Solar Irradiance Prediction using Deep Neural Networks

Philippe Beardsell
Université de Montréal et MILA
20139766

Jérôme Labonté
Université de Montréal et MILA
20144437

Marie St-Laurent
Université de Montréal et MILA
657930

1 Introduction

1.1 Summary

Global Horizontal Irradiance (GHI), the effective amount of solar irradiance that reaches an horizontal surface on planet Earth, determines the quantity in W/m^2 of solar power per surface area available to photovoltaic installations (i.e., solar panels). Accurate and reliable short-term GHI forecasting (or "nowcasting") is essential for solar energy providers to balance energy demand and supply effectively. While on-the-ground stations gather precise GHI measurements that can inform predictive models, local measurements require costly, maintenance-heavy infrastructure and only offer short-range coverage of limited scalability. To predict GHI across broad swaths of territory requires the kind of coverage offered by satellite imagery. For example, satellites positioned in the Earth's geostationary orbit to match its rotation period provide global uninterrupted coverage by locking onto a geographic point along the equator [Wikipedia, a]. While satellites do not measure GHI per se, they capture meteorological phenomena that can be used to infer and anticipate GHI.

For the current project, we trained deep neural networks to nowcast GHI based on geostationary satellite imagery of the American East Coast (GOES-East; NOAA). Predictions were evaluated against ground truth GHI measurements obtained from seven SURFRAD on-the-ground stations [SURFRAD]. Temporal sequences of geostationary satellite images were used to predict GHI at the moment the last image was captured (T0), as well as 1, 3 and 6 hours into the future (T1, T3 and T6). Our models were inspired by recent advances in deep-learning approaches to nowcasting meteorological phenomena like GHI, precipitations and storm formation. Models tested included 2-dimensional and 3-dimensional convolutional neural networks (CNN-2D and CNN-3D, respectively), a recurrent CNN (RNN-CNN), and a convolutional Long-Short-Term Memory network (ConvLSTM). Performance was evaluated against baseline predictions derived from a "clear sky" model that reflected an upper bound GHI expected under cloudless conditions [Ineichen and Perez, 2002]. Our best results were obtained with CNN-3D architectures that applied convolutions along the spatial and temporal dimensions of image sequences.

1.2 Related Work

Several recent efforts have been made to nowcast meteorological phenomena by modeling radar or satellite imagery data with deep learning techniques. Models differ in how they extract spatio-temporal information from sequences of images. While some models rely on simpler neural net architectures (e.g., Ameen et al. [2019]), most models typically extract spatial relationships among neighboring pixels using some type(s) of convolutional operations. For example, Siddiqui et al. [2019] processed sky-video images to predict solar irradiance with a slight variation on the first 3 blocks of

IFT-6759 Project 1 Report (Winter 2020), Université de Montréal, Canada.

VGG16 [Simonyan and Zisserman, 2014], a convolutional neural network (CNN) well-known for its good performance on image recognition tasks. Other groups have used a U-Net architecture, whose downsampling segment transforms input images into dense low-dimensional representations through layers of pooling and convolution, to output precipitation maps based on geostationary satellite [Lebedev et al., 2019] or radar [Agrawal et al., 2019] images.

While convolutions capture spatial image features predictive of immediate (T0) phenomena, short-term forecasting benefits from modeling change over sequences of images. Some models achieve this by adding a temporal dimension to 2-D image features with a 3-dimensional convolutional neural network (CNN-3D). For example, Zhao et al. [2019] have used a CNN-3D to predict direct normal irradiance, a component of GHI. Change over time can also be modeled using optical flow estimation algorithms (e.g., by estimating how wind direction determines the movement of precipitation fields Lebedev et al. [2019]), or by integrating CNNs into recurrent neural network architectures.

Mathe et al. [2019]'s PVNeT transforms each image from a sequence into a dense vectors through layers of convolution before processing their temporal structure through a Long-Short-Term-Memory (LSTM) model to nowcast GHI. Similarly, Siddiqui et al. [2019] processed images and auxiliary data in parallel with a two-tiers LSTM to predict solar irradiance. Zhang et al. [2019] also processed multi-source images with a CNN-3D before inputting the results into an two-layer LSTM to predict convective storm formation.

Another hybrid approach is the Convolutional Long-Short-Term-Memory (ConvLSTM) network introduced by Shi et al. [2015] to nowcast precipitations. ConvLSTM imposes a convolutional structure to an LSTM's input-to-state and state-to-state transitions, effectively stacking layers of recurrent convolutional layers on top of one another. This model has been adapted successfully by others to nowcast cloudage [Tan et al., 2019] and precipitations [Kim et al., 2017]. ConvLSTM's newer sibling, Trajectory Gated Recurrent Unit (TrajGRU; [Shi et al., 2017]), determines recurrent connections dynamically : it adjusts convolution neighborhoods per location and time stamp, based on current input and previous state, to capture location-variant transformations in the precipitation field.

Nowcasting performance has also been shown to benefit from complementing satellite or radar image time-series with observed or predicted auxiliary data. Such data can include measured temperature, humidity, solar altitude, or wind speed, or atmospheric or solar irradiance predictions, as shown by Mathe et al. [2019], Lebedev et al. [2019] and Siddiqui et al. [2019]. Auxiliary data and image features can be processed by parallel streams before being merged and passed through fully connected layers (e.g., Siddiqui et al. [2019]). Alternatively, they can be merged at the input stage at each time point so that time-specific metadata can nuance the information extracted from its corresponding image feature, for example by adding a metadata channel to each image as done here [Silver et al., 2017]. The models included in the current report combine characteristics from the models described above, including their approach to extract spatio-temporal information and to integrate metadata with image features.

2 Methods

2.1 Data Sources

GOES13 data To predict GHI, input features were derived from Geostationary Operational Environmental Satellite (GOES)-13 imagery made available from the US Department of Commerce's National Oceanic and Atmospheric Administration (NOAA). GOES 13 was the operational weather satellite for GOES-East from April 2010 to December 2017 [Wikipedia, b]. Its multiple sensors captured data at multiple wavelength bandwidths sensitive to red (visible spectrum) and infrared (invisible spectrum) light, water vapors, and CO2 emissions. Each image pixel corresponds to $4km^2$ of territory.

For training, satellite images from five channels [500-750 ηm, 3800 to 4000 ηm, 5800 to 7300 ηm, 10,2 to 11,2 μm and 3,0 to 13,7 μm] acquired every 15 minutes were available for the years of 2010-2015, inclusively. Raw NetCDF imagery was repackaged into one-day chunks of JPEG-compressed HDF5 8-bit files to facilitate storage and rapid access. A JPEG2000 16-bit version of the data was also available but due to time constraints and limited processing resources, we only used the 8-bit

version. Future works should definitely include running the best algorithms on the 16-bit imagery data to determine the effects of image quality on GHI predictions.

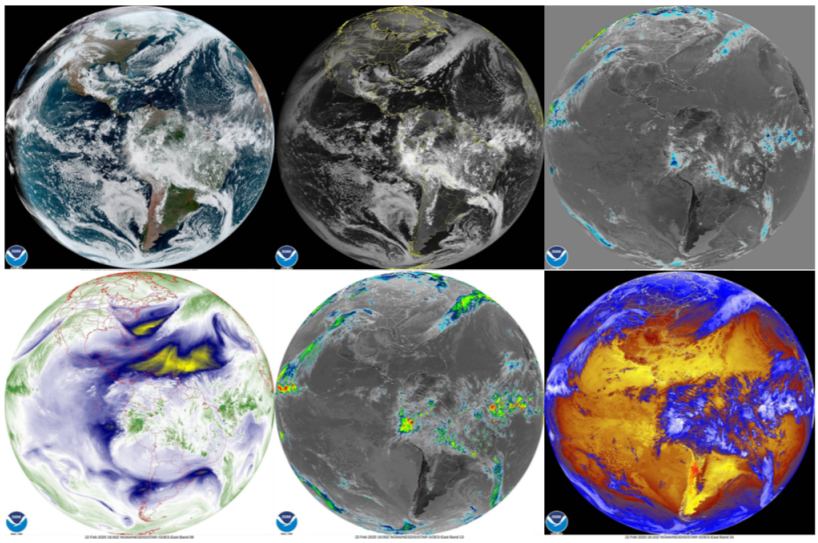


Figure 1: GOES-EAST images per channel, 22 Feb 2020 @16:00. Top: True color, band 2 (0.64ηm, red visible) and band 7 (3.9ηm, shortwave window - IR). Bottom: band 8 (6.2ηm, upper-level water vapor - IR), band 13 (10.3ηm, clean longwave window - IR) and band 16 (13.3ηm, CO_2 longwave - IR). IR = infrared. Source: https://www.star.nesdis.noaa.gov/

SURFRAD ground station data Ground truth GHI measures were available from seven Surface Radiation Budget Network [SURFRAD] ground stations located across the United States (Bondville, IL; Table Mountain, CO; Desert Rock, NV; Fort Peck, MT; Goodwin Creek, MS; Penn. State University, PA; and Sioux Falls, SD). SURFRAD stations collect precise ground-based measurements to help refine and verify satellite-based surface radiation estimates. These measurements are taken every minute but only the timestamps aligned with the GOES-13 images (every 15 minutes) were used as targets in our dataset.

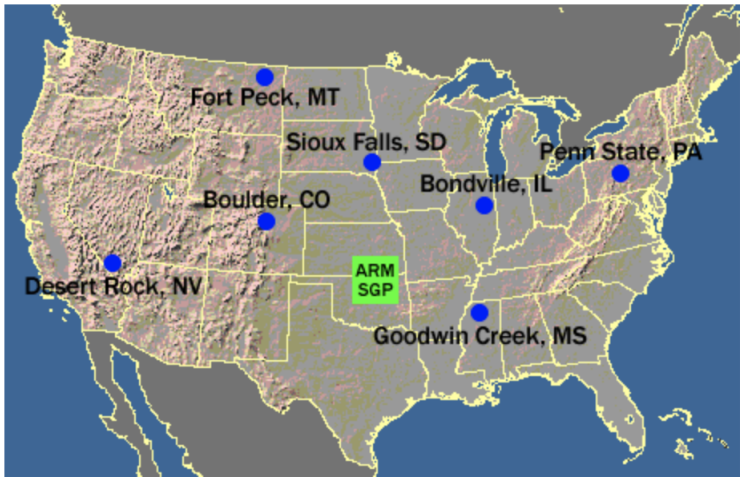


Figure 2: SURFRAD Stations. Source: https://www.esrl.noaa.gov/gmd/grad/surfrad/sitepage.html

Metadata Beside GOES 13 imagery, our models also relied on auxiliary input data to predict GHI. Auxiliary data included the day of the year (1-365), hour and minute of the day, a boolean daytime

indicator, and a "clear sky" GHI prediction estimated with a model from Ineichen and Perez [2002] implemented in python [pvlib python].

Clear sky predictions take into account a geographical location's coordinates (latitude, longitude and elevation), local time, and other meteorological factors (solar position, atmospheric pressure, temperature, Linke turbidity factor) to estimate GHI on a clear, cloudless day [Reno et al., 2014, Ineichen and Perez, 2002]. These estimates condense many sources of information that determine GHI, and provide models with an explicit "upper bound" GHI score, which it can learn to nuance with information extracted from satellite imagery.

2.2 Preprocessing and Dataloading

A patch size of 64×64 cropped around a station was chosen as a starting point. Since each pixel from the input represents a square of 4×4 km, our patch is equivalent to a 256×256 km square. We had the intuition that this was sufficient to predict up to 6 hours in advance while maintaining a reasonable dataset size, but future works should assess the effect of patch size on performance. Furthermore, a patch size of 64×64 prevented us from reaching the edge of the (650, 1500) GOES-13 image since the station FPK was located near the border at coordinates (607, 497) in pixel space.

For models taking past (pre-T0) sequences of satellite images as input (see 2.3), the period covered by the lookback window (up to 3 hours) and the frequency at which images were sampled (every 45 minutes) was decided based on a combination of practicality (size of the input) and common sense. For example, we chose a lookback window length of up to 3h or 50% of the forecasted period, which is comparable to the lookback length used in similar publications, which ranged from 33% to 150% of the forecasted period. Having more time, our models could likely be improved with experiments that optimize the lookback window length and the sampling frequency of pre-T0 input images.

Since each HDF5 file consisted of one-day image chunks, we converted our data to TFRecords to minimize I/O access and randomize the samples. Using a provided pandas dataframe (catalog.pkl) that listed time points and their offsets, we randomized the timestamps and generated preprocessed mini-batches, where a sample consisted of the pre-cropped image at T0 (64x64 pixels, 5 channels), 4 previous images at [T-3h, T-2h15, T-1h30, T-45m] (each 64x64 with 5 channels), the past metadata corresponding to each image (day, hour, minutes, day/night and clear sky predictions), the future metadata (clear sky predictions at T0, T1, T3 and T6), and the target GHI values at [T0, T+1h, T+3h, T+6h]. To minimize the memory usage of our dataset, the images were stored as 8-bit integers inside of TFRecords and were decompressed lazily to 32-bit float at train time.

The training and validation set included every date time of the catalog.pkl for which it was day time for at least one of the seven stations, and for which the image at T0 and the GHI values at [T0, T+1, T+3, T+6] were available. As a consequence, **our reported GHI values on the validation set underestimate the final RMSE on the test set** since some night time examples were included in the validation set. Past missing images were replaced by an array of -1, which corresponds to min-value imputation as we used min-max normalization in the [-1, 1] range for every image. To make the model even more robust to missing frames at test time, we used dropout with min-value imputation on the past images at train time (see 2.4).

For the min-max normalization of the images, we computed the min and max value per channel on the train data (2010-2014). We also used min-max normalization for the day, hour and minute of the image. The clear-sky values were not normalized to help the model output a GHI value in the same order of magnitude.

2.3 Models

Models were implemented in Tensorflow 2.0 using the Keras API [Keras]. Model architectures (e.g., the type, number and size of each hidden layer) were informed by other nowcasting models described in the literature, and adjusted and trained as described in 2.4. All models were trained with the Adam optimizer using the mean squared error (MSE) loss function.

Baseline Models To establish a performance baseline, we trained two simple neural networks to predict GHI strictly from the auxiliary metadata (see 2.1). The purpose was to determine how well a model could predict GHI based on these data without learning anything from the satellite images, by

learning to adjust the "upper bound" clear sky prediction by some weighting factor. Input included **past metadata** (day, hour, minutes, day/night and clear sky predictions captured at 45 minutes intervals within a 3-hour lookback window from T0), and **future metadata** (clear sky predictions at T0, T1, T3 and T6).

We trained a fully connected multi-layer perceptron (MLP; Figure 3) on a 1-D input vector of concatenated past and future metadata. The model had two 8-unit hidden layers (ReLU non-linearity, elastic-net kernel regularization L1 and L2 = 0.005, 10% dropout rate) and a 4-dim output layer with linear activation. To capture information about the temporal relationship between consecutive time points within the lookback window, we also trained a Long-Short-Term-Memory model (LSTM; Figure 4) on sequences of past metadata (each a 5-dim 1D vector including the day, hour, minutes, day/night and clear sky prediction for a single time point). The LSTM included two 8-unit hidden layers (tanh activation, sigmoid recurrent activation) whose output was concatenated with the future metadata, and fed to two fully connected hidden layers (8-units/layer, ReLU activation, 10% drop out, L1 and L2 kernel regularization = 0.005) and a 4-dim output layer with linear activation.

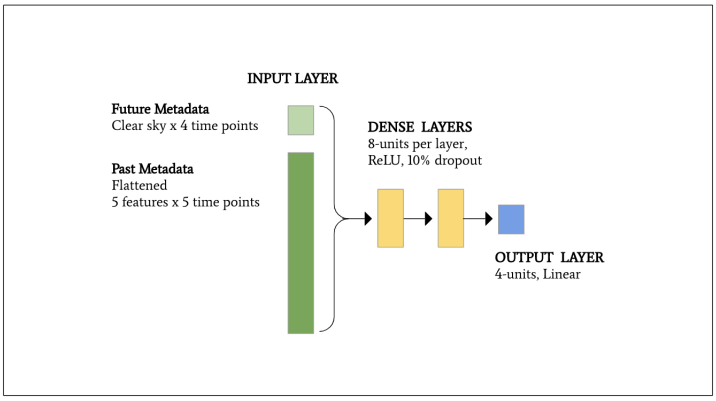


Figure 3: Baseline MLP model trained on auxiliary metadata.

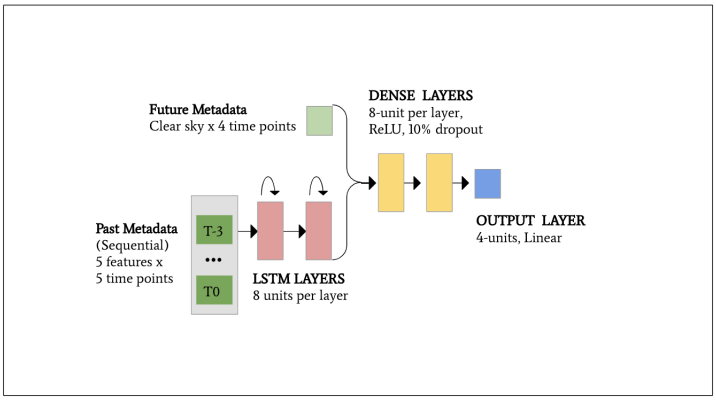


Figure 4: Baseline LSTM model trained on auxiliary metadata.

CNN-2D To determine how well a model could nowcast GHI at T0-T6 based strictly on information available at T0, we tested a simple 2-dimensional convolutional model (CNN-2D; Figure 5) that took as input a single cropped 5-channel image at T0 (64x64x5), as well as future clear sky predictions (T0-T6) and T0 metadata (day, hour, minute, daytime index and clear sky prediction). The model applied three convolutions (32@3x3, 64@3x3, 64@3x3) with ReLU activation to the input image, with 2x2 max pooling applied between each convolution. The result was flattened and concatenated with the current and future metadata. This vector then went through two fully connected layers of size 128 and 64 with ReLU activation, and one fully connected layer of size 4 that outputted a prediction at every target time. Note that this convolutional architecture served as a building block for the convolutional segment the CNN-RNN model described below.

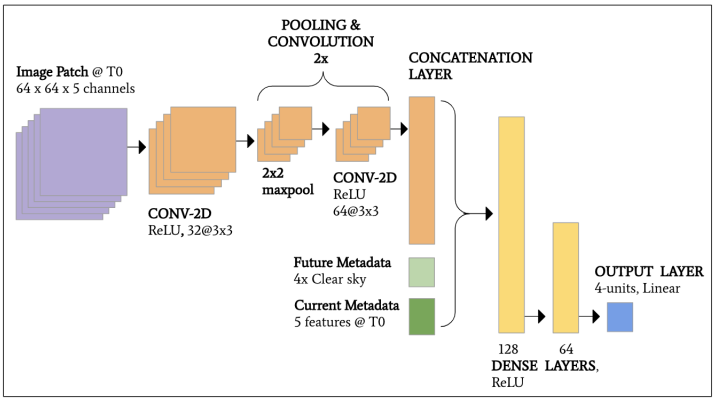


Figure 5: CNN-2D model architecture trained on GOES-13 image at T0 and auxiliary metadata.

VGG-3D We modeled spatio-temporal information from GOES-13 image sequences with a CNN-3D model adapted from Zhao et al. [2019] (Figure 6). Sequences of between 3 and 5 images (from T0) were concatenated (e.g., 3x64x64x5 for a 3-image sequence) and processed through a 3D convolution (32@3x7x7 convolution on first three dimensions, ReLU), and then converted into 2D space by squeezing it along its temporal dimension. The result was processed through two convolutions (64@5x5 and 128@5x5, both ReLU). 2x2 max pooling was applied between each convolution (2D or 3D), and results were flattened before being concatenated with the medatada. In a parallel input stream, sequences of past metadata (5-dim 1D vector per time point) were processed through a recurrent neural network (RNN; 1 layer of 8 units, tanh activation, 10% input and recurrent drop out). The processed metadata sequence (dim = 8) was concatenated with the flattened output from the convolution stream and the future clear sky predictions (dim = 4). This merged result was processed through a fully connected layer (ReLU) of size 64, and a size 4 output layer (linear activation).

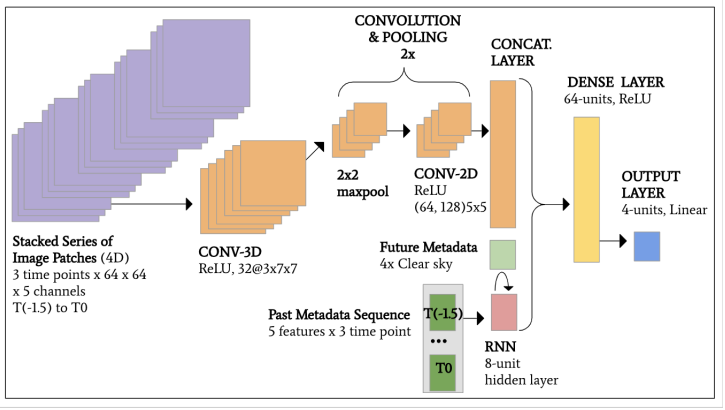


Figure 6: VGG-3D model trained on GOES-13 image sequence and auxiliary metadata.

ResNet-3D We tested another CNN-3D that implemented blocks of convolution inspired by the ResNet model [He et al., 2015] (Figure 7). Like VGG-3D, image sequences (length 3-5) were concatenated and processed through a 3D convolution (32@3x7x7 convolution on first three dimensions, ReLU), and then squeezed along its temporal dimension into 2D space. The result was processed through three convolution blocks based on the ResNet architecture [He et al., 2015]. Block 1 included three sub-blocks of [32@3x3, batch normalization (BN), ReLU, 32@3x3, BN, ReLU], Block 2 had four sub-blocks of [64@3x3, BN, ReLU, 64@3x3, BN, ReLU], and Block 3 had six sub-blocks of [128@3x3, BN, ReLU, 128@3x3, BN, ReLU]. Each sub-block was skipped by an identity shortcut connection. Down-sampling by a factor of two was applied at the end of the first sub-block from Blocks 2 and 3.

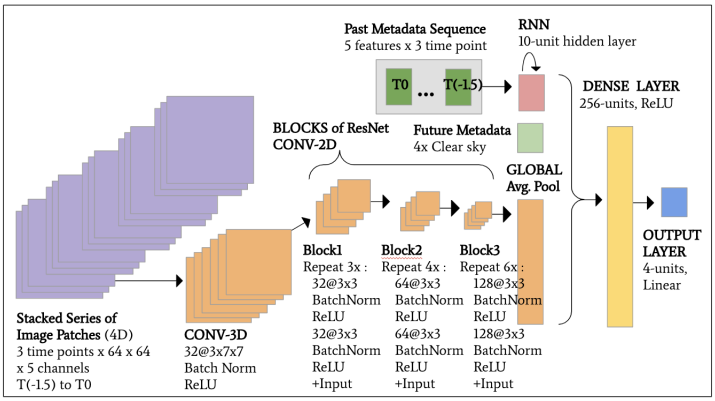


Figure 7: ResNet-3D model trained on GOES-13 image sequence and auxiliary metadata.

CNN-RNN We also modeled sequences of images with a convolutional recurrent neural net (CNN-RNN; Figure 8). At each input time point, the entering image (64x64x5) and its corresponding metadata (5-dim 1D vector) were processed through a recurrent cell. This cell applied a series of convolutions analogous to the CNN-2D model (32@3x3, 64@3x3 and 64@3x3, inter-spaced with 2x2 max pooling). The flattened convolution output was concatenated with the time point's metadata and sent to update a RNN hidden layer of size 512. Once the full sequence was processed, the RNN output was concatenated with the future clear-sky predictions (dim = 4) and processed through a fully connected linear output layer of size 4.

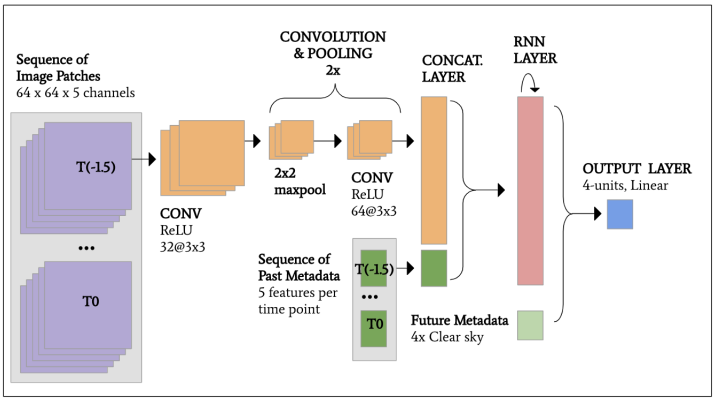


Figure 8: CNN-RNN model trained on GOES-13 image sequence and auxiliary metadata.

ConvLSTM Our last model processed sequences of 3-5 images via two 2 ConvLSTM layers that performed convolutions (each 64@3x3) on the input-to-state and state-to-state transitions of an LSTM (tanh activation, hard sigmoid recurrent activation), a method introduce by Shi et al. [2015] implemented in the Keras API (Figure 9). The result went through batch normalization and a layer of 2D convolution (1@3x3, sigmoid activation) before being flattened to a 1D vector. Past metadata (5-dim vector) were processed through a 2-layer LSTM (8 units/layer), and the result was concatenated with future clear sky predictions (4-dim vector) and the flattened processed image features. Merged data were passed through two fully connected layers (size 256 and 128, ReLU, 10% dropout) and one fully connected output layer (size = 4, linear activation).

2.4 Training and Evaluation

As GHI prediction is a regression task, model performance was evaluated with the root mean squared error (RMSE) based on the difference between the predicted and the ground-truth GHI score averaged across stations and time points (T0, T1, T3 and T6). For training and validation, we excluded every date time (T0) that corresponded to the night for **all** seven stations, whereas the evaluator.py at test

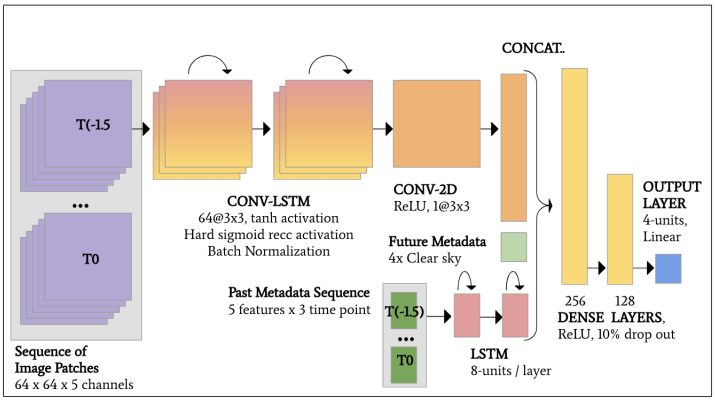


Figure 9: ConvLSTM model trained on GOES-13 image sequence and auxiliary metadata.

time uses a masking procedure to exclude individually every T0 (**per** station) that are during the night. As a consequence, our reported RMSE validation on the train and validation data is a lower bound on the RMSE exclusively at day time. Our models were trained on data from years 2010-2014, and validated on data from the year 2015. For the purpose of the course IFT-6759, our best model's official performance was obtained on a held-out test set of data points from 2016.

As a first phase, we trained each model listed in 2.3 to identify the type of architecture that obtained the lowest validation error. For efficiency, each model's architecture was adjusted based on cross-validated experiments conducted off-server on a small but representative portion (4%) of the training and validation sets. Final models were then trained (2010-2014) and validated (2015) on the full dataset on Calcul Québec's Helios3 cluster server using NVIDIA K80 GPUs [ComputeCanada] and the best model was identified based on those scores.

As a second phase, we performed a more extensive hyper-parameter search for the best model on Helios3 using the entire train set. We used a grid search approach to test how patch size and lookback window length influenced performance. We also tested whether rotating images from the train set helped the model generalize, by learning cloud movement from a variety of directions and avoid overfitting to weather patterns around the 7 stations. When data augmentation was used, each training image (and its corresponding channels) or series of images (as a single block) was rotated by either 0°, 90°, 180° or 270°, with a 25% probability for each position. We also added a 20% drop out rate to each image (excluding T0) from the train and validation set to help the best model handle missing data from the final test set. Dropped images were replaced by values of -1, the minimal pixel value after scaling.

Finally, for the third phase, we randomly generated a list of timestamps spanned across the year 2015, and we used the provided code in evaluator.py to replicate the settings at test time. We verified that the data loader at test time was functioning properly by making sure that our best model on the validation set of TFRecords had a similar RMSE on this task.

3 Results

3.1 Best Model Identification

Table 1 shows the training and validation error (RMSE) for each model. As anticipated, baseline neural nets (MLP and LSTM) performed better than the clear sky prediction, likely by scaling their output to reflect a more typical (not cloudless) day. Also, all models that used images and metadata performed noticeably better than the metadata baseline models.

The convolutional model that only received T0 data (CNN-2D) performed more poorly than models that extracted spatio-temporal information from image sequences. The best (lowest) validation score was obtained with the CNN-3D architectures (VGG-3D and ResNet-3D). Good scores were also obtained with CNN-RNN and ConvLSTM, although these models took a lot longer to train than

Table 1: RMSE per model

Model	# Parameters	Batch Size	# Epochs	Train RMSE	Validation RMSE
Clear Sky Model	NA	NA	NA	NA	207.94
Baseline MLP	348	32	35	172.24	154.40
Baseline LSTM	1204	32	35	171.59	154.84
CNN-2D	1,246,340		2	126.02	125.98
VGG-3D*	797,396	512	45	104.67	98.52
ResNet-3D*	2,110,628	256	21	98.89	**96.01**
CNN-RNN**	5,304,916		9	109.81	107.95
ConvLSTM*	1,541,541	32	2	113.30	110.95

*Image sequence length = 3
**Image sequence length = 5

the CNNs, for lesser results. ConvLSTM was especially slow and resource-intensive, taking several hours to complete a single epoch on the full training set.

Of note, VGG-3D had consistently produced the best results at the time we selected a "best model" for additional optimization. For this reason, VGG-3D was subject to a more extensive hyper-parameter search, as detailed in the next section. However, final tests revealed that ResNet-3D obtained the lowest validation RMSE, and it was selected as our final best model. In section 3.3, we present visualization results for both VGG-3D and RestNet-3D (both with image sequence length = 3).

3.2 Best Model Hyper-Parameter Tuning

Train and validation RMSE as a function of patch size and sequence length for model VGG-3D are shown in Table 2. A larger patch size systematically improved the results, suggesting that the model benefited from input that stretched farther around our point of interest to predict the future.

The length of the sequence (or depth of the stacked images, in this case) did not seem to have a large impact on the result. Only looking at a sequence length of 2 (T0 and T0 - 45 minutes) gave the best results on both patch sizes 32x32 and 64x64. One possible interpretation is that taking only 2 images is sufficient to infer linear cloud movement, and that the model either did not learn, or did not gain from non-linear movement patterns that can be extracted from longer image sequences. Also, taking fewer input features (shorter sequences) probably made it easier for the model to train more efficiently and to generalize better.

Early results indicated superior generalization in VGG-3D when data augmentation was applied. Without data augmentation, the model began overfitting the training set after around 10 epochs (Table 2). Therefore, data augmentation was applied to all models during hyper-parameter search (and to ResNet-3D whose results are shown in 3.3). The best VGG-3 model identified used a sequence of length 2 with 64x64 images.

3.3 Best Model Final Results and Visualization

We compare the results of our two best models, VGG-3D and ResNet-3D, on a subset of 2.4K random time points sampled from 2015. We only calculated error using daytime points for each station. This difference explains how RMSEs shown here are higher than those reported on our validation set, which contained some night predictions on which models must have performed well (Table 1). Overall, VGG-3D had a RMSE of 113.15 while ResNet-3D had a RMSE of 109.98.

Figure 10 shows a typical training curve for the VGG-3D. We can see that the train RMSE is always slightly higher than the validation. This difference may be due to the fact that data augmentation was only applied to the training set. The model does not seem to be able to overfit the data, which indicates that we could need an architecture with greater capacity.

Figure 11 shows the RMSE across stations for our best two models. We can see significant pair-wise differences between stations, as indicated by non-overlapping error bars (standard error of the mean for each station). Many factors could explain these differences, such as variability in GHI for a

Table 2: RMSE for VGG-3D Hyper-Parameter Search

# Patch size	Sequence length	# Epochs	Train RMSE	Validation RMSE
32x32	2	22	108.66	102.52
	3	19	106.95	102.71
	4	20	107.29	102.54
	5	23	107.37	102.82
64x64	2	45	104.67	**98.52**
	3	24	104.79	99.92
	4	29	103.73	99.35
	5	35	104/23	99.87
64*	5	10	97.06	101.22

*No data augmentation

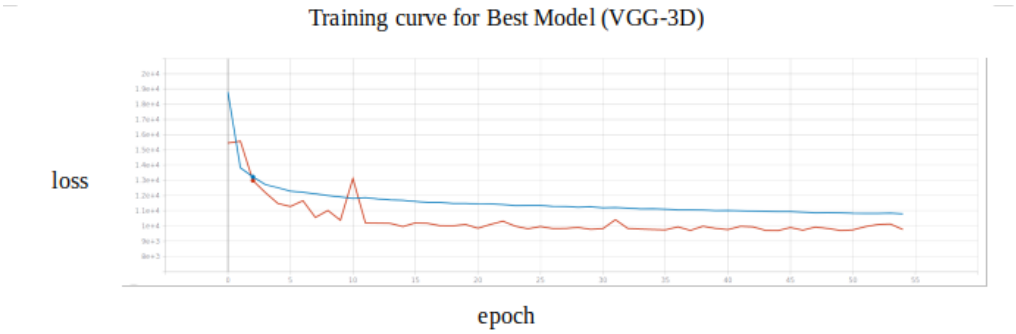


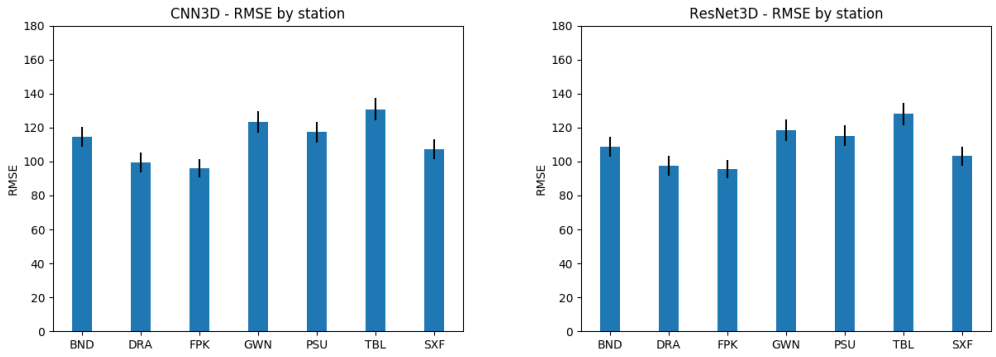
Figure 10: Training curve for VGG-3D. Loss is the mean squared error (MSE), plotted over epoch. Blue = Train MSE; Orange = validation MSE,

given station due to complex weather or topological conditions. For example, the station with the poorest predictions from each model is Table Mountain (Boulder, CO), which is set at the foot hills of the Rocky Mountains. Further investigation should explore whether additional auxiliary input like topology data can improve predictions. Interestingly, Desert Rock (NV) has low prediction errors despite being the closest point from the edge of the GOES-EAST map, suggesting decent signals despite the potential distortion.

Figure 12 shows the performance of both models in different cloudiness conditions. Cloudiness was determined with a heuristic model from Tina et al. [2012] that assigned time points to one of four categories (clear, slightly cloudy, cloudy and variable) based on the difference between measured GHI and its clear sky prediction. Prediction error is greater in cloudy conditions and lesser in clear conditions for both models, probably due to the fact that the clear sky scores given to both models as auxiliary input provides a good estimate of GHI on cloudless days.

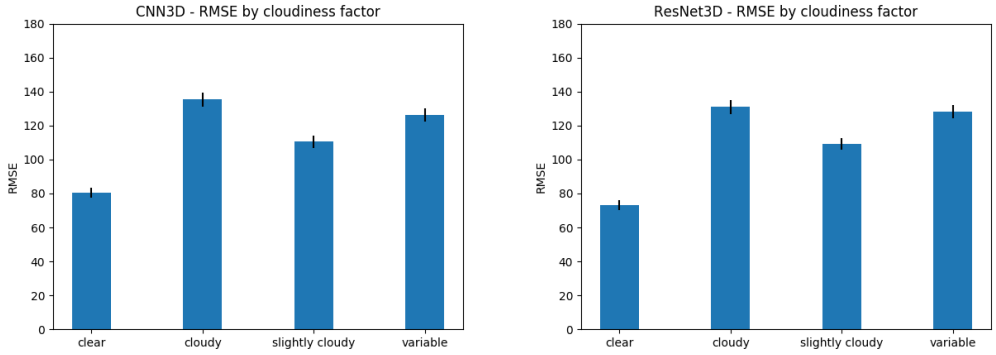
Performance per month (Figure 13) is better in winter and gradually decreases as we approach mid-summer for both models. The very similar plots suggest that this pattern is probably more related to the data than to the models per se. Results could be explained by the fact that GHI values are generally higher in summer in the US and that a constant relative error means a larger MSE for those months.

Figure 14 plots ResNet-3D predictions against clear-sky model predictions and ground-truth target GHI values (using code provided by IFT6759 course instructors). We chose the 3 of January to visualize the model's predictions at different cloudiness levels. For example, measured GHI values are perfectly aligned with the clear-sky predictions at the Desert Rock station (DRA), are at around half the clear-sky values for Sioux Falls and Table Mountain (SXF and TBL), and are low on a very cloudy day at stations Bondville and Goodwin Creek (BND and GWN). Overall, we see that our model performs well across these three different weather patterns, especially for predictions at T0. Interestingly, we observe that the predicted values at T+6 are often very close to the clear-sky values,



(a) VGG-3D (b) ResNet-3D

Figure 11: RMSE per SURFRAD station on sample from validation set. BND = Bondville, IL; DRA = Desert Rock, NV; FPK = Fort Peck, MT; GWN = Goodwin Creek, MS; PSU = Penn. State University, PA; TBL = Table Mountain (Boulder), CO; SXF = Sioux Falls, SD. Error bars = standard error of the mean.



(a) VGG3D (b) ResNet3D

Figure 12: RMSE per cloudiness condition on sample from validation set. Error bars = standard error of the mean.

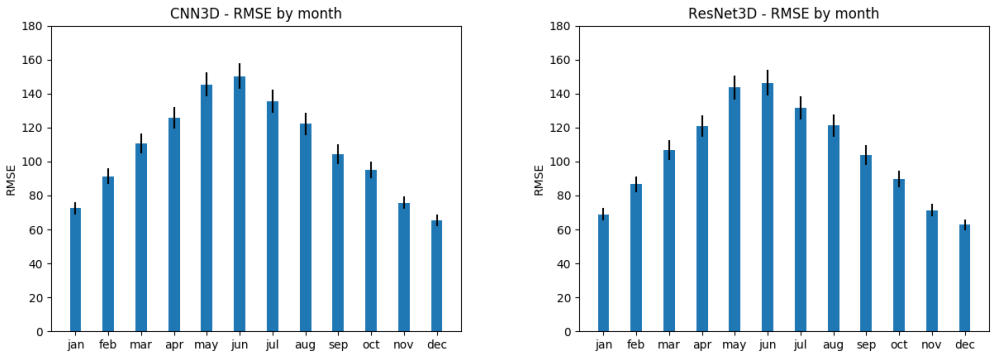
which seems to indicate that the model does not benefit much from the present and past images to predict 6 hours into the future.

4 Discussion

4.1 Successful Model Features and Future Steps

Among the models tested, 3-D convolutional models were more successful at extracting predictive temporal information from image sequences than convolutional-recurrent models. El-Gazzar et al. [2020] have contrasted strictly convolutional models against models that combine recurrence with convolutions to classify image time series (brain activation maps), and found that convolutions offer a computationally cheaper alternative to LSTMs. In our case, CNN-3D may have outperformed recurrent networks in a context where the number of training epochs was limited by the size and complexity of the dataset.

Within the parameters tested, models benefited from greater spatial information (i.e., larger patch size), while making limited use of temporal information (i.e., the number of images in a sequence). Hyper-parameter search revealed that CNN3D architecture mainly relied on linear transformations that could be estimated with only two time points. The VGG-3D learning curve also indicated that it



(a) VGG3D (b) ResNet3D

Figure 13: RMSE per month on sample from validation set. Error bars = standard error of the mean.

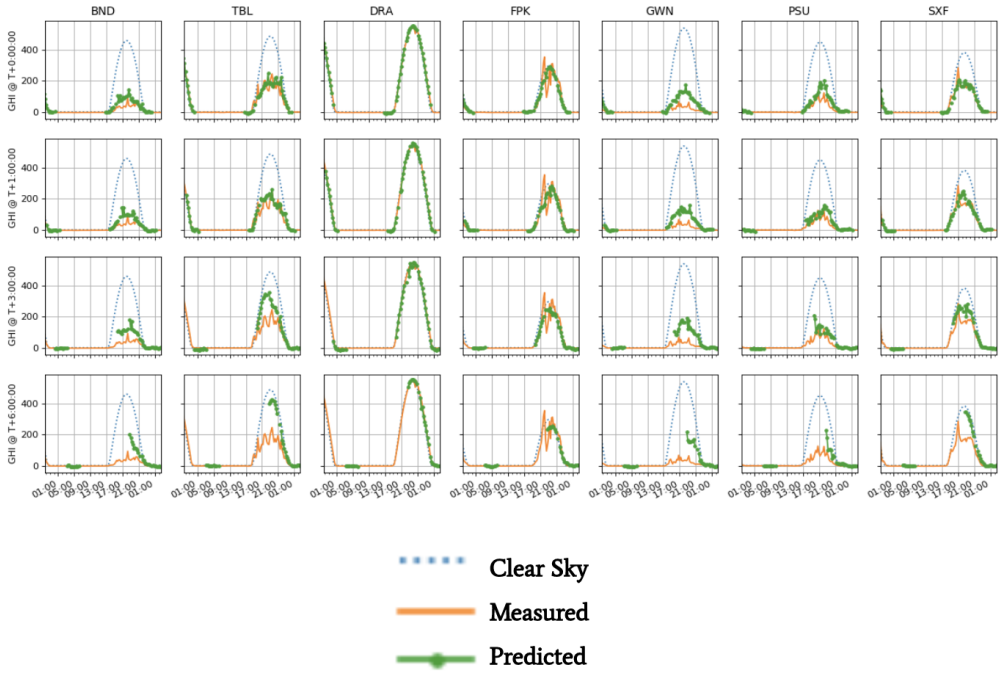


Figure 14: Visualization of the GHI values predicted by ResNet-3D (green) compared to ground-truth GHI target values (orange) and clear-sky model predictions (blue, dotted) at each station (column) for January 3, 2015. Each row corresponds to predictions at [T0, T1, T3, T6], respectively.

did not overfit, suggesting that architectures with higher capacity may be appropriate for this type of problem. Of note, very deep ResNet-3D had a slight advantage over much shallower VGG-3D, although it is unclear whether depth or capacity accounted for this difference. It is also possible that convolutions, which are location-invariant, may not be optimal to exploit non-linear motion over time to predict future GHI. Models with greater capacity, for example Trajectory GRU models that adjust convolutions dynamically [Shi et al., 2017], may be better adapted to extract such patterns.

Future efforts should experiment with a greater number of hyper-parameters that influence model capacity, including the number of convolution layers, kernel sizes, and the use of skip connections. Additional steps should also include a more thorough exploration of different lookback window lengths, as well as image sampling frequency rates, to assess their effect on nowcasting. In addition, our models relied on information from all five GOES-13 channels to predict GHI, and further analyses

should look into whether some of this information is unnecessary or redundant, with the goal of building leaner, better informed models.

Due to the limited number of locations with ground-truth GHI measures, one risk is that models with large capacity will learn the peculiarities of each site and skew their predictions accordingly, resulting in a model that does not scale well. Lebedev et al. [2019] have proposed using smaller patch sizes of satellite images to prevent overfitting to geographical areas (their small patches were 279x279km, compared to our 256x256km). Our results also indicate that measures like input drop out and data augmentation improved performance by promoting generalization. Future efforts could also evaluate the impact of normalization methods on learning rate and prediction.

Finally, our models appeared to benefit from auxiliary input data, especially the clear sky predictions. Early training attempts revealed poor, slower learning when clear sky predictions were scaled (min-max). These results suggest that models relied on clear sky estimates as an upper bound that anchored their predictions, and struggled to learn how they mapped onto the target scores if they were rescaled. Other groups have reported on the benefits of supplementing imagery data with auxiliary information to nowcast meteorological phenomena (e.g., Mathe et al. [2019], Lebedev et al. [2019], Siddiqui et al. [2019]). Future efforts should explore which additional information can help improve GHI nowcasting (e.g., example topological maps), and how best to integrate such features with image-derived features in a neural net architecture.

Of note, CNN-RNN was the only architecture that combined information from image features and metadata at each time point. Other models processed image and metadata features in separate streams. With the use of additional metadata, future works should assess how merging multi-modal data early or late along the processing stream affects performance, especially if the information is time-locked to a particular image.

4.2 Challenges and Limitations

The sheer size and complexity of the dataset created important challenges for project completion. For training to be time-effective, data required extensive pre-processing that introduced many opportunities for errors. The amount of time required to train each model also hindered our capacity to optimize our models within the project's time line. Despite having access to substantial computing resources, our final model performance would have benefited from additional time to explore different kinds of architectures and input combinations. As a next step, we would also wish to explore whether predictions are improved by the use of original NetCDF or 16-bit JPEG2000-compressed (max 1% loss) images, compared to currently used 8-bit JPEG-compressed images, a much lossier compression.

5 Conclusion

We successfully nowcasted GHI within a 6-hour time window using deep learning models. Best results were obtained with a 3CNN-3D architecture that processed multi-channel satellite imagery combined with auxiliary metadata, including clear sky GHI predictions. Our work identified techniques that can effectively be applied to nowcasting problems, and uncovered additional avenues of exploration to achieve more accurate predictions.

References

Wikipedia. Geostationary orbit, a. URL `https://en.wikipedia.org/wiki/Geostationary_orbit`.

NOAA. Noaa geostationary satellite server. URL `https://www.goes.noaa.gov/`.

SURFRAD. Surfrad (surface radiation budget) network. URL `https://www.esrl.noaa.gov/gmd/grad/surfrad/index.html`.

Pierre Ineichen and Richard Perez. A new airmass independent formulation for the linke turbidity coefficient. *Solar Energy*, 73:151–157, 09 2002. doi: 10.1016/S0038-092X(02)00045-2.

Bikhtiyar Ameen, Heiko Balzter, Claire Jarvis, and James Wheeler. Modelling hourly global horizontal irradiance from satellite-derived datasets and climate variables as new inputs with artificial neural networks. *Energies*, 12, 01 2019. doi: 10.3390/en12010148.

Talha Ahmad Siddiqui, Samarth Bharadwaj, and Shivkumar Kalyanaraman. A deep learning approach to solar-irradiance forecasting in sky-videos. *CoRR*, abs/1901.04881, 2019. URL http://arxiv.org/abs/1901.04881.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

Vadim Lebedev, Vladimir Ivashkin, Irina Rudenko, Alexander Ganshin, Alexander Molchanov, Sergey Ovcharenko, Ruslan Grokhovetskiy, Ivan Bushmarinov, and Dmitry Solomentsev. Precipitation nowcasting with satellite imagery. *CoRR*, abs/1905.09932, 2019. URL http://arxiv.org/abs/1905.09932.

Shreya Agrawal, Luke Barrington, Carla Bromberg, John Burge, Cenk Gazen, and Jason Hickey. Machine learning for precipitation nowcasting from radar images, 2019.

Xin Zhao, Haikun Wei, Hai Wang, Tingting Zhu, and Kanjian Zhang. 3d-cnn-based feature extraction of ground-based cloud images for direct normal irradiance prediction. *Solar Energy*, 181:510–518, 03 2019. doi: 10.1016/j.solener.2019.01.096.

Johan Mathe, Nina Miolane, Nicolas Sébastien, and Jeremie Lequeux. Pvnet: A LRCN architecture for spatio-temporal photovoltaic powerforecasting from numerical weather prediction. *CoRR*, abs/1902.01453, 2019. URL http://arxiv.org/abs/1902.01453.

Wei Zhang, Wei Li, and Lei Han. A three-dimensional convolutional-recurrent network for convective storm nowcasting. *2019 IEEE International Conference on Big Knowledge (ICBK)*, Nov 2019. doi: 10.1109/icbk.2019.00052. URL http://dx.doi.org/10.1109/ICBK.2019.00052.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 802–810. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5955-convolutional-lstm-network-a-machine-learning-approach-for-precipitation-nowcasting.pdf.

Chao Tan, Xin Feng, Jianwu Long, and Li Geng. Forecast-clstm: A new convolutional lstm network for cloudage nowcasting, 2019.

Seongchan Kim, Seungkyun Hong, Minsu Joh, and Sa kwang Song. Deeprain: Convlstm network for precipitation prediction using multichannel radar data, 2017.

Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Deep learning for precipitation nowcasting: A benchmark and a new model. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5617–5627. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7145-deep-learning-for-precipitation-nowcasting-a-benchmark-and-a-new-model.pdf.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–, October 2017. URL http://dx.doi.org/10.1038/nature24270.

Wikipedia. Goes 13, b. URL https://en.wikipedia.org/wiki/GOES_13.

pvlib python. Clear sky. URL https://pvlib-python.readthedocs.io/en/stable/clearsky.html.

Matthew Reno, Clifford Hansen, and Joshua Stein. Global horizontal irradiance clear sky models : implementation and analysis. 01 2014. doi: 10.2172/1039404.

Keras. Keras: The python deep learning library. URL https://keras.io/.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

ComputeCanada. Helios. URL https://docs.computecanada.ca/wiki/H%C3%A9lios.

Giuseppe Tina, Sebastiano De Fiore, and Cristina Ventura. Analysis of forecast errors for irradiance on the horizontal plane. *Energy Conversion and Management*, 64, 12 2012. doi: 10.1016/j.enconman.2012.05.031.

Ahmed El-Gazzar, Mirjam Quaak, Leonardo Cerliani, Peter Bloem, Guido van Wingen, and Rajat Mani Thomas. A hybrid 3dcnn and 3dc-lstm based model for 4d spatio-temporal fmri data: An abide autism classification study, 2020.