Solar Irradiance Nowcasting with Deep Learning

Himanshu Arora
Mila, University of Montreal

Guillaume Lagrange
Mila, University of Montreal

Swetha Srikari Maganti
Mila, University of Montreal

Maziar Mohammad-Shahi
Mila, University of Montreal

1 Introduction

With a recent increase in renewable energy sources such as solar photovoltaic (PV) systems, the exponential growth of solar energy poses challenges to power systems, mostly due to its uncertain and variable characteristic. The output of photovoltaic systems is highly dependent on solar irradiance, temperature and different weather related parameters. To deal with this issue, one can emphasize the importance of solar forecasting, as the surface solar irradiance has a direct influence on the output of solar power generation. The accurate prediction of solar energy available is important for PV energy plants in order to facilitate the planning and operation of photovoltaic systems, which in turn reduce the operation cost of these electric systems.

Solar irradiance on a surface, represented by the power per unit area (W/m^2) received from the sun, is ideally proportional to the incident sun rays, however external atmospheric factors, primarily clouds or other water particles, can occlude, reflect, refract or diffuse sun rays such that the effective irradiance, or Global Horizontal Irradiance (GHI), measured is much lower. For this reason, the problem of forecasting solar irradiance can be in part viewed as estimating the trajectory of clouds occluding the sky. Of course, other sensor data measuring water vapor or CO_2 may be valuable as well, as larger molecules of water vapor and carbon dioxide can absorb infrared radiant energy.

Capitalizing on the availability of publicly available sensor data, we are interested in a robust predictive model to provide GHI values using GOES-13 (Geostationary Operational Environmental Satellite) satellite imagery. GOES-13 is part of a constellation of satellites measuring atmospheric properties, with a collection of sensors capable of gathering data at multiple wavelengths from the visible spectrum for cloud motion information, to smaller wavelengths where water vapor can be captured. More specifically, we are interested in the task of nowcasting, i.e. predicting the GHI up to 6 hours in the future. For this project, we are interested in the GHI at four timesteps: T_0, T_0+1h, T_0+3h and T_0+6h. In order to have the GHI values associated with specific locations on the satellite imagery, we used the data collected by the 7 SURFRAD stations operated by the National Oceanic and Atmospheric Administration's Earth System Research Laboratory's Global Monitoring Division (NOAA). Additional information regarding the stations can be found in Appendix A. Although these stations collect many different measurements, we are only interested in the global horizontal irradiance. It is also important to note that the SURFRAD GHI values collected were pre-processed and smoothed using moving averages in order to be aligned with the 15-minute acquisition frequency of GOES-13.

Despite a large body of literature on GHI nowcasting or forecasting, the more recent experiences with deep learning techniques still seem to be an active area of research, and there is no definite solution to the problem. Moreover, we observe a lack of frameworks and established metrics to compare the different methods in a robust manner. In this paper, we establish a comparison between our different approaches with Ineichen & Perez's clear-sky model [1], and use the Root Mean Squared Error as the evaluation metric for our multi-output regression task.

In this paper, we explore different approaches to this problem using the geostationary satellite data at hand, and contribute a 3D convolutional neural network architecture that performs significantly better than existing persistent models. In Section 2, we provide an in-depth analysis of this data. Section 3 reviews previous literature related to the subject at hand. Section 4 elaborates the methodology. A discussion of our experimental results is available in Section 5, before the paper concludes in Section 6.

2 Data Analysis

While the amount of data at hand is significant, i.e. GHI measurements collected from the 7 SURFRAD stations and GOES-13 satellite imagery from the year 2010 to the end of 2015, it is also subject to a lot of perturbations. For every timestamp beginning January 1st 2010 until December 31st 2015, we have both the satellite imagery and station measurements at intervals of 15 minutes for a total of 210,336 records. Th GOES-13 images are made available to us in their raw (lossless) NetCDF form in 15-minute chunks, or repackaged in lossy HDF5 archives (8-bit or 16-bit) JPEG2000-compressed in 1-day chunks. In this section, we analyze both data sources to better understand the problem at hand.

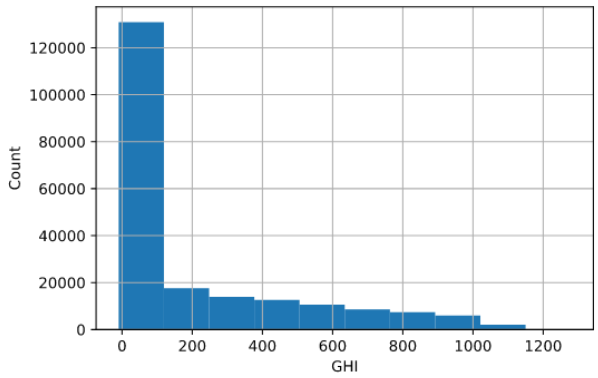


Figure 1: Distribution of GHI measurements for the Table Mountain (TBL) station.

A first glimpse at the GHI values, we noticed that the measurements can be spread across a very wide range of values, going as high as 1200 W/m^2. As shown in Figure 1, the distribution is skewed towards much smaller values, with a high concentration of values closer to zero because of the night times, around which the values are very small starting close to sunset and a little bit after sunrise. Even though we are displaying one of the stations here, this skewed distribution can be similarly observed with all of the SURFRAD stations.

Since the GOES-13 images are captured on a daily basis starting at 8:00 UTC, we do not have any images available for the first 32 timestamps. The satellite imagery is also unavailable at 10 specific timestamps throughout each day, and it can also be unavailable at other times, hence why we have 36,619 records missing their associated image, which is quite significant. On top of that, it is possible for the satellite imagery to have one of its 5 available channels missing, which can also be problematic when it's a channel of information we are using.

Table 1: Missing indicators (measured GHI, clear-sky GHI and cloudiness flag) at the different SURFRAD stations

Station	Target GHI	Clear-sky GHI	Cloudiness
BND	388	236	236
TBL	714	200	200
DRA	1503	304	304
FPK	823	256	256
GWN	5062	2735	2735
PSU	765	537	537
SXF	1321	563	563

Regarding the SURFRAD stations measurements, we see in Table 1 that some of the stations are more reliable than others. The missing values are either at night, which doesn't matter as much, if it's cloudy or otherwise have an undefined cloudiness flag. This is a little bit misleading though, because when looking at the records surrounding the timestamps where the GHI measurement is missing, the cloudiness flag is actually not cloudy. Thus, the cloudiness indicator cannot be trusted for the missing GHI values (`NaNs`).

Furthermore, because the missing values most likely arise from a sensor malfunction or maintenance, we originally noticed that there were multiple days (mostly surrounding the missing values) where almost all of the recorded GHI values were the exact same value, which can sometimes represent up to 96 values per day. These can be considered as corrupted measurements, where the Goodwin Creek (GWN) station was once again the most problematic. These

corrupted values have since been replaced by invalid values (NaNs) in the latest data table (a Pandas DataFrame) made available to us, for which the cloudiness indicator is also invalid as observed in Table 1. A sample illustrating the phenomenon observed in the original DataFrame can be seen in Table 2.

Table 2: Example of corrupted GHI measurements at station BND

iso-datetime	**Cloudiness**	**GHI**
2015-06-10 18:00:00	cloudy	NaN
2015-06-10 18:15:00	cloudy	NaN
2015-06-10 18:30:00	cloudy	NaN
2015-06-10 18:45:00	cloudy	NaN
2015-06-10 19:00:00	cloudy	75.0
2015-06-10 19:15:00	cloudy	75.0
2015-06-10 19:30:00	cloudy	75.0
...	...	...
2015-06-11 00:00:00	cloudy	75.0

In addition to the GOES-13 and SURFRAD stations data, we also have at our disposition the clear-sky model's predicted GHI values, as well as a cloudiness flag and a daytime flag for each of the stations. Although the clear-sky model only accounts for some effects of the atmosphere on solar irradiance, i.e. when the sky is clear of clouds, it is a good comparative baseline. The cloudiness flag is a categorical flag indicating the weather conditions at the station, which is helpful when analyzing the predicted values like the ones from the clear-sky model, and can be either "night", "cloudy", "slightly cloudy", "clear", or "variable". Since we are mostly interested in the daytime predictions, this can help us filter the records.

3 Literature Review

Many of the existing methods still used in practice for weather prediction or solar irradiance forecasting rely on physical and mathematical models, with strong assumptions on functional dependency of position and time. A popular type of model among these are clear-sky models. A clear-sky model is a grouping of formulae that are capable of producing an estimate of the solar irradiance arriving at a specific location. These models must be capable of accounting for the scattering, reflection and absorption that occurs due to atmospheric constituents such as water vapor, and atmospheric processes such as turbidity. The clear-sky model proposed by Ineichen & Perez [1] takes into account the zenith angle, air mass, atmospheric turbidity (Linke Turbidity) and elevation. It serves as a good comparative baseline when evaluating our different models, and can potentially serve as a complement to our model. Obviously, its main drawback is that it can predict the solar irradiance under the condition that the sky is not cloudy, and is therefore very unreliable under unpredictable weather circumstances.

More recently, advances in deep learning have been leveraged for solar irradiance nowcasting and forecasting, as well as weather forecasting, a task closely related to ours. Zhao *et al.* [4] proposed a 3D ConvNet to process multiple consecutive ground-based cloud images in order to extract cloud features including texture and temporal information. As opposed to the traditional 2D ConvNet architectures used to process static images like SolarNet by Feng & Zhang [5], which extracts features of a static sky image for 10-minute ahead GHI predictions, the proposed 3D ConvNet architecture of the authors makes use of multiple consecutive ground-based cloud images to be able to extract cloud motion information. This spatiotemporal information has been proven to be one of the main influence factors for Direct Normal Irradiance (DNI), which, along with Diffuse Horizontal Irradiance (DHI), is one of the components that form the Global Horizontal Irradiance (GHI).

In their paper, Siddiqui *et al.* [6] proposed a 2D convolutional neural network with fully-connected LSTM (*CNN+LSTM*) based architecture that capitalizes on auxiliary weather information such as air temperature, wind speed, relative humidity and barometric pressure for solar irradiance nowcasting. More specifically, they are interested in the solar irradiance nowcasting at the instance the frame is captured,as well as ahead-of-time irradiance predictions for a duration for up to 4 hours. Even though we do not have direct access to such auxiliary data, their approach is still interesting, with their first stage LSTM using the convolutional neural network features learned from sky-videos also being very much related to the ConvLSTM architecture proposed by Shi *et al.* [7]. In order to model the spatiotemporal relationships, Shi *et al.* [7] extend the idea of fully-connected LSTMs to ConvLSTM for precipitation nowcasting, which has convolutional structures in both the input-to-state and state-to-state transition. The main difference with the CNN+LSTM architecture proposed by Siddiqui *et al.* [6] and ConvLSTM [7] is that the CNN+LSTM architecture represents a two-step process: the 2D convolutional neural network acts as a spatial feature extractor, whose features

are multiplied by the LSTM cell to learn the temporal features. On the other hand, the core idea behind ConvLSTM is to replace fully-connected operations in the step-to-step transitions by convolution operations (2D states) [10], which captures underlying spatial features by convolution operations in multiple-dimensional data.

Following their proposed ConvLSTM [7], the authors expand their idea to propose *Trajectory Gated Recurrent Unit* (TrajGRU) [9], a model that can actively learn the *location-variant* structure for recurrent connections. It improves on previous ConvRNNs (ConvLSTM [7], ConvGRU [8], etc.), which apply a *location-invariant* filter to the input when used for capturing spatiotemporal correlations since the connection structure and weights are fixed for all the locations. As explained by Shi *et al.* [9], the local correlation structure of consecutive frames isn't the same for different locations and timestamps with motion patterns like rotation and scaling, which makes the use of convolutions inefficient to represent such *location-variant* relationship because of their *location-invariant* filter.

Interestingly enough, the authors in [4], [5], and [6] use the same NREL provided database [2] of Total Sky Imager (TSI) images for their task. Sky-camera image sources seem particularly useful, as they encode both incident light and atmospheric behaviour such as cloud and suspended particles. On top of the the TSI images recorded from the NREL's Solar Radiation Research Laboratory (SRRL) [2], Siddiqui *et al.* [6] also evaluate their proposed CNN+LSTM on a second dataset recorded at the Multiple Mirror Telescope Observatory (MMTO) [3], which uses a different camera to record the sky videos.

4 Methodology

4.1 Data Processing & Pipeline

Since the data available is often imperfect, it requires some pre-processing to improve the performance of the model. With two different data sources, we have to handle noise or invalid values both for our input images and target GHI values. Given the task at hand, we decided to skip any target datetime where T_0 is at night since our objective is to predict day time GHI values. For both training and evaluation, we also ignored the examples for which any of the target timesteps have invalid clear-sky prediction values (`NaN`s) since we imputed the clear-sky model's prior knowledge to our targets for some of our experiments. This is not a big issue since we have such a large number of examples available.

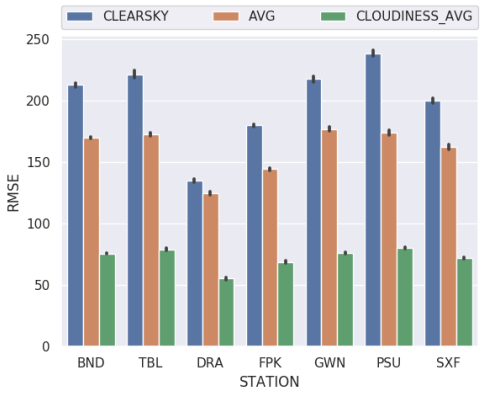


Figure 2: Evaluation of the different missing-data imputation methods for GHI values across the different stations.

Given the number of missing GHI values for daytime, we assessed three different imputation methods by randomly deleting a number ($n = 10000$) of the recorded values for each station from the dataset, and then evaluated how well the deleted values were predicted with the RMSE as shown in Figure 2. For a missing GHI value at timestamp T on a particular day of a month M, the `CLEARSKY` method replaces the missing value with the corresponding clear-sky GHI at the given timestamp T. The `AVG` method replaces the value with the average of GHI values at timestamp T of over the month M, while the `CLOUDINESS_AVG` method replaces the value with the average of GHI values at timestamp T for all days of the month M with the same cloudiness flag (similar weather conditions). Although the last method proposed is most accurate, the cloudiness flag cannot be trusted for the data at hand, on top of some of the missing GHI values having an invalid cloudiness flag (`NaN`) altogether. The first method could also be problematic as the latest data table has some missing clear-sky predictions for the same missing GHI measurements, so we used the `AVG` method as our replacement strategy.

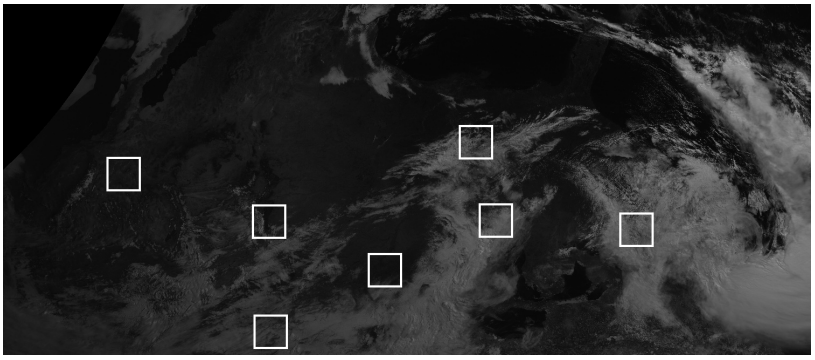


Figure 3: Example input for the visible spectrum with a patch size of 64 around the regions surrounding the SURFRAD stations.

Since we are interested in GHI values at SURFRAD stations, and each GOES-13 image covers the whole continental US, we decided to crop regions centered on the coordinates of those stations, which is a hyperparameter that can be changed. An example input is illustrated in Figure 3. The images also have five different channels available, with each channel providing information regarding different wavelengths. In some cases, there is a possibility of some channels missing, in which case we filled the missing channels with zeros. Since we want to predict a sequence of GHI values ahead of time, we also used past and present satellite frames, which help capture the multi-channel and multi-temporal aspect of the problem. This is usually referred to as the input sequence in the paper, which represent the past and present information useful for our task. All input images are normalized (per-image normalization) to have values in the range [0, 1]. We used the 8-bit compressed images for all of our experiments.

With a total size of hundreds of GBs of GOES-13 image data even in its compressed form, we also had to come up with an efficient data loading and processing pipeline, which is an important part of training any large-scale deep learning system. Between the various solutions used as the project went on, we had two efficient solutions available at hand: pre-processing the input images for the given patch size around the SURFRAD stations, saving a single file for each timestamp that we could load in memory during training, or batching the sequence of 256 images together along with other metadata needed for training (e.g. the clear-sky GHI) into separate HDF5 archive files. While the latter minimized input-output operations with less files, one could also use the first approach when a sufficient amount of RAM was available given the data used for the experiments. Both approaches could also be combined with Tensorflow's other optimization strategies.

4.2 Model Architectures

4.2.1 2D Convolutional Neural Network

The idea behind this architecture is to concatenate past and present images to have an input with the shape $H \times W \times (C \cdot T)$, where C is the number of channels and T the number of past and present images used. The architecture of our ConvNet is as follows: 2 convolution layers with 32 3×3 filters followed by a 2×2 max-pooling layer with dropout (0.25), then 2 convolution layers with 64 3×3 filters followed again by a 2×2 max-pooling layer with dropout (0.25), then a fully-connected layer of 128 units with dropout (0.5) connected to the final fully-connected layer with 4 outputs. All activation functions before the output layer in the network are ReLUs.

4.2.2 Convolutional LSTM

Based on the promising results presented by Shi *et al.*, we briefly experimented with a ConvLSTM architecture. The architecture is composed of three ConvLSTM layers, each with 64 3×3 filters followed by a batch normalization layer. The extracted features are then spatially reduced by a 4×4 max-pooling layer, and further reduced to a vector of size 128 by a fully-connected layer with ReLU activation. Finally, a fully-connected layer is used to obtain a final vector for our 4 nowcasting targets.

4.2.3 CNN+LSTM

Our CNN+LSTM architecture was greatly inspired by the works of Siddiqui *et al.* [6]. The first input convolution has 64 3×3 filters and reduced by a max-pooling layer of size 2×2 with stride (2,2). The model then has two convolution layers of 128 filters, and a further 2 layers of 256 filters, all of size 3×3. Each layer is followed by a

2×2 max-pooling layer just like the first convolution. All convolutions use a ReLU activation, are immediately followed by a batch normalization layer and are applied with appropriate padding and stride 1, thus there is no change in terms of size from the input to the output of these convolution layers. The extracted features are then reduced to a vector of size 512 by a fully-connected layer with ReLU activation, which are used as input for the long short term memory recurrent neural network (LSTM) learn a 128 vector representation of the historical frames. Finally, a fully-connected layer with dropout preceding it (0.5) is used to obtain a final vector for our 4 nowcasting targets.

4.2.4 3D Convolutional Neural Network

Inspired by the architecture proposed by Tran *et al.* [11] for various video classification tasks, we present a very similar 3D ConvNet architecture for our solar irradiance nowcasting task. The network is composed of three 3D convolution layers and three 3D max-pooling layers (each convolution layer has a ReLU activation and is immediately followed by batch normalization and max-pooling layer), a fully-connected layer of size 512 with ReLU activation and a batch normalization layer, and finally a fully-connected layer with dropout preceding it (0.5) is used to obtain a final vector for our 4 nowcasting targets. The number of filters for the convolution layers gradually increase from 64, 128 to 256 respectively. All convolution kernels have a homogeneous size of $3 \times 3 \times 3$, and all convolution layers are applied with appropriate padding (both spatial and temporal) and stride 1, thus there is no change in terms of size from the input to the output of these convolution layers. All max-pooling layers have a kernel size of $2 \times 2 \times 2$ with stride 1, except for the first layer which has a kernel size of $1 \times 2 \times 2$ with the intention of not merging the temporal signal too early.

4.3 Experiments

In order to have reproducible and comparable results, we set the random state (seed) to 128 for all of our experiments, with which the training set is always shuffled. For all of our experiments, we used the Mean Squared Error (MSE) loss function, with the RMSE as training and validation metric. We used the Adam optimization algorithm, as it has been proven to converge faster with its adaptive procedure compared to stochastic gradient descent (SGD). The default hyperparameters were maintained for the optimizer (learning rate = 0.001, beta_1 = 0.9, beta_2 = 0.99). All experiments were ran between 5 to 20 epochs based on the convergence of the models, with a batch size of 32 when using a small subset of the data, or either 128 or 256 for all of the data. Because of the lack of resources and time, the hyperparameter tuning was done manually when experimenting on a subset of the data, in an iterative fashion, which could be considered as a manual random search if such a thing has been previously defined. We experimented with different model architectures, previous sequence lengths, cropped patch sizes around the SURFRAD stations and target types. We always used all of the five available channels from the satellite imagery in our inputs.

Table 3: Different present and past image sequences used as inputs

	Input 1	**Input 2**	**Input 3**	**Input 4**	**Input 5**	**Input 6**	**Input 7**
Sequence 1	T_0	T_0- 0.5h	T_0- 1h	T_0- 1.5h	T_0- 2h	–	–
Sequence 2	T_0	T_0- 1h	T_0- 2h	T_0- 3h	T_0- 4h	–	–
Sequence 3	T_0	T_0- 1h	T_0- 2h	T_0- 3h	T_0- 4h	T_0- 5h	T_0- 6h

We initially tried a patch size of 82, but quickly reverted to 64 and 32 for all the experiments presented in this paper. For the target values, the measured GHI values at SURFRAD stations are used. Because the clear-sky model's predictions can always be obtained for a specific timestamp and location, we also tried to take advantage of that by imputing the clear-sky model's prior knowledge to our targets, using the difference between the measured GHI values and the predicted clear-sky values as our targets.

For quick assessment of the different architectures, patch sizes, sequence lengths and target values, we randomly selected 10000 timestamps between `2010-04-15` and `2011-04-15`, yielding a total of 34k examples for our initial experiments. For the validation set, we randomly selected 13 days throughout the year 2012 (across the different months of the year) for a total of 3.7k examples. As a comparative baseline, we evaluated the clear-sky model predictions for each target timestep, giving us a baseline RMSE of 185.64 on the training set and 159.17 on the validation set. Our validation set is easier than average, but it is still a fair baseline for our preliminary experiments to help us evaluate the different architectures and hyperparameters.

For our final experiments, we trained using the 2010-2013 examples, with the 2014 data for validation and the examples from 2015 for our test set. Based on the performances on the 2015 test set, we trained our best model on more data, using 2010-2014 for our final submission, which used the year 2015 as a validation set.

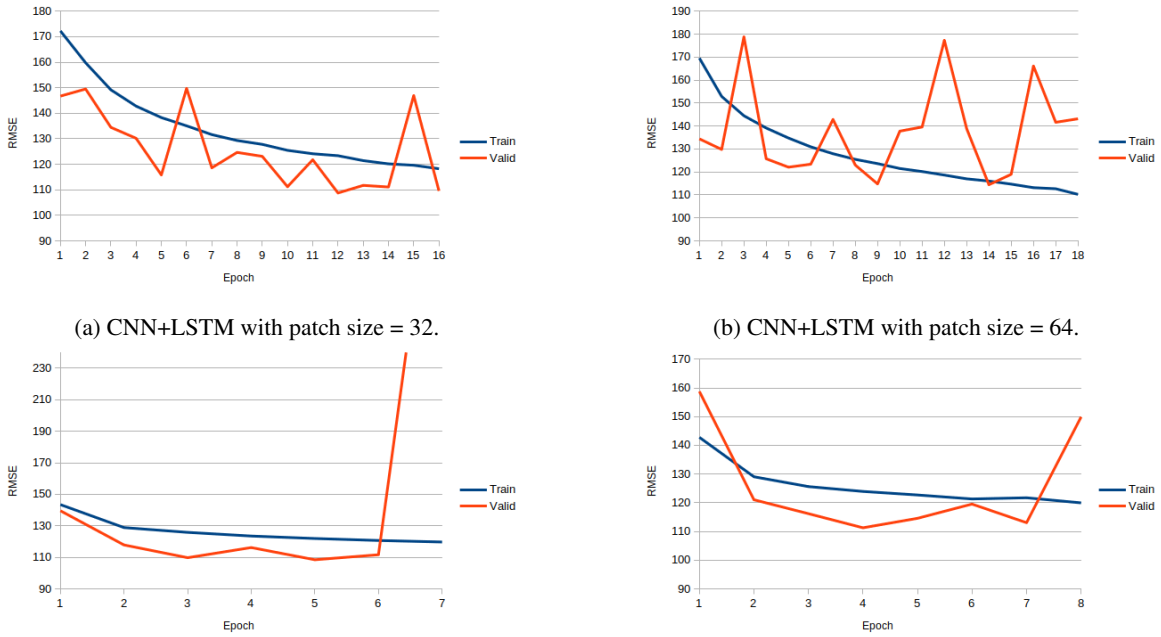
5 Results and Discussion

We first present the results of our various preliminary experiments on a subset of the data to justify the choice of our final models. Then, we present our final experiment results on the held-out test set as specified in Section 4.3. The results of our experiments are all presented using only the RMSE metric and not the MSE loss, since the loss follows the exact same trend, only with much higher values.

5.1 Preliminary Experiments

As mentioned previously, we first experimented with different model architectures, cropped patch sizes around the SURFRAD stations, sequence lengths and target types on a subset of the data to quickly iterate over the best configurations.

Between the different architectures presented in Section 4.2, we wanted to asses how each of them compared with similar hyperparameters: input image sequence, patch size, and target types. Before trying out the proposed CNN+LSTM architecture, we first started experimenting with our ConvLSTM architecture. This approach was quickly abandoned, as it was more difficult and expensive to train just on a subset of the data, and clearly overfitted our training set starting as soon as the first epochs. Perhaps using more data could have helped, but we could not even begin to imagine how long it would take to train on all of the dataset, and with the limited resources and time at our disposition we preferred to explore the other methods more thoroughly. Thus, although we initially experimented with the architecture, we will not present any results regarding this architecture as none of them were conclusive. For this reason, we switched to a simpler CNN+LSTM architecture, which we compared with our 3D ConvNet architecture.



(a) CNN+LSTM with patch size = 32.

(b) CNN+LSTM with patch size = 64.

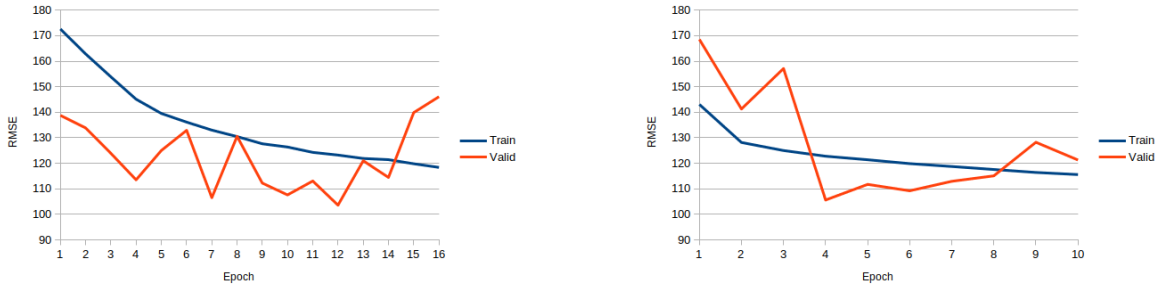
(c) 3D ConvNet with patch size = 32.

(d) 3D ConvNet with patch size = 64.

Figure 4: Train and validation RMSE for our CNN+LSTM and 3D ConvNet using Sequence 2 inputs (Table 3) and difference between clear-sky predictions and measured GHI as targets.

We first experimented using only the measured GHI at the SURFRAD stations as our targets for the task, trying out a patch size of 64 with an input sequence up to 2 hours behind (Sequence 1 in Table 3), or a patch size of 32 with an input sequence up to 4 hours behind. As illustrated by the results in Appendix D, we could difficultly achieve results similar to our clear-sky model baseline on the validation set (159.17). We tried to add more capacity to our models by increasing the number of fully-connected layers, but without success. For that reason, we decided to take advantage of the encoded information from the clear-sky model's predictions. Since this model takes into account different parameters that vary throughout the year (seasonality), we found that using the difference between the measured GHI values and the predicted clear-sky values as our targets to be extremely helpful to our models, as the input image sequences capture information (e.g. cloud motion) that is complementary to the clear-sky predictions.

Between both our 3D ConvNet and CNN+LSTM architecture, we can see in Figure 4 that for the same input sequence that uses current and past imagery up to 4 hours behind, using a bigger region of the image around the SURFRAD stations is not as impactful as one might think. Considering the fact that one pixel in the satellite imagery represents a surface of approximately 16 km^2, it might actually be detrimental to use an even bigger region around the stations because we might capture unwanted information that is not yet relevant to the specific points on the map we are interested in. In fact, using a smaller patch size of 32 usually yielded slightly better results, and to our advantage this is also more efficient in terms of training time, which also transfers for inference time. We also noticed that in both cases, the 3D ConvNet starts to overfit after 6 or 7 epochs, while the CNN+LSTM just seemed less stable overall. Overfitting our models to a smaller amount of data used in our preliminary experiments can actually be a desired factor, since using practically all of the data available for our final experiments will almost always yield a model that is better at generalizing to unseen data. The more examples of the same distribution, the closer the model predictions will get to that of the population.



(a) CNN+LSTM with patch size = 32.

(b) 3D ConvNet with patch size = 32.

Figure 5: Train and validation RMSE for our CNN+LSTM and 3D ConvNet using Sequence 3 inputs (Table 3) and difference between clear-sky predictions and measured GHI as targets.

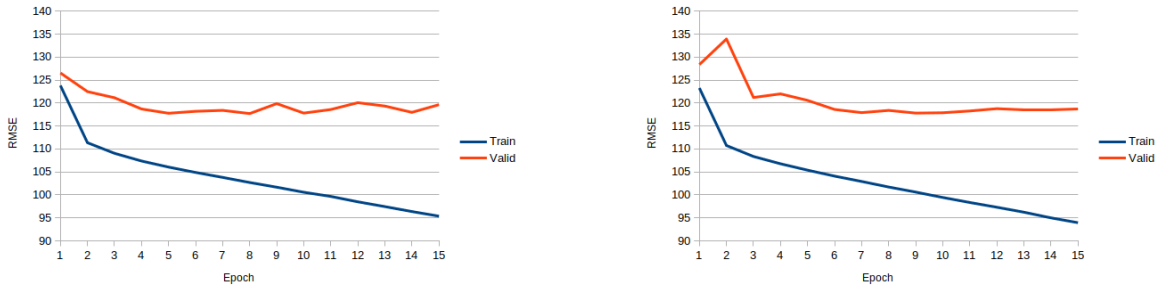
Next, we wanted to compare different input sequence lengths, looking back as far as 6 hours behind. Intuitively, since we are interested in predicting up to 6 hours ahead, we thought that perhaps capturing the temporal information for the same time window might help even more. As illustrated in Figure 5, where we can see the overall validation RMSE is slightly lower than in the plots of Figures 4c and 4a, our hypothesis seemed to have been confirmed. From Figure 5b, we also noticed that the 3D ConvNet seemed to generalize better when given a bigger input sequence as it does not seem to overfit, or at least not as quickly. Once again, this seemed to indicate that capturing the temporal information for the same time window as our predictions is helpful. This could also just be the case for the subset of the data we are using, which is why we experimented with both sequences for our final experiments presented in the following section.

We also compared the previously obtained results to our simple Conv2D architecture. Using the same configuration, i.e. a patch size of 32 and using images up to 4 hours behind, we found that it actually performed reasonably well, but plateaued due to its limited capacity as seen in Appendix E.

5.2 Final Results

Given the results found by our preliminary experiments, which all beat the clear-sky baseline established, we were able to efficiently converge to our best architecture with different hyperparameter configurations. As demonstrated in Section 5.1, using a smaller patch size of 32 produced better results while being faster to train, and our 3D ConvNet consistently performed better on the validation set than the CNN+LSTM, which was a bit more difficult to train to obtain stable results. For these reasons, we decided to use our two best 3D ConvNet configurations for our final experiments on the whole dataset splits, both using the difference between the clear-sky predictions and measured GHI as targets. For this, we used the data from 2010 to the end of 2013 for training, the samples from 2014 for validation and finally the examples from 2015 for our test set.

As illustrated by the curves in Figure 6, both models once again performed similarly, with the first one using an input sequence up to only 4 hours behind getting slightly better results on both the validation and test set as seen in Table 4. Both models seemed to plateau after just a couple of epochs, so we selected the best version of each with early stopping. As shown in Table 4, both of our proposed methods performed significantly better compared to our initial clear-sky baseline, which was expected given that our model is actually predicts the difference between the measured GHI and the clear-sky predictions, using the clear-sky values to get the final predicted GHI.



(a) Using Sequence 2 inputs (Table 3).

(b) Using Sequence 3 inputs (Table 3).

Figure 6: Train (2010-2013) and validation (2014) RMSE of our best 3D ConvNet configurations using the whole dataset.

Table 4: Final evaluation of our proposed 3D ConvNets

Model	**Input**	**Early stopping**	**Train RMSE** (2010-2013)	**Valid. RMSE** (2014)	**Test RMSE** (2015)
3D ConvNet	Sequence 2	Epoch 8	102.68	**117.69**	**109.63**
3D ConvNet	Sequence 3	Epoch 9	**100.59**	117.80	109.88
Clear-sky	–	–	187.52	197.79	194.00

Given the results presented in Table 4, we trained our proposed 3D ConvNet with an input sequence going back up to 4 hours (Sequence 2, as described in Table 3) on the years 2010 to 2014 for our final submission, using 2015 as our validation set and early stopping at the 10^{th} epoch to select the best version. The results for this final experiment can be found in Appendix B, with some visualizations of different activation maps of the network for a cloudy image in Appendix C.

6 Conclusion

After performing a lot of experiments using various model architectures, hyperparameters, pre-processing techniques and feature selection, we conclude that our 3D ConvNet model generalizes well to unseen data given its performance on the test set. We expect that good generalization on our test data will transfer to data with similar distribution for any year, including the year 2016 in the blind test set. We also expect our model to perform well regions outside of the SURFRAD stations. The proposed 3D ConvNet was trained using only the satellite images as an input, using the difference between the clear-sky model predictions and measured GHI for our targets. Given that the clear-sky model predictions can be obtained for any given position, this implies that, unless the images are of a different distribution (e.g. with a different terrain or cloud traversals), the model should generalize well to other locations.

The work presented was limited by the time and resources available. Since a major part of the time was spent on building an efficient data loading pipeline, there are a few recommendations that could be followed to enhance the performance of the models. For example, with the CNN+LSTM, we could have tried using various regularization techniques to get more stable validation results, or even reduced the learning rate (although Adam is usually well suited as it is adaptive). Futhermore, our initial ConvLSTM approach could also be revisited using the whole dataset to see how well it would compare to our 3D ConvNet as it has been shown to perform well on similar tasks. One could also explore a more complex Conv2D architecture, as it seemed to generalize fairly well. To further improve our overall results, future works could explore different data augmentation techniques as no data augmentation was used in this project, and can help with better generalization. One could also get a small improvement by using SGD instead of Adam as the chosen optimizer since it has been proven to achieve better performance in the long run.

References

[1] Ineichen, P. & Perez, R. (2002). *A new airmass independent formulation for the Linke turbidity coefficient*. (doi:10.1016/S0038-092X(02)00045-2)

[2] Stoffel, T. & Andreas, A. (2015). *NREL Solar Radiation Research Laboratory (SRRL): Baseline Measurement System (BMS)*. Golden, Colorado (Data). National Renewable Energy Laboratory. (doi:10.7799/1052221)

[3] Pickering, T. (2006). *The mmt all-sky camera*. In SPIE Astronomical Telescopes+ Instrumentation, pages 62671A–62671A. International Society for Optics and Photonics.
[4] Zhao, X., Wei, H., Wang, H., Zhu, T. & Zhang, K. (2019). *3D-CNN-based feature extraction of ground-based cloud images for direct normal irradiance prediction*. (doi:10.1016/j.solener.2019.01.096)
[5] Feng, C. & Zhang, J. (2020).*SolarNet: A Deep Convolutional Neural Networkfor Solar Forecasting via Sky Images*. The University of Texas at Dallas.
[6] Siddiqui, T.A., Bharadwaj, S. & Kalyanaraman, S. (2019). *A deep learning approach to solar-irradiance forecasting in sky-videos*. (arXiv:1901.04881)
[7] Shi, X., Chen, Z., Wang, H, Yeung, D-T., Wong, W-k. & Woo, W-c. (2015). *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting*. Part of: Advances in Neural Information Processing Systems 28 (NIPS 2015). (arXiv:1506.04214)
[8] Ballas, N., Yao, L., Pal, C. & Courville, A. (2015). *Delving deeper into convolutional networks for learning video representations*. In ICLR, 2016. (arXiv:1511.06432)
[9] Shi, X., Gao, Z., Lausen, L., Wang, H., Yeung, D-T., Wong, W-k. & Woo, W-c. (2017). *Deep Learning for Precipitation Nowcasting: A Benchmark and A New Model*. Part of: Advances in Neural Information Processing Systems 30 (NIPS 2017). (arXiv:1706.03458)
[10] Tran, Q-K. & Song, S-k. (2019). *Computer Vision in Precipitation Nowcasting: Applying Image Quality Assessment Metrics for Training Deep Neural Networks*. (doi:10.3390/atmos10050244)
[11] Tran, D., Bourdev, L., Fergus, R., Torresani, L. & Paluri, M. (2014). *Learning Spatiotemporal Features with 3D Convolutional Networks*. (arXiv:1412.0767)

Appendix

A SURFRAD Stations

Name	Code	Latitude	Longitude	Elevation
Bondville, IL	BND	40.05°N	88.37°W	230m
Table Mountain, CO	TBL	40.13°N	105.24°W	1,689m
Desert Rock, NV	DRA	36.62°N	116.02°W	1,007m
Fort Peck, MT	FPK	48.31°N	105.10°W	634m
Goodwin Creek, MS	GWN	34.25°N	89.87°W	98m
Penn. State University, PA	PSU	40.72°N	77.93°W	376m
Sioux Falls, SD	SXF	43.73°N	96.62°W	473m

B Final Submission

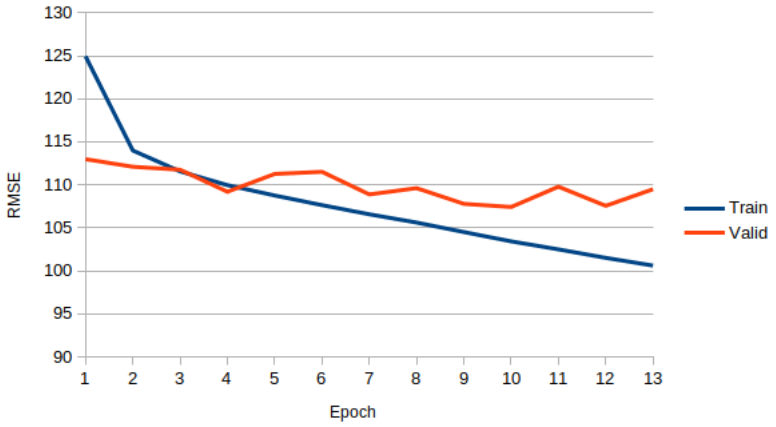


Figure 7: Final submission train (2010-2014) and validation (2015) RMSE of our best 3D ConvNet configuration using Sequence 2 inputs (Table 3).

Table 5: Evaluation of the proposed 3D ConvNet submitted

Model	Input	Early stopping	Train RMSE (2010-2014)	Valid. RMSE (2015)
3D ConvNet	Sequence 2	Epoch 10	**103.42**	**107.42**
Clear-sky	–	–	189.90	194.00

C 3D ConvNet Layer Activations

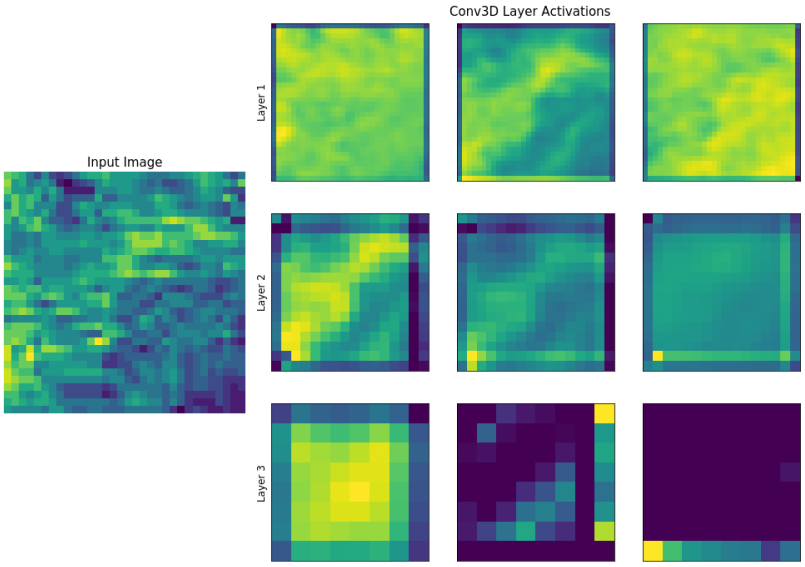
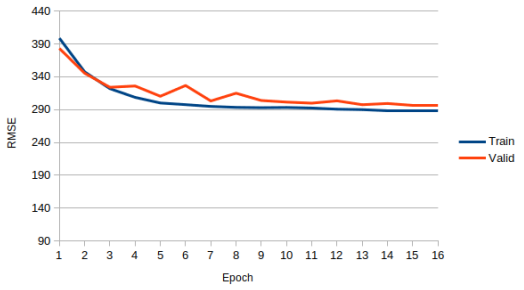
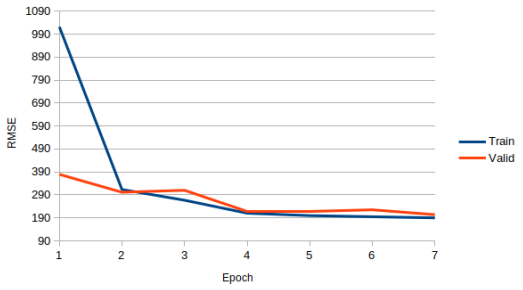


Figure 8: Random activation samples from 3D ConvNet layers for an image

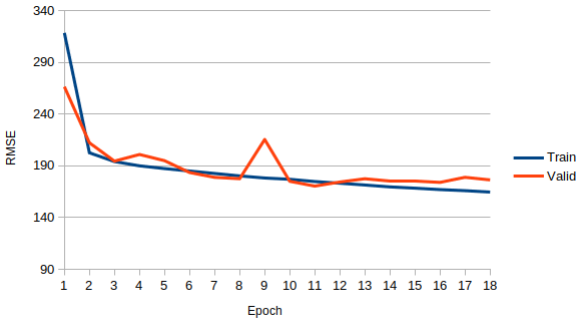
D Additional Experiments Using Measured GHI as Target



(a) CNN+LSTM using Sequence 1 inputs (Table 3). Patch size = 64.



(b) 3D ConvNet using Sequence 1 inputs (Table 3). Patch size = 64.



(c) 3D ConvNet using Sequence 2 inputs (Table 3). Patch size = 32.

Figure 9: Train and validation RMSE for our different experiments using the measured GHI as out targets.

E 2D ConvNet Experiment

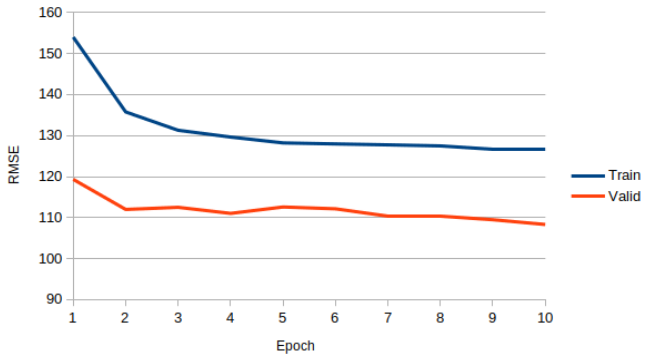


Figure 10: Train and validation RMSE for our comparative 2D ConvNet experiment Sequence 2 inputs (Table 3). Patch size = 32. Targets: difference between measured GHI and clear-sky prediction.