Final report - Solar Forecasting Service

Section 1: Introduction and objectives

1.1: Introduction

As the world is becoming more aware of the need to move away from non-renewable sources of energy and towards green solutions, solar power has become a popular choice. Once installed, it produces free energy with zero greenhouse gas emissions. Personal solar panel usage is increasing in Australia with more that 29% of homes having rooftop solar panels as of March 2021 [1]. The technology is also continuously improving to make it more efficient and cost effective to install, and so this number is sure to rise rapidly.

However, solar power is a variable source of energy. It is highly dependent on the weather, meaning a cloudy day can seriously reduce the amount of power a solar array will produce, on top of the fact that no energy is produced overnight. As a consequence, we still require energy from other sources to meet the demand. While it can be supplemented with other sources of renewable energy like wind power, these often face the same variability problems. This variability, which is often difficult to predict, can cause issues for the energy distribution companies that need to supply the extra power. Not supplying enough power can cause major disruptions for the population, on the other hand, having an excess amount of energy is a huge waste and very expensive. Therefore, in order to appropriately supplement the amount of power being generated by variable renewable energy sources, we generally require the use of dispatchable sources [2]. These are sources that can instantly supply electricity on demand, including hydroelectricity and batteries. However, the infrastructure requirements of such sources mean they are not always available.

Coal power, which is a non-variable energy source, is reliably used across Australia. However, it is generally not considered a good dispatchable source, as it cannot promptly or flexibly supply energy when demand changes [3]. Coal power plants can take up to 12 hours to reach full operating speed [4], and by that time the demand could have completely changed. Therefore, in order for it to provide adequate supplementation to a variable renewable energy source, a lot of warning is needed.

Our solution to this problem is to create a service that can predict the solar power generated by solar panels, so that the need for supplementary energy can be foreseen. Solar power is almost entirely determined by the local weather conditions [5], and thus we can utilise local weather forecasts to predict the power that will be generated. This will enable energy companies to make important decisions about how much power to produce, and thus decide when to turn on coal-fired plants.

Furthermore, once energy is generated it must then be transmitted to customers through the grid. The physical infrastructure of the transmission network dictates where energy can come and go [6]. Therefore, the questions are not just when and how much is required, but where it is required. A sunny day in Western Australia does not mean all the power plants in Queensland can be switched off. More specifically, the transmission network consists of a series of substations that each serve different regions. These substations are where the generated electricity is converted from high voltage to low so it can be put into powerlines. An example of these substation regions is shown in Figure 1. The proposed solution therefore needs not only to predict the amount of solar power that will be generated from the forecasted weather, but indicate where this energy will be produced. This will enable energy distributors to provide adequate energy that meets the demands of different regions.

Figure 1: AusGrid network substations. Adapted from [7]

1.2 Objectives

To specifically address these problems, we defined the following objectives

- To create a reliable forecast of solar power generation for a time period appropriate to making decisions regarding supplementary energy.
- To present this forecast in a way that immediately aids energy providers in deciding when and where supplementary energy will be required.
- To create a framework that is scalable and can adapt to any future changes to the solar power market.

Section 2: Design methodology and architecture

With the above objectives in mind, we followed CPS design principles and created a FBS decomposition to model our desired service.

2.1 Design principles

Our interpretation of the cyberphysical system Design Principles [8] in this application were:

- For *technical assistance and service orientation*, we wanted to create a service that was directly targeted at the intended user, who is an energy provider. This meant creating a user interface design where the information was presented in the most useful way when making decisions about power supply and distribution. Based on this, we wanted to present solar power forecasts on a map that show the energy being produced in different regions. The user can therefore see when and where supplementary energy would be required. We also wanted to choose a forecasting window that was most relevant to making these decisions. Since turning on power plants requires a 12-hour warm-up period, this meant forecasting 24 hours in advance.

- For *interoperability and interconnection*, we needed our service to connect with data from a range of sources to have the ability to make insightful forecasts. The most important of these are weather data and solar power generation data. We also wanted users to be able to connect with the service from any location.
- For *real-time capability*, we wanted to make use of the most recent weather forecasts. This would allow us to give an accurate prediction of power generation for the 24 hours after the time at which the service is being used. We therefore wanted this to be continuously updating.
- For i*nformation transparency/virtualization*, we wanted to feed such data into a model that accurately predicts solar power generation based on weather. This model could then be deployed virtually through the service.
- For *modularity*, we wanted to create a service that was easy to adapt and scale. In particular, we wanted to implement a 'black box' model that relates weather data to solar power generation. This would allow further improvements of the model to be directly implemented without affecting other components of the service. This model should be able to be applied to all solar panels we want to predict. In terms of scalability, adding additional solar panels to our model and visualisation should be easy. Furthermore, when implementing the visualisation, we wanted to ensure that additional features could easily be added, which meant implementing the user interface on mapping software that provided this flexibility.
- For *decentralization and autonomous decisions*, we wanted to create a service that, when integrated with other energy forecasts, creates a comprehensive prediction of energy generation and demand. From this, any gaps in the electricity supply could be calculated. This would have the potential for the service to make decisions like autonomously turning on power plants, and appropriately distributing energy. However, creating these other forecasts was outside the scope of the service at this stage.

2.2 Function-behaviour-structure decomposition

We modelled the implementation of this vision through a FBS decomposition. The diagram of this is shown in Figure 2, and is followed by a detailed explanation.

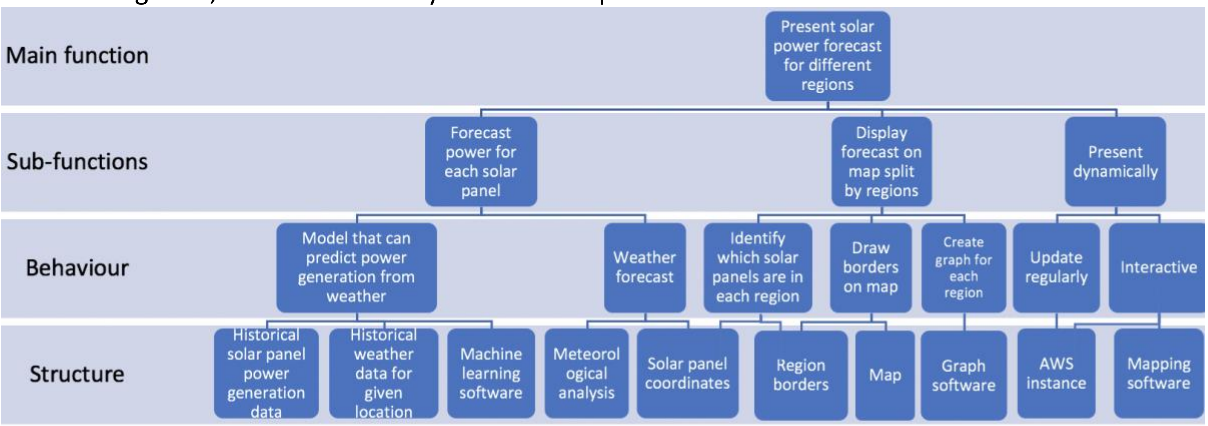

Figure 2: FBS for the main function of our service.

Our main function is to present solar power forecasts for different regions. Based on our identified problem, we wanted to design a service that provides a comprehensive overview of the amount of solar power that will be generated, rather than simply giving a specific prediction for individual solar panels. We therefore decided to present the information in a geospatial way, so that the forecasted power generation in different areas could be conveyed. By totalling the forecasted power across all solar panels in a given region, this would inform the user on whether that region is lacking or has excess energy. They can then decide if and where to turn on power plants, so that all areas of Australia have sufficient energy, without any waste.

The first sub-function to achieve this is to create an accurate forecast of power generation for individual solar panels. The foundation of this is a model that can predict power generation from weather. To train a model to do this, we needed machine learning software and historical data. More specifically, this would require historical power generation for a range of solar panels. To match this with the corresponding weather data, this data also needed to be timestamped and geotagged. We would then need to obtain weather observation station data for the given time period and location of our solar panels. By considering time-series weather data as an input sequence to a model, and the corresponding power generation as the output sequence, we would be able to use this to train a machine learning model. The other behaviour required to implement a forecast for the power generation of solar panels is to first forecast the weather. This requires the location of each solar panel being forecast, and a meteorological analysis of the surrounding weather patterns. However, weather forecasting is widely available online, so this behaviour would be performed externally to our service.

The second sub-function of our service is to display this information in a geospatial way to allow for power forecasts in different areas of Australia to be analysed. To implement this we would first need to separate the individual solar panels into distinct regions. Structurally, this would be based on the solar panel coordinates, as well as the region borders. We wanted to ensure the region zoning was of similar size to the substation network used by energy distributors, as these different regions work somewhat independently, in the way that some regions would have excess energy and some would be lacking. However, available information on these regions is limited. Therefore data for the borders would be obtained from the electorate zoning, as this ensures each region has a similar population size, and thus we can assume similar energy requirements. The next behaviour in implementing this would be to draw the region borders on a map, based on the map and bordering data. Finally, to display the power forecast for different regions, we would need to create a graph for each region, which requires graphing software.

The final sub-function is to present the visualisation dynamically. In particular, to implement this requires us to update our predictions regularly, so that they always forecast the subsequent 24 hours from the time the service is being used. This behaviour can be achieved on an AWS instance, where these updates are continuously run in the background. Dynamic presentation of the information also means allowing the service to be interactive. We wanted the amount of information presented at once to be minimised, to reduce information overload. Thus we wanted to present different modes of our visualisation that the user can swap between, so that they can see the information most relevant to them. This includes options like solar panel pins, that display information as the user interacts with them, or when interacting with regions, they display the total forecasted power for a given region. In particular, we wanted our visualisation to include a dynamic heat map, where a colour scale can be used to represent the different amounts of power that regions are forecast to produce. This would allow the user to focus their attention to regions of concern, such as regions that are significantly lacking in power, without needing to iteratively look through all predictions. Implementing all of these ideas can structurally be achieved using mapping software on the AWS instance.

The implementation of these functions is discussed in the following section. In particular, section 3 details the implementation of creating an accurate forecast of solar power generation through a machine learning model. Section 4 explains how we displayed these on a map split by regions, and presented the display interactively, through implementing a user interface. Each of these sections discusses our considerations in the methods chosen to implement them, details how the functions were ultimately implemented, and discusses the results our service achieved with reference to the function specified.

2.3 Architecture through the 5Cs

Our service architecture is summarised using 5C architecture [9] below.

- *Connection* is where machines/sensors are connected for the acquisition of data. In this, our approach was to access available data about physical components through web scraping. This involved accessing solar power data through AusGrid, and accessing weather data from weather stations. Historical versions of this data could be used to train a model, while forecasts, in particular weather forecasts, can be used to make a prediction.
- *Conversion* is where this data is converted into genuine information. For our service, this involves preprocessing of the data. For example, we must ensure both weather and solar time series data are in the same format (ie. having hourly time steps). In addition, we matched solar panel power generation data to the corresponding weather data based on the location and time period. Thus, the weather input and matching power output data could be used to train a model, or make predictions.
- *Cyber* involves analysing the data to obtain insightful information. In this, we created a model that found a relationship between weather data and solar power generation. This allows us to make predictions on the next 24 hours of solar power generation, based on the forecasted weather at the corresponding location. This layer also includes further data analysis, such as finding the total power generated for each region, based on all of the solar panels within its border.
- *Cognition* is where this information is presented to users in a way that offers insight and aids decision making. Our user interface design implements this by allowing users to remotely view these forecasts on a map. In particular, this allows users to see a graph of the hourly forecasted power output over the next 24 hours, for different locations. This visualisation also provides a heatmap, which gives users a visual summary of the important information.
- *Configuration* involves giving feedback back to the physical components of the system based on the analysis. In this, our service does not directly create any physical outcomes. It directs the user to areas of concern, such as regions that are lacking in power, so that supplementary energy can be arranged (ie. by turning on power plants). In the future, our system could be integrated with the energy provider to automatically trigger a decision like this.

Section 3-4: Implementation of model and visualization

Section 3: Model to forecast power generation

Before reaching the final implementation approach, several different alternatives for various sections of the solution were considered. The first of these were the variations in the solar modelling system, the core of the solution which was used to produce the meaningful outputs. The objectives set for this section of the program was a system that was reliable, repeatable, accurate, and able to produce easily understandable outputs. This led to the final design solution being chosen, however there are several other options which were initially considered, each with unique benefits and challenges.

3.1.1 Considerations: Inputs and Outputs

As discussed in the design section, the desired service should be able to accurately predict how much power will be generated by individual solar panels over the next 24 hours. This time frame was chosen to give energy companies enough time to make decisions due to the 12-hour warm up time for power plants required to bring generators up to synchronous speed [4]. While a longer time frame could be used, this would use less accurate weather forecasts which will affect the accuracy of solar generation forecast. The power generated by a solar panel over a 24-hour period was therefore chosen to be the output of our model. However, there was still choice in the exact form this would take. For example, we could choose it to represent the total, average, or maximum of this over the course of a day. Ultimately, because of the 12-hour startup time for a power plant, we wanted our output to show the exact time that power generation will start to drop, so that the target user knows exactly when to start preparing supplementary energy. Thus, we chose our output to be the 24-hour time series power generated by solar panels.

We then had several choices for the input to the model. Based on our reading, we found that solar power is influenced by a range of factors, including temperature, humidity and cloud cover, as well as the capacity and age of the solar cells [5]. It was clear weather data would be an input, and other properties of the solar panel should also be taken into account.

A unique approach that could have been taken in addition to utilizing weather forecasting could have been taking an input of a satellite imaging feed, varying the inputs and data feeds considered in the final implementation. This option was potentially viable, as it is possible to determine some specific weather patterns from analysing this imagery. This would have been an interesting approach to this problem, as it would have provided a very raw perspective of the factors that directly affect solar production, reducing the stages that data is processed through before reaching this solution, and hence potentially showing a truer picture of the conditions, with all possible biases from other systems removed.

This method was not pursued due to its derivation from the source, as although it may have been possible to identify relevant factors that affect solar panels from this data, in essence this information would have been used to create a weather forecasting model, which was far simpler to implement through access of accurate weather forecasting software or models, which are far more specific to purpose and hence would be more reliable and accurate in their modelling. It was deemed far more relevant to base the system inputs off of weather data, as it contains other useful information that imaging could not supply, as well as requiring less steps to be taken in the final system, instead offloading those steps to factions whose specific purpose is to create accurate and reliable information in these areas. We therefore decided the input to our model would be the time series weather data sourced from external meteorology providers that corresponds to the time series solar outputs.

3.1.2 Considerations: Modelling Architecture

There were several machine learning models that were considered for use in our modelling architecture. In particular, we considered starting with a K-nearest neighbours algorithm [10]. When given an input, this finds the most similar input sequences from a collection of examples, and combines the outputs of these to estimate the corresponding output. We initially thought this would be a good starting point as it is straightforward to train. However, we knew eventually that we would get best results for this forecasting problem using a neural network.

Ultimately, we decided that a better starting point would be to implement a simpler version of the modelling architecture while still learning to incorporate a neural network. Thus, we chose to use a Long short-term memory (LSTM) model which is a recurrent neural network that is appropriate for time series forecasting. The following section summarises how we implemented our simple model, which we eventually based our final model off.

3.2.1 Implementation: Individual Model architecture

The first model we designed was based on data from a single solar panel. In this, the model would need hourly historical weather data for the solar panel's location, and the corresponding hours of solar power generation. The model would take one day of such time series data as a single sample. Figure 3 depicts the LSTM model architecture. In this, the input x_i is the weather data at the i^{th} hour of the day. We wanted this to include metrics such as the UV index, cloud cover, humidity, precipitation, pressure, temperature, visibility and a Boolean value representing whether the sun is up based on the times of sunrise and sunset. The neural network is represented by the A block, which would simultaneously give the output power generation in the i^{th} hour (represented by h_i) as well as passing information to the subsequent time step.

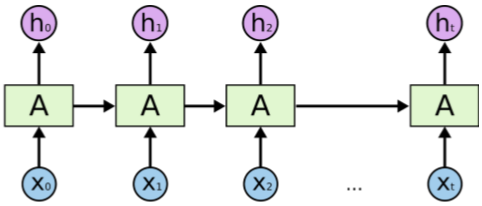

Figure 3: LSTM model architecture. Adapted from [11]

We experimented with this model until we were satisfied that hourly solar power generation could be accurately predicted using such architecture with the weather inputs described. The results of this initial model are discussed in the implementation/results section.

3.2.2 Execution of individual model

We used AusGrid's Solar home electricity data as the historical power generation data. This contains half-hourly timestamped data for 1 year for 300 customers. To train the individual model, we just used the first customer. The postcode of their solar panel is provided, and so we used a Python GPS querying module (Pgeocode) to find the closest town to the postcode. We then obtained the historical weather data by querying for the given time period and town on the WorldWeather API. This outputted hourly timestamped weather data, and so we converted our solar power generation data into hourly timesteps. Thus we were able to match the hourly weather data to hourly solar power data for the given solar panel, and split it up into 365 days worth of data. This data was normalised before applying it to the model training.

The 365 data samples were randomly grouped such that 70% was used for training, 20% was used for validation, and 10% was used for testing. We trained the model over 100 epochs using the training and validation data. The model was then used to make predictions for the test data, and compared to the actual observation. Overall, this had an average absolute error of 0.0698 kWatts, and RMS error of 0.1217 kWatts. Figure 4 shows this comparison for a random selection of days. Minimal turning from our initial results was performed, as this model had sufficiently proved to us that we could predict the hourly solar power generation from weather.

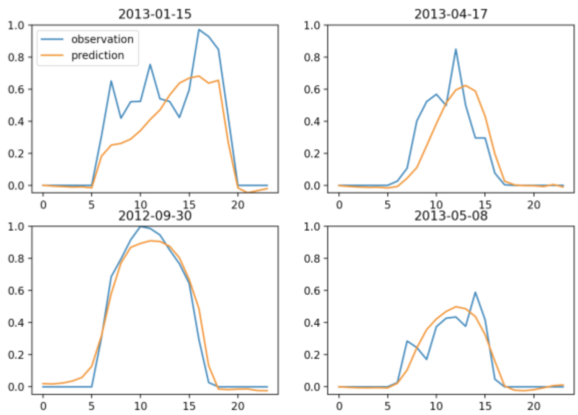

Figure 4: Observation vs prediction of testing data for model trained on one solar panel.

3.3.1 Implementation of Upscaled model architecture

Once we were sure that this model could make accurate forecasts for the one solar panel it was trained on, we needed to upscale it to make forecasts for panels in other locations. Different solar panels have different generator capacity values, which denote the power they can generate under full solar radiation. Furthermore, we hypothesised that different solar panels would respond differently to certain weather, based on the angle and direction they are positioned at, as well as whether any objects around them cause shade. Therefore, to accurately forecast different solar panels, we wouldn't be able to simply apply the model described above.

Subsequently, we considered individually training the model described above for individual solar panels. However, we agreed that training and deploying individual models was not very scalable. Furthermore, we knew we wanted our modelling to maximise the available information on how different weather affects solar power generation, which would mean making use of all weather and corresponding solar panel data available. This is particularly important with extreme weather events (eg. heat waves) which happen very rarely, meaning the data from an individual solar panel would not encompass all possible weather scenarios. Therefore, we wanted to make a model that both accounts for the unique properties of each solar panel, as well as using all available data on how weather affects power generation.

We subsequently designed a model based on encoder-decoder LSTM architecture. This is made up of two sub-models. The encoder takes the input sequences and converts it to a fixed-length internal representation, and the decoder interprets the internal representation to predict the output sequence [12].

In our design, the encoder layer would take information about the individual solar panels. This would include its generator capacity value, as well as a categorical variable representing the solar panel ID, which we chose to encode using one hot encoding [13]. The output of this layer would be a vector that internally represents this information. We knew we wanted to ensure that the size of this vector would provide enough resolution to convey the information given by each of the different solar panel IDs, as well as their generator capacity. This internal representation of the input would then be repeated 24 times, so that its size matches the number of steps in the daily time series data. Thus, the encoded vector would then pooled the time series weather data, and used as the input sequence for the decoder layer. The output of the decoder layer would then be the power generation sequence for the day.

This architecture was designed to allow the model to learn an overall relationship with weather data and solar output, but also learn how the unique properties of each solar panel related to each other. Although the encoder layer did not need to be an LSTM (as it doesn't work with any time series data), we chose to leave it in LSTM form as there were sufficient resources to implement it. With more experience using TensorFlow, we would have liked to implement a simpler model architecture for the encoder layer, such as a Dense layer, but getting the inputs and outputs into the correct shape to integrate into the rest of our model proved difficult. Nevertheless, the encoder layer is trained to encode specific details about the individual solar panel that inform how it responds to different weather. The decoder layer then allows us to use data across all of the different solar panels to learn how different weather affects solar power generation. This is particularly useful for events of extreme weather, where we have limited data on how they affect solar power generation. Thus, even if a particular solar panel has never seen such weather, data from other locations can still inform the prediction for that panel.

In terms of specific hyperparameters, a conscious decision was made for the length of the output of the encoder layer. This is an internalised representation of the input customer properties, and thus we chose a 600-element vector. This size provides enough resolution to convey the information given by the 300 customer IDs, as well as their generator capacity. Other parameters in tuning were not changed significantly from the example used as reference [12] as we believed our initial accuracy met the objectives of our service.

3.3.2 Execution of upscaled model

We used the same data described in the section on the individual model, except this time used data from all 300 customers. Each day for each solar panel was considered a single sample, such that overall we had 109,500 data sequences. This was randomly sampled such that 70% was used for training, 20% was used for validation, and 10% was used for testing. We tuned and trained the model over 100 epochs using the training and validation data.

3.4.1 Results and discussion of upscaled model

As outlined in the objectives of our service, we needed to create a reliable model for predicting solar power generation. To test the accuracy of our chosen model, 10% of the historical data was withheld for testing. The model was used to make predictions for the test data, and compared to the actual observation. This had an average absolute error of 0.063 kWatts, and RMS error of 0.161 kWatts. Figure 5 shows this comparison for a random selection of days and solar panels.

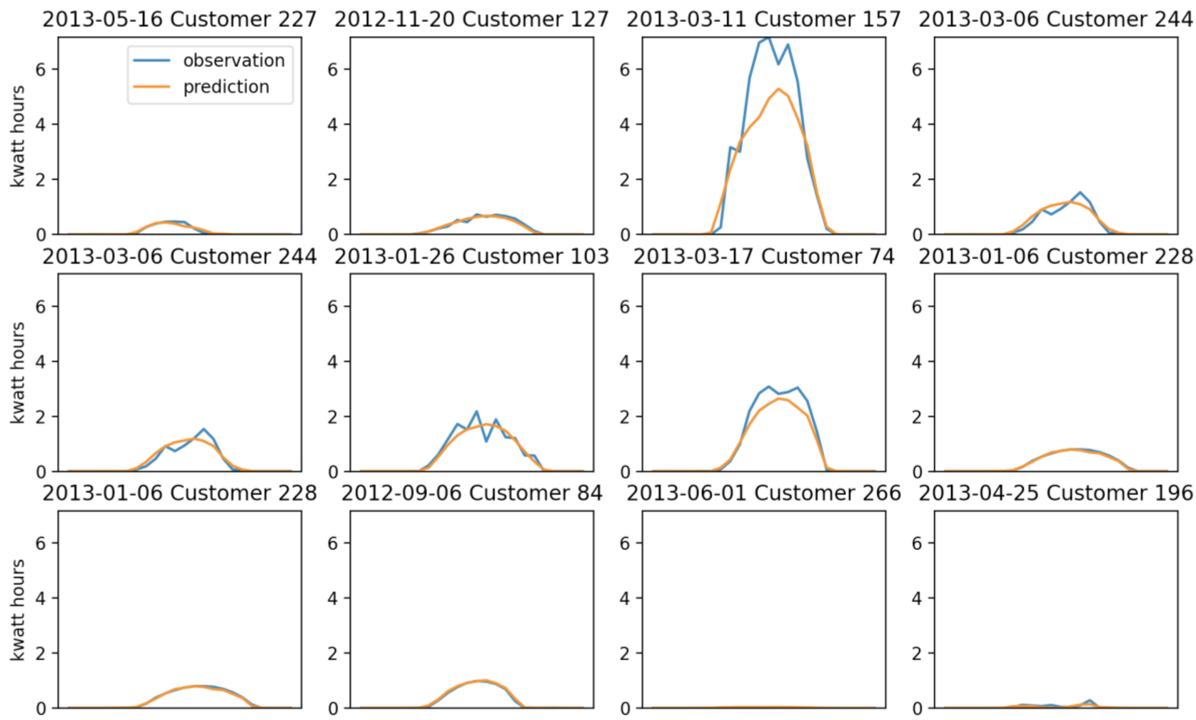

Figure 5: Observation vs prediction of testing data for model trained on all solar panels.

Since this error was very similar to that seen in the simple model for a single solar panel, we were confident our model architecture was behaving as designed. On average, the observation varies from the prediction by 0.063 kWatts. The maximum solar panel generator capacity out of all the solar panels used in this analysis was 5.9kWp, so our average absolute error is 1.07% of this. However, for solar panels with lower power generation capabilities, or for days with low sunlight, this error is more significant. In general, Figure 5 shows that the predictions match very closely with the observation for solar panels with different ranges, and for different levels of sunlight throughout the year.

The RMS error value of 0.161 kWatts is slightly higher, as this puts a heavier weight on larger deviations. A particular cause of large deviations are instances when a solar panel was observed to generate zero kWatts all day, despite the weather. It is likely that this data was incorrect. In the future, we could look to perform further preprocessing to remove suspicious looking data. This would help us train a more reliable model, as well as give a more meaningful error value. As well as improving the quality of the data, we could also improve the model's accuracy by further investigation into tuning the model's hyperparameters. This would optimise the model's performance by determining the right combination of hyperparameters for the model architecture. This could be done manually, or using an algorithmic method such as Grid Search.

Section 4: Visualisation and User Interface

4.1 Considerations of visualisation & UI

The remaining sub-functions of our service are about implementing the visualisation of these predictions through a dynamic user interface. In order to align this with the purpose of the whole system, it was important to prioritise a comprehensible interface that translates the complex data from modelling predictions into a quickly accessible format that allows for fast decisions to be made. However, these decisions also needed to be of the highest quality when relating to an extensive power grid, and hence more detailed information should be presented by the system when deemed necessary by the operator. These are some of the reasons that an interactive geospatial map was deemed the most beneficial option, however several other implementations for the visualisation were presented and considered.

The first consideration was to implement a simple, feed through output rather than a visualisation. In this, we would create an output that can simply be passed to a system control unit that is integrated into the power station facility. By doing so, autonomous decisions about where to provide supplementary power could be made. This would potentially be a more efficient holistic system, as it would remove the need for human input and interaction. However it would likely require significant research into forecasting energy demands, as well as the specific requirements of the power plants, and hence be a significant technological investment. Furthermore, it would be very specific to the needs of a single power company, and therefore not provide the modularity to implement the service Australia-wide. It was therefore determined that for the scope of our service, it was more viable to create a human centred design. In this, the forecasts from our model would be presented to the user for their interpretation.

In presenting this data, we also considered implementing a statistics-based format. This would have involved a large-scale graphing system, such as a dashboard of the different regions and subsets. The benefit of this would be to focus attention directly to the detailed time forecasting data. An interactive dashboard of the forecasts would also allow for numeric comparisons of power being produced across all solar panels, without needing to navigate to each of them. However, the downside of this solution would be the detriment to intuitive design, as it may be difficult for operators to comprehend such statistical data. Specifically, comprehending the locations of the solar panels through reading the place name or GPS coordinates is not intuitive. Without presenting a physical map, the user would need to have extensive location knowledge, or to manually search for a region of interest. As such, it was deemed more valuable to present this data in conjunction with geospatial relevance. A map style user interface allows the user to intuitively comprehend these solar locations, and thus to synthesise their locations and relationships to one another on a more complex level, ensuring the highest quality decisions are made.

Furthermore, in order to create a user interface that was intuitive and simple, yet displaying complex data, it was determined that linking the proposed solution into familiar systems would be most beneficial. This would allow users to feel comfortable with the system even on their first use, minimising the time investment required for a user to adjust to the foreign system, and instead being able to understand and engage with the software quickly. We therefore chose to implement an openstreetview style map, with interactive pins that display information on click. This is very similar to a Google Maps style user interface, which almost all users will have had past experience with, and hence are able to familiarise themselves with the system at first glance.

4.2 Execution of visualisation & UI

Our service is hosted through an AWS instance, which allows for remote computing power at a small cost. This means that users can access the service from any location, and view the same results. The webapp of our service is run through Flask, which allowed us to develop and test the page locally, and then easily rerun it on AWS. In particular, we made use of an add-in to this, called Flask-ApScheduler. This schedules a function to be run in the background of the app, and therefore we were able to update our forecast every hour. Updating our forecast both ensures that it is always predicting the subsequent 24 hours, but also that it uses the most recent weather forecasts to make the most accurate predictions.

The foundation of implementing our user interface was the use of a Leaflet JavaScript map. This provides vast flexibility with the way data can be integrated. We created this leaflet map through the use of the Folium module, allowing coordinate data to be utilised to introduce 3 distinct features; individual solar panel markers, region borders, and a choropleth heat map, all of which could then be directly added to the JavaScript map. The user can select which of these three layers they want to view, so to reduce information overload in presenting all of this information at once. The dropbox to select from these options is shown in Figure 6.

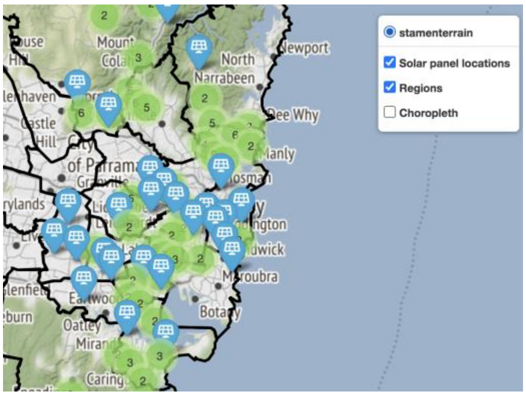

Figure 6: Dropbox to select different visualisations. Currently, solar panel pins and region borders are shown.

The first option for this is the individual solar panels. These are displayed at their coordinates, with clusters for solar panels that are too close to separate, as shown in Figure 7. Interacting with these displays information including the panel's generator capacity and its 24-hour forecast. For this, a graphing solution was integrated to retain and allow the user to access all the specific details of the forecast. These graphs are created while making predictions, and the output of which is stored in a *JSON* format. When adding coordinate markers to the map via Folium, these individual graphs are then paired with each marker, allowing for the user to open up their respective forecast on click. This allows users to investigate specific markers of interest and obtain detailed data, while minimising the amount of data initially cluttering the interface.

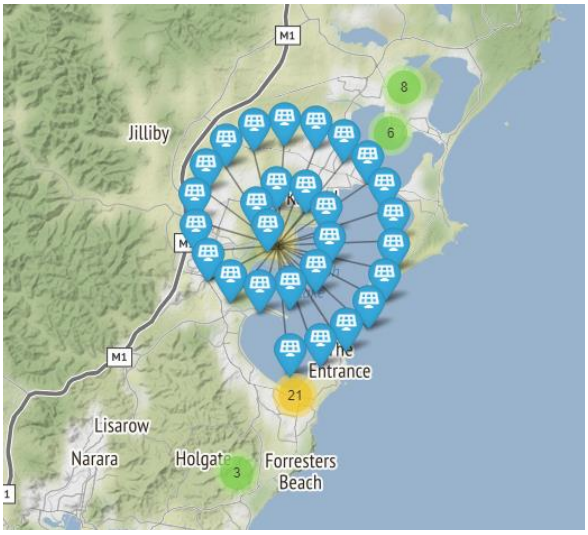

Figure 7: Solar panel pins and clusters

The second option in the user interface is the region zoning, which displays the summary characteristics of the solar panels within each region. This was a chosen design feature due to the relevance to industry, allowing supply lines or specific areas supplied by a single power station to be identified and categorized with their production levels. Specific power stations can therefore be adjusted for supply and demand changes. This was implemented through the use of polygon coordinate data. We scraped such data from government electoral voting regions. This was a simple way to obtain decent bordering, but may be limited in use to the user. This limitation is explored in more depth in section 5, which discusses our limitations and future work. However, our user interface implementation simply requires an input CSV file of the region borders, and so the regions can easily be changed simply by manually editing the file or providing a new dataset. Using the given data, we then autonomously determined which region each solar panel was contained within. Through this, it was possible to create a summation of the forecast from all solar panels within each region, and have this output displayed as a graph on click, as with the individual solar panels.

A similar method was adopted in order to achieve the choropleth heatmap functionality. Utilising the aforementioned region-specific subsets, the maximum power output of each region was found, and passed to the Choropleth attribute of the Folium map class. Through this, regions are assigned a colour on a scale that represents low to high maximum power. This allowed an efficient method for the user to quickly understand the region-specific forecasts.

4.3 Results and discussion of visualization & UI

The chosen implementation design is centred around a geospatial user interface, paired with specific graphs for solar panel modelling, able to be selected and viewed either for individual solar panels, or for specific regions. This resulted in a far extended value above that of just individual panels, as the various, isolated supplies were synthesised and used to create a high-level perspective on the production data. In particular, our heat maps, which were implemented with minimal technical effort, yield immense value to the end user. They provide understanding of the regions that are going to require power supplementation. Further investigation to the exact details of regions of concern can then be performed using the other two options, which present the forecasting data as graphs. An example of these visualisation options is shown in the screenshots in Figure 8.

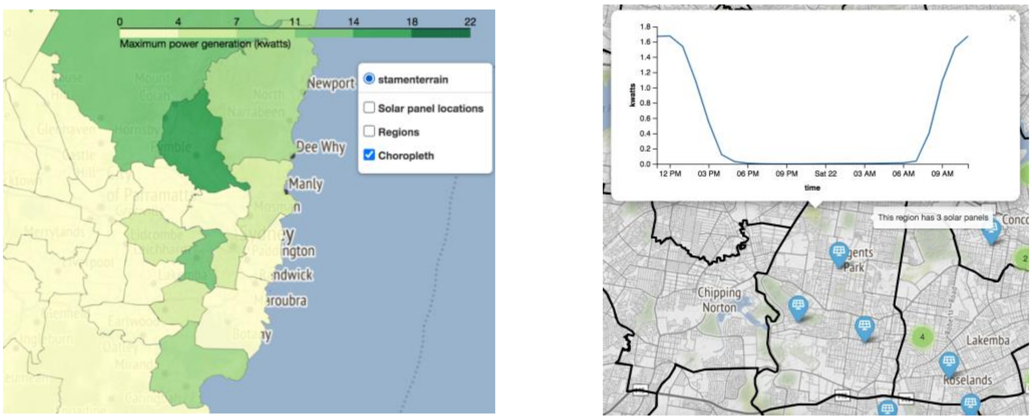

Figure 8 Left: Heat map window for regions within NSW, with the colour scale in the upper right corner. Right: Graph of overall forecasts of the selected region, overlayed on the map containing region borders and solar panel pins.

This implementation provided a comprehensive way to view a whole, dominant power supply system, whilst still retaining all the intricate details and nuances of individual solar panels that can be viewed if deemed necessary by the end user. Recalling the initial proposition of this solution, with a key objective being the presentation of solar power predictions in an immediately comprehensible format that allows the target user to make important decisions on when and where to supplement energy, it is clear that the chosen implementation aligns the solution to this.

Section 5: Limitations and Future Work

As we prioritised modularity in our design, the limitations of our service can be easily identified and addressed in the future. While we have provided a ready-to-go solution that can be currently implemented directly into industry use, future development and tailoring to the exact need of the target user will be essential moving forward. In this, the end user would be engaged in key consultation to determine the exact needs of the larger system, and how the most value could be created. Several aspects of the prototype's current limitations, and thus the potential for further improvements is detailed here. These include upscaling to all solar panels in Australia, improving the model with other relevant inputs, utilizing more relevant region zoning, and ultimately the implementation of an autonomous supplementary energy system.

The first priority of future work would be the scaling up of this system to create a service that accounts for all solar panels in Australia. Although a clear proof of concept has been demonstrated for the entire AusGrid NSW power supply network, the flexibility of this solution ties immense value to its ability to be implemented in a wide variety of geographical locations, and would be a key development in the future application and expansion of this system. We have ensured that such scaling would require minimal change, as the system is built on scalable infrastructure, for example the weather forecasting process is able to provide relevant information for any geographical location on the globe. The key data set that would need to be sourced and provided would be the historical solar performance data, which would be used as a crucial input into determining the characteristics of new regions, and producing the most accurate forecasting models.

Another key expansion opportunity would be in relation to the inputs taken by the forecasting model. As demonstrated in this report, the modelling and testing conducted for this system showed that the accuracy of the modelling fell within an acceptable range. However, there are a variety of extra inputs that could be considered to further improve the model's accuracy. For example, we have proposed the option for satellite imaging data of the solar panels' location. Other inputs that could be relevant to the system include the age and quality of the solar panels, as well as environment data that could be identified at their installation, such as a dusty outback compared to city air quality. Considering these inputs would create a more comprehensive system that is able to build its model not only on a foundation of weather forecasting, but also a wide variety of conditions that could be affecting the solar power generation within a certain region. However, while this could greatly reduce the forecasting error, there would be a cost of both time investment and a large computational model, and would only be investigated when the current accuracy is determined insufficient.

As mentioned previously, a particular limitation of our system is the preset region borders, whose data we obtained from electorate zoning. These regions were chosen due to their population moderation, which allowed us to assume relatively similar demand characteristics. However, a more suitable choice would be the energy distributor's substation regions. While we couldn’t access this data due to the proprietary nature of this information, in a real-world implementation it would be sourced directly from the target user. This would allow the user to understand the different levels of supply and demand that will occur in different substations. For example, if there are certain regions that are supplied by a single power station, knowing their specific supply needs would allow an immediately informed response on whether to turn on the plant in the next 12 hours. This would be more beneficial than our current system, which relies on synthesis between stations in order to supply the correct amount to the currently set electoral regions.

Finally, the long-term potential of this service has been alluded to throughout the report, which would be a system that can autonomously produce the required supplementary energy. This would integrate energy generation forecasts with energy demand forecasts, to identify the gaps that coal energy will need to fill. It could then autonomously decide which power plants to turn on or off. Our current service provides a step in this direction, by providing more effective and efficient tools for human decision making. However, there are many other requirements for this vision to be fully embodied, in particular, we would need comprehensive forecasts for energy that will be produced from all variable sources as opposed to just solar, requiring the industry as a whole to take a technological step forward into Industry 4.0.

Section 6: Conclusion

Overall, the objectives of the service were met to a high standard.

The forecasting model provides an accurate prediction of the solar power generated within the next 24 hours. This time period was deemed most relevant to the preparation for generating coal electricity, and thus our predictions will aid decision making in how to supply this supplementary energy. However, there is room for improvement with the accuracy of this model, with the further addition of other data.

Our visualisation of this data was achieved using a map as the user interface. This allowed the user to easily see where all the solar panels were located, and the region bordering could be easily used to determine the overall supply of energy in relevant regions. The choropleth overlay allows energy companies to see where energy is in abundance and where it is lacking in a very simple and easy to understand way. This allows the user to make immediate decisions on when and where to supply supplementary energy.

Furthermore, the service is highly modular. The system can easily be adapted to install improvements to the model, change the region zoning, or add extra solar panels. We therefore have created a solution that is scalable and adaptable to the ever-changing solar power market.

By providing this service, we hope coal power can be used as a more dynamic source of supplementary energy. In foreseeing the amount of energy required, we hope energy companies will be able to reduce wastage of energy while still ensuring outages are avoided. However, the most useful potential of our service is the opportunity to integrate it into a system that autonomously provides supplementary. This long-term goal will further reduce error and costs.

Section 7: References

[1] Australian Government. 2021, "Solar PV and Batteries." energy.gov.au. https://www.energy.gov.au/households/solar-pv-and-batteries (accessed 28 May 2021).

[2] AEMC. 2021, "The Value of Dispatchability in the NEM." AEMC https://www.aemc.gov.au/news-centre/perspectives/value-dispatchability-nem (accessed 30 May 2021).

[3] M. Diesendorf. 2018, "Is Coal Power 'Dispatchable'?" Renew Economy. https://reneweconomy.com.au/is-coal-power-dispatchable-71095/ (accessed 30 May 2021)

[4] US Environmental Protection Agency, Office of Air and Radiation. 2014, "Assessment of Startup Period at Coal Fired Electric Generating Units." https://www3.epa.gov/ttn/atw/utility/matsssfinalruletsd110414.pdf (accessed 30 May 2021)

[5] E. Gordo, N. Khalaf, T. Strangeowl., 2015, "Factors Affecting Solar Power Efficiency." New Mexico Supercomputing Challenge. https://www.supercomputingchallenge.org/14-15/finalreports/88.pdf (accessed 26 March 2021)

[6] T. Wood. 2020, "Explainer: what is the electricity transmission system, and why does it need fixing?" The Conversation https://theconversation.com/explainer-what-is-the-electricity-transmission-system-and-why-does-it-need-fixing-147903 (accessed 30 May 2021)

[7] Ausgrid, 2019, "Distribution and Transmission Annual Planning Report." Ausgrid. https://www.ausgrid.com.au/-/media/Documents/Reports-and-Research/Network-Planning/DTAPR2019/DTAPR-2019.pdf?la=en&hash=7E4EF497497565A04B44C94A4899DF06DBFF0EB9 (accessed 30 May 2021)

[8] I-Scoop "Industry 4.0: the fourth industrial revolution – guide to Industrie 4.0" https://www.i-scoop.eu/industry-4-0/ (accessed 31 May 2021)

[9] J. Jiang. 2018, "An Improved Cyber-Physical Systems Architecture for Industry 4.0 smart factories." *Advances in Mechanical Engineering.,* vol. 10, no. 6. pp. 1-15, 2018 [online]. Available: https://journals.sagepub.com/doi/pdf/10.1177/1687814018784192#:~:text=The%205C%20architecture%20consists%20of,connect%2D%20ing%20sensors%20to%20machines (accessed 30 May 2021)

[10] F. Martinez. 2020, "Time Series Forecasting with KNN in R: the tsfknn Package." Cran. https://cran.r-project.org/web/packages/tsfknn/vignettes/tsfknn.html (accessed 25 March 2021)

[11] C. Olah. 2015, "Understanding LSTM networks." colahs blog. https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed 30 May 2021)

[12] J. Brownlee. 2018, "Multi Step LSTM Time Series Forecasting Models for Power Usage." Machine Learning Mastery. https://machinelearningmastery.com/how-to-develop-lstm-models-for-multi-step-time-series-forecasting-of-household-power-consumption/?fbclid=IwAR1Qry2p6BJphKM18x5EaCQdDTpSgYyx7lrsSMpgsphL-bJMJ4KYTRTm_oc (accessed 30 May 2021)

[13] J.Brownlee. 2019, "3 Ways to Encode Categorical Variables for Deep Learning." Machine Learning Mastery. https://machinelearningmastery.com/how-to-prepare-categorical-data-for-deep-learning-in-python/ (accessed 30 May 2021)