if we take as input values from 09:00 to 9:55, our task is to predict PV yield from 10:00 to 10:55.

$$\hat{y}_{t=10:00} = \hat{y}_{t=10:05} = ... = \hat{y}_{t=10:55} = y_{t=09:55} \tag{4.2}$$

RNNs and CNNs - using prior PV only

The task is to predict a sequence of values, and the first two models take in PV data only. In this group, two models are considered, the first of which takes input as an LSTM, and the second as a CNN. Both models in this group have multiple possible layers of CNN and LSTMs, followed by a fully connected layer with a minumum of 12 neurons to correspond to the 12 time sequence values predicted.

The first model is a CNN, which uses multiple levels to try to learn abstract features. After each convolutional layer, a MaxPooling layer is used to reduce the dimensionality and learn the essential features. This is followed by two dense layers to resize the output to predict a sequence of values. The CNN treats the sequence as one vector.

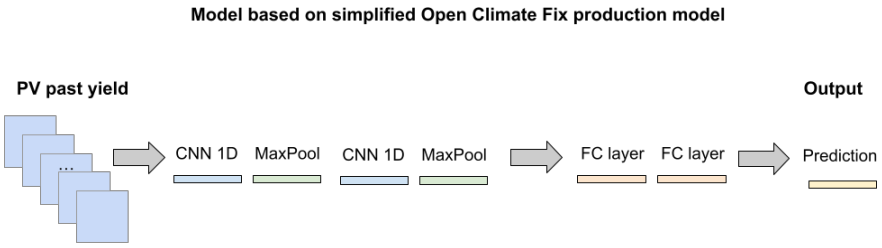


Figure 4.1: CNN model

The second model uses a recurrent neural network, an LSTM, again with multiple levels, to process the input as a sequence. The difference with the LSTM is that it passes each value through the sequence and updates the state as it progresses through the sequence. For the first two layers, it returns the full sequence of outputs, but on the last layer, it returns only a single series of predictions.

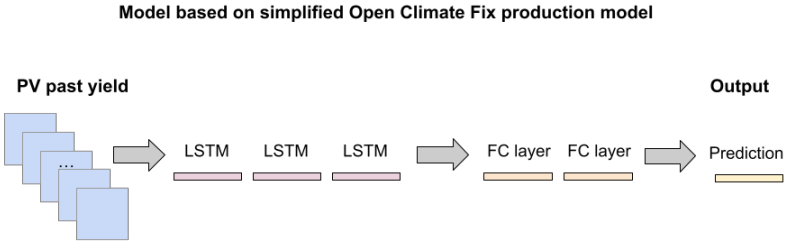


Figure 4.2: CNN model

OCF Conv3D model - sat and past PV inputs

This report now turns to models which include satellite data. For images, our main comparison is between CNN and ConvLSTM, and we are also interested in the comparison showing to what extent providing the prior PV data provides additional information. These models are based on the Conv3D architecture used by Open Climate Fix, comprised of four layers of Conv3D, and then fully connected layers combining the prior PV data. Dropout [17] is used to reduce overfitting, a technique where a set fraction of weights are randomly selected and omitted during training. The dropout rate is set using hyperparameter tuning.

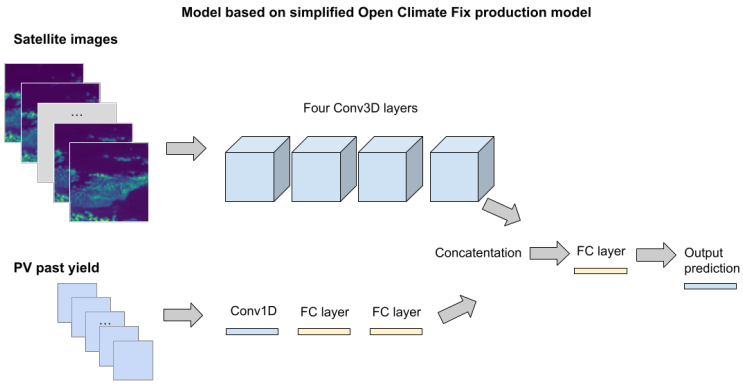


Figure 4.3: Model based on OCF 3D CNN, taking both satellite images and past PV yield, using convolutional components

Custom ConvLSTM model - sat and past PV inputs

The final model is one designed as part of this resarch to be similar to OCF's Conv3D, but to incorporate recurrent components - ConvLSTMs to process the images, and LSTMs to process the PV input sequence. Again dropout is used, and tuned as for the Conv3D.

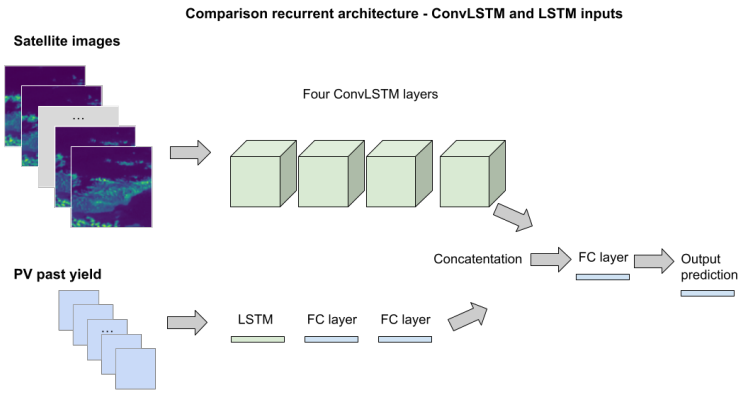


Figure 4.4: Custom ConvLSTM model, taking using both satellite images and past PV yield, with LSTM components

4.2 Model training and dataset split

The model is trained on 22 months of data from 1/1/2020 to 7/11/2021. The data is split into train, test, and validation datasets. This is often done on a split of 80%:10%:10% between the three sets. When data is mutually independent, the subsets are often generated through random sampling.

However, as pointed out in Schultz et al [45], meteorological data constitutes a continuous time series with auto-correlation on different scales. As a result, randomly drawn samples could overlap, and information from the training dataset would leak into the test and validation datasets. This would impair our ability to discern whether the model was able to generalise, making hyperparameter tuning less helpful, and could lead to overly optimistic results as the test set effectively includes information used in training.

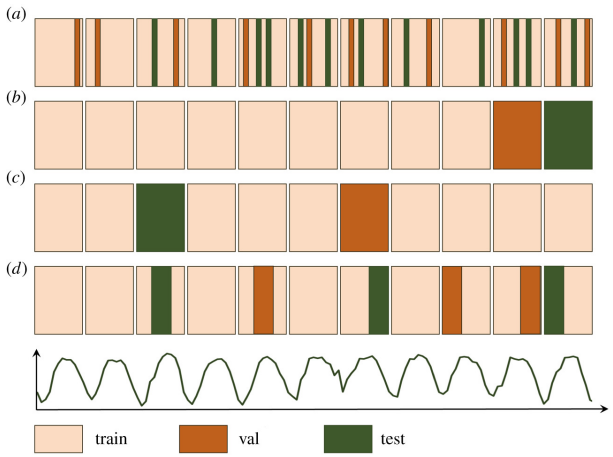


Figure 4.5: Different train-val-test strategies taken from time series data to avoid autocorrelation by avoiding contiguous datasets, from Schultz et al [45]

Figure 4.5 shows different splitting strategies for training, validation, and testing. Case (a) depicts random sampling, violating the independence assumption, and cases (b-d) show variants of random block sampling which avoid spurious correlations.

After speaking to Open Climate Fix, they advised an approach similar to case (b) above, but leaving a gap of a day between each section. The reason is that weather patterns often last for several days, and a contiguous split would in effect lead to data leaking from the test into the training data.

Figure 4.6 shows the split used to generate the datasets in this report. Following the strategies described in Schultz et al [45], and maintaining the conventional 80%:10%:10% split, days 1 to 20 are for training, days 22 to 24 for validation, and days 27 to 29 for testing.

4.3 Hyperparameter training

There are several hyperparameters which can be varied. For all models, the models are trained until convergence on the training data, and the learning rate is varied between [0.01, 0.001, 0.0001, 0.00001], and dropout rate is varied between [0.05, 0.5] in steps of 0.05.

For the CNNs and the ConvLSTMs, the kernel size varies between [3,5,7]. For ConvLSTMs the number of filters varies between [16,32,64]. For the fully connected layers the number of nodes are

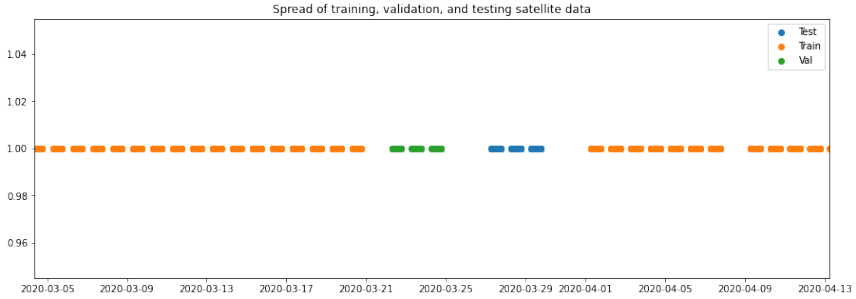


Figure 4.6: Train-val-test approach used. The x axis is the range of timestamps of the dataset, and the dashes show the clusters in which train, test, and validation datasets are sampled. Spacing of at least two days keeps the samples non-contiguous.

varied in the set [12,24,48].

4.4 Optimisation

The models are trained until convergence using mean squared error, defined as follows.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{\mathbf{y}}_i - \mathbf{y}_i)^2 \tag{4.3}$$

The MSE is a good loss function because it is a smooth differentiable function, though it will tend to penalise large errors very highly. For this reason, when it comes to evaluation, the analysis will use two frequently used evaluation metrics in machine learning, Mean Absolute Error (MAE), and Root Mean Square Error (rMSE).

$$rMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{\mathbf{y}}_i - \mathbf{y}_i)^2} \tag{4.4}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|(\hat{\mathbf{y}}_i - \mathbf{y}_i)| \tag{4.5}$$

There is an important difference between RMSE and MAE, making the former more sensitive to outliers. RMSE effectively works with the square root of the mean of the square errors. MAE does not square the errors before summing them. This leads to different results. To give an example, the errors $(1, -1, 20)$ would give an RMSE of $\sqrt{\frac{1}{3}(1^2 + (-1)^2 + 20^2)} \approx 11.6$, but MAE of $\frac{1}{3}(1 + 1 + 20) \approx 7.3$. This will mean that rMSE will be more sensitive to outliers.

The experiments also use the Adam, or Adaptive Movement Estimation optimiser [28] rather than stochastic gradient descent, as the algorithm for backpropagation as this is often used in deep learning to improve performance. The Adam optimiser is an algorithm often used in deep learning and computer vision, as a modified form of gradient descent. The optimiser combines two techniques: gradient descent with momentum, and nonuniform step sizes across different dimensions. Momentum means that the direction of gradient descent has some persistence between updates, to avoid gradient descent skipping over minima. Non-uniform step size allows for the weights to update by different amounts in different dimensions on the same step, but to still be influenced by the gradient of the loss function.

4.5 Tests for proposed hypotheses

The literature review in chapter 2 highlighted the debate over whether CNNs or RNNs are better at time series forecasting, and how deep learning multiple weather predictions incorporate satellite data. The experiments aim to test these two ideas, formulated as the following two hypotheses.

- **Hypothesis 1**: Explicitly modelling the data as a sequence improves performance.
- **Hypothesis 2**: Satellite images contain information which improves prediction accuracy.

The experiments will compare the mean absolute error and root mean square error for each model, over a range of time horizons similar to those used in solar PV nowcasting in practice: 1, 2, 3, and 4-hour time series predictions. The predictions will be for a sequence of values because grid operators are interested in the changing shape of the energy supply, not just a single point value.

Below are the comparisons used in each hypothesis. This will provide two sets of data points for each comparison, and enables an ablation study by comparing the effectiveness with and without satellite imagery.

- **Hypothesis 1 (sequences)**: Compare 2D CNNs and LSTMs, then compare 3D CNN and ConvLSTM
- **Hypothesis 2 (satellite images)**: Compare 2D CNN and 3D CNN, then compare LSTM and ConvLSTMs

Chapter 5

Results and analysis

Chapter summary: This chapter compares the performance of the models across the tasks (5.1), and evaluates the hypotheses in light of this evidence (5.2). Are there systematic errors in the predictions, and does this vary over time? The chapter explores patterns in the prediction errors (5.3). And how might the model derive its predictions? Activations are visualised in a similar convolutional network (5.4). Finally, was it all worth it? A cost/benefit of emissions generated and averted is given (5.5).

5.1 Results

This section presents the results of the experiments. The task was to predict a sequence of values of variable lengths, from 1 hour to 4 hours. The charts below show MSE and RMSE by model over different prediction windows.

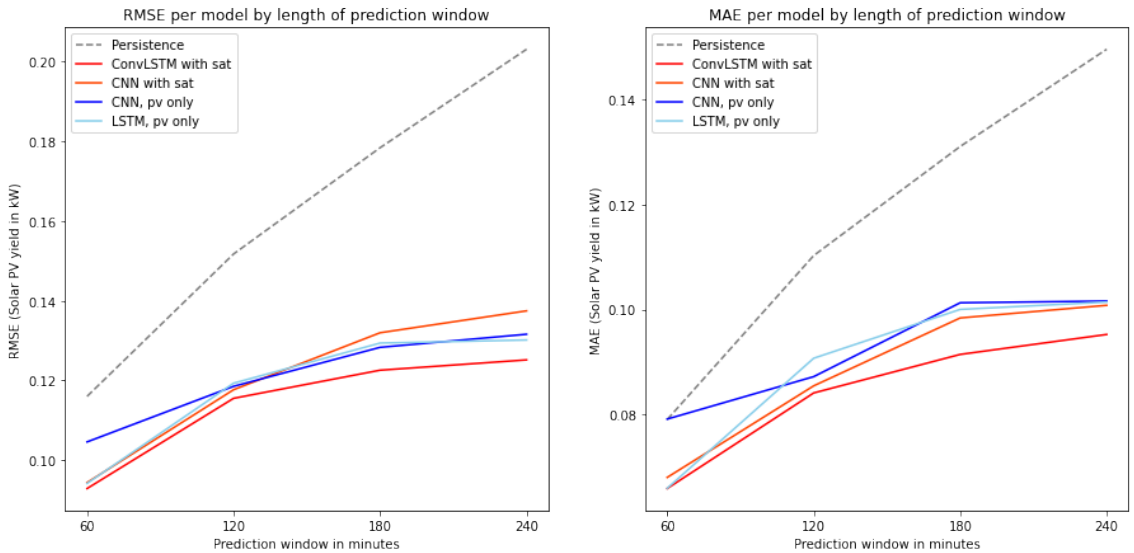


Figure 5.1: Charts comparing the RMSE and MAE for selected models, over a time period between 60 and 240 minutes. ConvLSTMs outperform other models, especially over longer time scales but otherwise the results are fairly mixed.

Key findings

The first finding is that all machine learning models presented outperform the baseline of a persistence model. The performance of this model also worsens over time as solar PV yield tends to vary more over several hours than shorter time periods.

Over time horizons up to 120 minutes, or 24 prediction frames, all models except the PV only CNN (in dark blue) have a similar performance on both metrics. Surprisingly, even the PV only LSTM has performance competitive with the models which also take satellite images as input.

As the time range increases past two hours, the ConvLSTM outperforms the CNN and all other models. On the RMSE measure, the CNN with satellite input actually performs worse than the two models which take satellite images as input.

		Baseline	PV only		Sat and PV	
Time	Metric	Persistence	CNN	LSTM	Conv3D	ConvLSTM
60 mins	RMSE	11.60%	10.46%	9.42%	9.44%	**9.29%**
	MAE	7.91%	7.92%	**6.60%**	6.81%	**6.60%**
120 mins	RMSE	15.18%	11.86%	11.93%	11.77%	**11.55%**
	MAE	11.03%	8.72%	9.07%	8.55%	**8.41%**
180 mins	RMSE	17.84%	12.83%	12.94%	13.20%	**12.26%**
	MAE	13.10%	10.13%	10.00%	9.84%	**9.15%**
240 mins	RMSE	20.32%	13.16%	13.02%	13.75%	**12.52%**
	MAE	14.95%	10.16%	10.14%	10.08%	**9.53%**

Table 5.1: Prediction error over different time horizons, with best results highlighted in bold. ConvLSTM is the best performing model.

5.2 Evaluation of hypotheses

5.2.1 Hypothesis 1 : Time series modelling improves accuracy

The first hypothesis is that modelling the data explicitly as a time series would improve prediction accuracy. The difference between recurrent neural networks and convolutional neural networks is that recurrent networks take both their inputs at the current time step, and one or more state vectors from the previous time step. For a LSTM network, there are two vectors which $\mathbf{h}_t$ and $\mathbf{c}_t$, which can be interpreted as short-term state and long-term state respectively [18].

For the PV only inputs, taking the four time periods and two error metrics, the LSTMs outperform CNNs on five out of eight comparisons. In three of these examples, the difference is less than 0.2%. For the PV and satellite inputs, the ConvLSTM model, which uses a recurrent architecture, outperforms the 3D CNN model on eight out of eight metrics. On this comparison the performance difference increases over time, from 0.2% differences at 60 and 120 minutes to 1% differences at 180 and 240 minutes.

This provides moderate evidence that the recurrent network approach outperforms convolutions, and that the state and memory used in explicit time series modelling helps to improve predictive accuracy. The finding seems to be more true over longer time periods and when the models include satellite images as inputs.

5.2.2 Hypothesis 2 : Satellite images improve predictions

The second hypothesis is that satellite images contain predictive information, so that by including satellite data as an input, we can improve our predictions. The theoretical justification would be that the main uncertainty in short-term PV yield is from cloud movement, and so if we can anticipate their movement using a sequence of satellite images, the model is able to exploit this information to improve predictions.

Comparing CNNs between those with PV only input, and those with PV and satellite input, for the four time horizons and two error metrics, on six out of eight comparisons, the models which take satellite input have a better performance. The differences are variable between 1% and 0.1% with no clear trend.

Comparing the recurrent architectures - the LSTM against the ConvLSTM - on seven out of eight comparisons the satellite image models perform better, plus one comparison where the results are tied. The improvement is fairly stable around 0.5%.

Overall this is also moderate evidence that satellite images improve predictions, but this seems more robustly true when the comparison leverages the recurrent network architecture of the LSTM.

5.3 Analysing the predictions

This section explores the predictions for the best model, the ConvLSTM model, over a one-hour time horizon. It begins by comparing the predictions against true values for three randomly chosen days in the test set.

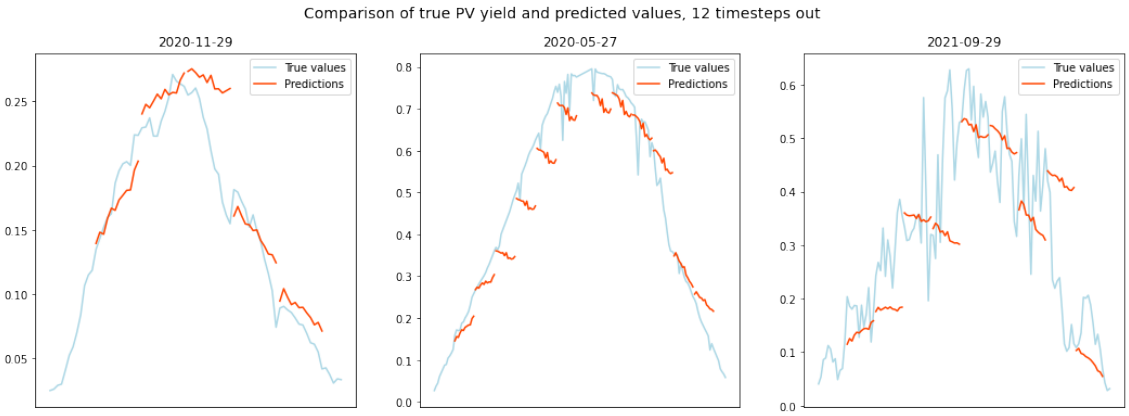


Figure 5.2: Predicted and observed PV yield across three randomly chosen days in the test set. Predictions an hour ahead are in orange and observed values in light blue. Note the variation in y values (solar yield) across days. The model tends to under-predict sunny and cloudy days.

Interestingly, it appears that the ConvLSTM generally predicts the overall gradient of the prediction correctly - if the prediction window is getting brighter or darker. However, in the middle day shown above, the model consistently predicts that the PV yield will decrease. The middle day has PV readings up to 75% of the two-year maximum, and is much brighter than the other two examples which peak at around 30% and 60%. This suggests that the model does not deal well with sunnier days, and might under-predict on those days.

The plot below shows the total relative prediction errors per mean prediction error over the sequence, across a two-year period. Since there are 12 predicted values with a range of [0,1], the largest possible magnitude of error is 12. The models optimise for mean square error, which is not sensitive to the sign of the error. This plot shows the direction of the error.

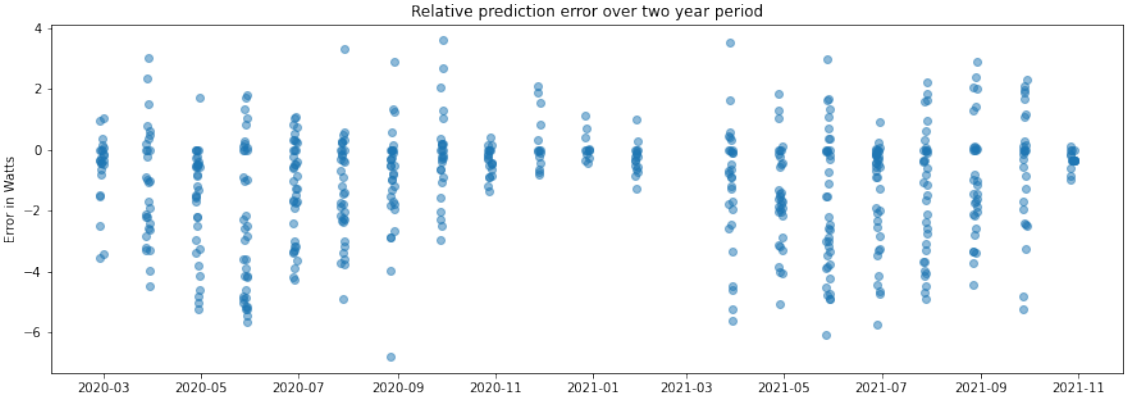


Figure 5.3: Plot of relative prediction error over two years - mean error over sequence. Overall the model tends to have a negative error - unpredicts PV yield. This is more extreme in summer, with smaller errors in winter.

It appears that the model tends to have the errors with the largest magnitude in summer, lower magnitude errors in November, December, and January. Since the test set days are three days chosen from the end of each month, the values appear in lines which span the test set ranges.

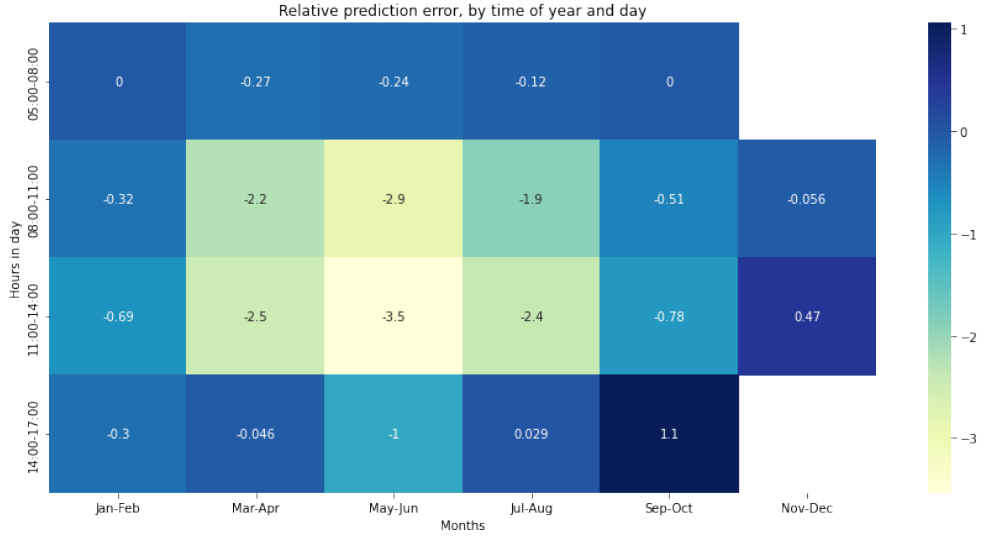


Figure 5.4: Heatmap of relative prediction error - mean error over sequence. This shows that the model tends to underpredict PV yield overall, with the biggest mean errors in the middle of the day in the summer months.

Finally this heatmap represents the same data shown above. We can see that the model tends to under-predict mostly strongly during the early afternoon on summer days, which further shows that the model tends to find high PV yield and bright conditions anomalous, and reverts towards mean PV values.

5.4 CNN Interpretability

A frequent criticism of neural networks is that it is hard to interpret how they determine their results. This section examines the activations in a similar neural network, to understand what drives the results, using an filter visualisation approach inspired by Chollet [7].

This report has investigated the relationship between a sequence of observations and a sequence of pv readings. Interpreting the results would require viewing the activations across the sequence. For simplicity, a simpler neural network was trained to infer a single PV reading from a single image. The model architecture is similar to the CNN, and is comprised of four convolutional layers, but with only 16 filters per convolution, again for simplicity. Experimentation using ReLU revealed that most of the images were dark. In order to make the activations easier to see, Leaky ReLU is used as the activation function, so a wider range of activation values would be visible. Leaky ReLU [34] shows small values for the gradient when the activation is less than zero. This also suggests another direction for further research (see 6.4) using different activation functions.

The model was trained for 10 epochs on a randomly selected sample of 1000 satellite images, and it achieved a test set performance of a 2% mean square error. Three images are shown from the test set, which have different characteristics and will hopefully lead to different activations. These are described below.

- **Case A**: The left hand image is from a morning in summer and shows a clear sky and the land area is in clear relief to the sea, which has a high standardised PV reading of 0.60. The predicted value is 0.59.
- **Case B**:The middle image has comparatively less contrast between land and sea, and is taken from early in the morning in spring, and is likely to be darker overall. Standardised PV reading around 0.09. The predicted value is 0.12
- **Case C**: The right image is taken from a summer morning, at a similar time to Case A but a different year. It has patchy cloud cover across the image, leading to a PV reading of 0.25. The predicted value is 0.32.

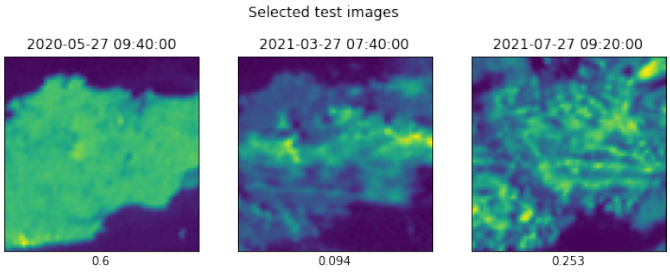
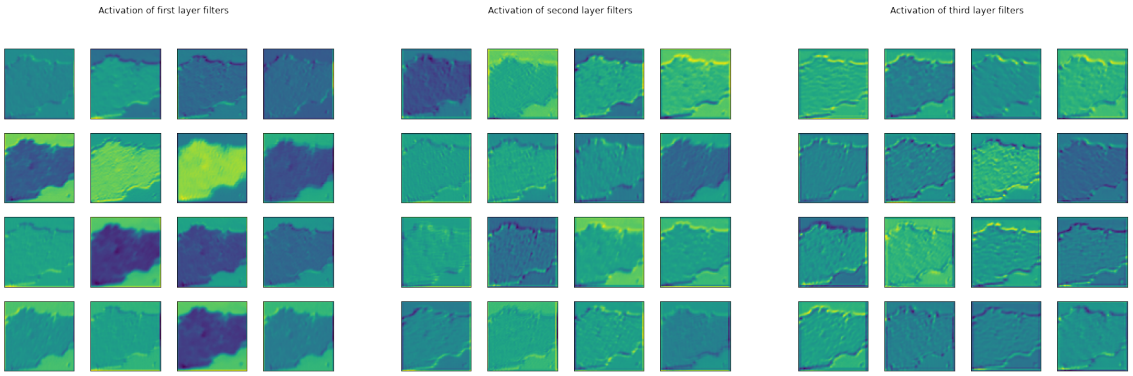


Figure 5.5: Select test images for interpretability, with their time stamps and observed PV yield values

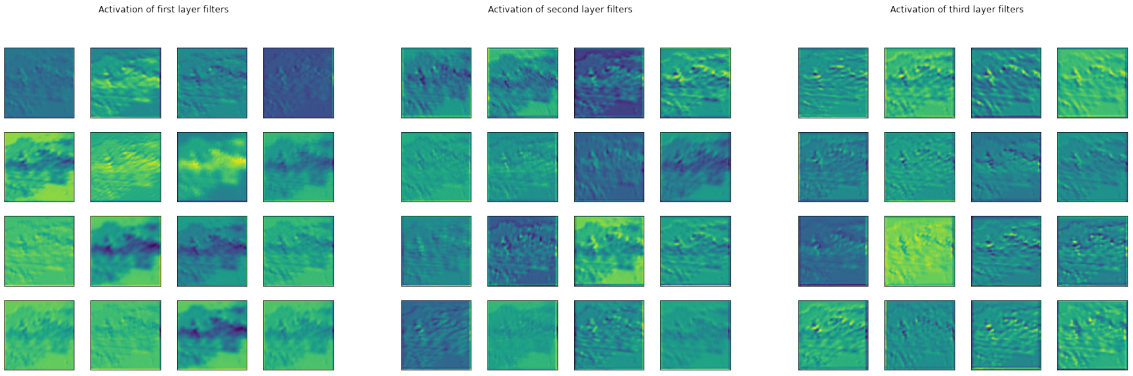
Overview of simplified model filter activations

This section overlays the activations from the CNNs on the three test images chosen earlier. Each row deals with a single image, and each column represents a convolutional layer. In theory, there should be simpler features lower in the model - on the left hand side, and more complex features higher up in the model - on the right.

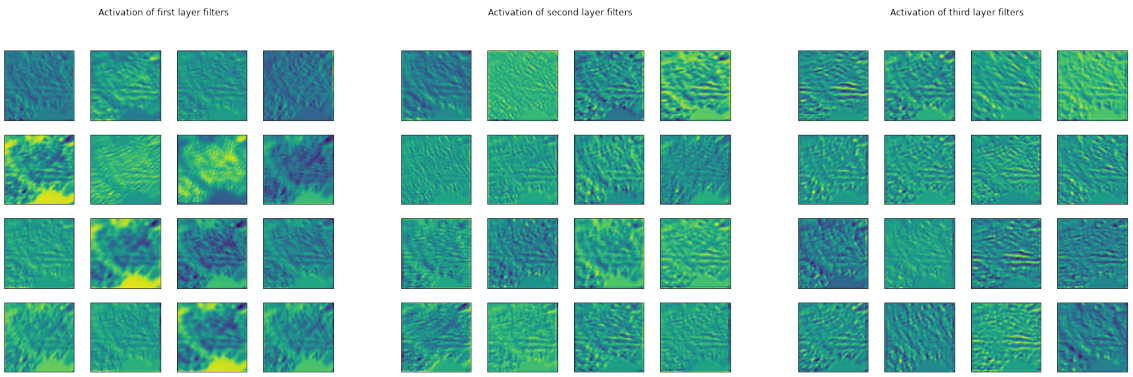
Activations for Case A



Activations for Case B



Activations for Case C



Brighter values indicate higher activations in those regions of the image. From inspecting the activations below, there appear to be a few general features:

- Some first level filters appear to detect sea, land, or edges

- For Case A, the second and third layers appear to build up a clear view of the coastline, which would indicate high relief
- It is harder to discern which, if any, filter are recognising cloud - it could be filter 7 which has fairly high activation across all cases
- For Case C, there does not appear to be any particularly highly-activated filter, which might represent a cloudy image overall

To summarise, it seems that this neural network is focusing on recognising an image with high relief between land and sea, and not separately recognising cloud. This makes sense because times when there is a high contrast between land and sea will be when the land is visible and highly reflective - so when the sun is high in the sky and there are few clouds. This idea is revisited in the final chapter 6.

5.5 Computational and carbon cost

Researchers in machine learning are beginning to explore the energy requirements and carbon intensity of large models. For example Strubell et al [50] find that a large transformer model could result in 284 metric tonnes of carbon emissions, compared to an individual on a 6-hour flight resulting in 0.9 metric tonnes. Does the increased forecast accuracy of superior models improve forecast accuracy enough to justify their energy requirements? This section estimates the energy requirements of the models from this dissertation. Several assumptions are stated and revisited them in the limitations section below.

Each of the models was run on a Tesla T4 GPU available through Google Colab, assumed to be in Northern Europe, the region closest to UCL, which has a carbon efficiency of 0.21 $kgCO_2eq/kWh$. Energy usage simulations were conducted using the Machine Learning Impact calculator presented in [33]. The report uses the computational time required to train and make inferences over a two-hour prediction window and compare the ConvLSTM and 3D CNN. It is assumed that the predictions are generated for each of the 330 UK grid supply points, and are generated every hour, from 06:00 to 20:00, and for half of the year, giving 330 * (20-6) * 365 * 0.5 = 843,150 forecasts per year.

OCF estimates that better PV forecasts would reduce emissions in the UK by up to 100,000 tonnes per year [16]. Based on conversations with domain experts, they estimate the UK spinning gas reserve could be reduced by 40%, or 200 megawatts per day for approximately half of the year. Gas reserve produces approximate 1,000 tonnes of CO_2 per megawatt. Together this offers 200 MW of saving * 1/2 the year * 1,000 tonnes of CO_2, or 100,000 tonnes per year.

It is hard to know exactly how much increased forecast accuracy would lead to reduced gas turbine usage, as this depends on the grid operators' confidence in the models. A range is presented from where the increased accuracy relative to baseline translates into carbon savings either at a ratio of 1:10,000 to 1:10.

	CNN	**ConvLSTM**
Time to train model (s)	916.39	673.11
Time for inference, one forecast (s)	0.16	2.21
$\implies$ Total time for year (training + 843,150 * inference) (hrs)	37	517
Emissions generated (kg CO_2-equiv) from [33]	**0.54**	**7.6**
MAE (watts)	13.75%	12.52%
Improvement over persistence (watts)	6.57%	7.80%
Emissions averted (kg CO_2-equiv)	**657-657,000**	**780-780,000**
Net emissions averted (kg CO_2-equiv)	**656-656,999**	**772-779,992**

Table 5.2: Estimates of emissions generated and reduced

.

It seems that even under pessmistic assumptions, an improvement in the PV forecast could lead to at least a 100 times more emissions averted than generated.

Limitations of computational and carbon cost analysis

The above analysis has many limitations. From the emissions generated side, the energy production from the data centre can come from many different sources. This analysis uses a calculator provided by Lacoste et al [33], which takes available information about data centre energy supply. However, it is not clear when using Google Colab in which region the calculations are performed, and the energy mix will also vary throughout the day and year.

Perhaps there are even more limitations in terms of estimated benefit. Benefit is estimated using approximate calculations from Open Climate Fix. In order to meaningfully reduce carbon emissions, a forecast PV model would not need to have a lower MSE and RMSE - it would need to convince the grid operators at ESO National Grid. Only if they were more confident in this model, and then reduced spinning gas turbines as a result, then can we argue that improved accuracy led to reduced emissions. So the relationship between increased accuracy of a predictive model and emissions averted is not likely to be linear.

In a 2021 paper published in *Nature*, researchers from Deepmind worked with expert meteorologists from the UK Met Office to develop a deep generative model to predict precipitation [41]. They found that the meteorologists were interested in models which led to predictions which looked better and represented possible weather conditions, not simply models with the lowest error measures. Grid operators may similarly be more interested in realistic PV predictions, but not necessarily the most accurate ones, which can then feed into their own decision-making.

Chapter 6

Discussion and conclusions

Chapter summary: This chapter discusses the meaning and implications of the results. It considers CNNs and ConvLSTMs in relation to the task of solar PV nowcasting using satellite imagery (6.1), and explores the results, including from analysing prediction errors and the convolutional activations (6.2). The following section (6.3) looks at implications for machine learning and solar PV nowcasting. Finally, the chapter consider directions for future research (6.4) and reflects on the project's achievements (6.5).

6.1 Discussion of main results

The main finding of this research is that there is moderate evidence that time series modelling in recurrent networks has higher predictive accuracy, especially over longer time periods and when taking satellite images as input. Satellite images also improve predictions, especially when using the recurrent ConvLSTM architecture.

Why might LSTMs outperform CNNs on this specific task?

The difference in RNNs is that they pass the information back in to update their predictions, includes a temporal aspect absent from CNNs. This is sometimes described as the 'memory', and is based on a biological model of information processing, where the value in a time series is processed at each moment, but in combination with memories and context.

CNNs were developed for machine vision, and assume translation invariance. This means CNNs assume that it does not matter where in the data object a particular feature appears. This makes sense in vision, because the range of perspectives and distances between an object and the perceiver means we should still recognise a particular feature regardless of where it appears. When CNNs are applied to a block of images, the model effectively flatten the entire network into one high-dimensional vector. This means the time dimension is treated in the same ways as any other dimension.

But in RNNs, the sequence and ordering does matter. But in weather prediction, the time dimension will matter more because later images are likely to have more predictive power. LSTMs are a step better than RNNs because they carry forward a richer thread of information than a single state vector. LSTMs can remember some information, but forget others. As discussed in the literature review in Chapter 2, there is debate over whether LSTMs outperform CNNs, such as in Bai et al [2].

The experiments in this project found that the LSTMs had a more significant improvement over CNNs for longer time horizons. A potential explanation is that the LSTM is able to learn context vectors which say whether the day is getting brighter or darker, and these vectors are able to be remembered better and take more predictive importance.

Why might ConvLSTMs outperform 3D CNNs on this task?

This research generally found that satellite imagery improved predictive accuracy, but the biggest differences were over longer time scales. Why might the ConvLSTMs outperform CNNs over longer time scales?

One reason could be that weather predictions images include both constant features and temporary aspects. For example, an important and constant feature might be a darker image overall, such as in the evening; and temporary aspect could be transient cloud pattern which might might build to provide cover, or might dissolve, leaving a clearer sky. The combination of states and updating in LSTMs and ConvLSTMs might provide further capacity to learn which features are constant and which ones are transient.

Also, A 3D convolutional neural network will start by treating all dimensions in the three-dimensional block as equally important, and will need to use some of its neurons to learn the time dependency. By including time explicitly as a dimension in time series, additional information is provided, so the neural network does not need use up its training data learing the importance of time, and instead can focus on other features important for predicting PV yield.

Patterns in prediction errors - a gloomy model

From analysing prediction error, it is apparent that the model consistently under-predicts PV yield, especially so in the early afternoon in summer. This could be caused by the model having insufficient training data for sunny days, and under uncertainty to revert towards the mean. This could be further explored by training different models on summer and winter, and seeing if the same effect persists. MSE equally penalises over- and under- predictions, and since the early morning and evening tend to have low or near-zero PV yield, there are many cases where predicting a low value is a safe bet. This could also be explored by altering the loss function to more heavily penalise under-predictions.

6.2 Interpretability

What is the model learning?

In terms of interpretability, this report briefly examined a simpler model using only a CNN structure. The lack of activation for cloud patterns, and instead higher regions of activations for land, sea, and edges suggests that this model is learning to recognise a high-relief image of the land. This makes sense because a high constrast image of the land against the sea would imply a bright sun and minimal cloud cover, because the land is much more reflective than the sea. Perhaps the model is not learning to recognise cloud dynamics, but instead simply recognising when the land and the sea have strongly contrasting values. As discussed earlier, future research could explore the activations in the time series models. Since the ConvLSTM model outperforms an LSTM model, this suggests there is some additional information learned through the satellite images.

6.3 Implications

Implications for machine learning

The main finding in this dissertation was that explicitly modelling data as a time series improved predictive accuracy when there was relevant long-term information. Rich Sutton [51] argued that the 'bitter lesson' of AI research is that general methods that leverage computation are ultimately the most effective and accurate. Sutton claims that general approaches are better because of the 'arbitrary, intrinsically-complex, outside world' but he does not rule out the use of time as a dimension. In weather forecasting, time seems to be a particularly important dimension - with a higher degree of predictive importance in solar PV nowcasting than finding patterns in other areas.

Since 2017, the transformer architecture [52] has become a dominant approach in computer vision. The transformer architecture is general and defines a broad hypothesis space in terms of time-dependency. We found that explicit recurrent representation of time series improved predictions. This suggests it may be possible to improve on general architectures by leveraging recurrent relations.

From examining the convolutional layer activations for a simpler model, model appeared to be recognising the contrast between the land and the sea, and not necessarily learning about cloud dynamics. This is a case of 'getting what you ask for, not what you want', a perennial issue in machine learning. Only by examining the activations was this apparent. If a model like this was deployed on a different geographical region with no coast line, its predictive accuracy would likely be far lower. This highlights the importance of interpreting how machine learning models derive their predictions, and ensuring that the datasets used in training are representative of real data in deployment.

Implications for solar PV nowcasting

This dissertation focused on predicting solar PV in the UK, and found that several relatively simple model was able to outperform a persistence baseline. In fact models which only took past PV readings as input had competitive performance up to two hours. This suggests that relatively computationally cheap models, which only take in past PV data, could be useful in anticipating solar PV dynamics. It also highlights the importance of having timely PV yield data made easily available.

This research also found that even with relatively pessimistic assumptions, the benefits of reduced carbon emissions from more accurate solar PV readings outweight the associated carbon emissions with running the models, on a scale of at least 1:100. However, as models become larger and improvements in accuracy become more marginal, there may be false economies. In order to reduce environmental impact, larger models should try to run as efficiently as possible, and investigate the energy sources behind their data centres.

Inequalities in access to data

One limitation is that solar PV nowcasting research disproportionately focuses on a small number of wealthier countries. In the meta-analysis of papers on solar PV nowcasting, Martins et al [36] found that studies using data from the US outnumber any other country, and only 3 of 45 papers are identified as using datasets from non-OECD countries.

Energy inequalities

Moreover, the UK has relatively low potential for energy solar PV than other countries. The chart below, based on data from the World Bank [3] shows the potential energy achievable through solar PV against the proportion of per-capita energy demand met through solar energy. The UK is highlighted in green, as one of the countries with the lowest potential for solar energy due to its latitude and geography (on the y axis), but among the highest proportion of solar PV as a proportion of energy demand (x axis).

Figure 6.1: A graph showing solar capacity against solar potential for individual countries - the UK, highlighted with a large green circle towards the bottom, has among the worst potential for PV but one of the highest portions of its energy provided by solar PV. The graph was made for this report using the World Bank dataset [3].

In the influential *Sustainable Energy* textbook [35], the late David MacKay argued that energy from solar PV could be a way to reduce your personal average electricity consumption, but that in the UK the combination of high population density and high latitude would make it hard to deploy enough solar to put a get enough energy to put a significant dent in total energy consumption for the UK.

"*It would quite probably be better to put the panels in a two-fold sunnier country and send some of the energy home by power lines* ([35], p. 41)"

In the 2009 book, MacKay mentions the DESERTEC plan [8] which uses larger concentrating solar power, rather than solar panels, in sunnier Mediterranean countries, and then delivers the energy using long-distance high-voltage direct-current transmission lines. This project appears to have stalled in the subsequent fourteen years, but the point remains that the UK might not be the best country to deploy large amounts of solar energy.

If further research aims to reduce carbon emissions, it might be fruitful to focus on relatively under-studied countries with a higher potential for PV, where the scope for integrating more energy from solar PV into the electrical grid is greater.

6.4 Future work

There are multiple possible directions in which these results could be improved. The most straightforward would be to repeat the experiment with larger and different datasets. This dissertation focused on a subset of the UK, but a model could be trained on many randomly selected points. Open Climate Fix divide the UK into approximately 330 sections, corresponding to the grid supply points (GSPs) used by National Grid. In addition, this model has only looked at the UK but it would be useful to perform experiments on other countries with lots of installed solar PV capacity, and with a greater potential for PV, such as Spain, Morocco, and Egypt. Even the data within the UK could be significantly improved, as this model used 1300 solar PV stations, when there are at least 260,000 solar PV installations where the best available datasets estimate 86% coverage [49].

The analysis of prediction errors in 5.3 showed that the model tended to demonstrated worse performance on sunnier days. This could be because the dataset has more cloudy and dark days than sunny days. The dataset could be augmented to increase the number of sunny days. More generally, the model tended to under-predict PV yield, so future work could include a loss function to more heavily penalise under-predictions.

Visualisation of the activations in 5.4 of a similar CNN showed that many of the activations were dark using the ReLU activation function. This was only discovered at a late stage in the report. Leaky ReLU was used as an alternative activation function, and showed more activations in a simpler model. Future research could explore whether the use of the Leaky ReLU activation could improve predictive accuracy on the main models.

In terms of more advanced models, Xu et al [53] develop a combination of a GAN and an LSTM for prediction of cloud images. Since around 2017, attention-based models have been used for sequence prediction tasks [1] and would provide another way to examine dependencies in time series data, though as mentioned earlier, it might be helpful to include explicit representations of time series information to reduce the search space. As mentioned earlier, another approach could be using generative models. An unpublished paper by Gong et al [20] uses Stochastic Adversarial Video Prediction to predict temperature, and finds that they outperform ConvLSTMs. Further research could try using a similar approach to predict PV yield.

A different direction could be develop a model to predict a sequence of frames, and then use a second model to infer PV yield from these images. Jack Kelly from Open Climate Fix mentioned this as an area of potential future work, with the reason that there is larger dataset for the first task - all satellite images - and that the main source of uncertainties in forecasts is cloud dynamics. Therefore segmenting the task into separately predicting cloud dynamics from PV yield could be a promising area for further research.

6.5 Achievements

This section assesses the project goals set out at the beginning of the project, in section 1.3.

6.5.1 Project goals

Goal	Status	Progress
Specify a task	✓	Identified solar PV forecasting as a computational task building on concepts covered in my course at UCL
Data collection	✓	Identified two real-world datasets used to train models used in industry
Preprocessing	✓	Examined and selected subsets of the data, and split into valid test/train/validation sets, then aligned, normalised and cleaned data
Model selection	✓	Learned how ANNs, CNNs, RNNs, and ConvLSTMs are used, and what approaches are used by OCF, then implemented my own versions of these models
Results evaluation	✓	Trained models on datasets, tuned hyperparameters, and critically assessed results in relation to findings in the literature and at OCF

Table 6.1: Progress against project goals set in section 1.3

.

6.5.2 Personal goals and reflection

In this project I sought to work with on a real world machine learning problem, learn about contemporary approaches in computer vision, and critically appraise the results. Machine learning for weather forecasting is a large and rapidly developing research field, and I have tried to cover as many of the approaches described in the review article by Schultz et al [45]. I have learned about and implemented approaches in computer vision up to 2017, and researched state-of-the-art approaches such as using GANs [1]. I have enjoyed this project and am proud of my achievements on the MSc Computer Science course at UCL and in this dissertation.

[1] I am also going to the computing centre in Julich, Germany from 27-30 September 2022 to attend the MAELSTROM Boot Camp 2022: Machine Learning for Weather and Climate, organised by Dr Bing Gong, the lead author of the Gong et al preprint

Appendix A

Code and video presentation

A.1 Code

All of the code from this dissertation was written in Python, and is available at `https://drive.google.com/drive/folders/1IgZpWxcXFueEvTHJptnb-CgdJP6xr_ld`. The contents of the folder are as follows:

- `data_pipeline`: pipeline and preprocessing of satellite and PV data
- `model_evaluation`: comparison of performance across models and time periods
- `model_files`: saved model files after training
- `unused_mode_notebooks`: as described in Chapter 1, I followed an Agile approach and experimented - this folder contains each increment of the models as my understanding developed over the project
- `Models 20, 40, 41, 42, 43`: models as described in experimental design
- Multiple `Report` files, used to generate predictions, calculations, and figures shown in this report

A.2 Project presentation video

The project presentation is available at `https://www.youtube.com/watch?v=r7wUlhLoIRs`.

Bibliography

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *3rd International Conference on Learning Representations, ICLR 2015,* 2015. URL: http://arxiv.org/abs/1409.0473.

[2] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.* URL: https://arxiv.org/abs/1803.01271.

[3] World Bank. *Solar Photovoltaic Power Potential by Country.* 2020. URL: https://www.worldbank.org/en/topic/energy/publication/solar-photovoltaic-power-potential-by-country.

[4] Akansha Singh Bansal, Trapit Bansal, and David Irwin. "A moment in the sun: solar nowcasting from multispectral satellite data using self-supervised learning". In: *e-Energy '22: Proceedings of the Thirteenth ACM International Conference on Future Energy Systems* (2022), pp. 251–262. DOI: https://doi.org/10.1145/3538637.3538854.

[5] Kyle Barron. *suncalc.* URL: https://pypi.org/project/suncalc/.

[6] Christoper Bishop. *Pattern Recognition and Machine Learning.* Springer: New York, NY, 2006.

[7] Francois Chollet. *Deep Learning with Python (Second Edition).* Manning, 2021.

[8] *DESERTEC Foundation.* 2022. URL: https://www.desertec.org/.

[9] Natioal Grid ESO. *Future Energy Scenarios: July 2022.* 2022. URL: https://www.nationalgrideso.com/future-energy/future-energy-scenarios.

[10] National Grid ESO. *Future of GB's Electricity National Control Centre.* 2019. URL: https://www.nationalgrideso.com/news/future-gbs-electricity-national-control-centre.

[11] National Grid ESO. *The Road to Zero Carbon.* 2022. URL: https://www.nationalgrideso.com/future-energy/our-progress/road-zero-carbon/report.

[12] National Grid ESO. *What is Frequency?* 2022. URL: https://www.nationalgrideso.com/electricity-explained/how-do-we-balance-grid/what-frequency.

[13] EUMETSAT. *Meteosat Second Generation.* 2022. URL: https://www.eumetsat.int/meteosat-second-generation.

[14] EUMETSAT. *SEVIRI.* 2022. URL: https://www.eumetsat.int/seviri.

[15] Open Climate Fix. *EUMETSAT Dataset.* 2022. URL: https://huggingface.co/datasets/openclimatefix.

[16] Open Climate Fix. *Forecasting*. 2022. URL: https://www.openclimatefix.org/projects/forecasting/.

[17] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, 2016, pp. 1050–1059.

[18] Aurelien Geron. *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow*. O'Reilly (Second Edition), 2019.

[19] F.A. Gers and J. Schmidhuber. "Recurrent nets that time and count". In: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*. Vol. 3. 2000, 189–194 vol.3. DOI: https://doi.org/10.1109/IJCNN.2000.861302.

[20] B. Gong et al. "Temperature forecasting by deep learning methods". In: *Geoscientific Model Development Discussions* 2022 (2022), pp. 1–35. DOI: https://doi.org/10.5194/gmd-2021-430. URL: https://gmd.copernicus.org/preprints/gmd-2021-430/.

[21] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. Cambridge, MA, USA: MIT Press, 2016.

[22] Google. *Google Colaboratory*. 2022. URL: https://colab.research.google.com/.

[23] Alex Graves. *Generating Sequences With Recurrent Neural Networks*. Aug. 2013. URL: https://arxiv.org/abs/1308.0850.

[24] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: https://doi.org/10.1162/neco.1997.9.8.1735.

[25] IPCC. *AR6 Working Group II: Impacts, Adaption, and Vulnerability - Summary for Policymakers (SPM)*. 2022. URL: https://www.ipcc.ch/report/ar6/wg2/.

[26] IPCC. *AR6 Working Group III Report: Mitigation of Climate Change - Summary for Policymakers (SPM)*. 2022. URL: https://www.ipcc.ch/report/sixth-assessment-report-working-group-3/.

[27] Andrew Jaegle et al. "Perceiver IO: A General Architecture for Structured Inputs and Outputs". In: (2021). DOI: https://arxiv.org/abs/2107.14795.

[28] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *ICLR (Poster)*. 2015. URL: http://arxiv.org/abs/1412.6980.

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[30] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: https://doi.org/10.1109/5.726791.

[31] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444. ISSN: 0028-0836.

[32] Alex X. Lee et al. "Stochastic Adversarial Video Prediction". In: (2018). URL: https://arxiv.org/abs/1804.01523.

[33] Sasha Luccioni et al. “Quantifying the Carbon Emissions of Machine Learning”. In: (2019). URL: https://www.climatechange.ai/papers/neurips2019/22.

[34] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. “Rectifier nonlinearities improve neural network acoustic models”. In: *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. 2013.

[35] David MacKay. *Sustainable Energy - without the hot air*. Green Books, 2009. URL: https://www.withouthotair.com/.

[36] Bruno Juncklaus Martins et al. “Systematic review of nowcasting approaches for solar energy production based upon ground-based cloud imaging”. In: *Solar Energy Advances* (2022). DOI: https://doi.org/10.1016/j.seja.2022.100019.

[37] Warren S. McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *Bulletin of Mathematical Biophysics* (1943), pp. 115–133. DOI: https://doi.org/10.1007/BF02478259.

[38] Michael A Nielsen. *Neural Networks and Deep Learning*. Determination Press. URL: http://neuralnetworksanddeeplearning.com/.

[39] Christopher Olah. *Understanding LSTM Networks*. 2015. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[40] Rachel Prudden et al. *A review of radar-based nowcasting of precipitation and applicable machine learning techniques*. 2020. URL: https://arxiv.org/abs/2005.04988.

[41] Suman Ravuri et al. “Skilful precipitation nowcasting using deep generative models of radar”. In: *Nature* 597 (2021).

[42] Matt Reynolds. *Bad weather forecasts are a climate crisis disaster*. 2021. URL: https://www.wired.co.uk/article/solar-weather-forecasting.

[43] Frank Rosenblatt. “The perceptron: A probabilistic model for information storage and organization in the brain”. In: *Psychological Review* 65 (1958), pp. 386–408. DOI: https://doi.org/10.1037/h0042519.

[44] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323 (1986), pp. 533–536.

[45] Martin Schultz et al. “Can deep learning beat numerical weather prediction?” In: *Philosophical Transactions of The Royal Society A* 379 (Feb. 2021). DOI: https://doi.org/10.1098/rsta.2020.0097.

[46] Xingjian Shi et al. “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf.

[47] Ian Sommerville. *Software Engineering*. 9th ed. 2010.

[48] Liam Stoker. *‘Unprecedented’ events send UK power market into negative pricing for six hours straight*. 2019. URL: https://www.current-news.co.uk/news/unprecedented-events-send-uk-power-market-to-negative-pricing-for-six-hours-straight.

[49] Dan Stowell et al. “A harmonised, high-coverage, open dataset of solar photovoltaic installations in the UK”. In: *Scientific Data* 7 (2020). DOI: https://doi.org/10.1038/s41597-020-00739-0.

[50] Emma Strubell, Ananya Ganesh, and Andrew McCallum. "Energy and Policy Considerations for Deep Learning in NLP". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019). DOI: 10.18653/v1/P19-1355.

[51] Rich Sutton. *The Bitter Lesson*. 2019. URL: http://www.incompleteideas.net/IncIdeas/BitterLesson.html.

[52] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[53] Zhan Xu et al. "Satellite Image Prediction Relying on GAN and LSTM Neural Networks". In: *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. 2019, pp. 1–6. DOI: https://doi.org/10.1109/ICC.2019.8761462.