Solar Irradiance Prediction

Alexander Peplowski
Université de Montréal
alexander.peplowski@gmail.com

Harmanpreet Singh
Université de Montréal
harmanpreet.singh@umontreal.ca

Marc-Antoine Provost
Université de Montréal
provoma1@hotmail.com

Mohammed Loukili
Université de Montréal
mohammed.loukili.1@umontreal.com

ABSTRACT

Solar irradiance is the power per unit area (watt per square meter, W/m^2), received from the Sun in form of electromagnetic radiation. [26] Many external factors such as physical and external factors can affect how much solar irradiance is available at a given location on earth. The effective irradiance is called Global Horizontal Irradiance (GHI) [26]. In this paper, we propose a deep learning approach to solar irradiance nowcasting. The experimental results show that the proposed method achieves forecast skill of about 110 GHI RMSE for up to six hours in the future.

1 INTRODUCTION

The present paper describes a study that was made on satellite imagery (GOES-13)[27] measured by seven SURFRAD [2] stations in the continental US. The goal was to use satellite imagery in order to build a robust predictive model that would provide Global Horizontal Irradiance (GHI)[26] values at specific points on a given map for both present and future times. Specifically, the prediction horizon interest is $T_0, T_0 + 1h, T_0 + 3h, T_0 + 6h$, i.e, predicting GHI values for a specific timestamp as well as for 1, 3, and 6 hours in the future. Such models can have a significant impact on the future of large-scale renewable energy plants as it is critical to power system planning, management, and operations. Many existing GHI prediction solutions rely on physical and mathematical models. However, these solutions do not always capture complex temporal phenomena (clouds, winds, etc.). The hypothesis of our project is that satellite imagery for nowcasting can be used to help model these complex phenomena and seek to develop models leveraging open data sources. As for the metrics we used to evaluate our model, we used Mean Squared Error (MSE) as loss function to optimize our model and kept track of the Root Mean Squared Error (RMSE) [28] to evaluate the performance of our model, see paragraph 4.3.

2 LITERATURE REVIEW

By definition GHI is the sum of the diffuse radiation incident (DRI) on a horizontal surface plus the direct normal irradiance (DNI) projected onto the horizontal surface, which can be formultaed as follow [20]:

$$GHI = DRI + DNI * cos(z)$$

where z is the solar zenith angle of the Sun.

We have found tremendous articles that tackle the solar irradiation forecast using Artificial intelligence models. Before, going through these models, it worth highlighting that solar irradiation models can be divided in 3 main categories [7][15] which are:

- Numerical Weather Prediction models which are deterministic models, based on current weather conditions [25]. This kind of model gives better prediction from 6 hours on-wards [7].
- Statistical models, which are based on the on-site time-series measurements. This kind of models have better performance for prediction up to 6 hours [7].
- Hybrid models, is yet another kind of models that combine different algorithms from the previously cited models, in order to improve the prediction accuracy. We can cite these references [5], [4] for more details.

Most of the papers make a comparison with a simplistic model that is Persistent model. its also known as naïve model [8]. It is recommended to use it as a comparison before implementing complex models. The persistence model supposes that global irradiance at time t is best predicted by its value at time $t-1$:

$$\hat{X}_{t+1} = X_t$$

There is also the Clear Sky model which is a deterministic model that estimates the global irradiance in clear sky conditions at any given time. Clear sky models are also used to calculate a cloudiness index or clearness index [14]. There are a large variety of clear sky models, ranging from simplest models which are only a function of the solar zenith angle to more complicated models that use many atmospheric parameters, such as aerosols and precipitable water [14].

In the recent decade, the Artificial Neural Networks (ANNs) models used successfully to represent complex functions. They have been used successfully as well for GHI predictions. These models can take either time series data, time-series images or both. Mellit and Pavan [16] proposed a Multilayer Perceptron (MLP) which can forecast GHI for 24 hours ahead. It accepts as input parameters mean daily irradiance and mean daily air temperature; Similarly Kemmoku and al. [12] used a multistage ANN to predict GHI of the next day. The input data to the network are the average atmospheric pressure, predicted by another ANN and various weather data of the previous day. Sfetsos and Coonick [22] used an ANN to make one-step predictions of hourly values of GHI.

Cloud imagery based models are interesting because cloud imagery can capture details that classical approaches can not [8]. These models can detect the motion of cloud structures using motion vector fields like proposed by Lorenz and al. [6]). The errors of satellite data and sky images based forecasts proposed in the literature increase drastically under low sun elevations, high spatial variabilities and low irradiance conditions [8]. Hammer and al. [9] demonstrated achieved 17% RMSE in satellite imagery for

30 minutes cloud index forecasts and 30% RMSE at 2 h forecast horizons.

In the last years, more modern and sophisticated models have emerged. Mukhoty and Maurya [18] used a recurrent neural network (RNN) and Long Short Term Memory (LSTM) model architecture. The advantage of this model is his ability to capture long term dependency in the data, which is common in time series forecasting. The data used was captured through weather satellites in hourly basis. The experiment shows that the LSTM architecture gives better result than RNN or classical ANN architecture. Zhao and Wei [30] used a new Convolutional Neural Network (CNN) with 3D kernel to forecast direct normal irradiance (DNI), based on consecutive ground-based cloud (GBC) images. This model gives 17.06% improvement over the persistent model. Kamadinata et al. [11] combined two ANNs to predict GHI in 1 to 5 minutes in advance from sky images. The first model take as input red blue ratio (RBR) from sky images to predict cloud movement, and then feed a second ANN model to make GHI prediction. They also experimented a simpler model which make GHI prediction based on sky image extracted features. The results show that the accuracy of their model is comparable with the existing prediction techniques with less expensive instruments and less computational efforts. An interesting approach is used by Xingjian et al. [23] by using a convolutional LSTM model (ConvLSTM) to predict the future rainfall intensity in a local region over a relatively short period of time. While the subject is not related to GHI prediction, the approach could be useful as it uses images as input and LSTM model to memorize time dependencies. We finish the literature review with this elegant approach from Haixiang Zang et al. [29] where they used two CNN-based models, that is residual network (ResNet) and dense convolutional network (DenseNet). They introduced also a new data preprocessing method to construct input feature maps for the two novel CNNs, and they used a meta learning strategy, based on multi-loss-function network, to train the two deep networks. The authors claimed a coverage error rate less than 5%, while specially the error rate of ResNet is smaller than 1%.

3 DATA ANALYSIS

3.1 Data sources

We were provided with two main data sources, such as GEOS-13 imagery and some metadata stored as a Pandas dataframe[19]. Since the raw NetNCDF imagery of the Geostationary Operational Environmental Satellite provided by the National Oceanic & Atmospheric Administration (NOAA) was over 800GBs, it was split into 15-minute chunks. More specifically, it was provided in three different versions;

- The original, 16-bit, GZIp-compressed (lossless) NetNCDF files, in 15-minute chunks.
- Repackaged, 16-bit, JPEG2000-compressed (lossy) HDF5 archives, in 1-day chunks.
- Repackaged, 8-bit, JPEG-compressed (lossy) HDF5 archives, in 1-day chunks.

Each GOES-13 satellite imagery has five channels, each one of them corresponding to different wavelength. Keep in mind that the satellite is geostationary, i.e it is stationary in relation to a fixed point on Earth. As for the metadata, each column in the dataframe specifies an attribute for every 15-minute entry. Some of these attributes include the absolute path on the remote server (Helios) to the GEOS-13 imagery files, binary flag indicating whether it is day time or not at the station, a cloudiness categorical flag indicating the weather conditions at the station, the clearsky GHI [20], and the real measured GHI at the station.

3.2 Data analysis

By doing exploratory data analysis, after remapping string nans to numpy NaNs, we observed that there were some missing values for certain features (Table 1). Note that the total number of observations is 210 336.

Features	Missing percentage
ncdf_path	17.4098%
hdf5_8bit_path	0.015%
hdf5_16bit_path	0.015%
BND_GHI	0.086%
TBL_GHI	0.2444%
DRA_GHI	0.5980%
FPK_GHI	0.2696%
GWN_GHI	1.2242%
PSU_GHI	0.1084%
SXF_GHI	0.362753%

Table 1:
Relevant percentage of missing values per feature over the whole DataFrame

Based on the utility module provided, we observed that even if the percentage of missing hdf5_path files was lower than the percentage of missing ncdf_path files, we should not take the occurrences where there are hdf5_path available but no ncdf_path available because on such occurrences, the hdf5 files may correspond to another timestamp where the data was available for both files. Then, we explored if the distribution of nighttime and daylight observations were equal for all of the seven stations. It turns out that effectively, the distribution of nighttime and daylight observations is almost equal for all stations (around 50-50). We also wanted to confirm our hypothesis that the location of the station would affect the daylight GHI values. Indeed, our hypothesis was valid, as a station like DRA, (Figure 1) which is in Nevada receives more sunlight than a station like PSU (Figure 2), which is located in a northern state (Pennsylvania).

Next, we wanted to verify if the clearsky_ghi values provided in the dataframe were hard limits or if the true GHI values could exceed them. As observed, GHI values are not bounded by clearsky_ghi values (Figure 3). This gave us a good indication to not bound our predicted values by the clearsky GHI.

As per the underlying structure of the data, it is important to note that an image is 650 pixels per 1500 pixels for each of its channels, that one pixel is equal to 16km^2 and that each GOES-13 image consists of five channels.

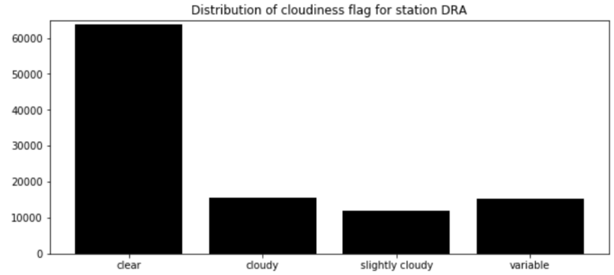

Figure 1: Cloudiness distribution for DRA station

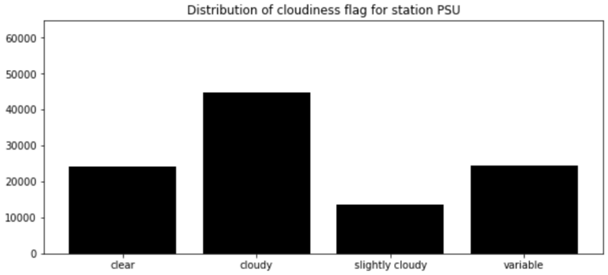

Figure 2: Cloudiness distribution for PSU station

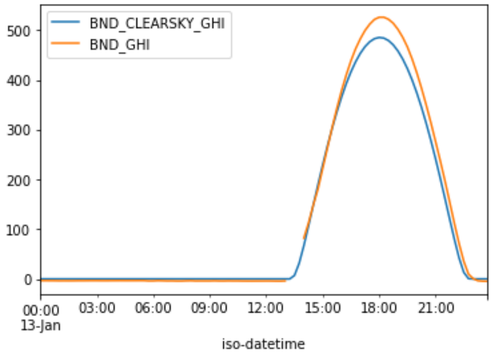

Figure 3: True vs clearsky GHI values at the BND station for one day

3.3 Handling missing data

As shown in Table 1 we have missing GHI values during daytime along with missing images. We have implemented methods to handle this issue. All the details of these experiments are shown respectively in the following notebooks `Data_analysis.ipynb` and `Image_analysis.ipynb` located in Notebook folder in our github repository [1]. To summarise these experiments, the GHI value was interpolated with two methods. First, we used the nearest available values from previous and future timestamps, and interpolate linearly all missing values in between. This approximation is good enough for little gaps of a couple of timestamps of 15 minutes. But one it comes to big gaps of more than 3 hours, then the linear interpolation is not anymore realistic. To handle this situation, we have used the GHI values of the day before and the day after to calculate the mean of GHI value for each missing timestamp. This method gives a better approximation that linear interpolation as it makes a smoother transition from the day before to the day after.

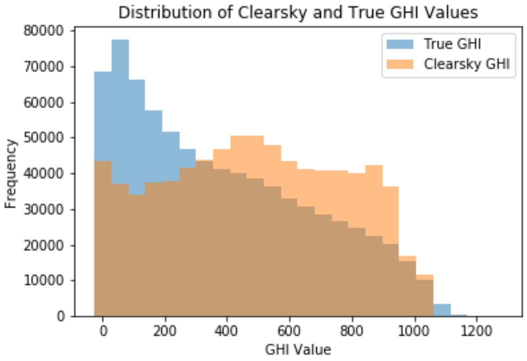

Figure 4: Daytime GHI Distributions

Regarding the images, we have used the same principal that is taking the linear interpolation for the missing image from the previous timestamp and the next timestamp. As shown in the notebooks, the interpolated images looks realistic. We discuss in more details the methodology adopted for handling missing data in the Data processing section.

4 METHODOLOGY

4.1 Data processing

4.1.1 NaN Values. The first step of our data processing was to handle NaN values. In fact, we converted string nans to numpy NaN and for each station that had missing nightly clearsky GHI or true GHI, we filled those with 0 since there is no solar irradiance at night. Then, we interpolated missing GHIs based on both previous and future values in the dataframe. For the reasons discussed above in the data analysis section, we dropped the dataframe observations where they were no path to the NCDF files and for the .H5 files. Then, we shuffled the dataframe so our model could remain as general as possible. All this preprocessing activities are shown in our `Data_analysis.ipynb` notebook in our github repository [1]

4.1.2 Image Cropping. Having preprocessed the dataframe, the next step was to crop the images for training and validation set. Our training set spanned from 2010 to 2014 and our validation set consisted of the images from 2015. The goal was to crop the GOES-13 images centered on the coordinates of the stations so our data loader would be more efficient. Before doing so, we needed to map the station coordinates to their pixel value. In fact, while we have the station coordinates, e.g., BND which sits at 40.05192, -88.37309, this does not correspond to the pixel 40.05192 in our image which has a shape of (650, 1500). The cropping experiment and mapping the stations coordinates on the images are shown in our `Image_analysis.ipynb` notebook in our github repository [1]. After this mapping was applied, we observed that one particular station, FPK, is located at the 607th pixel on the x-axis. It is an important information that we considered further along our data processing when defining the cropping window hyperparameter, otherwise the crop would be out of bound.

4.1.3 Image Sequencing. While cropping, we also defined a sequence length of previous images to consider alongside T_0. This sequence length consisted of the previous $T_{0-30\text{min}}, T_{0-1\text{h}}, .., T_{n-30\text{min}}$. We only considered a sequence valid if T_0 was during daytime. If one of the timestamps in the defined sequence length was missing, we imputed this particular timestamp by the image at T_0. Thus, for every timestamp T_0 during daytime, we cropped each channel's image for each station as well as the defined sequence length of images at 30 minute intervals. We then stacked those channels together to get a single image and standardized it. Since the range of the pixel values varied across the images and channels (channel 1, 2, 3, 4, and 6), we retrieved the mean and standard deviation of each channel in order to accomplish the normalization.

4.1.4 Cyclical Solar Data. Another processing step that we did was to encode cyclical continuous features such as the day of the years and the time of the day such that our model knew they were cyclical. We did so by transforming the data in 2D using a sin and cosine transformation.

4.2 Data Loading

Once we had the cropped images, we wanted to save those images so our data loader could access them. For each T_0 timestamp alongside its previous sequence of images, we defined a mini-batch of 256 that we saved in a .H5 file in addition to the true GHI, the clearsky GHI and the daytime flag for T_0, T_1, T_3, T_6. Also, in this .H5 file, we stored the station ID and the true datetime for each station, that converts the global timestamp to its own local timestamp according to the timezone of each station. These additional features were useful for the other models tried, such as the ConvLSTM [23] mentioned in the literature review. Those files were stored in our team directory on Helios server and served as input to our model.

4.3 Model description

4.3.1 Loss function. We define the residual clearsky ghi as:

$$k_{true} = \frac{ghi_{true}}{ghi_{cs}}$$

The MSE loss function is calculated as:

$$\text{loss} = (k_{true} - k_{pred})^2$$

The metric used for early stopping:

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{1}$$

4.3.2 Input Features.

- Set of 3 cropped images centered around station coordinates (resolution 82x82x5)
- Clearsky values $ghi_{cs} \in \mathbf{R}^4$ for $T = \{0, 1, 3, 6\}$
- Nighttime flags to prevent training on nighttime examples $\in \mathbf{R}^4$
- Cyclical time of day $\in \mathbf{R}^2$
- Cyclical time of year $\in \mathbf{R}^2$

Cyclical time of day and time of year are defined as:

$$t_d = \sin\left(\frac{\text{sec in day}}{\text{sec per day}} * 2\pi\right)$$

$$t_y = \sin\left(\frac{\text{day in year}}{\text{days per year}} * 2\pi\right)$$

$$\text{cyclical time of day} = (\sin t_d, \cos t_d)$$

$$\text{cyclical time of year} = \left(\sin t_y, \cos t_y\right)$$

4.3.3 Regularization. For regularization, we used batch normalization (BN) [21]. It is a technique used to normalize activations in intermediate layers of deep neural networks in order to alleviate internal covariate shift, i.e, the tendency of the distribution of activations to drift during training, thus affecting the inputs to subsequent layers [21]. BN accelerates training, enables higher learning rates, and improves generalization accuracy [3].

4.3.4 Hyperparameter Optimization Strategy. Due to time restriction, manual hyperparameter tuning was done. We did so by comparing performance on the cross-validation data set with the best baseline model and the model with the hyperparameter changes. The impact of the change of hyperparameter is compared based on the before/after performance on the validation set.

4.3.5 Model. We implement a feed-forward CNN with architecture similar to VGGnet [24] and AlexNet [13]. To summarize, our model consists of:

- One 3D CNN Layer
- 5 2D CNN Layers
- 3 max pooling layers
- Batch Normalization [21] after every layer
- RELU activation function after each batch normalization operation
- 3 dense layers after CNN portion of model
- Sigmoid output activation function

Refer to figures 5 and 6 for an illustration of the final model design. Refer to table 3 for a description of the convolutional layers used by the model.

Note that after the convolutional layers, we stack normalized clearsky GHI values and time of day and time of year cyclical features with the output of the CNN layers, and pass these inputs on to the dense layers.

Also of note: because of the sigmoid output function, the model will effectively predict k_{pred} which represents a percentage of the clearsky GHI value ghi_{cs}. Therefore, we can recover the prediction of the true GHI value by multiplying it with the clearsky GHI value as follows: $ghi_{pred} = ghi_{cs} * k_{pred}$

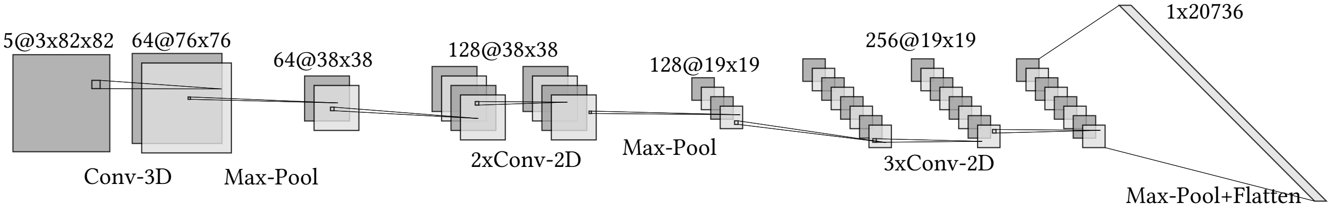

Figure 5: 3D-CNN architecture

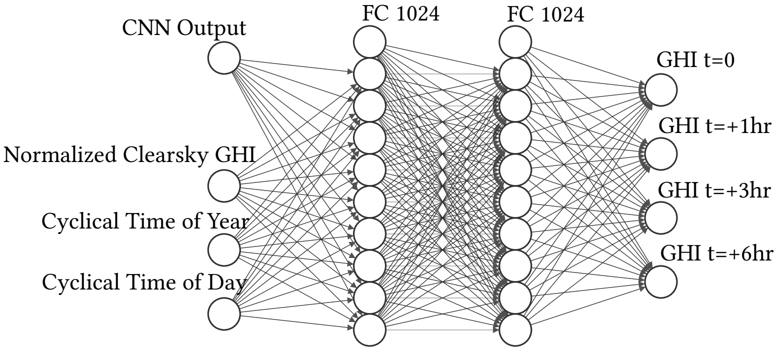

Figure 6: Dense layer of the 3D-CNN

Layer	Kernel	Output shape
(Input layer)	N/A	5x3x82x82
Conv3D	64x(3x7x7)	64x76x76
Max pooling	(2x2)	64x38x38
Conv2D	128x(3x3)	128x38x38
Conv2D	128x(3x3)	128x38x38
Max pooling	(2x2)	128x19x19
Conv2D	256x(3x3)	256x19x19
Conv2D	256x(3x3)	256x19x19
Conv2D	256x(3x3)	256x19x19
Max pooling	(2x2)	256x9x9
Flatten	N/A	1x20736

Table 2:
Architecture of the 3D-CNN

The output of the convolutional network is then stacked with additional input data. This data includes the cyclical time of day, cyclical time of year, and the normalized clearsky GHI value.

5 RESULTS AND DISCUSSION

5.1 Cross-Validation Techniques

The model is trained on all data from 2010-2014 and it is validated on data from year 2015 only. The objective of this choice of cross-validation split is to ensure that the data is distributed evenly. The weather over one year is an annual cyclical event, so to ensure an even distribution of data, we aim to select data for training and validation from periods of time that cover this whole cyclical event. Also note that GHIs that occur while 'night_flag' is set are excluded from scoring calculations as to not have the model learn on irrelevant data.

5.2 Final Hyperparameters

After hyperparameter optimization was performed, the following were used in the 'best model': the number of cropped images is set to 3, where the time difference between image captures is 30 minutes. The final batch size is 256 units, whose choice was a trade-off between performance and memory limits. The learning rate used was (0.001) with the Adam optimizer. Some hyperparameters that were fixed without substantive tuning are the dropout rate (0), the crop window size (82x82) and the max_k GHI. Note that max_k GHI is defined to be the maximum clearsky ratio, so for example, a value of 1.1 would imply a maximum prediction of 110% of the clearsky value at that time.

Some experimentation was performed with dropout vs batch normalization and it was found that better performance was obtained with batch normalization. Additionally, the number of dense nodes was varied by multiples of 2. In the end, best performance was obtained with 1024 nodes in all 3 dense layers.

5.3 Result presentation

Refer to figures 7 and 8 for the training curves which represent the best model's loss and GHI RMSE performance as a function of training epoch. Note that only 3 epochs of training are required for convergence. Note that the small number of epochs required is also a function of the number of training images available, which is large since our training set spans many years and includes data which might share similar characteristics with other random samples from the training set.

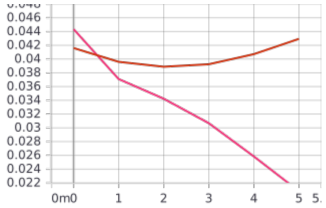

Figure 7: Loss over training epochs for the best model

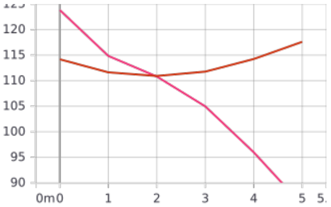

Figure 8: RMSE performance over training epochs for the best model

Model Description	Val RMSE
Best model (ref hyperparmeter section)	110.2
Model with sequence of 5 images	112.2
Dense layers have 2048 Nodes	111.5
Add onehot station IDs	110.9
Clearsky baseline	163.0

Table 3: Performance Summary of Various Models

While our model beats the clearsky deterministic model, it does not achieve state of the art performance. We suggest few improvements such as data augmentation, i.e rotating, flipping, and zooming the images. In addition, as stated above, manual hyperparameter search was conducted. Other than typical grid search and random search, Bayesian optimization could be tried to improve the performance of the model.

6 CONCLUSION

In this study, a CNN-based model utilizing GOES-13 satellite imagery and related metadata for nowcasting GHIs prediction was proposed. The main advantage of the 3D-CNN model we propose is its faculty to represent complex spatial and temporal phenomena like the displacement of atmospheric conditions such as clouds. We demonstrate the validity of the model's capability of such representation by comparing its performance with that of the clearsky deterministic model baseline.

6.1 Future Work

We were constrained due to tight project deadlines. Given more time, we would have liked to explore many things such as:

In the development of our model, as stated in the methodology, we manually tuned hyperparameters. Future work might include performing a more sophisticated hyperparameter search with Orion [17] or using some other parallel hyperparameter searching tool.

In the preprocessing step, we may want to experiment with other ways of handling missing data. One approach which would be interesting to interpolate missing images, that is, instead of copying the image from t_0 to fill missing images, we would interpolate the missing data from temporally adjacent images. In addition, an interesting idea that we would have liked to test is data augmentation, i.e rotating, zooming, and flipping images.

Also, we would like to run further experiments with image sequences of differing lengths or spacing. Currently, we only performed experiments with a set of 3 and 5 images, with a temporal spacing of 30 minutes apart. It would be interesting to see if performance is maintained or improved with fewer images, or if the spacing plays a large factor in determining model performance.

Our cross-validation technique relied on only one particular split of our data. Given more compute resources, we would be able to better validate the performance metrics by performing some version of k-fold cross validation. This would allow us to be more confident in the hyperparameters and models which we finally selected.

Finally, it would be interesting to implement a different set of models than the ones we tried (CNN and LSTM). One such architecture that might be interesting to explore could be implementation similar to ResNet [10].

ACKNOWLEDGMENTS

This paper was jointly written and edited by all the authors.

REFERENCES

[1] Team 08. winter 2020. Github repository. https://github.com/notAlex2/Solar-irradiance-Team08-IFT6759.git

[2] National Oceanic Atmospheric Administration. 2019. Earth System Research Labotary. https://www.esrl.noaa.gov/gmd/grad/surfrad/

[3] Johan Bjorck, Carla Gomes, Bart Selman, and Kilian Q. Weinberger. 2018. Understanding Batch Normalization. *arXiv e-prints*, Article arXiv:1806.02375 (May 2018), arXiv:1806.02375 pages. arXiv:cs.LG/1806.02375

[4] J.C. Cao and S.H. Cao. 2006. Study of forecasting solar irradiance using neural networks with preprocessing sample data by wavelet analysis. *Energy* 31, 15 (2006), 3435–3445. https://doi.org/10.1016/j.energy.2006.04.

[5] Shuanghua Cao and Jiacong Cao. 2005. Forecast of solar irradiance using recurrent neural networks combined with wavelet analysis. *Applied Thermal Engineering* 25 (02 2005), 161–172. https://doi.org/10.1016/j.applthermaleng.2004.06.017

[6] Marco Girodo Detlev Heinemann, Elke Lorenz. [n.d.]. Forecasting of Solar Radiation. *Oldenburg University, Institute of Physics, Energy and Semiconductor Research Laboratory, Energy Meteorology Group* ([n. d.]). https://uol.de/fileadmin/user_upload/physik/ag/ehf/enmet/publications/solar/conference/2005/Forcasting_of_Solar_Radiation.pdf

[7] Hadja Diagne, Mathieu David, Philippe Lauret, and John Boland. 2012. Solar irradiation forecasting: state-of-the-art and proposition for future developments for small-scale insular grids. (05 2012).

[8] Maimouna Diagne, Mathieu David, Philippe Lauret, John Boland, and Nicolas Schmutz. 2013. Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renewable and Sustainable Energy Reviews* 27 (2013), 65 – 76. https://doi.org/10.1016/j.rser.2013.06.042

[9] A. Hammer, D. Heinemann, E. Lorenz, and B. Lückehe. 1999. Short-term forecasting of solar radiation: a statistical approach using satellite data. *Solar Energy* 67, 1 (1999), 139 – 150. https://doi.org/10.1016/S0038-092X(00)00038-4

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *arXiv e-prints*, Article arXiv:1512.03385 (Dec. 2015), arXiv:1512.03385 pages. arXiv:cs.CV/1512.03385

[11] Jane Oktavia Kamadinata, Tan Lit Ken, and Tohru Suwa. 2019. Sky image-based solar irradiance prediction methodologies using artificial neural networks. *Renewable Energy* 134 (2019), 837 – 845. https://doi.org/10.1016/j.renene.2018.11.056

[12] Y KEMMOKU, S ORITA, S NAKAGAWA, and T SAKAKIBARA. 1999. Daily insolation forcasting using a Multi-Stage Neural Network. *Solar Energy* 66, 3 (1999), 193 – 199. https://doi.org/10.1016/S0038-092X(99)00017-1

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[14] Joshua S. Stein Matthew J. Reno, Clifford W. Hansen. [n.d.]. Global Horizontal Irradiance Clear Sky Models: Implementation and Analysis. *Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation* ([n. d.]). https://prod-ng.sandia.gov/techlib-noauth/access-control.cgi/2012/122389.pdf

[15] Adel Mellit and Alessandro Massi Pavan. 2010. A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected PV plant at Trieste, Italy. *Solar Energy* 84 (05 2010), 807–821. https://doi.org/10.1016/j.solener.2010.02.006

[16] Adel Mellit and Alessandro Massi Pavan. 2010. A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected PV plant at Trieste, Italy. *Solar Energy* 84, 5 (2010), 807 – 821. https://doi.org/10.1016/j.solener.2010.02.006

[17] MILA. winter 2020. Orion repository. https://github.com/Epistimio/orion

[18] B. P. Mukhoty, V. Maurya, and S. K. Shukla. 2019. Sequence to sequence deep learning models for solar irradiation forecasting. In *2019 IEEE Milan PowerTech*. 1–6.

[19] Pandas. 2018. Pandas DataFrame. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html

[20] Roland L. Hulstrom Richard E. Bird. 1981. A simplified Clear Sky Model for Direct and Diffuse Insolation on Horizontal Surfaces. *NREL Technical Report* (1981).

[21] Christian Szegedy Sergey Ioffe. 2015. Batch Normalization: Accelerating Deep Network Training by reducing Internal Covariate Shift. *International Conference on Machine Learning (ICML)* (2015).

[22] A. Sfetsos and A.H. Coonick. 2000. Univariate and multivariate forecasting of hourly solar radiation with artificial intelligence techniques. *Solar Energy* 68, 2 (2000), 169 – 178. https://doi.org/10.1016/S0038-092X(99)00064-X

[23] Xingjian SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. 2015. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 802–810. http://papers.nips.cc/paper/5955-convolutional-lstm-network-a-machine-learning-approach-for-precipitation-nowcasting.pdf

[24] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:cs.CV/1409.1556

[25] Wikipedia. [n.d.]. Numerical weather prediction. https://en.wikipedia.org/wiki/Numerical_weather_predictio

[26] Wikipedia. 2017. Solar irradiance. https://en.wikipedia.org/wiki/Solar_irradiance

[27] Wikipedia. 2020. GOES-13. https://en.wikipedia.org/wiki/GOES_13

[28] Wikipedia. 2020. Mean Squared Error. https://en.wikipedia.org/wiki/Mean_squared_error

[29] Haixiang Zang, Lilin Cheng, Tao Ding, Kwok W. Cheung, Zhinong Wei, and Guoqiang Sun. 2020. Day-ahead photovoltaic power forecasting approach based on deep convolutional neural networks and meta learning. *International Journal of Electrical Power Energy Systems* 118 (2020), 105790. https://doi.org/10.1016/j.ijepes.2019.105790

[30] Xin Zhao, Haikun Wei, Hai Wang, Tingting Zhu, and Kanjian Zhang. 2019. 3D-CNN-based feature extraction of ground-based cloud images for direct normal irradiance prediction. *Solar Energy* 181 (2019), 510 – 518. https://doi.org/10.1016/j.solener.2019.01.096