University of Nottingham
UK | CHINA | MALAYSIA

G53IDS
Predicting Solar Irradiance

Submitted 14th December 2018, in partial fulfillment of
the conditions for the award of the degree **BSc Hons Computer Science and Artificial Intelligence with Year in Industry.**

Timothy R. Cargan

School of Computer Science

University of Nottingham

I hereby declare that this dissertation is all my own work, except as indicated in the text:

Signature _____

Date _____ / _____ / _____

I hereby declare that I have all necessary rights and consents to publicly distribute this dissertation via the University of Nottingham's e-dissertation archive.

Abstract

Adoption of renewable energy sources, is a key step in curbing the effects climate change. Moving away from traditional energy generation processes presents many challenges. Unlike coal and natural gas many renewable energy sources and specifically solar operate intermittently and have abrupt changes in output. This variability presents a problem both for the generators, trying to meet contractual obligations, but also the grid operators endeavouring to ensure a consistent load. Having a system capable of producing accurate predictions of solar irradiance would be of use to both parties. This project aims to build such a system. Leveraging advancements in both the domains of machine learning and big data processing.

Contents

Introduction

The Intergovernmental Panel on Climate Change (IPPC), an international body studying the effects of climate change, recently released a report outlining what is needed to be done to keep to global climate change below 1.5C. Whilst they say the goal is not yet out of reach, in order to achieve it "rapid, far-reaching and unprecedented changes in all aspects of society" are needed [1]. One of the point's raised by the IPCC is that "renewables are estimated to provide up to 85% of global electricity by 2050" [1]. One issue holding back the adoption of renewables, as an energy source is cost. Energy producers do not want to invest in something that will not make them money [2].

In order to increase investment in renewables, these ventures must be as profitable as possible. One way to improve profitability is by reducing operational costs for generators. Unlike traditional generators, solar generation is variable and peaks during the middle of the day, a time of relatively low demand for energy. The weather can also change rapidly causing supply to cut off abruptly [3]. Inaccurately forecasting output can cause the generator to over or under deliver on their commitment to the grid, unbalancing the amount of energy on the grid. In order to maintain stability, the grid operator is forced to intervene, finding alternative energy sources to make up the shortfall or requiring other producers to reduce their output. The cost of this action, the balancing cost, is passed onto the offending producer [4]. Having an accurate prediction of power generation, granular enough to be of use in both the long and short term, enables more effective balancing of the grid and reduces costs to suppliers. Sine the grid, in the short run, usually trades in 30 minute blocks [4] having a prediction only a few hours into the future can be enough to make an impact. Reliably announcing their future output producers can avoid incurring production penalties and the grid manager can more easily balance the grid [4].

Specifically, this project will focus on building predictive models for use within the solar industry. At their core, all solar base power generation technologies convert solar radiation into power. Given perfect generation circumstances power output, for a photovoltaic (PV) panel, can simply be thought of as a function of solar irradiance and system efficiency. The main source variability in power output is the amount of solar irradiance that falls on the generator [5]. One of the biggest factors that effects irradiance (excluding constants such as time of day, day of year and latitude) is cloud cover. Having an accurate picture of the clouds location, density, direction and speed of movement would help facilitate the creation of a predictive solar irradiance model.

Today solar farms are continuously producing lots of data. Measures of solar irradiance, power output, detailed telemetry for each component. This data is usually high resolutions, often sub minute and can be streamed. The volume and speed of data can necessitate the use of "big data" paradigms such as Apache Spark [6]. It is also fairly easy to access highly granular and localised weather data for historical actuals, current observations and forecasts. Commercial providers such as Weatherbit, Dark Sky and OpenWeatherMap claim to offer accurate observations for virtually any point on the planet [7–9]. One can also access rich radar and satellite imagery covering large areas of the globe available at various update frequencies (our source provides updates every three hours) [9, 10]. This kind of data lends itself very well to machine learning (ML). Since the vast majority of the data is time series, meaning there is temporal sequence, it can be possible to show correlations between variables (i.e cloud cover and solar irradiance). There are many interesting ML approaches that have been show to work leveraging the sequential properties of this kind of data to make predictions.

There are many examples in the literature of people applying ML techniques to predict solar irradiance. Paoli et al show that use of traditional ANNs perform comparably to more conventional statistical approaches specifically ARIMA, Bayesian inference and Markov chains [11]. Alzahrani et al have demonstrated use of deep learning, in the form of a RNN, to forecast solar irradiance at sub second resolution [12]. Marquez et al have used meteorological data from the US National Weather Services and an ANN to predict solar irradiance up to 6 days out [13].

Thanks to an industry partner, Elastacloud, access to a broad range of data is possible. Metrics from numerous solar installations across the United Kingdom, both industrial scale solar farms and home solar installations. Additionally, access to a repository of rich weather data as described above.

1.1 Aims and Objectives

The main aim of the project is to build a system capable of accurately predicting solar irradiance. The predictions should be at a high enough granularity to be of practical use as described above. The irradiance predictions can in turn be used to predict power output of solar generators such as photovoltaic solar farms or home installations, providing

a commercial advantage. The performance of the model(s) can be measured against existing industry approaches to predict this value.

- Investigate current state of the art approaches for predicting both solar irradiance and cloud movements. Also investigate approaches to image / video frame imputation.
- Design and develop a model to predict future location of clouds from satellite and ground based observations
- Design and develop a model to interpolate "missing frames" in an image sequence of clouds
- Design and develop a model to predict solar irradiance based on location / movements of clouds
- Combine all 3 models to give a more granular sequence of predictions
- Create Visualisations to provide a deeper understanding of how predictions are being made

Related Work

In the process of developing a fully functional system there are many issues that need to be addressed. Data has to be efficiently processed and stored. Systems and endpoints have to be set up in oder to integrate with existing systems and make data accessible. However, from a research perspective, the project can be split into three main problems that need to be addressed to reach its goal.

This section will cover the following topics:

- Regression — Predicting Solar irradiance given the state of the weather (Section 2.1)
- Nowcasting — Predicting the weather data in (short term) future time steps (Section 2.2)
- Imputation — Increasing the granularity of the raw data (Section 2.3)
- Performance — Approaches for measuring the perform-ace of the model(s) (Section 2.4)

2.1 Regression

Solar irradiance predictions can be thought of as a regression problem. The goal of regression is to create a predictive model, f, that takes as input meteorological state x and outputs a scalar y representing the prediction of solar irradiance. A commonly used approach to build these kind of predictive models are artificial neural networks (ANNs). They are effective a modelling complex relationships and dealing with high dimensionality input data. They can also be expanded in ways to make them better at dealing with both sequence and spacial data.

Section 2.1.1 gives a brief overview of ANNs. More advance neural network approaches are addressed in sections 2.1.2 and 2.1.3. Section 2.1.2 introduces recurrent neural networks, an approach for dealing with sequential data. Section 2.1.3 discusses convolutional neural networks, a technique for dealing with spacial data i.e images.

2.1.1 Feedforward Neural Networks

A neural network is comprised of multiple nodes called neurons. Although inspired, in part, by the biological neuron they are distinct. Each neuron takes multiple values (x) inputs, performs a weighted sum and applies an activation function on the result to determine the value of its output. The classical ANN is a Feedforward Neural Networks, simply meaning there are no cycles in the network (graph) of neurons. The network is built by combining multiple neurons into layers and stacking layers on top of one another. The first layer (layer 0) takes its values from the input, for layer n the results of the activation functions of all neurons at layer $n-1$ are taken as inputs [14].

Appropriate values for all the weights (ω) of the network are usually "learned" through back propagation. There is a large body of existing work exploiting ANNs to perform regression for various use cases [14].

2.1.2 Recurent Neural Networks

A major limitation of standard neural networks is their assumption of independence among examples. After an example is processed, all state is lost. This presents a problem if the data is related in time ore space. Recurrent neural networks (RNNs) aim to address this issue for time sequence data. By having the ability to "selectively pass information across sequence steps, while processing sequential data one element at a time," [15] they can model non-independent sequences of elements. Simply put, a RNN takes as input both an element from time step T_n as well as its previous sate when processing element from step T_{n-1}.

By taking in its state from pervious time steps RNNs are able to efficiently model sequence and time dependencies as information can be passed from one step to another. This is distinctly deferent from approaches where multiple time steps are concatenated together. They set a fixed bound on the learnable sequence, thus "precluding modelling long-range dependencies " [15]. RNNs avoid this issue and can model arbitrary length dependencies.

2.1.3 Convolution Neural Network

In a traditional neural network all neurons are fully connected, each neuron takes as input all outputs from the preceding layer. A neuron at layer n receives inputs from all nodes at the pervious layer $(n-1)$, and so requires a corresponding number of weights. A [10X10X3] image as input would results in 300 weights per neuron in the first layer. As images increase in resolution the number parameters will increase accordingly. Using images of a more realistic resolution ([100X100X3]), combined with many neurons per layer and multiple layers, the number of parameters for the network (ω) will quickly grow to an unreasonable size. Having too many parameters increases training complexity and can lead to overfitting [16].

Convolutional Neural Networks (CNNs) exploit the spacial nature of images to reduces the number of connections and so parameters. Rather than looking at all inputs, each neuron of a convolution layer looks at (takes as input) a $3D$ subsection of the previous layer. Additionally, pooling layers can be used to perform downsampling, reducing the overall dimensionality of the data [16].

2.2 Nowcasting

Nowcasting, in the meteorological context, is a process that "maps the current weather, then uses an estimate of its speed and direction of movement to forecast the weather a short period ahead; assuming the weather will move without significant changes" [17]. Various approaches for nowcasting exist and the specific one used depends on the timescale and the meteorological measure to be forecast [18]. A major challenge of nowcasting is dealing with the chaotic nature of the weather.

As described above, taking a set of know states to feed through an ANN in order to predict solar irradiance is, whilst not easy, a fairly trivial problem to obtain usable results with. This, however, relies on the data used to make the prediction already existing. As described in the motivation, being able to predict how much irradiance a solar farm will receive *now* is of little use. By combining the approach stated above with nowcasted values, we can provide accurate values at least a few hours into the future.

2.2.1 Existing Approaches

One approach for short range weather predictions is to use numerical models [19]. By combining images and other measurements of the weather at T_0 and running them through a simulation of the physical equations in an atmospheric model it is possible to calculate expected values for $T_1, T_2 \dots T_n$. However, this is computationally expensive and often reserved for longer term predictions [18].

Another approach is to use to use optical flow. Taking weather images, ether from radar or satellite and extrapolating movement forward in time [20]. This approach is explored in more detail in further sections.

2.2.2 Novel Approaches

Deep learning and other machine learning techniques have been used, with success for nowcasting [18]. An interesting example of this is the use of a Convolution LSTM network [19]. Weather radar image sequence can be thought of having both spacial and temporal aspects. LSTMs are know to be effective at modelling temporal sequences [15]. The use of a fully connected LSTM, in this instance however, results in "redundant" connections. Adding a convolution in the input to state, and state to state transitions of the model, the spacial aspect of the data can also be represented. Using this technique with weather radar data from airports they were able to outperform existing approaches. The authors also discuss how this approach can extend to more generic spatiotemporal sequence predictions.

2.3 Imputing / Interpolating Data

The granularity of the raw data, for some sources is much lower than the target output. For example, the satellite imagery is only updated once every three hours, a much lower frequency than the required output granularity of one hour. There are two ways to approach this, add a "flag" to the predictive model so that it can learn to adjust its predictions for data that is misaligned to its output target time. Alternatively, one could attempt to impute (fill in) the missing data.

For the sake of compartmentalisation and expandability imputation is the approach taken (it is also a more interesting solution). This makes it possible to assume the input data is of the same frequency as the output for the predictive irradiance model, greatly simplifying it. This enables improvements to be made independently to each model whilst also reducing each models respective complexity. This kind of ensemble learning, combining multiple models to perform a more complex task, has been shown to be effective [14]. It also means that improving overall system performance can be done in multiple ways.

This approach also has the advantage of potentially simplifying switching to alternate, more accurate or granular data sources down the line. If for example, a new source of cloud cover imagery was used that had a much higher granularity it could be simply be swapped in place of the imputation.

There are various statistical methods for imputing sequence data. Filling in using the last value, taking the midpoint between $t-1$ and $t+1$. However, for this use case they have two major issues. The cloud cover data, as well as any other meteorology data that might be used, is a 2D matrix. They required a point in the future, e.g $t+1$, to be known. Whilst this is possible at training time, any production system would not have that luxury and so would have to use a predicted value (either nowcasted by the system using the approaches above or from external forecast providers). This approach could be practical however it would add more uncertainty to the system. A more ideal solution would be use a single model that can take in the lower frequency raw data and output nowcasted values in the target frequency (potentially using its previous output as input).

An interesting comparison that addresses both the 2D and the temporal nature of the data is video. A lot of work, in recent years, has been put into both next frame prediction and intra frame prediction. This comparison was the inspiration for [19] and avenue explored as a potential solution.

The following sections will look at approaches that leverage the sequential nature of the data to attempt to impute the missing values. Section 2.3.1 gives an overview of optical flow. Section 2.3.2 explores the use of Generative Adversarial Networks.

2.3.1 Optical Flow

Optical flow is a technique commonly used in computer vision (CV). "The goal of optical flow estimation is to compute an approximation to the motion field from time-varying image intensity" [21]. By taking in a sequence of images, a $2D$ motion vector in the image plane can be calculated for each point. Using these motion vectors, each points next position can be calculated and the next frame generated. This approach can be used to predict the future movements of objects within a scene based on their existing trajectories. Optical flow has been applied to many CV problems such as object detection and tracking, movement detection and robot navigation [22] Optical flow has also been used for nowcasting, predicting the next "frame" of weather radar images [19, 20].

Whilst not a perfect fit, it would be possible to take the lower frequency input and produce values with a higher granularity. One potential shortfall of this approach is that optical flow does not account for the chaotic nature of weather.

2.3.2 Generative Adversarial Networks

A relatively new approach for image generation are Generative Adversarial Networks (GANs). Initially described by Goodfellow et al, a GAN is comprised of two networks a "Generator" and a "Discriminator" [23]. The Discriminator D attempts to predict the probability that an image was created by Generator G or drawn from real examples. Both the Generator and Discriminator are trained simultaneously. The Generator learns by attempting to output images that D cannot confidently classify, if G is successful D's confidence for all images is 0.5. At the same time the Discriminator learns to differentiate images produced by G and those drawn from "true" sample space.

By using complex and novel architectures for both G and D, GANs have successfully been used to generate highly photo realistic images [24].

Mathieu, Comprie, and LeCun have extended the use of GANs for video frame interpolation and next frame prediction [25]. They also attempt to address a perceived fuzziness of generated images by combining both a standard GAN with an additional loss to "based on the image gradients, designed to preserve the sharpness of the frames." By combining the confidence level of D with this new loss to train G they were able to generate objectively sharper images. This approach is interesting as, for cloud imputation, the additional loss could be a measure of prediction accuracy.

There are two major differences between the uses above and the proposed use:

- The time scales a vastly different. In the intra frame generation, the differences are a less than a second. The cloud images are over hours.
- There is no simple measure of ground truth. Only data from time step T_{3n} is available.

2.4 Performance Measures

Having clearly defined perforce metrics that can be used for all models is vital for comparing them. It is also important to make sure that the metrics encompass various aspects of the models performance, e.g accounting for outliers etc, whilst being easy to understand and objectively compare. Both RMSE and MAE are potential performance metrics however they only tell half the picture. Since the data is highly sequence dependent another metric to use could be the coefficient of determination (R^2).

2.4.1 Industry Metrics

Some of the metrics used to evaluate model performance in industry are:

- Coefficient of Determination
- Mean absolute scaled error
- Error monthly rate — the absolute difference between predicted and actual irradiation is scaled accordingly to the maximum average irradiation for that month
- Error range distribution — the absolute difference between predicted and actual irradiation
- Error percentage per day — can only be used if the train/test split is not random

2.4.2 Synthetic Experiments

Another way to gain insight into various models relative performance is through the use of synthetic experiments. An interesting synthetic benchmark proposed in the literature is a moving MNIST [19]. The idea is to have a sequence of images containing digests drawn from the MNIST dataset. At each time step the digits move within the frame.

Design

There are three main components to the system design. The data collection system, this includes the pipelines that fetch the raw data as well as data life cycle management. The predictive model and training processes, including any

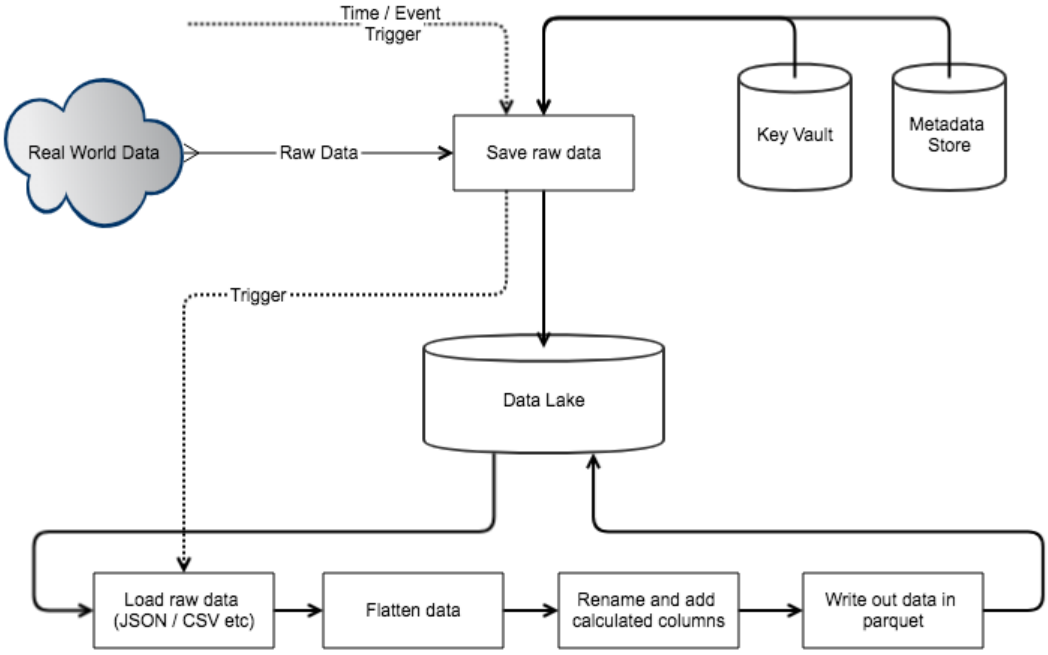

Figure 1: Detailed view of data collection pipelines

data pre-processing steps as well as model tuning, evaluation and exportation. The production system, comprised of any APIs that are exposed, real time feeds and production databases, CI/CD etc. The main technology stack is built on the Microsoft Azure cloud, so many of the systems leverage functionality provided by products offered by Microsoft.

3.1 Data Collection

The data collection system, at present is fairly simple. It ingests all raw data and lands it into the data lake, a central repository for all data that can scale as needed. It is composed of two separate systems. A set of Azure Function that run periodically, every 30 minutes collecting weather data and every three hours satellite imagery from various sources.

The second process runs as a scheduled Databricks notebook (Apache Spark). It loads the raw data, reshapes it, and saves it in parquet format. This pipeline can also be extended to run and export predictions to a database or other "sink".

An overview of the pipelines can be seen in figure 1.

3.2 Predictive Model(s)

This one of the main goals and a key output of the project. Designing the model is still an ongoing process. However, based on the outcome of the literature review my current thinking is as follows. Use ensemble learning to combine multiple models to build the system. The system would be comprised of two models. A GAN to both nowcast and impute data. A regression model that takes the GANs output and predicts irradiance values for the next hour.

The GAN will generate data to fill in the gaps in granularity and nowcast. Since the cloud cover imagery could be thought of as a video with missing frames. It makes sense to extend the work done Mathieu et al by and use of the method of combined loss [25]. The losses combine to build a description of expected aspects the missing data. In the case of imputing cloud cover, the output should look like clouds, there should be some kind of sequential aspect.

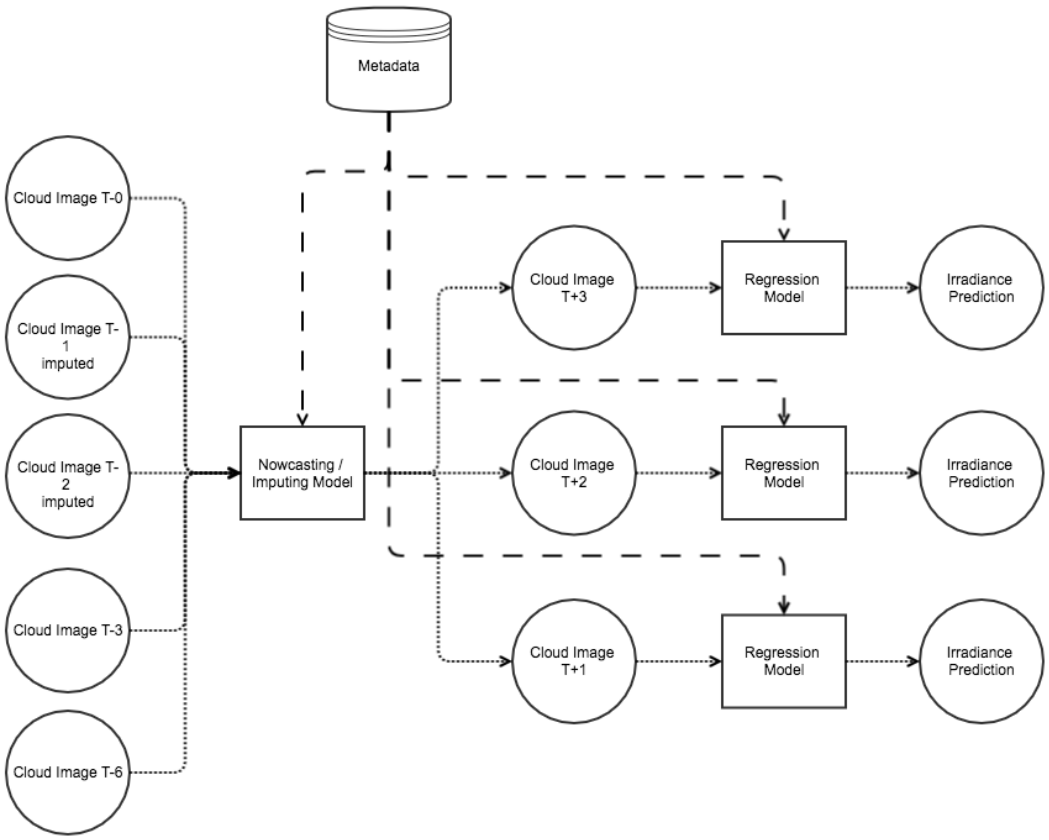

Figure 2: Overview of the model(s)

The output of the GAN would then be feed into the regression model to generate the final irradiance predictions as shown in figure 2.

The regression model will produce the final value, as input it takes:

Meteorological data Cloud cover although could be extend to add temperature, pressure, rainfall etc

Metadata Hour of day, Clear Sky value (calculate irradiance if conditions were prefect)

3.3 Production System

The main goal of the project is to output irradiance predictions in a usable format. For now, very little effort has been put into designing the final production system as it is secondary to a performant model. A high level overview of how the various pipelines and models fit together can be seen in figure 3. Additionally, there are some high level NFR and FR for the system. Other aspects that have to be taken into account are version control and model management.

3.3.1 Requirements

Non Functional Requirements

- Minimal practical latency between new data entering the system and predictions being available
- Easy expandability of the model, adding new data sources or expanding the scope of its predicting e.g a new location

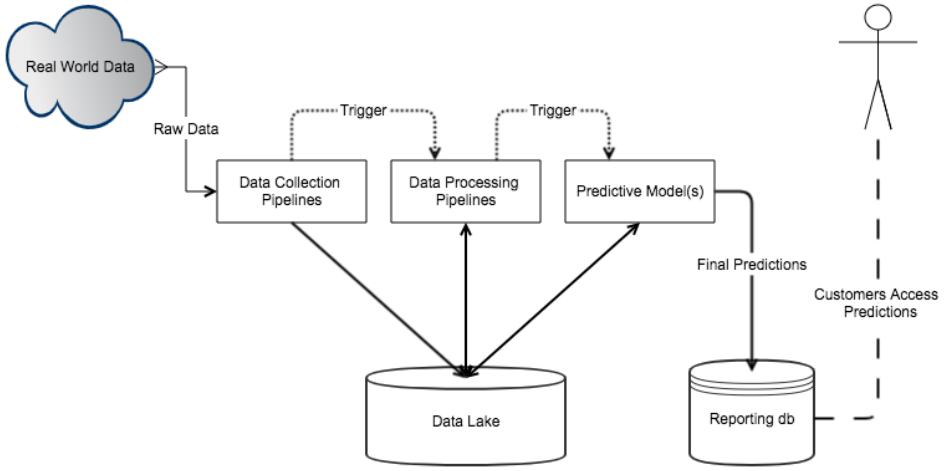

Figure 3: Overview of the main components of the system outlining the data flow

- Comparable performance to existing approaches

Functional Requirements

- Provide irradiance predictions for all locations with existing predictions
- Automatically fetch and process data
- Provide visualisation capabilities for both predictions and performance measures
- Exported model be able to function in a distributed framework

Progress

This section outlines the progress that has been made in the project to date. It gives an overview of work not necessarily coved in the sections above. It also contains critical review of the current state of the project.

4.1 Project management

The project has been run with a waterfall like approach, with tasks being systematically done moving down the GANTT chart from the project proposal.

4.1.1 Data Feeds

An important part of this project is data wrangling. Setting up and scheduling feeds to collect appropriate data. Building out pipelines to reshape, restructure and reformat the raw data. Implementing data life cycle processes, ensuring data is catalogued and easily accessible. A lot of work in the first few weeks was spent setting up and ensuring the stability of the pipelines. Specially the weather data and satellite imagery as access to historical can be limited. Without a large enough dataset, the project cannot provide any practical results beyond synthetic benchmarks. Getting this work done was vital for success.

Moving forward, I hope to automate some of the data collection directly from solar farms across the UK. Additionally, dependant on external factors, there may be data from smart meters attached to home solar installations from over 10,000 houses across the UK.

4.1.2 Data Analysis

Another important first step undertaken, was to perform some basic analysis of the raw weather data. The goal of this was to look for trends and validate some of the "industry wisdom". Overall the outcome was mixed.

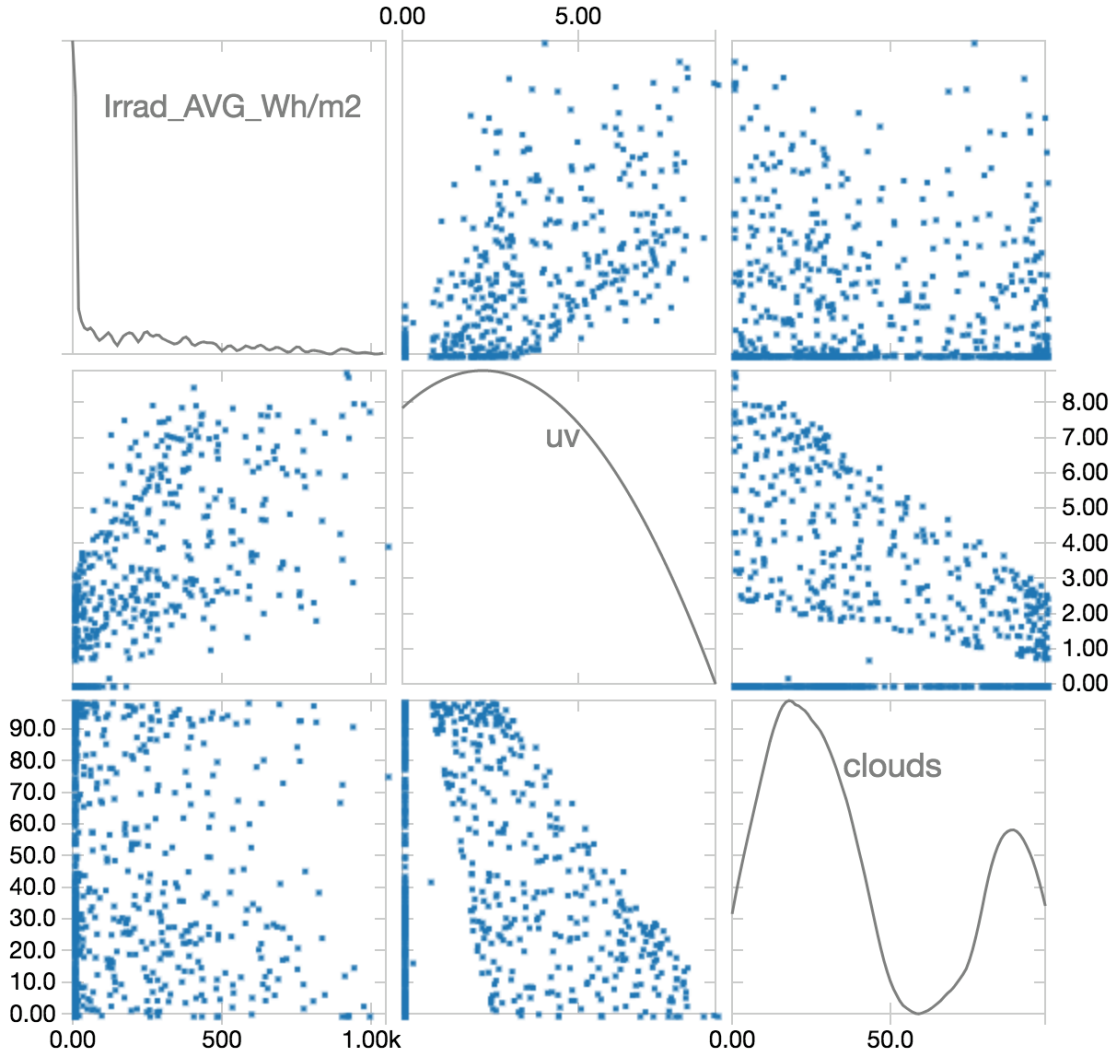

Figure 4: Scatter plot of Cloud Cover vs Irradiance vs UV Index

A simple plot of irradiance, cloud cover and UV index clearly demonstrates many of the finding, as shown in figure 4. When looking at all the data, there is a clear correlation between cloud cover and UV index. There is also a correlation between UV index and irradiance. Unexpectedly there seems to be an only slight correlation between cloud cover and irradiance. There are a few explanations for why the expected correlation was not seen:

- Cloud cover is not a major factor in Irradiance, which would go against industry wisdom
- There is a time factor, a lag between cloud cover measurements and reporting / update time.

- The data is wrong. As with all the date used, it is provided by external commercial sources. There are no guarantee that their equipment is working or how accurate the data is, an no practical way to validate it.

After digging deeper into the data it became apparent that although weather data is available in hourly intervals its veracity at this granularity is questionable. A good example of this is can be seen in figure 5. This data is an image representation of point cloud cover measurements over the UK showing a clear change in resolution. This change in quality as well as the slower than expected updates could explain some of the anomaly seen.

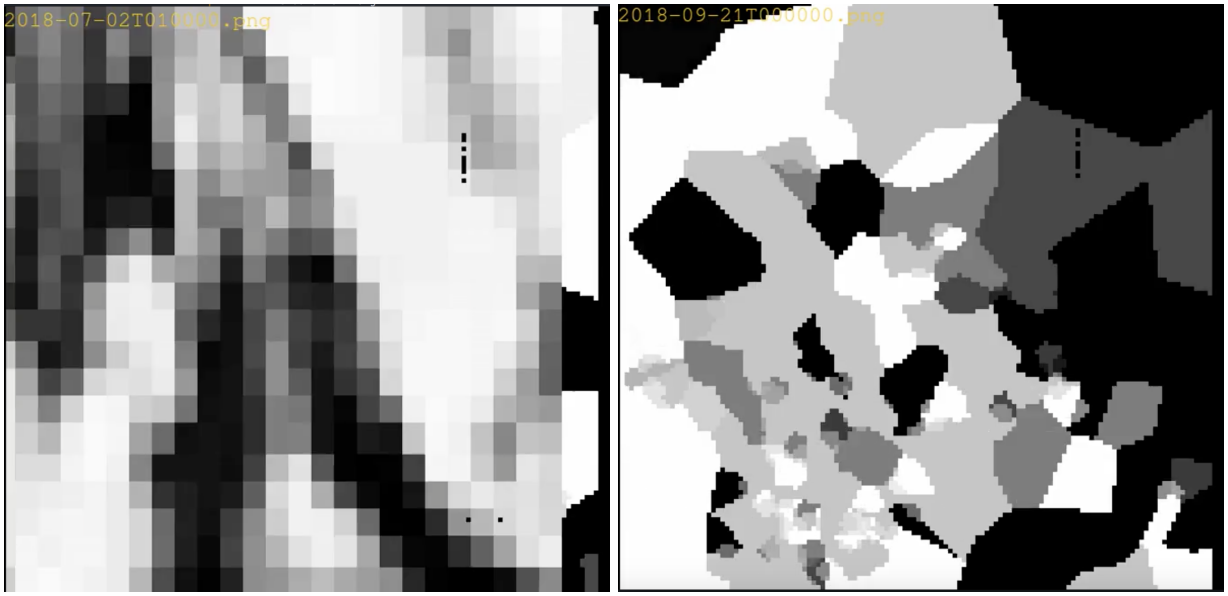

Figure 5: Example of cloud cover over the UK for Left: 2018-07-02, Right: 2018-09-21

One of the providers used for raw weather data is *weatherbit.io*, who recently published a blog outlining work they have done to improve rainfall accuracy by adding new satellite based sources [26]. This highlights a potential issue with using these kind of providers and should be to taken into account when anomalies arise in the data.

After analysing the cloud cover data, I came to the conclusion that the cloud cover measure is derived from satellite imagery whereas the UV index is a ground based measure. If the cloud cover is derived from satellite imagery, then all of the issues previously stated with satellite data will apply and can explain the messy correlation.

4.1.3 Preliminary models

As an exploratory exercise, some preliminary models were built. The main goal of this exercise was to gain familiarity with both Tensorflow and Keras as they will be heavily used to build the final solution. In addition to gaining familiarity with the aforementioned frameworks, new cloud based technologies were used and evaluated. This will be helpful as training on the cloud will likely be vital to enable rapid iteration of model design and evaluation next semester. Because this work was conducted primarily as a learning exercise, no experimental rigger was used and as such no results will be reported.

The simple models used a subset of the data to train and where much smaller than the envisioned final solution. Given this restriction and lack of any tuning, although no objective result can be provided, their initial performance appeared promising.

4.2 Contributions and Reflections

Whilst a lot of initial progress has been made, overall the project is not where I had hoped it would be. Although disappointing this is not surprising. Doing a 70 / 50 credit split I knew there would be conflicting priorities for my time and the higher than normal workload would affect my progress. The anticipation of slippage can be seen in the GANTT

chart in my proposal. I was perhaps a little over-optimistic in the amount of progress that would have been made before the Christmas break, but an ample amount of time next semester was set aside to make up or expand scope depending on progress.

This said, it is important to recognise that a lot of groundwork has been done. Many of the data pre-processing pipelines have be built out and are running. The data analysis has validated some of my assumptions, and initial processes for visualising the data have been built. Preliminary models have been built using (in part) real data. Most importantly, review of existing work has reviled promising avenues for exploration.

Moving forward, after the exam period, I hope to be able to put much more time into focusing solely on dissertation and speed up the work. Early on I need to flesh out an initial topology for my model as well as build versions of the models described in the literature. Completing this work is vital to allow enough time to train, run experiments and evaluate results.

I will also change how I work to a more SRUM like approach with defined sprints, sprint goals and clearer tasks. This will enable better time management and ensure work progress by setting clear targets and deadlines. It will also give me more flexibility to adapt to potentially new data sources as well as adjust based on experimental results and suggestions from stakeholders.

4.2.1 GANTT

Figure 6 is the GANTT chart from the proposal, figure 7 is the revised GANTT chart. The major changes made are:

Coursework catchup (N1) was expanded, and took three weeks instead of the planed one. Whilst not unexpected, it did slow down progress pushing the project behind ideal scheduled. The knock-on effect can be seen in the slippage of most tasks after Nov-19. That said, a lot of useful content was learned that can feed back into my work moving forward. Especially from the machine learning model, specifically around experimental methodology and model evaluation and comparisons.

Real Data (D3) expanding the model to use real data has been extended. Although progress has been made in that regard, the preliminary models were using real data. Running real experiments with them using both irradiance data and the cloud imagery has yet to be done.

Build Literature Models (D5) A new task to build the models described in the literature. This was suggested by my supervisor and will enable comparisons in synthetic benchmarks.

Coursework (N4) More time has been allocated for catching up with coursework next semester. One of the major issues this semester has been keeping up with other modules. The expected workload next semester is lower, but ensuring there is enough time to keep up is vital to success.

References

[1] Matt McGrath. Final call to save the world from 'climate catastrophe'. `https://www.bbc.co.uk/news/science-environment-45775309`, October 2018.

[2] Geraldine Ang, Dirk Röttgers, and Pralhad Burli. The empirics of enabling investment and innovation in renewable energy. *OECD Environment Working Papers*, No. 123(123), 2017.

[3] Paul Denholm, Matthew O'Connell, Gregory Brinkman, and Jennie Jorgenson. *Overgeneration from solar energy in California: a field guide to the duck chart*. National Renewable Energy Laboratory Golden, CO, 2015.

[4] Aurora Energy Research. Intermittency and the cost of integrating solar in the gb power market. `https://www.solar-trade.org.uk/wp-content/uploads/2016/10/Intermittency20Report_Lo-res_031016.pdf`, September 2016.

[5] Ian McKenzie Smith Keith Brown Edward Hughes, John Hiley. Electrical energy systems. In *Hughes Electrical and Electronic Technology*, chapter 39, pages 824–828. Pearson/Prentice Hall, 2008.

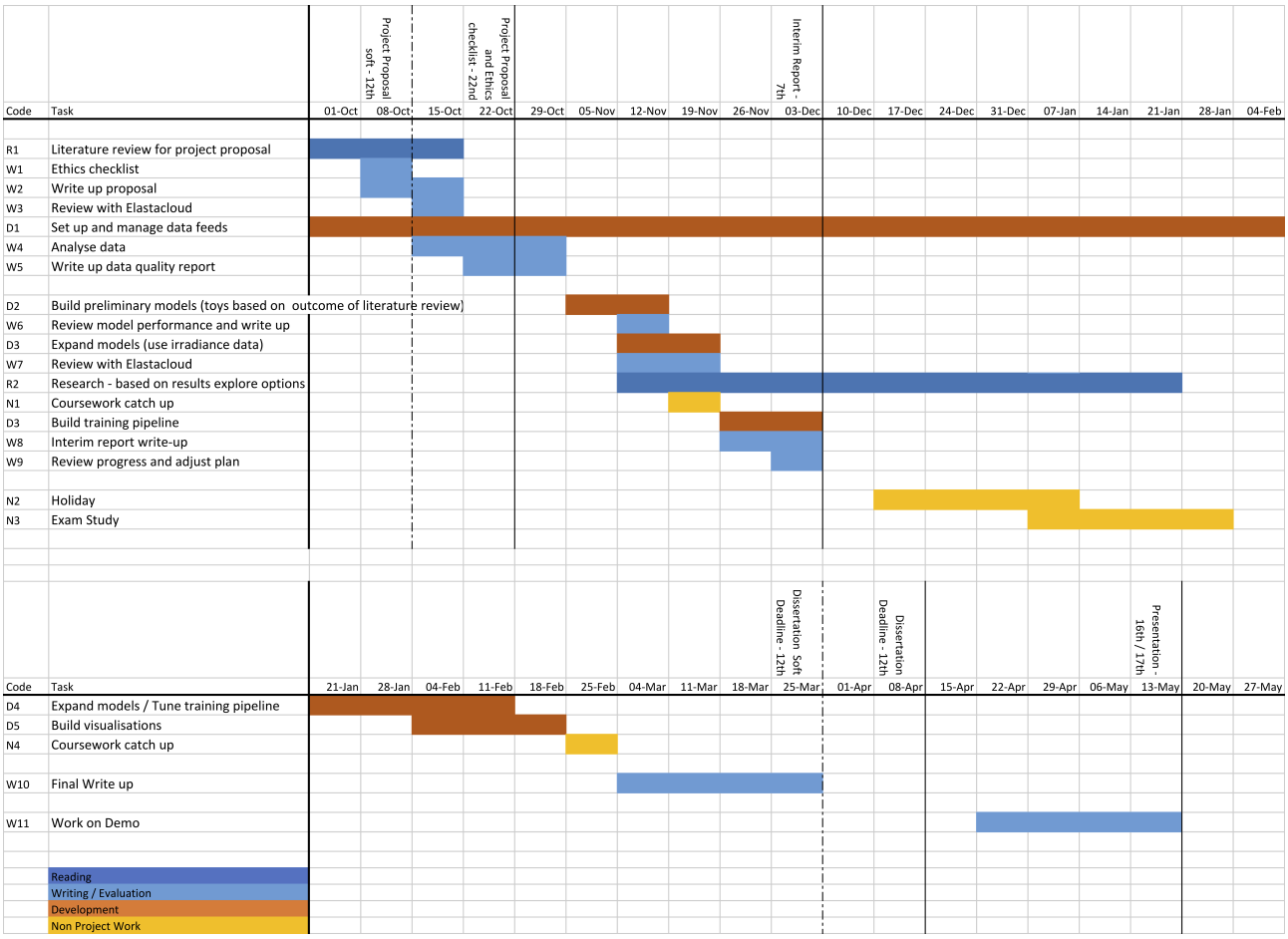

Figure 6: GANTT chart submitted in the proposal

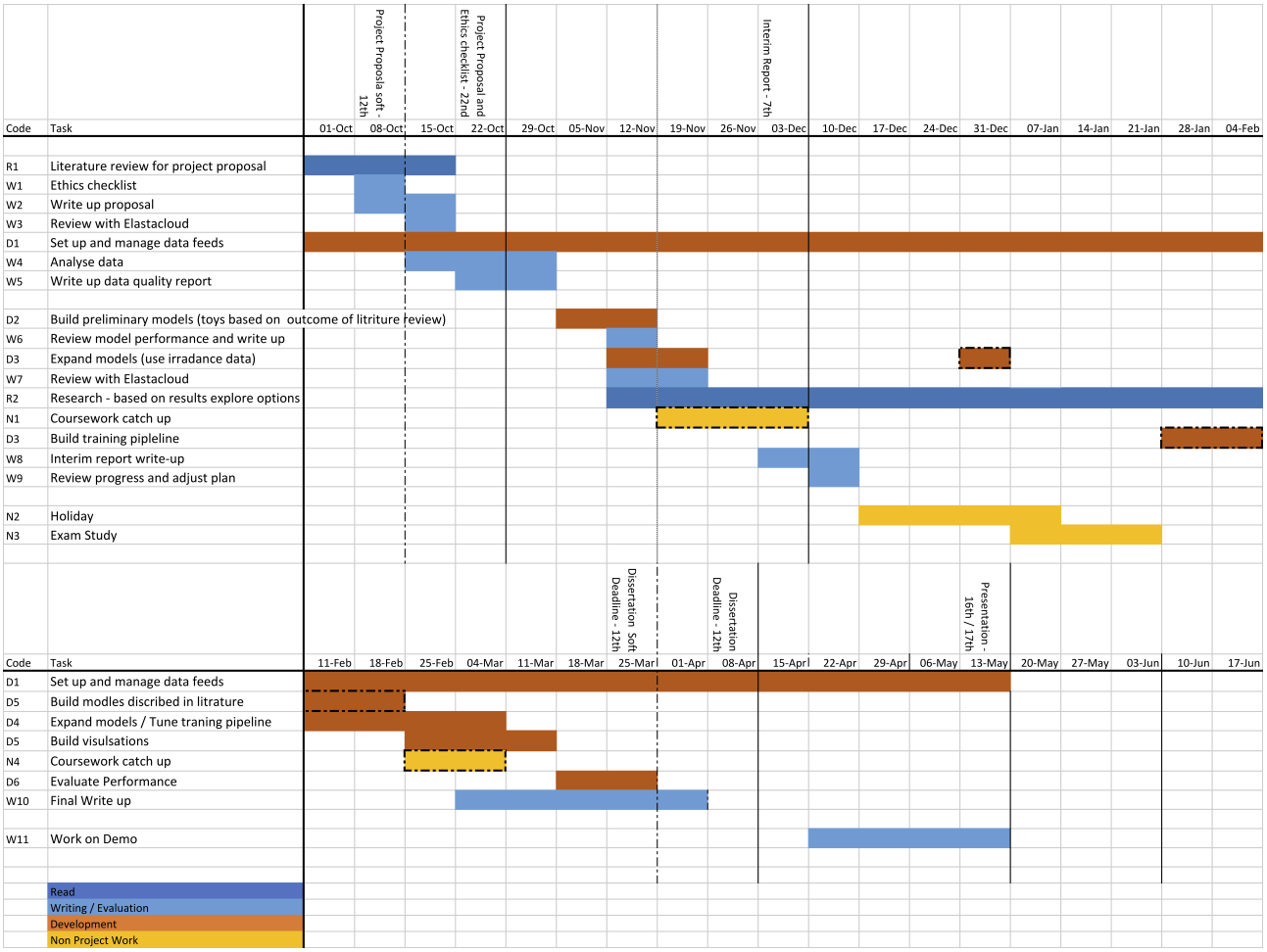

Figure 7: Updated GANTT chat with the progress adjustments and new work added. Big changes have been highlighted

[6] Richard Conway and Sandy May. Using azure databricks, structured streaming, and deep learning pipelines to monitor 1,000+ solar farms in real time. https://databricks.com/session/using-azure-databricks-structured-streaming-deep-learning-pipelines-to-monitor-1000-solar-farms-in-real-time, October 2018.

[7] Data source - weatherbit. https://www.weatherbit.io.

[8] Data source - darksky. https://darksky.net.

[9] Data source - openweathermap. https://openweathermap.org.

[10] Data source - met office. https://www.metoffice.gov.uk.

[11] Christophe Paoli, Cyril Voyant, Marc Muselli, and Marie-Laure Nivet. Forecasting of preprocessed daily solar radiation time series using neural networks. *Solar Energy*, 84(12):2146 – 2160, 2010.

[12] Ahmad Alzahrani, Pourya Shamsi, Cihan Dagli, and Mehdi Ferdowsi. Solar irradiance forecasting using deep neural networks. *Procedia Computer Science*, 114:304 – 313, 2017.

[13] Ricardo Marquez and Carlos F.M. Coimbra. Forecasting of global and direct solar irradiance using stochastic learning methods, ground experiments and the nws database. *Solar Energy*, 85(5):746 – 756, 2011.

[14] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[15] John Berkowitz Zachary Chase Lipton and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019, 2015.

[16] Cs231n convolutional neural networks for visual recognition. http://cs231n.github.io/.

[17] Met Office. Nowcasting. https://www.metoffice.gov.uk/learning/making-a-forecast/hours-ahead/nowcasting.

[18] Cyril Voyant, Gilles Notton, Soteris Kalogirou, Marie-Laure Nivet, Christophe Paoli, Fabrice Motte, and Alexis Fouilloy. Machine learning methods for solar radiation forecasting: A review. *Renewable Energy*, 105:569 – 582, 2017.

[19] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, 2015-January:802–810, 2015.

[20] W.-C. Woo and W.-K. Wong. Operational application of optical flow techniques to radar-based rainfall nowcasting. *Atmosphere*, 8(3), 2017.

[21] David Fleet and Yair Weiss. Optical flow estimation. In *Handbook of mathematical models in computer vision*, pages 237–257. Springer, 2006.

[22] Kelson R. T. Aires, Andre M. Santana, and Adelardo A. D. Medeiros. Optical flow using color information: Preliminary results. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, pages 1607–1611, New York, NY, USA, 2008. ACM.

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[24] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016.

[25] Michaël Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *CoRR*, abs/1511.05440, 2015.

[26] colinc. Historical weather api update - historical satellite based precipitation. https://www.weatherbit.io/blog/post/historical-weather-api-update-historical-satellite-based-precipitation, June 2018.