Department of Computer Science

Submitted in part fulfilment for the degree of MEng

Predicting Short-Term Solar Photovoltaic Outputs: An Investigation of the Transformer Model's Accuracy and Adaptability

Dema Williams

28 April 2025

Supervisor: Dr Poonam Yadav

ACKNOWLEDGEMENTS

Special thanks to Dr Poonam Yadav and Dr Olufemi Olayiwola who's continued support helped tremendously throughout the project. Without both of their guidance I would not have been able to complete this dissertation of mine.

Additional thanks to Dr Ahsan Khan whose feedback was greatly appreciated and was very useful for me.

STATEMENT OF ETHICS

All data used throughout this paper, which includes the PVGIS synthetic solar PV dataset, the OpenMeteo historical dataset and the Institute for Safe Autonomy's real world solar PV dataset are all open and available to use for this project with no ethical concerns.

These datasets contain no personal or human data, and the data was used for a non-commercial research-based project with no conflict of interests.

The project did not focus on unethical practices nor did it collaborate with any 3rd parties that could bring the University into disrepute. Additionally, the project has no restrictions on the dissemination of the results discovered.

All data and code used for this project, has been provided to allow for replication and further research if desired, details of which are in the appendix.

TABLE OF CONTENTS

CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

EXECUTIVE SUMMARY

Solar energy is increasingly becoming a more common means of energy generation. However, the main disadvantage of solar energy is the fact that power output is dependent on highly changeable conditions. This makes predicting solar power outputs for the short term extremely difficult. This is important due to the fact that if short term predictions can be made accurately, solar energy will become a more reliable means of energy generation, which in turn will then make solar energy a greater means to combat climate change. In the past, research has mainly focused on using recurrent neural networks to make these forecasting predictions, but few studies have determined how effective the transformer architecture is for forecasting solar power outputs.

The main research gap that this project sets to address is determining whether transformer machine learning (ML) models are able to be effectively used on weather and solar irradiance data to make short term forecast predictions for photovoltaic (PV) outputs. This is an important aim because if solar PV yields can be predicted more accurately then solar power can become a more reliable means of electricity generation. Current research has determined that transformers are suitable for PV predictions, but have focused on solar data located within tropical or Mediterranean climates. This project determines whether transformers can be used to predict solar PV within a temperate climate and compares these to current industry standard machine learning models (CNN, LSTM, etc), which are used as baselines for all comparisons. Finally, there is very little research into transformers' ability to transfer learnt knowledge from one solar dataset to another. Therefore, the final research gap this project aims to address is to determine how effective pretrained transformers are at PV predictions when given a new solar dataset and compare the predictions generated to the same industry standard ML models. All these objectives will help conclude whether the transformer architecture is suitable to base future solar PV forecasting models upon.

The methodology was split into two key sections; the training of the models on a large dataset and then finetuning the pretrained models on a smaller different solar PV dataset. Due to the limited amount of data within the real-world dataset used in this project, it was decided to utilise a large synthetic dataset to initially train the transformer and baseline models. These models were tested and analysed to determine how successful they were at PV predictions on the synthetic test set. Then each model was given the new unseen smaller dataset from a real-world solar farm that was used to fine-tune the model. Again, the models were tested and the results analysed on the new test set.

The key findings from the first section of this project discovered that the two transformers were consistently able to outperform the baseline models in terms of accuracy. Then in the second transfer learning section it was discovered that baseline models benefitted from pretraining. However, the transformer models performed worse than if they had just been trained from scratch on the new dataset. But interestingly, despite the smaller size of the new dataset, the non-pretrained transformers were able to provide a more accurate forecast compared to any of the baseline models whether or not the baselines underwent pretraining. Despite the clear advantage in terms of accuracy, the transformers only achieved these high accuracies at a huge computational cost when compared to the baseline models.

These results indicate that transformer models are extremely capable at making accurate predictions in terms of solar PV forecasting within a temperate climate. However, the huge computational cost as well as the transformers' inability to adapt to a new dataset indicates that they are not quite the clear choice when creating an ML model to predict solar PV yields across countless solar farms. Instead, the disadvantages need to be understood and subsequently minimised if in the future a solar power plant wishes to use the transformer architecture to predict PV yields at their solar plant.

1 Introduction

In order to decarbonise power generation, the use of renewable energy sources is an effective and popular method to produce clean energy. One of the most common and cheapest renewable energy sources (RES) is solar power. Solar energy comprised just over 30% of all renewable energy generated worldwide in 2022, with the other 70% comprising mostly of wind or hydropower [1]. Now, despite RES's success, power sources that utilise fossil fuels are still the primary means of power generation worldwide. As a result, solar energy only accounted for 3.6% of all power generated in 2022 around the globe [1]. However, as shown in the graph, it's predicted that by 2050 solar will become the second largest energy source (behind only wind), contributing to 25% of the earth's total energy needs [23].

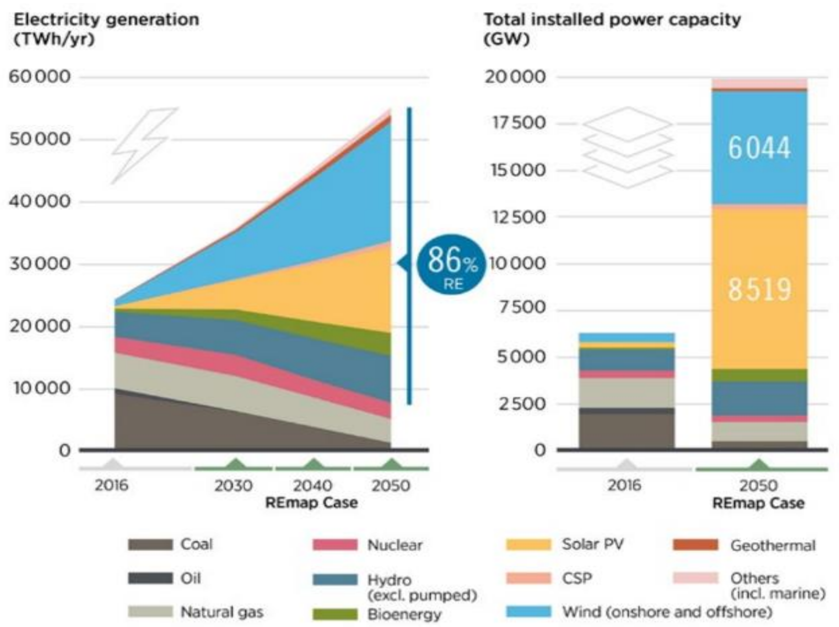

Figure 1-1*: Future estimations for the power generation broken down by energy type [23], CSP = Concentrated Solar Power.*

Despite the clear trend where solar PV is heading, techniques to accurately predict solar photovoltaic (PV) yields are in their infancy. A lot of research specialises in long-term forecasting, e.g., over a year. Long-term forecasts are very useful for planning where solar farms should be built. However, in order for the energy grid to be able to depend on solar energy, there needs to be a reliable method to predict the short-term energy production. This is because accurate predictions are crucial when the energy grid is trying to balance power generation throughout the country at peak times [2], as they are used decisively in electricity markets and power system operations [24].

Accurate predictions will allow for power plant operators to decide whether to ramp up or down production in order to keep the grid stable.

Historically, the research into short-term PV predictions has mostly focused on using recurrent neural networks such as Long Short-Term Memory or Convolutional Neural Network models. There are few experimental studies looking into how effective transformer models are at accomplishing accurate predictions for solar PV generation, and almost all focus on solar data gathered in non-temperate climates. However, almost all studies utilising transformers for PV predictions from these arid and tropical climates have concluded that they can consistently outperform the previous industry standard Recurrent Neural Network (RNN) models. Therefore, further research should also determine whether transformers can also outperform RNN models in a temperate climate. Furthermore, almost all research into transformers within the solar PV context only train and test the model on one dataset gathered from one solar plant and don't analyse how effective transformers are at transferring learnt features across different solar PV datasets that have been gathered from separate PV plants. In order to become an effective model to forecast PV, the ML model will need to make accurate predictions on distinct solar datasets from different solar plants. This will allow one highly generalised ML model to be used on a nationwide scale to help effectively balance the electrical grid.

Further research into these key aspects will allow for the decisive conclusion on whether the transformer architecture can be suitably applied to solar PV datasets in order to make accurate predictions across several solar plants within a temperate climate. This in term can then help conclude whether the transformer architecture is suitable for a generalised ML model for solar PV forecasting on a nationwide scale.

Research Questions

The main purpose of this project is testing how accurate transformer models are at short term solar PV predictions, when compared against industry standard machine learning (ML) models. Therefore, the first research question is:

- Can transformer models predict short term PV forecasts, when trained on a large synthetic dataset from a temperate climate, to a higher standard when compared to current best practices?

This then leads us to the next thought process which is whether transformers trained on synthetic data are still effective when given real world data. There has been very little research into determining transformers' ability to transfer knowledge from different datasets within a solar PV context. Therefore, the second question is:

- How effective are transformers at PV predictions when given a small real-world dataset after being trained on synthetic data, when compared to the current industry standard models?

These research questions will allow us to determine how feasible transformer models are at providing a machine learning model that can effectively predict PV outputs, which can then be used by the National Grid to correctly balance the electricity demand. This in turn will prove to make solar power a more reliable method of electricity generation, which will allow for the expansion of solar as a means of carbon neutral electricity generation. This then helps make solar power a more popular method of stable electrical power generation to then be used to help combat the effects of climate change.

Project Outline

The report contains several key sections detailing the work done throughout the project. The first section is the background and literature review. This is where the current literature is analysed to provide the reader with background knowledge to where research has focused on in the past. This section also concludes by identifying the research gap which then helps justify the key aims of this project.

The next section is the methodology and implementation. These are typically separate sections in other reports, however due to their similarities it was decided to merge them. This section outlines the key decisions made and the justifications for all choices throughout the project.

Then it's the results and evaluation section. Here, we present the results of our experiments by explaining how each of the models performed in separate scenarios. Finally, we end the report with the conclusion which is where we summarise the key findings, determine how effective transformers can be utilised for our goals and then explain possible areas for further work.

2 Background and Literature Review

In this section, we outline the current literature that uses machine learning for forecasting purposes within the field of solar energy. We first look into recurrent neural networks (RNNs), which historically is where most of the research into solar PV forecasting was specialised in.

But since the transformer architecture was debuted in 2017 [12], more research into utilising transformers for solar PV predictions has been published. Therefore, the second section of the literature review looks to understand what discoveries have already been made relating to transformers and solar energy.

The finally, this section concludes by identifying the research gap that currently exists and the outlines how this project aims to address that research gap.

2.1 Long Short-Term Memory (LSTM)

One promising development for short-term predictions is using a Long Short-Term Memory (LSTM) neural network (NN). An LSTM is similar to a regular NN however the hidden layers contain memory cells. These memory cells regulate what the LSTM “remembers” or “forgets” [3]. This allows the LSTM to learn much faster than other similar machine-learning techniques [3].

One paper [14] used an LSTM on publicly available historical data from a solar system located in Cyprus. The model predicted forecasts with up to 15-minute time intervals, a very good time horizon that can be used effectively for power grid management. The authors analysed the results and concluded that the model provided good predictions without overfitting the data set. In [15] the authors used an LSTM on weather and historical data to predict solar irradiance over a large area (Atlanta) in difficult weather conditions, i.e. where cloud cover is erratic. The authors concluded that the LSTM was able to work well with the inputted data to provide accurate results. Another paper [16] just used historical power data, but trained 5 separate LSTM models with slightly different architectures. They then compared the results of the LSTMs with statistical methods and other basic neural networks. The authors concluded that the LSTMs consistently outperformed the other methods. However, not including weather data limits the validity of the models if the desire is to be used in managing power grids. The authors in [17] used an LSTM on data from the weather, the PV systems (panel temperature) and historical data regarding power outputs from the PV system. The short forecast horizon of 15 minutes is very impressive. The authors then tested their model and concluded that it could accurately predict the PV output of the solar PV system over the course of a day’s energy production. One paper of interest used an LSTM and synthetic weather forecasts to predict photovoltaic (PV) production [4]. In this paper, the authors go on to state that they used weather data such as the solar irradiance, temperature, wind speed, and relative humidity along with power generation from previous hours. The authors then compared their LSTM to other machine learning models and analysed the results. It was determined that the LSTM network is consistently more accurate than other machine learning models such as Recurrent Neural Networks (RNN) and

General Regression Neural Networks (GNN) [4]. This indicated that LSTMs are able to provide promising predictions.

Another interesting paper proposed 2 deep learning models, but the results showed that only the LSTM showed promising results [5]. Here, the authors proposed to use an LSTM for precise hourly predictions. The authors used meteorological data such as temperature, humidity, and radiation for the inputs in order to predict PV power outputs. The authors evaluated this proposed LSTM against several other models and consistently found that the predictions made were among the most accurate. Additionally, when the LSTM was used to predict the hourly changes to PV power outputs over a day, which the author states is a considerably more difficult task. The proposed LSTM showed impressively low errors for these days.

2.2 Convolutional Neural Network (CNN)

Another machine learning model is the Convolutional Neural Network (CNN). CNNs are used commonly in image recognition models. A CNN consists of convolutional layers, pooling layers, then ending in fully connected layers [6]. At the convolutional layer, the CNN consists of filters that are called kernels [6]. The kernels scan the entire input, and the dot product between the input and the filters is produced as the output. Then the pooling layers will then shrink the output maps keeping only the key features [6]. In a CNN there can be multiple convolutional and pooling layers until the final fully connected layers are reached. Here, the output of all the previous layers is fed into a standard neural network to produce the final output of the CNN [6]. CNNs are most useful when the data has a grid-like topology, for example time-series data has one dimensional topology, while an image has 2-dimensional topology [7].

One paper that utilised a CNN to predict PV power outputs, used the CNN to analyse the cloud coverage of images taken of the sky to try and predict the power outputs in the next 15 minutes [8]. The paper used a solar farm located at Stanford University for the PV data. Additionally, there was a camera nearby that took images of the sky, and this was used for the data regarding cloud coverage [8]. Overall, the paper focuses on determining the best CNN architecture and how the authors discovered said architecture. However, the analysed results of the predicted PV output is the most relevant aspect of the paper. Here, the authors determined that the proposed best solution determined through extensive hyperparameter tuning performed better than the current CNN system the researchers had previously worked on [8]. However, a major limitation of this paper was that the primary focus wasn't to determine how accurate the CNN can produce short-term forecasts from a wide range of weather conditions, but was instead describing the techniques used to determine the best CNN architecture. As such, determining how feasible this model would be if it were used to aid power generation forecasts is omitted from the paper.

Another paper utilised multiple CNN models to in order to determine how feasible they were at predicting PV outputs is [18]. The data consists of power data from just one solar panel located at the Wroclow University of Science and Technology. Additionally, the authors used weather data such as wind speed and ambient temperature measured from a nearby sensor. The forecasting was done in 15-minute intervals, a very good time horizon for managing power grids. The authors used a sliding window approach to the data [18]. This is where time series data is able to be converted into vectors. This is useful as it reduces the computational cost of training the model. The authors tested these CNN models and found they all consistently outperform traditional statistical methods for solar PV prediction.

2.3 CNN-LSTM hybrid

In recent years, the most exciting developments in short-term PV forecasts have quite often come from the utilisation of CNN-LSTM hybrid models. Here, researchers use CNN and LSTM architectures in combination to create one model. CNN-LSTM hybrids are powerful as CNNs specialise in feature extraction while LSTMs focus on the context of data over time. Therefore, using CNN-LSTMs in time series data leads to exciting results.

One paper that used a CNN-LSTM hybrid [19] used a CNN to classify weather type, and then an LSTM to predict PV generation. They used 3 years of power data collected from a power system in Korea, as well as weather data collected from the Korea Meteorological Administration. The authors concluded that their proposed model was able to adequately predict the actual power outputs. However, this paper did not compare the proposed model with other machine-learning techniques. Another paper [20] uses data from a solar system in Abu Dhabi, and managed to highly accurately predict solar irradiation. Although this is not directly predicting solar power, it is still useful to see how a CNN-LSTM is used within the context of solar farms. Another paper [21] used publicly available data from a PV system in Australia. They also used weather data such as wind speed, temperature and humidity among others. The authors concluded that the CNN-LSTM hybrid model performs better than the CNN or LSTM models that it was tested against.

Another interesting paper that utilised a CNN-LSTM hybrid used the CNN architecture to extract features from a one-dimensional time series data input. The authors state that the CNN is ideal for this aspect because it is able to extract unseen data features that can be useful to determine PV outputs. Then the output of this was fed into the LSTM in order to discover connections and try and create predictions on the data [7]. Then before the final step a fully connected layer is utilised in a way for the CNN-LSTM to interpret the final output. The authors used power output historical records, weather data (temperature, wind-speed), and also the facility power consumption of the solar system. The authors then compared their proposed CNN-LSTM to five other machine-learning models [7]. They found that the CNN-LSTM achieved the lowest mean absolute error. The authors state that the proposed model predicts global characteristics of energy production better than the traditional models and also does a decent job at predicting local characteristics [7]. However, as promising the results for long-term predictions are, the real challenge is short-term predictions which the paper implies that the CNN-LSTM is not as effective in the area. Overall, the authors' proposed method provides the most accurate results compared to the comparison models. However, they don't provide any forecasts for anything less than 3 days, which means that this specific model will not have many practical uses in helping balance grid loads on an hour-to-hour basis.

Another paper that utilised the CNN-LSTM hybrid approach aimed to combine the two architectures in order to integrate the advantages of both into one model [9]. The paper uses real-world historical data from a PV plant in Belgium. The authors remove any data recorded at night, as PV output is negligible [9]. The resolution of the data is 15 minutes [9]. This is a positive and unusual as other papers typically have a minimum resolution of an hour. However, the model does not take into account any weather data, which if included typically increases the accuracy of the predictions. Instead, the model just uses the previous 60 PV output data points to predict the next one [9]. The authors then compared the results of this model and found that it was consistently more accurate than other methods. Overall, this paper is useful to prove the effectiveness of a CNN-hybrid approach. However, the decision to not include weather data led to the overall accuracy being less than necessary if the CNN-LSTM is to be used to help manage power grids.

2.4 Transformers

The Transformers' basic architecture is that they are comprised of an encoder and decoder structure [12]. The input sequence is mapped to a sequence of continuous representations. The decoder then generates an output sequence, and at each element generated, the transformer uses the previously generated sequences as input for the next element [12]. Another key aspect of the transformer is called attention. This is a mechanism that allows the model to focus on the key information in the data.

Transformers have several different versions, each focusing on different problems at hand. Most commonly transformers have been used in natural language processing [40]. However, there are additional use cases such as computer vision [41] and text to image generation [42]. However, for solar yield forecasting we are interested in transformers that specialise in time series data.

One such paper [13] utilised a timeseries transformer to try and predict the PV outputs from using weather data and historical data from the largest solar farm in the Southern Hemisphere, located in Australia. The transformer performed better than all the other models that it was tested against. Another paper that used transformers [22] used data from a PV system located in China. Here the authors concluded that amongst the models compared, the transformer model achieved the best predictions for both the short- and medium-term time horizons. In [43], the authors collected data from several power plants in Italy and focused on the effect that different sources of weather data had on the models. Nevertheless, they were able to showcase that transformers can accurately predict PV generation. In [44] the authors developed their own transformer model and utilised solar irradiance data from two solar farms located in South Korea. They discovered that their model was the most accurate but all transformers used were more accurate than the baseline RNN models. In [45], a transformer was trained on solar and weather data to then make short-term predictions in PV output for the farms. When compared to the baseline models the transformer was able to outperform all other models. In paper [46], the authors used data from several different solar plants (likely located in China but they don't specify). The paper utilised a transformer-based architecture that was trained on a specific solar farm's dataset and then transferred the learnt features to attempt predictions on another farm's data. The authors concluded that the transformer was able to outperform all baseline models. The research seen in [47] used a transformer model to make medium and long-term predictions for solar PV generation. They utilised both PV and weather data gathered from the solar power plant and discovered that the transformer model achieved the highest accuracy among all models. The authors highlighted that the transformer had excellent accuracy for long-term (90+ day) predictions. The paper [48] utilised a transformer to predict yields with a time horizon of 1-7 days. Data was collected from 18 regions in India. The authors concluded that their transformer was able to forecast more accurately than other machine learning models they tested. The authors who published [49] utilised a transformer to predict solar radiation with a forecast window of up to 24 hours. They discovered that their proposed transformer model was able to achieve a higher accuracy compared to standard LSTM regression models. The paper [50] used real world solar PV data from farms located in Australia to build their own transformer architecture in order to predict the next day's power production on an hourly basis. They discovered that their model was able to outperform the state-of-the-art baseline models when provided with the same dataset.

2.5 Research Gap from Current Literature

***Table 2-1**: Comparisons of Transformer papers mentioned in Literature Review*

Trained model on synthetic data?	Model uses real world data?	Data is from a temperate climate?	Compared transformer to other best practices?	Analysed model's ability to transfer knowledge to other unseen solar PV datasets?
No transformer papers in literature review performed this	All papers in literature review	No papers in literature review	All papers in literature review, except [22]	Only [46]

Shown in Table 2-1, research into transformers' ability in predicting solar PV yields in a temperate climate is extremely limited, therefore this project's main focus is to determine whether transformers can be utilised for short-term power generation forecasts for solar farms within a temperate climate. The transformers will be compared against current best practice models (CNN, LSTM and CNN-LSTM) in order to determine how effective transformers are, at predicting short term PV forecasts in a temperate climate. This is common practice amongst the papers referenced, and this will help contextualise how effective transformers are compared against current industry standards.

Again, shown in Table 2-1, there is limited research on training transformers on synthetic data to then be fine-tuned on a smaller real world data set. The ability to transfer knowledge to another dataset is a crucial property for an ML model within solar PV predictions. This will allow models to be first trained on a generic dataset, then fine-tuned on a smaller real-world dataset from the solar plant. This is important as it would allow for the greater adoption of utilising models to predict PV yields, as the barrier for entry is lowered with a smaller need for real world data (which can take many years to properly gather for each plant). This can lead to solar energy becoming a more reliable power source. Therefore, the model will then will also be fine-tuned on a real-world solar PV dataset based in a temperate climate in order to determine the adaptability of the transformer models when compared to the baselines. This will help determine whether transformers are suitable at generalising not just to unseen data within the same solar plant, but also at other solar plants that would have their own likely smaller datasets.

3 Methodology and Implementation

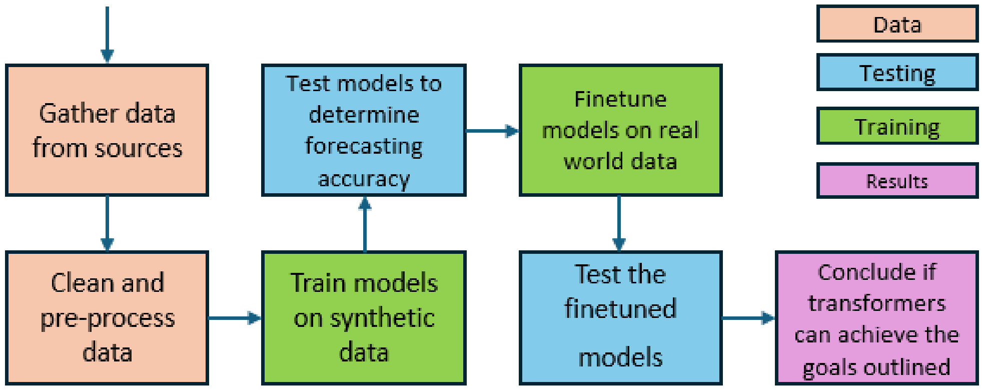

***Figure 3-1**: The methodology overview*

This section outlines the methodology followed and the specific implementations used for this experiment. These two sections were combined due to the similarities between them in order to produce a concise report that does not included any needless repetition. Each subsection outlines the steps taken and justifies the reasons for any decisions made.

The two primary goals of the experiment can be concisely described as; training ML models on a large synthetic dataset and finetuning these models on a smaller real-world dataset. As a result, the methodology followed was structured in such a way to effectively achieve these goals and answer the research questions set out.

Figure 3-1 outlines the basic stages of the methodology that this project followed. Each subsection below provides more details on each stage.

3.1 Data Sources

This subsection outlines the origin of the data for each of the two primary goals of the experiment. The data is split into two sections: a mainly synthetic dataset that was used for training the models, and then a smaller real-world dataset that was used for the transfer learning stage of the project.

For the training data, a reason to use mainly synthetic data is that with more data available the models have a greater likelihood to generalise patterns rather than overfit the training data. Since the real-world data was determined to not have a great enough quantity; solar PV and weather data was gathered elsewhere.

The PV data source in question was the Photovoltaic Geographical Information System (PVGIS). This is an online tool developed by the European Commission that provides free and open access estimates for solar PV generation from any location [39]. Research has indicated that the benefit of using more data outweighs the downside of using data that's synthetic [4], therefore using this dataset is a massive benefit for our project. However, it is important to note that due to the synthetic nature of the data, there is less "noise" present which could lead to inaccuracies when the model is used on real world noisy data. The granularity of the data is on hourly intervals, which is a useful time granularity because it is small enough to allow for short term predictions, while also giving us enough data for our models to learn from. This will provide the power generation data and some minor weather information too. Each time step has 9 variables that the model will use; which includes some basic weather data, but the most important ones are the solar irradiance and the power output (the second of which is the models target variable and what it aims to predict).

However, in order to improve the accuracy of the models, additional weather data was utilised too. This will be provided by Open Meteo, which is an open-source API for providing highly localised historical weather information [25]. This is a good addition to the dataset because it provides more weather information than just the standard PVGIS data, which is useful for the deep learning models as they will have a greater chance to learn patterns, from the more in-depth information on the weather dataset. Another benefit of this dataset is that the data is gathered at a resolution of 1km on a regional scale, which in turn means we can use weather data close to the site's location rather than the nearest weather station, which could be much further away. This will help provide a more accurate picture of the data. The granularity of which can be customised, but it was determined to use hourly intervals to easily match the PV dataset. However, a big downside to using this data is that the hourly weather data only goes back until late March 2021. This means that despite the PV dataset going back almost a decade, we are only able to use data from March 2021 to December 2023, where the PV dataset ends. Despite the fact that more data would be a nice luxury, the almost 3 years of data will be more than enough to train on.

These two data sources were combined into one dataset. In total, there were just over 24,000 time steps of data with 16 individual variables each. The target variable, which the models will aim to predict, was the power output of the farm at each timestep. The data set was split 70% training, 15% validation, and 15% testing.

For the transfer learning stage, the data was also gathered from two different data sources; the power data was gathered from a real-world data set that included data on voltage, electrical current and power output of the solar panels. The granularity of the data was every few minutes starting in late October of 2023 and ending in mid-February 2025. However, the data did not include values for nights. This is not a major issue as the power output would be zero anyway, but slight data pre-processing was necessary, which is explained in more detail later.

The weather data was gathered from sensors located at the solar farm, therefore giving us a very accurate picture of the weather conditions that would impact solar power generation. Similarly, the granularity of which is every few minutes starting in late September 2024 and ending in late January 2025. Both real world datasets were gathered by the Institute of Safe Autonomy at the University of York and are open to use for this project [53].

The data was then merged to form one dataset with just over 2500 time steps (about 10% of the previous synthetic training data, therefore it was justified to first use the larger synthetic dataset for training, then this one for finetuning), with each time step containing 55 variables that were used by the model to perform its predictions. Again, the data set was split 70% training, 15% validation, and 15% testing for the finetuning of the models.

From an ethical stand point, PVGIS [39] and OpenMeteo [25] data sources are open and available for use under the Creative Commons Attribution 4.0 International (CC BY 4.0) licence. While the real-world data is open to use and available from the Institute of Safe Autonomy at the University of York [53].

3.2 Data pre-processing

This subsection outlines methods and specific implementation for cleaning the data ready for the models to take as inputs for training and finetuning.

The first step in the data pre-processing pipeline would be to ensure that there are no NAN/missing values. This would cause errors in the model which would prohibit learning. For the initial training data, the NAN values were counted and there were roughly 10 records missing out of the 25,000 data points. Therefore, the simplest approach is to just remove these time steps from the data, because it was determined that this is the simplest method and would have a very minimal impact.

However, the transfer learning dataset caused more issues. Because the initial models would be trained on hourly data, and the real-world data contains data every minute, the best point of approach was determined to average the data for each hour. This works well for the data in the day. However, in the real-world data, any data at night was omitted. This meant if the data is averaged over the hour, it will introduce a lot of NAN values corresponding to the night-time values. The best course of action was to set all NAN values to 0. This was decided as it would impact the data loaders the least, as they will require a full day's worth of data to work at their optimum. This is because the data loaders assume a full continuous set of timesteps so it was determined the best course of action was not to introduce large time gaps between datapoints. However, this would mean that the model would more easily be able to predict when the farm would output 0 power. This was determined to not be a major concern as the main point of interest in solar PV forecasting is to determine the peak power during the day, and not the power outputs at night. Additionally, this was a similar approach taken by the work in [9], therefore we can also justify this approach as being grounded in previous research.

Once the missing values are dealt with the next step in the pipeline is to create data structures that would represent the 5-time steps worth of data (4 previous hours, and the current hour the model is aiming to predict). This is simply achieved by looping through the data, and at each time step the previous 4 hours' worth of data is added to an 2D array with the current time step's data. From studying the literature most models utilise a larger historical time window by feeding the model possibly up to a day's worth of data. This is different to our proposed time window of 5 hours. However, these models were also aiming to predict PV generation for many time steps in the future. Since our experiment only predicts the next time step, the larger time window was deemed unnecessary. This is because the larger data input will greatly increase the computational cost of the model, but will not provide further benefit because the model is only predicting one time step. By making the historical time window 24 hours their models are able to determine when the mornings and evenings are, which in turn would then help the model make accurate predictions for the next 24 hours. Since our models only predicts one time step, data from several hours prior to the target hour is irreverent and has very little impact to the current PV yield. Therefore, the decision was made to utilise just 4 previous hours because this would allow for a reduced computational complexity while also providing enough data for the model to make a prediction one time step into the future. Then the data structure would also contain data on the current time step, such as the weather data, which would help it in its power predictions. However, one problem we have to be aware of is data leakage. Data leakage is where a model has access to information that it would not have access to in a real-world application [31]. In our case, since we are asking the model to predict PV power output, we need to take extra care to not include the power data for the time step we are trying to predict. This is achieved by ensuring that the power data for the predicting time step was set to 0. This way the model would not be able to "cheat" by looking at the target variable from its inputs. This hidden PV output becomes the target value for the model to try and predict, and is used within the loss function to train the model.

The next step in the pipeline is to scale the data. In some instances, there are variables that would contain values in excess of 20,000. Scaling the data is an important step for ML. This is because the ML algorithms used in training are sensitive to the scale of variables [32]. Therefore, scaling would help the model to converge more effectively. The built-in scikit-learn function MinMaxScaler was used. This sets each variable within the range 0 to 1. Where the smallest value is 0 and the maximum value is 1 [33].

3.3 Model Architecture and Hyperparameters

This section outlines how each model's architecture was determined and the justification of each model's parameters. All models were implemented in Python using the PyTorch library. This was a good decision for this project as PyTorch provides a wide number of built-in methods with lots of documentation to help build our models. Additionally, PyTorch is widely adopted in research therefore this project follows convention by further utilising the library.

CNN

The CNN is based upon the weather time series CNN 3 in [29]. This model is a good choice to base our model upon because the authors determined that their model was effective within a time series dataset for power generation. However, there are a few changes. Firstly, rather than inputting a full day of data to then try and predict the next 24 hours of data, our model will only take 5 timesteps for reasons outlined in the previous subsection. Similarly, rather than having 24 outputs for the prediction, the model used in our project will have just one output to predict the next hour's PV output. Seen in Figure 7-1 (in Appendix A) is the architecture of the CNN. A kernel size and stride of 3 in convolutional layers and 2 in max pool layers was used. Padding was set to 0 on first convolutional layer and 1 on second. The activation function was RELU and dropout was set to 0.5. These hyperparameters were chosen for our solar PV model because the authors in [29] determined through several experiments with different hyperparameters that these values created a model with the greatest accuracy for timeseries power generation. Therefore, in order to assure our model would also be the industry standard for CNNs the choice was to not stray too far from the author's discoveries, except for when necessary as explained above.

LSTM

The LSTM model is based upon the work in [5]. This LSTM is a good choice to base our model upon because the work in [5] determined this architecture was able to make accurate predictions within solar PV. However, there are a few key differences. Firstly, the data used for their model had just 6 weather input variables, therefore the model used for our comparisons will be changed to accept the 16 input conditions from the weather and PV dataset. Additionally, the author's model had an input time window of 14 hours. For our experiment, only 5 timesteps will be included for the LSTM to train on. This will help reduce our model's complexity as well as make training quicker. The other major change was the output layer changing from a size of 14, to just one. This way the model would predict the next hour we wanted rather than the next 14 hours. After each LSTM layer, the activation function tanh was used and dropout was set to 0.2. This is because the author of [5] determined these parameters to produce a model that accurately predicted solar PV on their dataset. Therefore, in order to try and ensure our model would also have accurate predictions these parameters were kept the same. Figure 7-2 (in Appendix A) shows the LSTM architecture for our experiment.

CNN-LSTM

The CNN-LSTM model was based upon the work available in [34] [35], because this model displayed accurate short-term forecasts within a wind power context. Despite the fact that this model was for time series in wind power prediction, it can still be effectively applied to solar time series because the datasets have common similarities (both are power generation that are dependent on weather conditions). The main difference is changing the input size to 5, to signify how many time steps are included in the data. Additionally, the paper didn't specify how many hidden layers or the size of the LSTM layers, so from reading the papers in the literature review a best guess of 3 hidden layers and 64 neurons per LSTM layer was decided as these were common parameters seen across models, as well as being the same parameters used in the LSTM model used in our experiment. Seen in Figure 7-3 (in Appendix A) is the architecture for this model. For the CNN-LSTM a kernel size of 2 on first convolutional layer, 1 on the rest all with a stride of 1 was used. The padding was 0 on first convolutional layer and 1 on the rest. The activation function used was SELU. These hyperparameters were chosen for our model because the work outlined in [34] determined these parameters to have the best convergence rate discovered.

Simplified Transformer Encoder (STE) Layer

The first transformer was based upon the work in [36], which in turn based their work on simplifying the encoder layer in the original transformer paper [12]. The key differences between this transformer and the original paper is that there is no input embedding, residual connections or normalisation within the architecture. Additionally, there is a fairly simple fully connected linear layer both at the input and output ends of the model. Dropout was set to 0.1 because the work within [36] determined this value resulted in a model that achieved accurate predictions. Figure 4-2 shows this model's architecture.

It's important to add that transformer's inherently have no method to determine positional sequence within an input. Therefore, transformers utilise positional encoding which allows the transformer to learn which values correspond to which time step within an input [38].

The transformer encoder layer, is the PyTorch implementation [51] of the standard encoder layer from the original transformer paper [12]. This is a standardised function used for transformers. It consists of feed forward network and self-attention of 4 heads.

Attention within a transformer is a technique that allows the model to figure out the importance of certain sections within an input sequence [12]. This is necessary as transformers have no method to determine position from a sequence, as explained above. Multi-head attention (used in this transformer), has several attention heads (4 in our case), to perform the attention process several times, with each head independently learning different representations of the input, at different positions [12]. 4 heads were chosen because this was the number utilised in [36] therefore in order to ensure our model was also suitable for timeseries data it was determined to keep the same number of heads for our datasets.

Full Transformer Encoder (FTE) Layer

The second transformer is based upon the work in [37], which in turn fully implemented the encoder layer from the paper [12]. The main difference was that the input size was changed in order to fit our own data. Dropout was set to 0.1, because seen in [37], it was determined that this was the best dropout value to use from the author's experiments. Seen in Figure 4-1 is the model's architecture.

Before the positional encoding stage, this model also uses input embedding. Input embedding is more useful for natural language processing, as this is a princess where the input words are converted to vectors of the same dimension as the model [12]. However, for time series the data is already numerical therefore the primary purpose of input embedding within this model is to ensure that the input will match the dimensional size of the transformer encoder layer. However, in our model we utilise a convolutional layer within this input embedding which will capture any local patterns for each timestep. Therefore, the secondary purpose is feature extraction to help the transformer layers, later on. This was done because seen in [37] the author concluded that this technique was the PyTorch implementation of input embedding for timeseries data that closely followed the input embedding from the paper [12].

This model then passes the input to the positional encoding stage (explained in the simplified transformer model), before heading to the multi-head attention transformer encoder layer. This model only utilises one encoder layer, unlike the transformer explained above.

This transformer also utilises the residual connection and normalisation before and after the FC layers. Residual connection acts as a shortcut for the data, providing a pathway for the data to skip some layers and allowing the data to reach latter parts of the network [52]. This has been proven to be effective in larger machine learning models to help the model converge quicker. Layer normalisation essentially centres and scales the data around the mean, which also helps the model converge quicker.

3.4 Training

Training took place on an Intel i7-1255U CPU, with 16GB of RAM available. At each epoch the model was evaluated against the validation set. This allowed for the model to be tested against an unseen dataset so the performance metrics could be analysed per epoch. Additionally, this allowed for the implementation of early stopping into the training of the models. This is where if there is no improvement in the validation loss after 10 epochs the training ends. This helps protect against overfitting. These values were used for all models during training:

***Table 3-1**: Training hyperparameters*

Batch size	Optimiser	Learning rate	Weight Decay	Loss function
5	Adam	0.0001	0.001	Mean Squared Error (MSE)

A batch size of 5 was used because the size of the data as well as the complexity of some of the models would have an impact on the limited hardware available for training. Therefore, utilising a low batch rate was determined to help reduce the computational cost of training.

From the papers referenced in the literature review, several papers such as [45], [46], and [50] utilised the Adam optimiser to great effect. The value of the learning rate and weight decay was also determined by analysing the available literature and these specific values were commonly used to successfully train both RNN and transformer models. The MSE loss function was used because this is extremely common practice from the papers we looked at.

3.5 Transfer Learning

This is the second aim of the project, where each model is analysed in their effectiveness to make predictions on the smaller real-world dataset. Here, the project took the pretrained models and fine-tuned them against the real-world data. This is where the models are trained on a smaller number of epochs on the data (20 epochs in this case), in order to gain a basic understanding of the specifics of the new data. Learnt from the training section, 20 epochs was enough to provide a semi-accurate model while also not providing enough training time to fully converge on the optimum solution. Therefore, since we are analysing each model's ability to adapt to a new dataset not the model's full capability for the new dataset it was determined to use 20 epochs for the finetuning section.

The previous data pipeline was followed, this time outputting the training, validation and test datasets for the new data. Since the new data contained a different number of variables per time step (52 instead of 16), the input dimensions of the models had to change in order to accept the new input size. Then the pre-trained model's weights were used for each layer excluding the input layer for the new model. As a result, the input layers are the only difference from the pretrained model and the new model to be used on the real-world data.

Since the new model has a slightly different architecture, two tests were performed. One fine-tuned the new model with all the pretrained weights initialised. Then the other fine-tuned the new model with no weights initialised. The performances of both were analysed by testing the model on the test set, which included specific day scenarios relating to high and low cloud cover days. This analysis was then used to determine how effective each model type is for transfer learning on a new dataset.

3.6 Testing

In addition to the MSE loss of the test set, 2 more metrics were determined to be used for the comparisons of the models. These are the models' Root Mean Square Error (RMSE), and the Mean Absolute Percentage Error (MAPE).

The RMSE is essentially a measure of the model's standard deviation [27]. A lower RMSE means the error in the accuracy is smaller. RMSE is calculated as:

$$RMSE = \sqrt{\left(\frac{\Sigma(y_i - \hat{y}_i)^2}{N - P}\right)}$$

Where y_i and $\hat{y}_i$ is the actual and predicted value for the i^{th} value, P is the number of parameters estimated and N is the number of observations. RMSE is useful as it is a single metric that states how accurate a model is, making for simple comparisons. Additionally, it is sensitive to outliers as these will result in a much greater score when the RMSE is calculated [27] and this is important in the context of solar farms as accurate forecasts on easily predictable days are not the main aim, rather it's the cloudy more complicated days that are the true problem for machine learning models. Therefore, the RMSE is a good metric to use as it will punish any model that performs poorly in regard to predicting forecasts on the cloudy days.

The other metric is the Mean Absolute Percentage Error (MAPE). This is a percentage equivalent to the mean absolute error, therefore it highlights the average magnitude of the error in a model [28]. The MAPE is defined as:

$$MAPE = \frac{\Sigma\left(\frac{|A - F|}{A}\right) * 100}{N}$$

Where N is the number of fitted points, A is the actual value, and F is the forecast value [28]. A lower MAPE indicates a more accurate model. MAPE is useful in our context as it is very useful in analysing regression models that map inputs x to an output y [28]. Additionally, it is also useful for ongoing analysis on the model [28], which is useful for when the model will be tested against the real-world data. For these reasons the MAPE was utilised for our project.

To compare these metrics with other models in a standardised way the skill score will be utilised. The skill score is not a quantitative metric that analyses the model itself, but is instead a technique to compare model accuracies with a baseline model, and then the result of this determines if the model performs better or worse than the baseline [26]. In that paper, the authors defined the skill score as:

$$Skill\ score = \frac{A_f - A_r}{A_p - A_r}$$

Where A_f is accuracy of the forecast for the system of interest, A_r is the baseline reference accuracy and A_p is the accuracy if the forecast was perfect [26]. For our analysis of experiments, we will be using the same skill score formula. If the skill score is greater than 0, the the model outperforms the baseline. If it is 0, then the model is the same as the baseline. Finally, if the score is less than 0, the model is worse than the baseline.

However, despite the usefulness of these metrics, in order to fully understand the bigger picture further analysis was necessary. This was performed by analysing the model's effectiveness for scenarios relating to different types of days. The first scenario was the basic completely sunny, very low-level cloud cover percentage. This would test the model's performance on easy to predict days, and the expected result is that the model will be able to achieve a high accuracy on these types of days.

However, in order to provide an effective forecast, the model will need to perform well on days where there is a high cloud cover percentage. This scenario had two further specifications. The first being days that consistently have 100% throughout the day. Then the second will test the model against days where cloud cover varies significantly. These two specifications will test how effective the model is with regard to scenarios where conditions are hard to make predictions in or consistently changing hour by hour, which is useful to determine how effective a model is at forecasting as outlined in our goals

4 Results and Evaluation

In this section the results from the two experiments are presented. We first present and analyse the effectiveness of the transformers in regards to training on the large synthetic dataset. All the metrics are then compared to determine how well the transformer managed to forecast PV generation in relation to the baseline models. This then allows us to determine how well transformer models can forecast short term PV outputs.

Then this section presents the results from the second aim of assessing the ability of transformers to transfer knowledge across different datasets within a solar PV context. Again, the metrics from this experiment is presented which allows for comparisons between the baseline models and the transformers. Which in turn allows us to conclude the effectives of transformers within the field of solar PV transfer learning.

4.1 Summary of Testing the Models trained on the Synthetic dataset

For the first aim of the experiment, the models were tested against the test dataset. The MSE, MAPE, and RMSE of each were recorded in order for quantifiable comparisons on the baseline models to be made.

Baseline Models' Results

The 3 baseline models were trained and then subsequently tested.
The CNN was trained until early stopping was triggered at 28 epochs, for a total training time of 1750 seconds (29mins, 10 secs). On the test dataset, the CNN recorded an MSE loss of 0.001485, an RMSE of 0.0271 and an MAPE of 1.78%. Plotted in Figure 7-4 (in Appendix A) is the (MSE) Loss, RMSE and MAPE per epoch for the CNN's training.

As evident in the loss and RMSE graphs the model trained well and converge effectively, as both training and validation metrics decrease fairly similarly. However, the MAPE graph consistently suggests that the model was actually overfitting the training dataset, which is evident from the fact that the validation MAPE does not converge as well compared to the training MAPE.

The LSTM was trained for a total of 1808 seconds (30 minutes, 8 seconds) when early stopping was triggered at 40 epochs and achieved a test MSE of 0.000352, an RMSE of 0.0139 and an MAPE of 1.00%. As shown in Figure 7-5 (in Appendix A) the model converges extremely well and there is no evidence of overfitting from the model:

The CNN-LSTM was trained for 29 epochs (again early stopping was triggered) over 3310 seconds (55 minutes, 10 seconds). Against the test dataset the model achieved an MSE of 0.000244, and RMSE of 0.0147 and an MAPE of 1.25%. As evident from the test and validation metrics in Figure 7-6 (in Appendix A), the model converges effectively, before beginning to slightly overfit which is where training was halted.

Seen in Figure 7-7 (in Appendix A) are the baseline models' performances for the Sunny days. The blue bars represent the cloud cover percentage, which is useful to see how cloudy a certain hour was. The blue line is the actual PV output. The dashed lines represent the respective baseline models' predictions.

As evident from the graph the CNN performed the worse, unable to accurately predict the peak power and performing poorly at night time predictions. The LSTM and CNN-LSTM however performed much better, with the CNN-LSTM being the most accurate during the day, but performed worse at nights which is reflected in the metrics. This is evidence of why further analysis was needed as the metrics suggest the LSTM model is the most accurate, but purely on daytime predictions the graph suggests the CNN-LSTM is actually the more accurate model. These two models were able to almost predict the correct peak power and they both provided accurate night predictions.

Shown in Figure 7-9 (in Appendix A) are the baseline models on the mixed cloudy days. As evident from the graph the CNN again performed the worse unable to make reliable predictions. The LSTM and CNN-LSTM both performed well (the CNN-LSTM again slightly better during days but worse during nights).

Finally, seen in Figure 7-8 (in Appendix A) is the graph for the full cloud cover day. The CNN again performed the worse providing a forecast with a poor accuracy. The LSTM and CNN-LSTM again performed much better. Because of the scale change it is clear to see the CNN-LSTM's error offset for night-time predictions negatively impacts the test metrics. The LSTM is more accurately apple to predict nights which is reflected in the overall test metrics for this dataset. However, the CNN-LSTM still has the most accurate day predictions.

Transformers' Results

The STE was trained for 33 epochs over 25143 seconds (6 hours, 59 minutes, 3 seconds). Against the test data set it recorded an MSE of 0.000173, an RMSE of 0.0110, and an MAPE of 0.87%. As seen in Figure 4-4, the model converged extremely effectively then spent the majority of the training time with slight improvements. The fact that the validation metric is well below the test metrics implies that the model has not overfit the training data and actually generalises extremely well.

The FTE was trained for 33 epochs over a total of 50427 seconds (14 hours, 27 seconds). Tested against the test dataset the model achieved an MSE of 0.000277, and RMSE of 0.0123, and an MAPE of 0.84%. The performance mapped against epochs seen in Figure 4-5 shows that the model converged quickly, before appearing to overfit the data, which was then reversed at around epoch 15 where the model continued to make minimal gains until training terminated due to early stopping.

Shown in Figure 4-13, Figure 4-14, and Figure 4-12 are the transformers' predictions for each of the day scenarios. The STE model is able to effectively predict the next hour's PV output to an excellent standard. No matter what the weather conditions are, the model is able to make accurate predictions. However, similar to the CNN-LSTM, there is an error offset for night-time predictions. The FTE model is able to provide more accurate predictions than the previous transformer when it comes to the test, sunny and mixed clouds testing data. This is also evident in the graphs provided showcasing that the transformer is able to almost predict an almost exact match compared to the actual PV outputs for the timestep. However, when it comes to the days with full cloud cover the predictions are not as accurate compared to the simple transformer, and are arguably the worse predictions on this subset from all models except the CNN. Overall, though, both transformers would make for effective forecasting models.

Comparisons of Models Trained on Synthetic Data

Presented below are all the metrics for the full test set (black), the sunny test subset (orange), the mixed cloudy days subset (green) and the fully cloudy days subset (red). The best metric for each is ***highlighted.***

Table 4-1: Test metrics for the models trained on the synthetic data

	CNN	LSTM	CNN LSTM	Simplified Transformer Encoder	Full Encoder Transformer
MSE Loss	0.001485 0.003371 0.003089 0.001207	0.000352 0.000327 0.000581 ***0.000089***	0.000244 0.000276 0.000361 0.000167	***0.000173*** 0.000265 ***0.000242*** 0.000091	0.000277 ***0.000245*** 0.000255 0.000266
RMSE	0.0271 0.0491 0.0423 0.0244	0.0139 0.0148 0.0164 ***0.0076***	0.0147 0.0160 0.0170 0.0129	***0.0110*** 0.0154 0.0142 0.0091	0.0123 ***0.0125*** ***0.0127*** 0.0117
MAPE	1.78% 2.90% 2.58% 1.87%	1.00% 0.90% 0.98% ***0.60%***	1.25% 1.21% 1.24% 1.23%	0.87% 1.18% 1.04% 0.82%	***0.84%*** ***0.73%*** ***0.80%*** 0.91%

Therefore, from just quickly looking at the table it is clear to see the forecasting abilities from the transformers are the best. In all but one data subset, one of the transformers provided the most accurate forecast. The only instance where a different model beat both transformers was where the LSTM had a better accuracy for the sunny day subset.

The table below shows the calculated skill score of each metric for each test case. If a skill score is positive this means that the new metric is better than the old baseline reference. The metrics used as baselines and comparisons are the overall metrics for the entire test dataset, not the sunny/cloudy subsets.

Table 4-2: Skill Scores for the transformer models. STE = Simplified Transformer Encoder, Full Transformer Encoder

Model	CNN			LSTM			CNN LSTM		
Metric	MSE	RMSE	MAPE	MSE	RMSE	MAPE	MSE	RMSE	MAPE
STE	0.8835	0.5941	0.5112	0.5085	0.2086	0.1300	0.2910	0.2517	0.3040
FTE	0.8135	0.5461	0.5281	0.2131	0.1151	0.1600	-0.1352	0.1633	0.3280

As evident from the skill score calculations both transformers are better than almost all the baseline models, strongly indicating that the transformers are able to make more accurate and more reliable forecasts for the next hour's PV generation from the solar farm, compared to the baseline models.

However, it must be addressed that although the transformers did have the more accurate forecasts, the training time for both transformers were extremely long in comparison to the other baseline models. Seen in Figure 4-6 are the training times and seen in Figure 4-7 are the inference times for all the models. The transformers are many times slower at both training and inference. Therefore, even though the transformer models are able to achieve the best accuracies, they do so at a massive computational cost compared to the baseline models.

4.2 Summary of the Transfer Learning Results

For the second aim of the project, the pre-trained models were fine-tuned on a small real-world dataset. Then the model's predictions were tested and analysed against certain datasets. These results were then used for the skill score calculations to easily determine the most accurate models.

Transfer Learning on the Baseline Models

CNN: Shown in Figure 7-10 (in Appendix A) you can see that the pre-trained CNN model was able to converge fairly effectively on the new dataset. Compare this to Figure 7-11 (in Appendix A) which is the same CNN model but with all weights randomly initialised, it is clear then the pre-trained model is able to effectively predict outputs for this dataset.

LSTM: Seen plotted in Figure 7-12 (in Appendix A) is the loss, RMSE and MAPE of the pre-trained LSTM over 20 fine tuning epochs on the real data. Comparing this to the LSTM of the same architecture that did not undertake pre-training seen in Figure 7-13 (in Appendix A) it is clear to say that the pretrained model has greatly benefited from learning the previous dataset as it was able to converge more effectively on the new data.

CNN-LSTM: As seen in Figure 7-14 and Figure 7-15 (in Appendix A) the pre-trained and second non-pretrained graphs, both models converge at similar rates, both reaching their optimum within a few epochs. However, the pretrained model's metrics are more impressive and the model achieved slightly higher accuracies compared to the non-pretrained model.

Shown in Figure 7-16 and Figure 7-17 (in Appendix A) are baseline models' sunny day and cloudy day data predictions for their respective subsets. All models performed better on the sunny day predictions, while predictions were less accurate on the cloudy days.

On the test dataset of the real-world data the CNN achieved an MSE of 0.002099, and RMSE of 0.0237 and an MAPE of 1.26%. The CNN achieved adequate results on the new real-world data when attempting to predict peak power, but underestimated the total output significantly. Similarly, to the training data, the accuracy of the CNN is not enough to be effectively used in a real-world solar farm.

On the test dataset, this pre-trained LSTM achieved an MSE of 0.000324, an RMSE of 0.0110, and an MAPE of 0.76%. From the graphs the pre-trained LSTM has adapted to the new dataset extremely well. Peak power predictions are mostly accurate, with a few exceptions for harder conditions. However, this model also does not manage to accurately predict night-time outputs.

On the entire test dataset, the pre-trained CNN-LSTM achieved an MSE of 0.000354, an RMSE of 0.0149 and an MAPE of 1.22%. As evident in the day graphs, the CNN-LSTM is able to make fairly accurate predictions for sunny days, only just underestimating the peak power again. For cloudy day conditions, the daytime predictions are mostly accurate, however, the night-time predictions are completely inaccurate. However, as discussed previously, daytime predictions are of greater importance to forecast correctly.

Transfer Learning on the Transformers

As seen from the pretrained STE model's graph that shows the testing and validation MAPE at each epoch (Figure 4-3) and then the non-pretrained STE model's graph at each epoch (Figure 4-10) the models both converge at similar rates, however, the pre-trained model's validation metrics oscillate less suggesting it learned more generalised features from the training data that it was able to use on the new dataset. From the graphs presented in the subsequent figures, it's evident that there was little difference between the pretrained FTE model (shown in Figure 4-9) and the non-pretrained FTE model (shown in Figure 4-8). They both converged at similar rates and also finished training at around the same metrics for both training and validation.

On the test dataset the STE model achieved an MSE of 0.000256, an RMSE of 0.0140, and an MAPE of 1.08%, while the pretrained FTE model when tested against the test set achieved an MSE of 0.000754, an RMSE of 0.0173, and an MAPE of 1.29%. As seen in Figure 4-11 and Figure 4-15, the graphs suggest the STE model is able to predict well for both sunny and cloudy days. However, the STE model is completely inaccurate on night-time predictions, which is especially obvious on the cloudy day predictions where the model has predicted negative power outputs for those hours, which isn't possible on a real solar farm. The graphs also show the pretrained FTE model performs well on the sunny days, predicting the morning, evenings and nights to a near perfect accuracy. However, on the cloudy days the FTE model does struggle presenting some of the worst accuracies across all models.

Comparisons of Models Fine-tuned on Real World Dataset

Presented below are all the metrics for the full test set (black), the sunny test subset (orange) and the cloudy days subset (red). The best metric for each is ***highlighted***.

Table 4-3: The test metrics for the pretrained and non-pretrained models on the real-world data. STE = Simplified Transformer Encoder, Full Transformer Encoder

		MSE			RMSE			MAPE		
CNN	Pre trained	0.00234	0.00319	0.00199	0.0399	0.0463	0.038	3.03%	3.20%	3.03%
	Non Pre trained	0.01357	0.04228	0.00164	0.0555	0.1236	0.0217	3.45%	7.24%	1.68%
LSTM	Pre trained	0.00032	0.00043	0.00009	0.0110	0.0146	0.0073	0.76%	0.92%	0.62%
	Non Pre trained	0.00143	0.00234	0.00027	0.0240	0.0372	0.0133	1.76%	2.58%	1.13%
CNN LSTM	Pre trained	0.00035	*0.00034*	0.0002	0.0149	0.0156	0.0136	1.22%	1.18%	1.22%
	Non Pre trained	0.00049	0.00065	0.00057	0.0184	0.0202	0.0207	1.53%	1.54%	1.78%
STE	Pre-trained	0.00026	0.00036	0.00018	0.0140	0.0156	0.0125	1.08%	1.08%	1.03%
	Non-Pre trained	0.00023	0.00055	0.0001	0.0119	0.0175	0.0095	0.94%	1.30%	0.79%
FTE	Pre-trained	0.00075	0.00049	0.00102	0.0173	*0.0128*	0.0235	1.29%	*0.84%*	1.80%
	Non-Pre Trained	***0.00025***	0.00037	***0.00007***	***0.0099***	0.0143	***0.0064***	***0.65%***	0.91%	***0.45%***

From pre-training the model, you would expect that the model will achieve a greater accuracy on the new data compared to if you just trained from scratch. This is true for the CNN, LSTM and CNN-LSTM. But interestingly enough, both transformers did not benefit from pre-training and actually had a worse accuracy (in all but one metric). Furthermore, it was discovered that training one of the transformers (FTE) from scratch on just the smaller real-world data produced a model that was more accurate than any of the baseline models, with or without pre-training taking place. This result was not expected and might be due to the fact that the transformer's architecture is inherently more complex than the baseline models, which might result in the model needing more time to be fine-tuned to start producing more accurate forecasts when compared against a transformer model that has not been pretrained.

Presented below are the skill scores, for just the metrics on the entire test set (sunny/cloudy day metrics were excluded as they're a subset). The baseline model metrics are the pretrained ones as they had the highest accuracies. On the skill score comparisons, it is clear how much of a negative impact the pretraining had on the transformers. Both non-pretrained transformers achieve higher accuracies than almost all baseline models, while if the transformers were pre-trained they performed worse against the baselines.

***Table 4-4**: Skill Scores for the pretrained and non-pretrained transformer models. STE = Simplified Transformer Encoder, Full Transformer Encoder*

Model	CNN			LSTM			CNN LSTM		
Metric	MSE	RMSE	MAPE	MSE	RMSE	MAPE	MSE	RMSE	MAPE
Pretrained STE	0.8889	0.6491	0.6436	0.1875	-0.2727	-0.4211	0.2571	0.0604	0.1148
Non-Pretrained STE	0.9017	0.7018	0.6898	0.2813	-0.0818	-0.2368	0.3429	0.2013	0.2295
Pretrained FTE	0.6795	0.5664	0.5743	-1.3438	-0.5727	-0.6974	-1.1429	-0.1611	-0.0574
Non-Pretrained FTE	0.8932	0.7519	0.7855	0.2188	0.1000	0.1447	0.2857	0.3356	0.4672

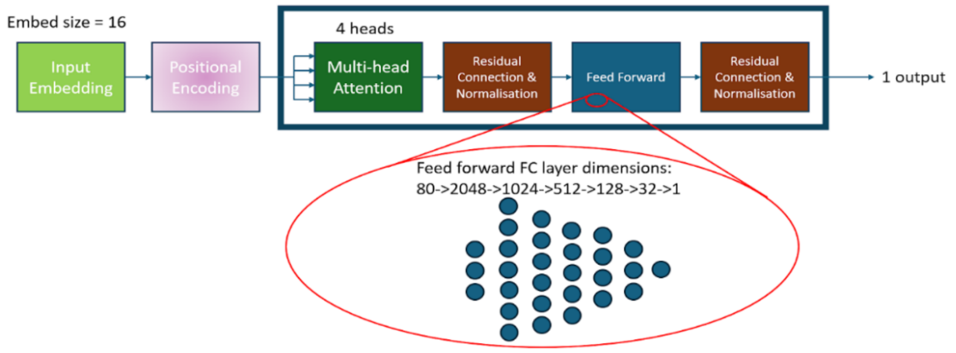

***Figure 4-1**: The Full Transformer Encoder architecture used in this experiment*

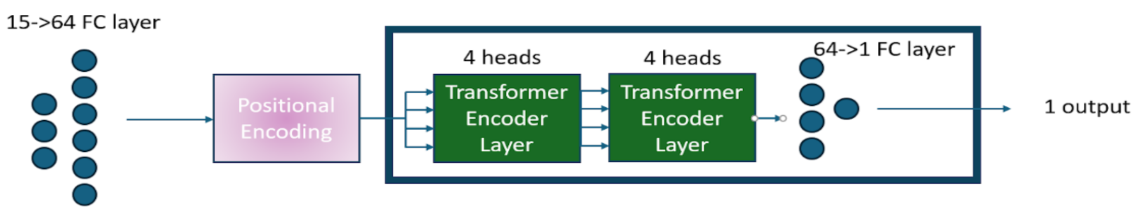

***Figure 4-2**: The Simplified Transformer Encoder architecture used in the experiment*

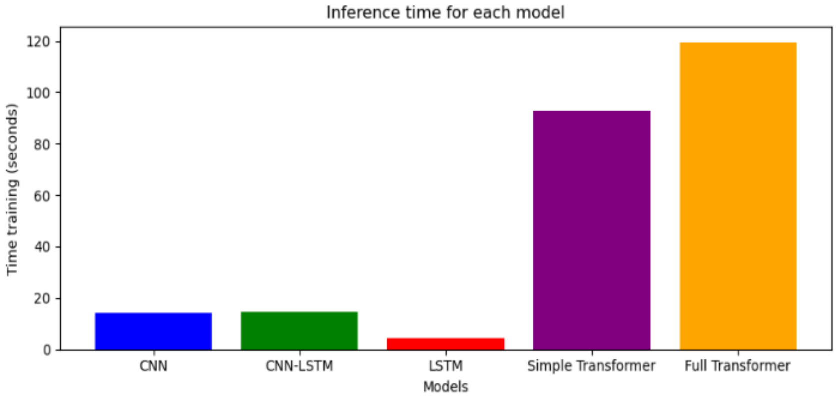

Figure 4-7: *The time for each model to make its predictions on the test dataset during evaluation*

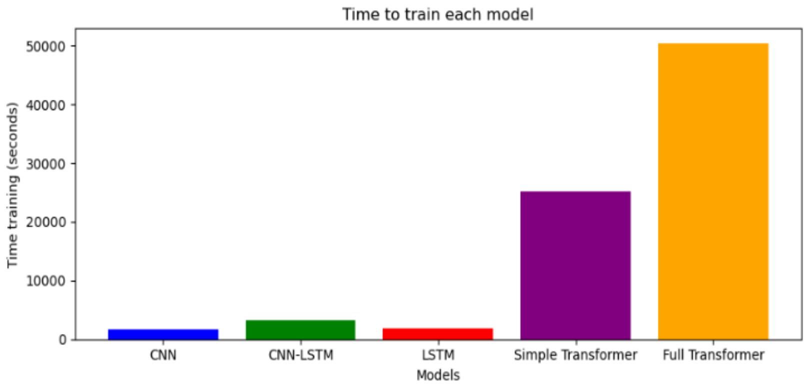

Figure 4-6: *The time in seconds to train each model*

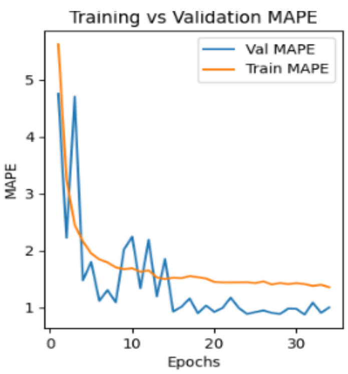

Figure 4-5: *FTE's MAPE per epoch during training*

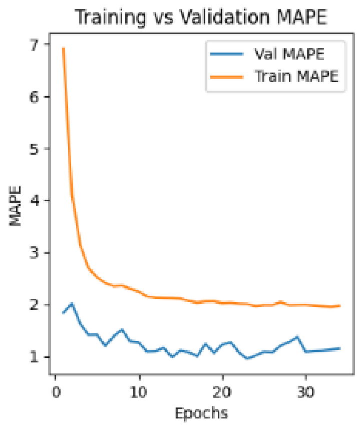

Figure 4-4: *STE's MAPE during training*

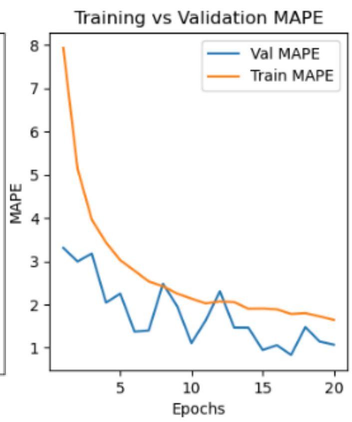

Figure 4-3: *Pretrained STE's MAPE during finetuning*

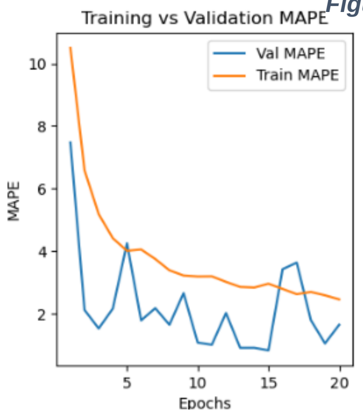

Figure 4-10: *Non pretrained STE's MAPE during finetuning*

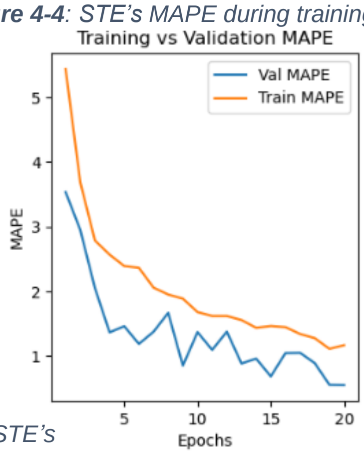

Figure 4-9: *Pretrained FTE's MAPE during finetuning*

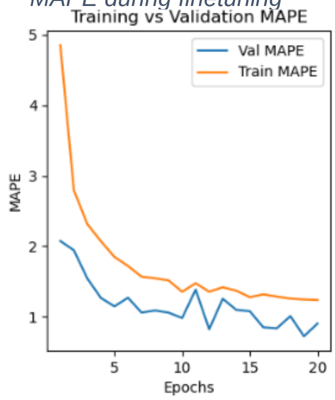

Figure 4-8: *Non pretrained FTE's MAPE during finetuning*

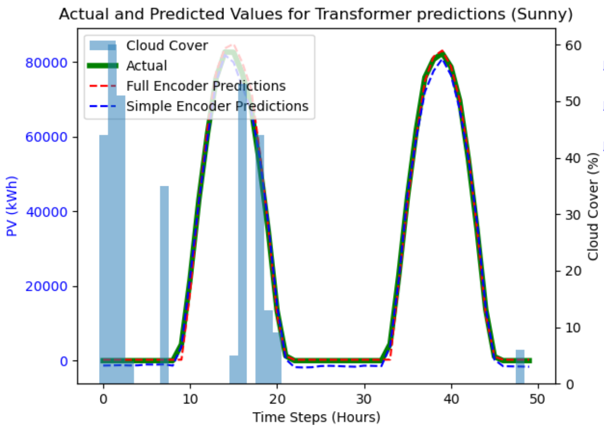

Figure 4-14: *Transformer predictions on Sunny days*

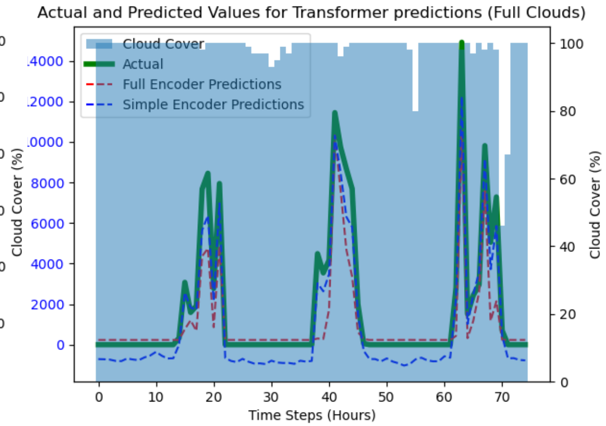

Figure 4-12*: Transformer predictions on Full cloud*

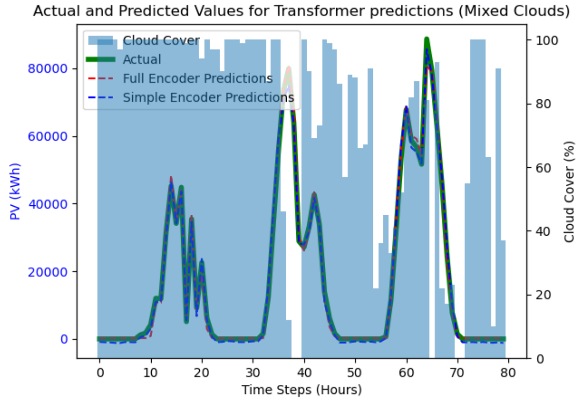

Figure 4-13*: Transformer predictions on Mixed cloud*

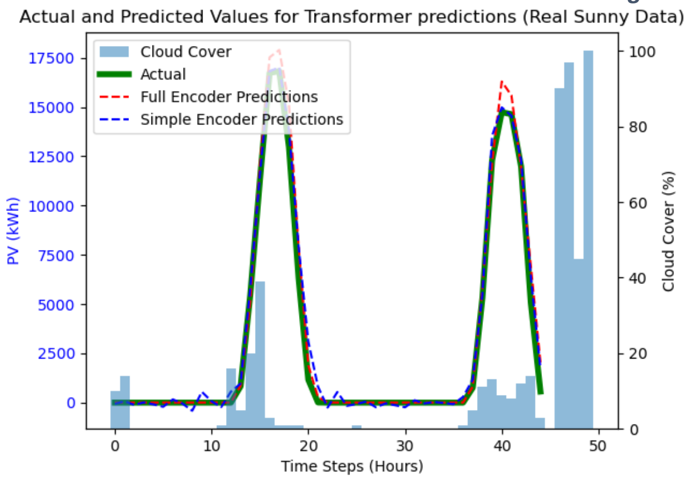

Figure 4-15*: Transformers on real data for sunny days*

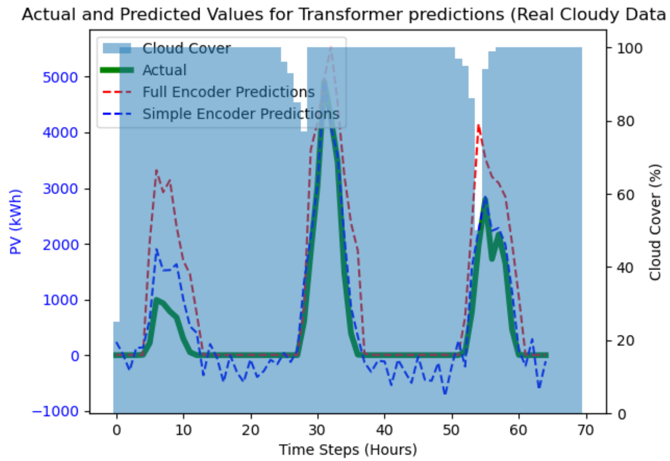

Figure 4-11*: Transformers on real data for cloudy days*

5 Conclusion

From the results, it's clear that transformers are able to produce the most accurate forecasts compared to current best practices. Both transformers were able to consistently outscore the baseline models in almost all metrics. Both transformers produced highly accurate forecasts in a range of conditions, ranging from the easy and predictable sunny days to the constantly changing conditions of cloudy days. However, little work was spent on hyperparameter finetuning. To eliminate the possibility that the hyperparameters chosen negatively impacted the baseline models, further work could look into adapting the hyperparameters to see whether the accuracies of all models can be further improved upon.

From the transfer of knowledge by pretraining the models, we learnt that the baseline models are able to produce more accurate results on new real-world data through slight finetuning. However, the transformer models were not able to repeat this and in fact were negatively impacted by undertaking pretraining on a similar dataset. Despite this, the non-pretrained transformer was still able to produce the most accurate forecast, beating almost all metrics from all the pretrained baselines. As a result, further work should be done with finetuning transformers to determine whether transformers require additional finetuning time or if in fact are not able to be adapted effectively on new datasets.

Despite all the positives for the transformers, it must be noted that they only achieved the most accurate results after undergoing training for extremely long times, much longer than any of the baselines. Therefore, in an ideal world with vast data, exceptional hardware and unlimited time, transformer models are the clear choice to utilise for short-term solar PV forecasts. But for our world, the ability to quickly train and easily adapt models on different data sets is a crucial necessity for creating a machine learning model that can make accurate predictions for all solar farms. This ideal model would then have the greatest impact in helping to decarbonise the electricity grid. As such, stating whether transformer models are clearly better is a little more nuanced, as the efficiency and transferability of the baseline models still beat the transformers. Overall, however, the accuracy of the transformers is still extremely impressive and as such it can be stated that transformers are definitely able to help forecast solar PV outputs to the necessary accuracy needed to help in the fight against climate change. But this project stops short at concluding whether transformer models can decisively be used for a generalised ML model for solar PV predictions at the nationwide scale necessary to help load balancing the electrical grid.

6 Bibliography

[1] Hamed H. Pourasl, Reza Vatankhah Barenji, Vahid M. Khojastehnezhad, Nov 2023 “Solar energy status in the world: A comprehensive review”, Volume 10, Pages 3474-3493, https://www.sciencedirect.com/science/article/pii/S2352484723014579

[2] Akhter, M. N., Mekhilef, S., Mokhlis, H., & Shah, N. M., Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques. *IET Renewable Power Generation*, *13*(7), pages 1009-1023. https://doi.org/10.1049/iet-rpg.2018.5649

[3] S. Hochreiter, J. Shmidhuber, Long Short Term Memory, 1997, https://www.bioinf.jku.at/publications/older/2604.pdf

[4] M. S. Hossain and H. Mahmood, "Short-Term Photovoltaic Power Forecasting Using an LSTM Neural Network and Synthetic Weather Forecast,” vol. 8, pages 172524-172533, Sept 2020 https://ieeexplore.ieee.org/document/9200614

[5] Lee, Donghun, and Kwanho Kim, "Recurrent Neural Network-Based Hourly Prediction of Photovoltaic Power Output Using Meteorological Information" *Energies* 12, no. 2: 215, Jan 2019 https://www.mdpi.com/1996-1073/12/2/215

[6] Alzubaidi, L., Zhang, J., Humaidi, A.J. *et al.* Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data* **8**, 53 Jan 2021, https://link.springer.com/article/10.1186/s40537-021-00444-8

[7] Ali Agga, Ahmed Abbou, Moussa Labbadi, Yassine El Houm, Imane Hammou Ou Ali, CNN-LSTM: An efficient hybrid deep learning architecture for predicting short-term photovoltaic power production, Electric Power Systems Research, Volume 208, July 2022, https://www.sciencedirect.com/science/article/pii/S0378779622001389

[8] Vignesh Venugopal, Yuchi Sun, Adam R. Brandt; Short-term solar PV forecasting using computer vision: The search for optimal CNN architectures for incorporating sky images and PV generation history. *J. Renewable Sustainable Energy* 1 November 2019; 11 (6): 066102, https://pubs.aip.org/aip/jrse/article/11/6/066102/284756/Short-term-solar-PV-forecasting-using-computer

[9] Mingliang Bai, Xinyu Zhao, Zhenhua Long, Jinfu Liu2, Daren Yu, Short-term probabilistic photovoltaic power forecast based on deep convolutional long short-term memory network and kernel density estimation, July 2021
https://arxiv.org/abs/2107.01343

[10] Smola, A.J., Schölkopf, B. A tutorial on support vector regression. *Statistics and Computing* **14**, 199–222, Aug 2004,
https://link.springer.com/article/10.1023/B:STCO.0000035301.49549.88

[11] H. S. Jang, K. Y. Bae, H. -S. Park and D. K. Sung, "Solar Power Prediction Based on Satellite Images and Support Vector Machine," in *IEEE Transactions on Sustainable Energy*, vol. 7, no. 3, pp. 1255-1263, July 2016,
https://ieeexplore.ieee.org/abstract/document/7437475

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, June 2017, Attention Is All You Need
https://arxiv.org/pdf/1706.03762

[13] Xuetao Jiang, Meiyu Jiang, and Qingguo Zhou, Aug 2023, Day-Ahead PV Power Forecasting Based on MSTL-TFT
https://arxiv.org/abs/2301.05911

[14] M Konstantinou, S Peratikou , AG Charalambides, Solar Photovoltaic Forecasting of Power Output Using LSTM Networks. *Atmosphere*. Jan 2021; 12(1):124
https://www.mdpi.com/2073-4433/12/1/124

[15] Y. Yu, J. Cao and J. Zhu, "An LSTM Short-Term Solar Irradiance Forecasting Under Complicated Weather Conditions," in *IEEE Access*, vol. 7, pp. 145651-145666, 2019,
https://ieeexplore.ieee.org/abstract/document/8864021

[16] Abdel-Nasser, M., Mahmoud, K. Accurate photovoltaic power forecasting models using deep LSTM-RNN. *Neural Comput & Applic* **31**, 2727–2740, July 2019,
https://link.springer.com/article/10.1007/s00521-017-3225-z

[17] V. De, T. T. Teo, W. L. Woo and T. Logenthiran, "Photovoltaic Power Forecasting using LSTM on Limited Dataset," *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, Singapore, pp. 710-715, Sept 2018
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8467934

[18] V Suresh, P Janik, J Rezmer, Z Leonowicz, Forecasting Solar PV Output Using Convolutional Neural Networks with a Sliding Window Algorithm. *Energies*. Feb 2020; 13(3):723.
https://www.mdpi.com/1996-1073/13/3/723

[19] S-C Lim, J-H Huh, S-H Hong, C-Y Park, J-C Kim, Solar Power Forecasting Using CNN-LSTM Hybrid Model. *Energies*. Oct 2022; 15(21):8233
https://www.mdpi.com/1996-1073/15/21/8233

[20] N Elizabeth Michael, M Mishra, S Hasan, A Al-Durra. Short-Term Solar Power Predicting Model Based on Multi-Step CNN Stacked LSTM Technique. *Energies*. March 2022
https://www.mdpi.com/1996-1073/15/6/2150

[21] Kejun Wang, Xiaoxia Qi, Hongda Liu, Photovoltaic power forecasting based LSTM-Convolutional Network, Energy, Volume 189, Dec 2019,
https://www.sciencedirect.com/science/article/pii/S0360544219319206

[22] Jiahao Wu, Yongkai Zhao, Ruihan Zhang, Xin Li, Yuxin Wu, Application of three Transformer neural networks for short-term photovoltaic power prediction: A case study, Solar Compass, Volume 12, Dec 2024
https://www.sciencedirect.com/science/article/pii/S2772940024000237?ref=cra_js_challenge&fr=RR-1

[23] IRENA, Future of Solar Photovoltaic: Deployment, investment, technology, grid integration and socio-economic aspects (A Global Energy Transformation: paper), International Renewable Energy Agency, 2019,
https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2019/Nov/IRENA_Future_of_Solar_PV_2019.pdf

[24] C. Wan, J. Zhao, Y. Song, Z. Xu, J. Lin and Z. Hu, "Photovoltaic and solar power forecasting for smart grid energy management," in *CSEE Journal of Power and Energy Systems*, vol. 1, no. 4, pp. 38-46, Dec. 2015,
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7377167

[25] Zippenfenig, P. (2023). Open-Meteo.com Weather API [Computer software]. Zenodo, available: https://open-meteo.com/

[26] E.Wheatcroft, December 2018, LSE Research Online, Interpreting the skill score form of forecast performance metrics, available:
http://eprints.lse.ac.uk/91134/1/Wheatcroft_Interpreting-the-skill-score.pdf

[27] S.Olumide, August 2023, Arize, Root Mean Square Error (RMSE) In AI: What You Need To Know, Available: https://arize.com/blog-course/root-mean-square-error-rmse-what-you-need-to-know/

[28] A. Roberts, February 2023, Arize, Mean absolute percentage error (MAPE): what you need to know, available: https://arize.com/blog-course/mean-absolute-percentage-error-mape-what-you-need-to-know/

[29] I. Koprinska, D. Wu and Z. Wang, "Convolutional Neural Networks for Energy Time Series Forecasting," *2018 International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, Brazil, 2018, pp. 1-8, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8489399

[30] Amazon Web Service, What is Transfer Learning, [Online], [Accessed March 2024], https://aws.amazon.com/what-is/transfer-learning/

[31] T Mucci, IBM, Sept 2024, What is Data Leakage in Machine Learning , [Online], [Accessed March 2024], https://www.ibm.com/think/topics/data-leakage-machine-learning

[32] W H Khoong, Medium, Jan 2023, Why Scaling your data is important, [Online], [Accessed March 2024], https://medium.com/codex/why-scaling-your-data-is-important-1aff95ca97a2

[33] SciKit learn Documentation, MinMaxScaler, [Online], [Accessed April 2024], https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[34] Honghai Kuang et al, 2021, IOP Conf. Ser.: Earth Environ. Sci. 702 012019, Short-term wind power forecasting model based on multi-feature extraction and CNN-LSTM https://iopscience.iop.org/article/10.1088/1755-1315/702/1/012019/pdf

[35] GitHub Repository of Short-term wind power forecasting model based on multi-feature extraction and CNN-LSTM [Online], [Accessed March 2024], https://github.com/ozancanozdemir/CNN-LSTM

[36] J Heaton, T81-558: Applications of Deep Neural Networks, [Online], [Accessed March 2024], https://github.com/jeffheaton/app_deep_learning/blob/main/t81_558_class_10_3_transformer_timeseries.ipynb

[37] B Keck, *Timeseries Transformer*: Application of Basic Transformer Encoder Layer to Timeseries Forecasting, [Online], [Accessed April 2024],https://github.com/BrandenKeck/pytorch_fun/blob/main/timeseries_transformer/forecasting_model.py

[38] N C Paleti, Medium, July 2024, Positional Encoding Explained: A Deep Dive into Transformer PE, [Online], [Accessed April 2024], https://medium.com/thedeephub/positional-encoding-explained-a-deep-dive-into-transformer-pe-65cfe8cfe10b

[39] PVGIS User Manual [Online], [Accessed Feb 2024], available: https://joint-research-centre.ec.europa.eu/photovoltaic-geographical-information-system-pvgis/getting-started-pvgis/pvgis-user-manual_en

[40] A. Gillioz, J. Casas, E. Mugellini and O. A. Khaled, "Overview of the Transformer-based Models for NLP Tasks," *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, Sofia, Bulgaria, 2020, pp. 179-183, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9222960

[41] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2022. Transformers in Vision: A Survey. ACM Comput. Surv. 54, 10s, Article 200 (January 2022), 41 pages, https://dl.acm.org/doi/full/10.1145/3505244

[42] S. Naveen, M. S. S Ram Kiran, M. Indupriya, T.V. Manikanta, P.V. Sudeep, Transformer models for enhancing AttnGAN based text to image generation, Image and Vision Computing, Volume 115, Nov 2021, https://www.sciencedirect.com/science/article/abs/pii/S026288562100189X

[43] Gabriele Piantadosi, Sofia Dutto, Antonio Galli, Saverio De Vito, Carlo Sansone, Girolamo Di Francia, Photovoltaic power forecasting: A Transformer based framework, Energy and AI, Volume 18, Dec 2024, 100444, ISSN 2666-5468, https://www.sciencedirect.com/science/article/pii/S2666546824001101

[44] Jimin Kim, Josue Obregon, Hoonseok Park, Jae-Yoon Jung, Multi-step photovoltaic power forecasting using transformer and recurrent neural networks, Renewable and Sustainable Energy Reviews, Volume 200, Aug 2024, 114479, ISSN 1364-0321, https://www.sciencedirect.com/science/article/pii/S1364032124002028

[45] Q. -T. Phan, Y. -K. Wu and Q. -D. Phan, "An Approach Using Transformer-based Model for Short-term PV generation forecasting," *2022 8th International Conference on Applied System Innovation (ICASI)*, Nantou, Taiwan, May 2022, pp. 17-20 https://ieeexplore.ieee.org/document/9774491

[46] Zhao, X. A novel digital-twin approach based on transformer for photovoltaic power prediction. *Sci Rep* **14**, 26661 (Nov 2024), https://www.nature.com/articles/s41598-024-76711-4

[47] Zhou Ning , Shang Bo-wen , Zhang Jin-shuai , Xu Ming-ming, Research on prediction method of photovoltaic power generation based on transformer model, Frontiers in Energy Research, Volume 12, Aug 2024, https://www.frontiersin.org/journals/energy-research/articles/10.3389/fenrg.2024.1452173/full

[48] Pospíchal J, Kubovčík M, Dirgová Luptáková I. Solar Irradiance Forecasting with Transformer Model. *Applied Sciences*. Aug 2022, https://www.mdpi.com/2076-3417/12/17/8852

[49] Dong H, Tang P, He B, Chen L, Zhang Z, Jia C. Multi-step solar radiation prediction using transformer: A case study from solar radiation data in Tokyo. Journal of Building Physics. Jan 2024, https://journals.sagepub.com/doi/10.1177/17442591231218831?icid=int.sj-abstract.similar-articles.7

[50] Matteo Tortora et al, MATNet: Multi-Level Fusion Transformer-Based Model for Day-Ahead PV Generation Forecasting, Jun 2023, https://arxiv.org/html/2306.10356v2

[51] PyTorch Documentation, TransformerEncoderLayer, [Online], https://pytorch.org/docs/stable/generated/torch.nn.TransformerEncoderLayer.html

[52] W Wong, Medium, What is residual connection?, Dec 2021, [Online], [Accessed April 2024],https://medium.com/data-science/what-is-residual-connection-efb07cab0d55

[53] University of York, Institute for Safe Autonomy, [Online], [Accessed April 2024] https://www.york.ac.uk/safe-autonomy/

7 Appendix A

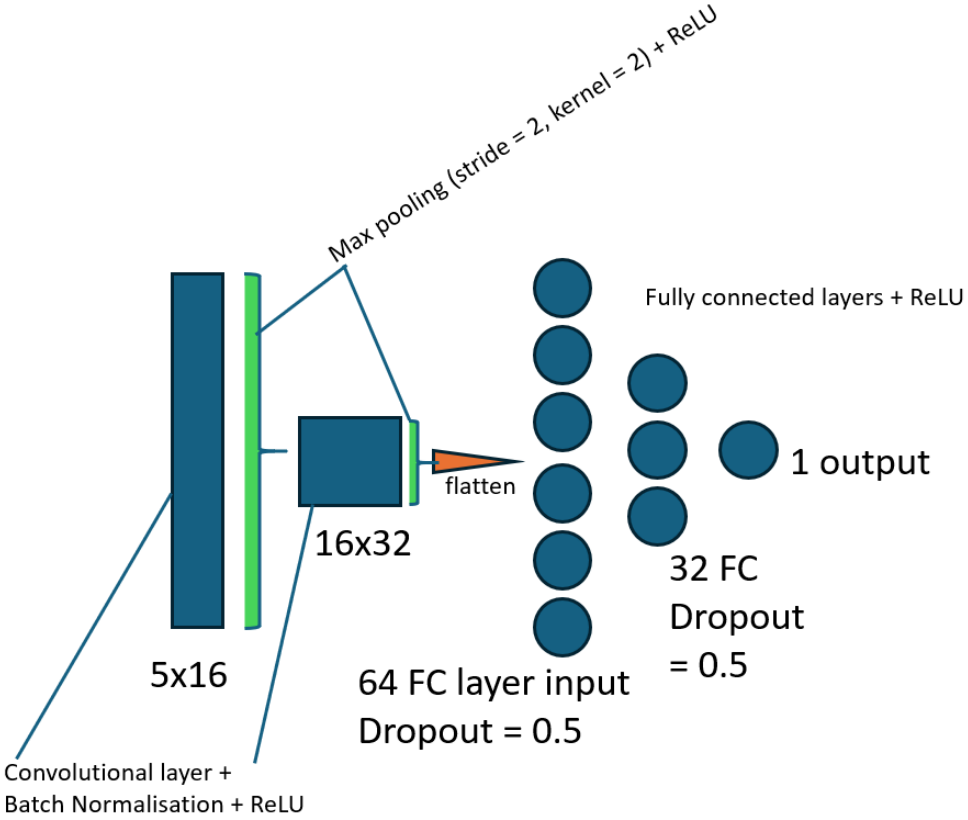

Figure 7-1: The CNN architecture used for this experiment

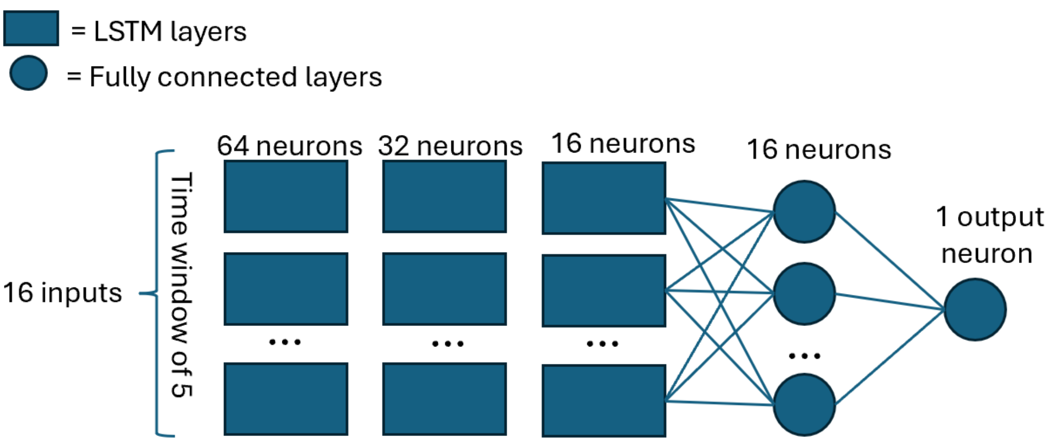

Figure 7-2: The LSTM architecture used for this experiment

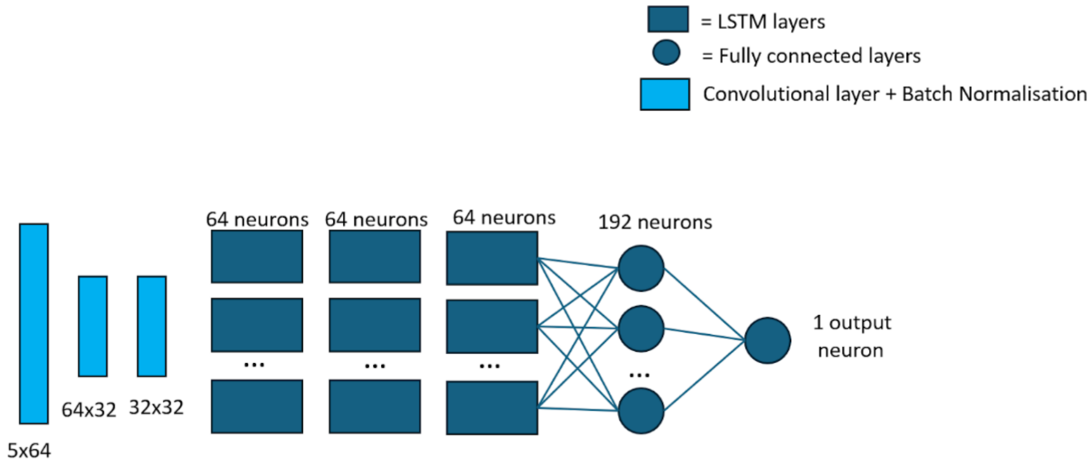

Figure 7-3: The CNN-LSTM architecture used for this experiment

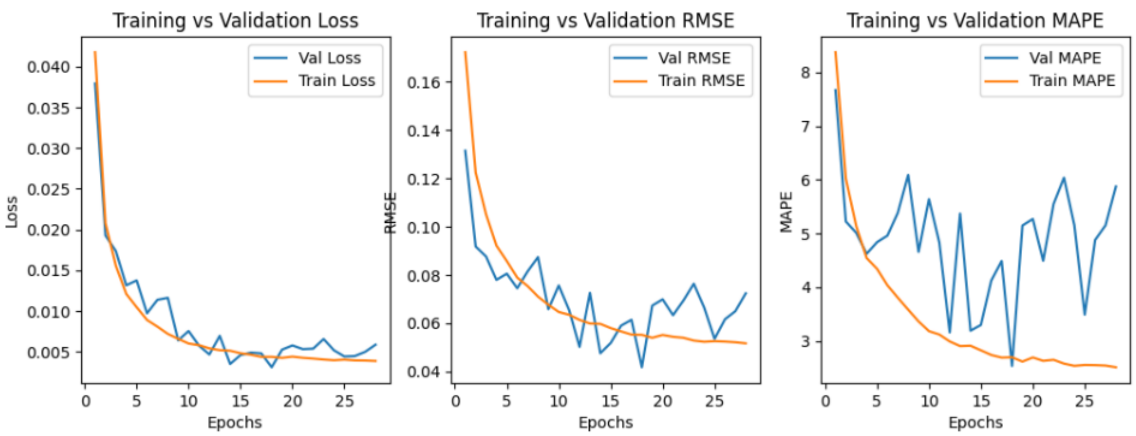

Figure 7-4: The testing and validation metrics plotted for each epoch during training for the CNN

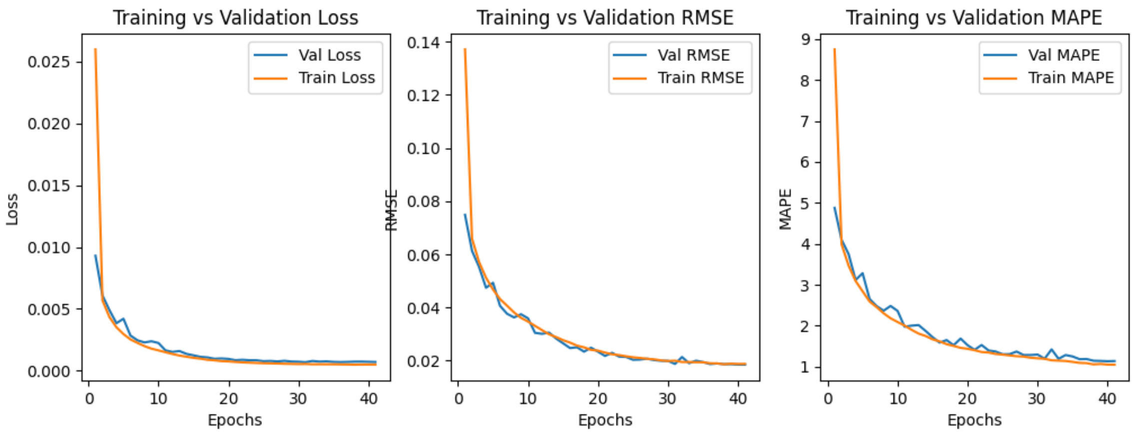

Figure 7-5: The testing and validation metrics plotted for each epoch during training for the LSTM

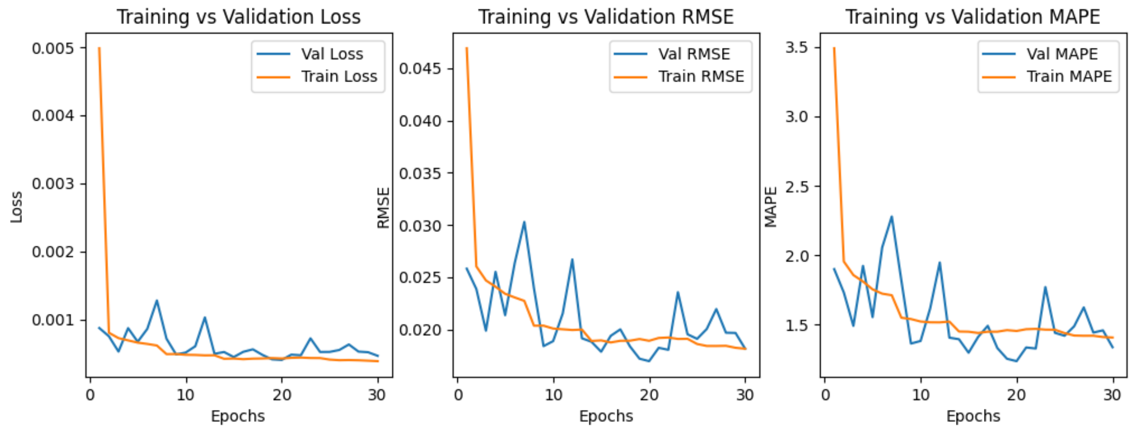

Figure 7-6: The testing and validation metrics plotted for each epoch during training for the CNN-LSTM

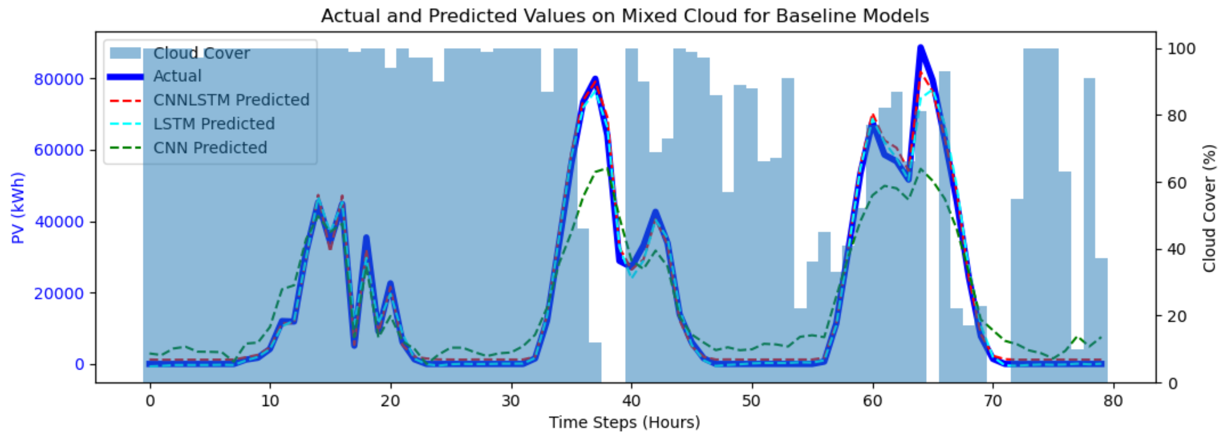

Figure 7-9: The predictions from the baseline models for the mixed clouds data subset (dashed lines), plotted against the actual value (solid blue line)

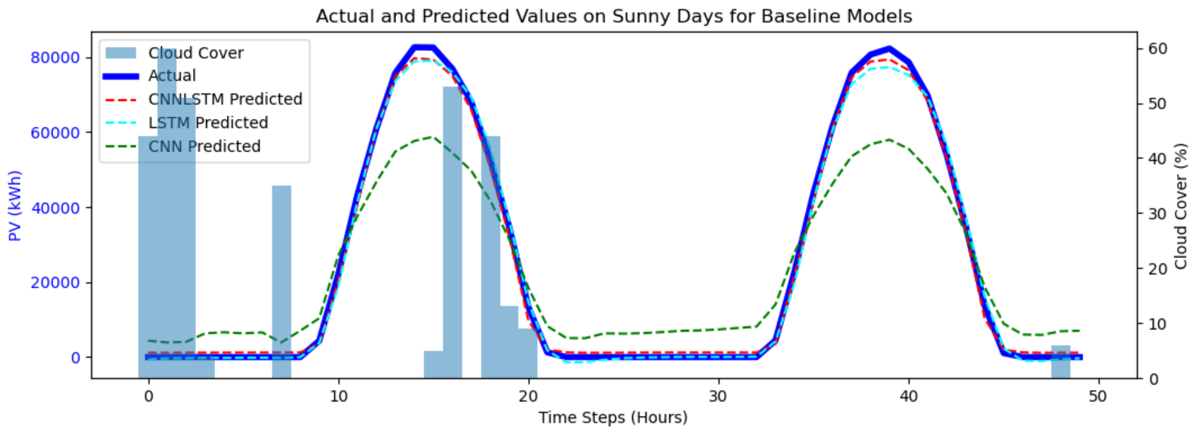

Figure 7-7: The predictions from the baseline models for the sunny data subset (dashed lines), plotted against the actual value (solid blue line)

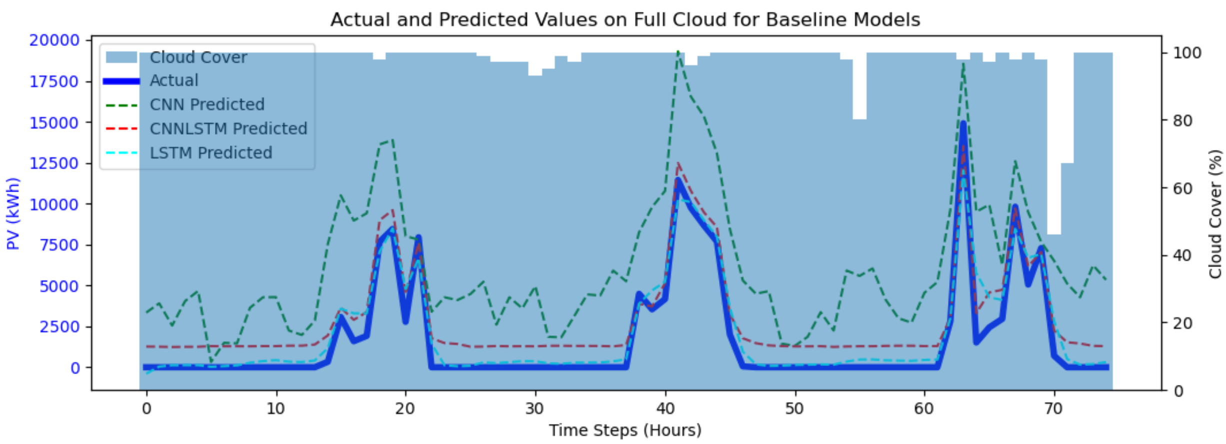

Figure 7-8: The predictions from the baseline models for the full clouds data subset (dashed lines), plotted against the actual value (solid blue line)

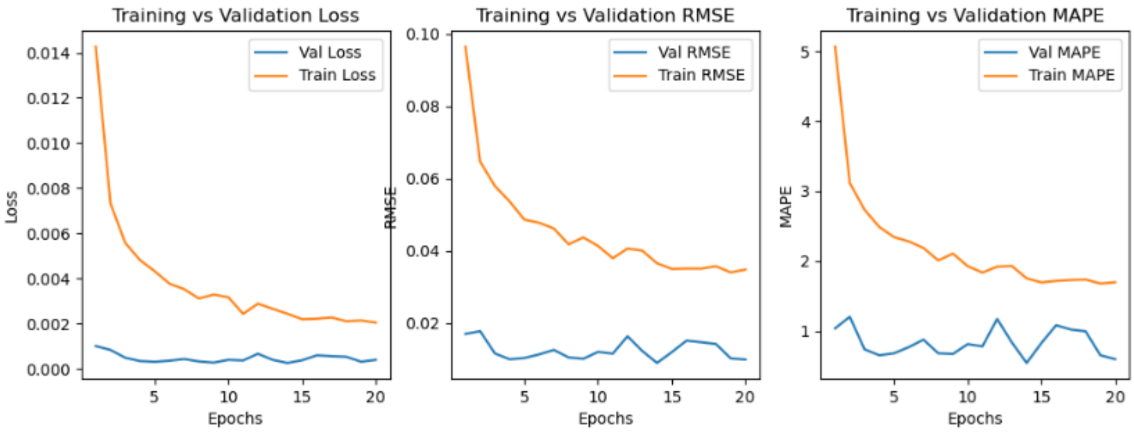

Figure 7-10: Pretrained CNN's test and validation metrics per epoch during finetuning

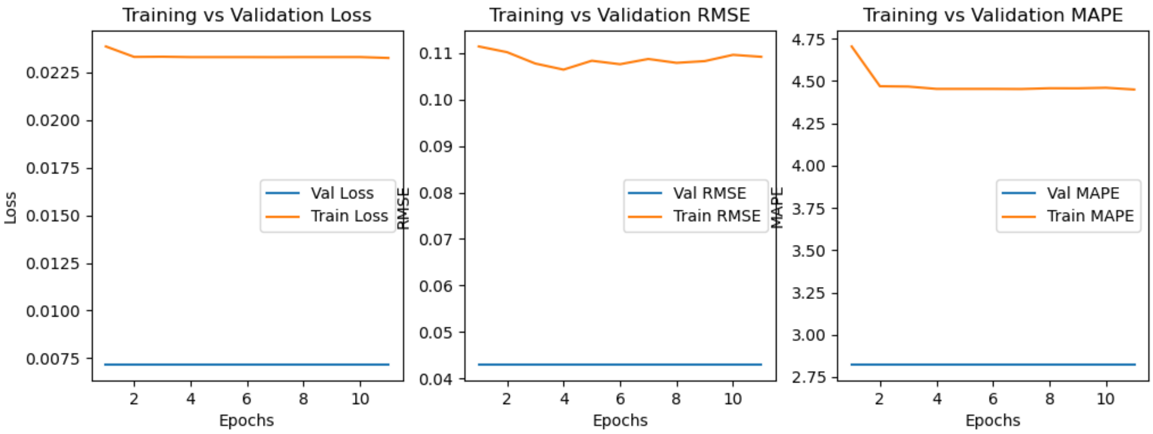

Figure 7-11: Non Pretrained CNN's test and validation metrics per epoch during finetuning

Figure 7-12: Pretrained LSTM's test and validation metrics per epoch during finetuning

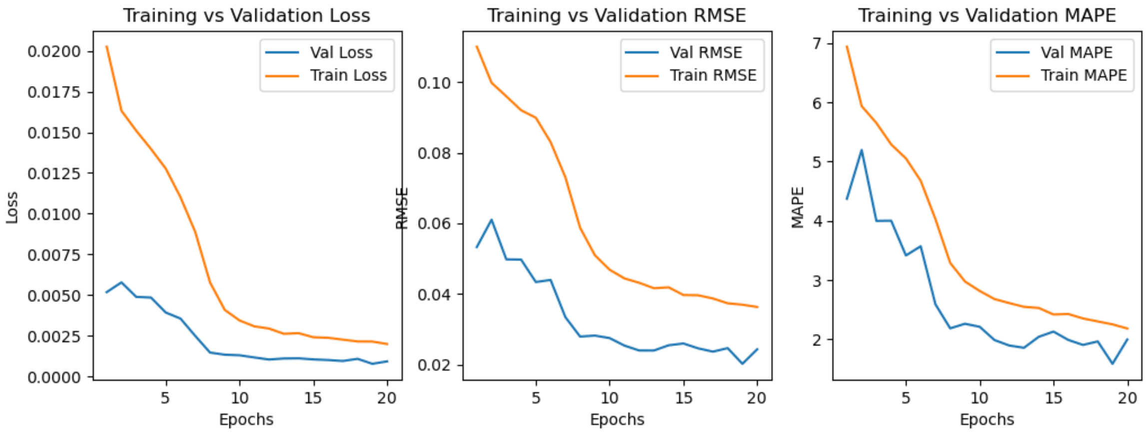

Figure 7-13: Non Pretrained LSTM's test and validation metrics per epoch during finetuning

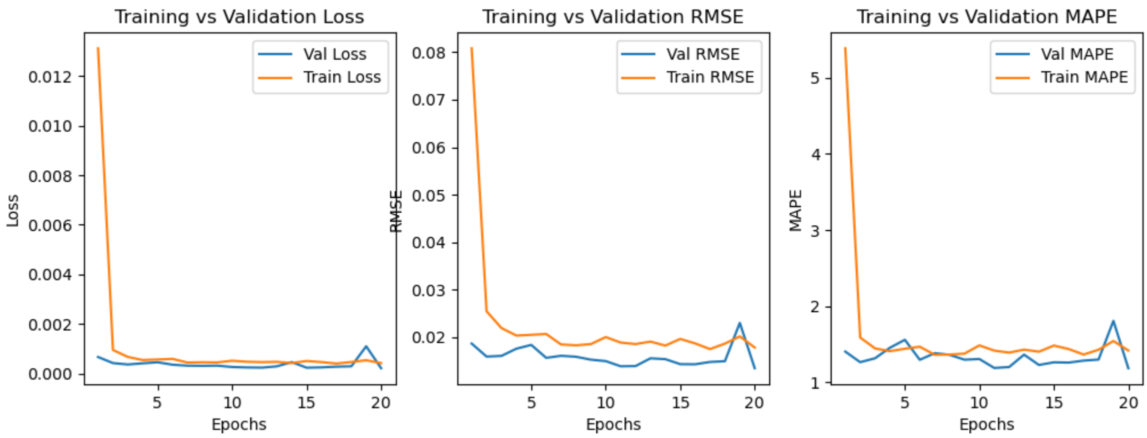

Figure 7-14: Pretrained CNN-LSTM's test and validation metrics per epoch during finetuning

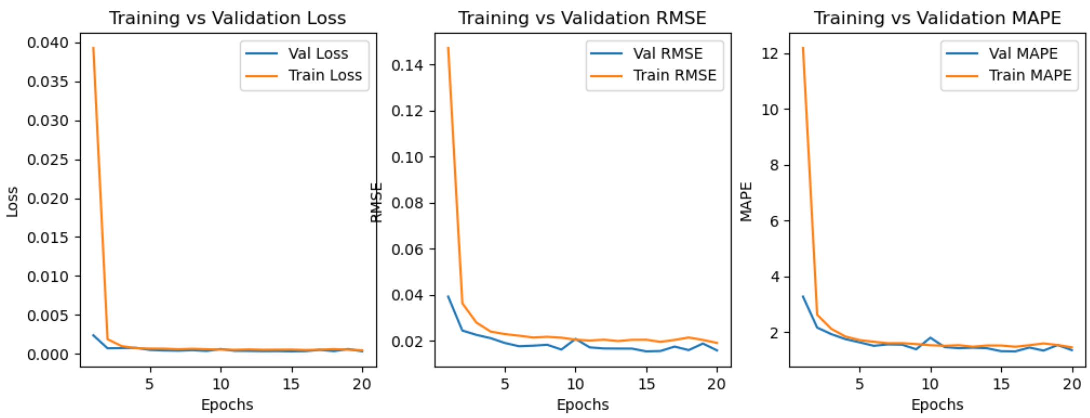

Figure 7-15: Non Pretrained CNN-LSTM's test and validation metrics per epoch during finetuning

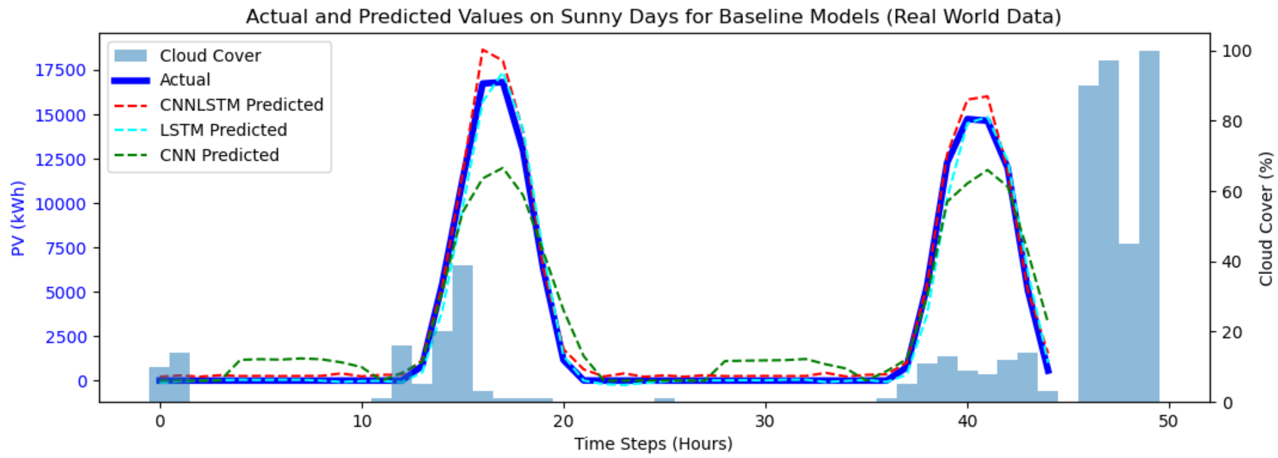

Figure 7-16: The predictions from the pretrained baseline models on the sunny days data subset

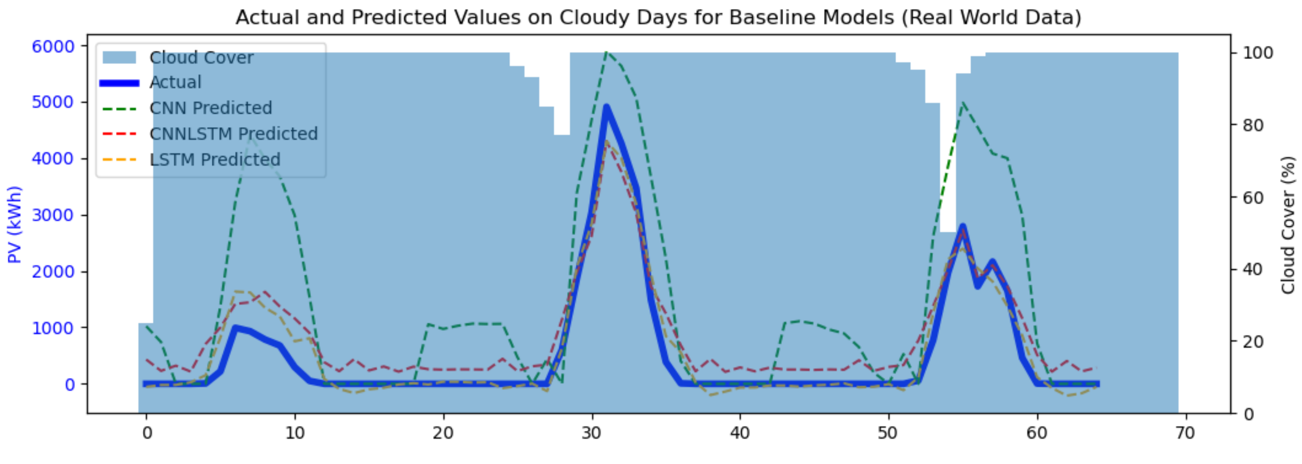

Figure 7-17: The predictions from the pretrained baseline models (dashed lines) on the cloudy data subset, plotted against the actual value (solid blue line)

8 Appendix B

The code and data used for this project can be found at:

https://github.com/dema1537/ProjectSolarPV

The data used can be found in the Data folder.

The OpenMeteoData.csv is the weather data for the training of the models.

The PVGISdata.csv is the synthetic PV data for training.

The façade_solar_data.csv is the PV data for the real world farm operated by the Institute of Safe Autonomy at the University of York. This was used for the finetuning.

The façade_weather_data.csv was the weather data for the real solar plant listed above.

The CloudDataForRealWorld.csv contained just the cloud cover for the time period the real world data covered. This data was not used during training or finetuning, but was instead only used to plot cloud cover on the real world prediction graphs.