Generation of High Resolution Climate Change Projections with Recurrent based Convolutional LSTMs

Nidhin Harilal
Undergraduate
IIT Gandhinagar
nidhin.harilal@iitgn.ac.in

Prof Udit Bhatia
Assistant Professor
IIT Gandhinagar
bhatia.u@iitgn.ac.in

ABSTRACT

There has been a rise in the incidents of extreme climate events primarily due to the impacts of climate change. There is a need for the estimation of the level of intensity and duration of such extreme events. Earth System Models(ESM), which are run at spatial resolutions, helps in generating climatic projections. Typically, such models have a resolution of 150-300 km. Most of the extreme impacts require information at scales of 50 km or less, so some method is needed to estimate such localized effects. Low-scale projections could be developed by deriving statistical relationships between observed small-scale variables and large-scale GCM values. Deep Learning has drawn a lot of attention in generating fine-resolution climatic projection owing to the Spatio-temporal nature of the data. However, Recent attempts towards Statistical Downscaling are mostly focused on utilizing merely the spatial dependencies present in data. However, none of these works have considered combining the benefits of utilizing both the spatial as well as the temporal dependencies present in the GCM data. In my work, I present a Recurrent Convolutional LSTM based approach towards generating fine-resolution projections from the GCM climate variables. Also, I address the problem of Concept Drift that exists with the current models and a comprehensive set of factors why my proposed model outperforms the currently existing ones.

KEYWORDS

Climate Statistical Downscaling, Deep Learning, Daily Precipitation, Super-Resolution

1 INTRODUCTION

The impact of global warming and climate change are already observed throughout the world and has led to temperatures increase, increase in intensity of extreme events, and rise in sea levels[9]. The health as well as the security of human beings have become a concern due to its vulnerability towards extreme weather events with increasing intensity, duration, and frequency. Estimation of such extreme events have also become important to answer the questions related to sustainability of our present infrastructure and technology.

Global Circulation Models (GCMs) are physics-based mathematical models for simulating our Earth's atmosphere and generating climatic projections. These models take into account the atmospheric, land, and ocean effects along with the physics and climate dynamics while generating the projections. Archived GCM outputs are used in academia as well among researchers for weather forecasting, understanding the climate, and forecasting climate change. Owing to GCMs high-computation requirements, their projections are limited to a very coarse resolution. Using GCM as a means for assessing the local impacts have complications as GCM have a coarse resolution, and they also fail to capture the climatic variations at a regional scale.

Downscaling is a technique of generating fine-resolution projections and it could be broadly classified into two categories i.e, Dynamical and Statistical techniques. Dynamical downscaling act as regional climate models [3], are dynamical models similar to the GCMs but they operate at the regionals level, thus, generating high-resolution projections of climatic variables restricting to that particular region. The complication with such models is that they are computationally expensive and also the same model could not be scaled to other regions owing to the differences in the region climate dynamics. Statistical downscaling on the other hand, attempts to learn the statistical relationship between the low-resolution and the high-resolution[3]. This statistical approach may include methods such as conventional machine learning [2], Convolutional Network [8] and even basic Recurrent nets [4]. Methods which uses deep learning for statistical downscaling has been found very promising owing to the spatial and temporal dependencies present in the data. Despite the availability of so many techniques, there has not been work done which explicitly captures both the spatial as well as the temporal dependencies in the low-resolution GCM data for generating fine-resolution projection. The lack of work on an effective Statistical downscaling using deep learning approach motivated me to research in this problem.

1.1 Key Contributions

The key contributions are as follows:

- I present a Recurrent Convolutional LSTM based approach towards Statistical Downscaling of climatic data from coarse-resolution GCM to fine-resolution Observation data.
- I attempt to address many of the limitations and false assumptions with the current deep learning approaches towards Statistical downscaling.
- My methodology along with my proposed model attempts to solve many of these addressed limitations, thus, making it more reliable than the existing ones.
- The proposed model has the potential to outperform the current state-art deep learning approach towards statistical Downscaling in terms of predictive performance and generalisability for future predictions.

2 CURRENT WORK & ITS LIMITATIONS

Due to the high significance of generating fine-resolution projections from GCM projections, there has been much work done in statistical downscaling. However, the ones which involves deep learning models has been found the most effective in terms of both their performance. It is because the spatio-temporal nature of the climate system motivates the adaptation of deep learning techniques to statistical downscaling. However, There has not been much work which uses deep learning based approach for Statistical Downscaling.

In this section, I will address two of the main works which consists of models with Neural Networks attempting to generate high resolution projections from low-resolution GCM data.

2.1 SD with Single Image Super-Resolution[8]

This work is about DeepSD[8] which consists of stacked super resolution convolutional neural network (SRCNN)[1] framework for statistical downscaling of climate variables. DeepSD augments SRCNN with multi-scale input channels to maximize predictability in statistical downscaling.

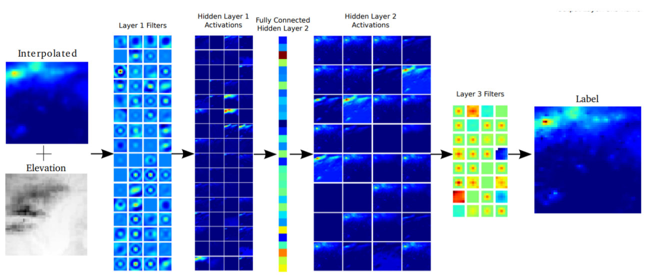

Figure 1: DeepSD's proposed SRCNN approach[8]

However, the basic architecture that this work uses consists of repeated SRCNN blocks which has three convolutional layers, where each block is responsible for partial increase in resolution. So, Four such blocks leads to four times up-scaled projection. Inorder, to compensate the false data points of GCM, an auxiliary climatic variable *elevation* (height from sea-level) was given as another channel along with the GCM data. This guided the model in finding patterns in the grid where the GCM data points varied largely from that of observation data.

2.2 Limitations of DeepSD

One of the most critical point to note in this work is that, they have envisioned the problem of Statistical Downscaling as that of a *Single Image Super-Resolution*. In this type of model there is a one-to-one mapping made while training. That is, in each training instance, the input would be a low-resolution projection of a particular day mapped to the high-resolution one. The reason behind this was because, this model consisted merely of convolutional layers which were responsible for considering only the spatial features and relations between the data points.

The problem that I discovered with this implementation was the problem of concept drift. Climatic data, especially precipitation, are spatiotemporal, and it changes over time with a particular pattern. The concept drift was the assumption of a static relationship between the input and the output, i.e identically independent data points.

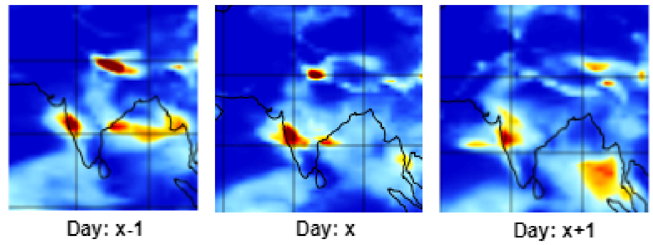

Figure 2: Existence of temporal dependencies in the data

Figure. 2 shows how precipitation distribution of a particular day depends on the previous days. In such case, using a model that considers only the spatial dependencies present and neglecting the temporal part, has a adverse effect on the predictive capabilities of the model.

2.3 SD with Recurrent LSTMs[4]

This particular work attempted to solve Statistical downscaling of precipitation using long short-term memory recurrent neural networks[4]. LSTMS are commonly used for processing sequential inputs and are preferred over vanilla RNNs as they solve the problem of vanishing gradients to some extent[6]. However, one of the essential requirements for LSTMs is that it requires a one-dimensional data stream. So, in order to fit this with the problem, this work[4] flattened the image to form a single dimensional array and passed it to LSTMs.

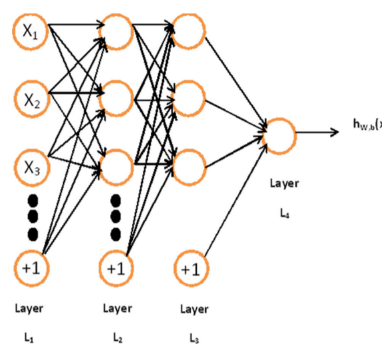

Figure 3: Recurrent LSTM based model for SD[4]

They have stacked three LSTM layers as shown in **Figure.**3. The first and the second LSTM layers return sequences over different timestamps, so that, the next followed LSTM layer could be able to process. However, the last layer returns sequence of only one timestamp (one timestamp here depicts one day).

2.4 Limitations of using Vanilla LSTMs

LSTMs perform well in cases where data is time-series and one dimensional[6]. In case of Statistical Downscaling, the climatic data points also have a spatial dependency. So, If given a flattened data to the LSTMs, the information about spatial features would get lost, and, it will only consider the temporal dependency present in the data. This model may work with a somewhat decent accuracy on a very small region (where spatial dependency could actually be ignored). However, it would perform poorly if scaled for the entire Indian region. One more actual problem is that this particular could be scaled for the region as big as India. LSTM iterate over the entire sequence in order to preserve the long-term dependencies. When this model is applied to small region, the flattened data has a sequence length in the range of hundreds. However, when scaled up for big region such as India, this effective length becomes in the range of thousands. Therefore, it is not practical for LSTM to handle such large sequences.

3 PROPOSITIONS & ATTEMPTS TOWARDS SD

3.1 Proposition 1: Inclusion of more Auxiliary Climatic Variables

Since, GCM are itself mathematics and physics based models, therefore, there may be instances where the produce afalse values. After actual observation, GCM values had incorrect distribution most of the time. Addition of extra information such as Elevation by DeepSD[8], was one of the ways of tackling this. The hypothesis was that there must exist some patterns on which the GCM models produce those incorrect data over the grid. And giving extra information to the model would help it in deciding how to manipulate with the data without completely blindly trusting it.

My proposition is that why should we restrict ourselves from providing only the elevation data. There are other climatic varibles apart from precipitation that GCM models produce. If we provide that data points to the model, than the model might find existing patterns in any of those climatic variables. Even Climate Research shows the relationship between different climatic variables that are produced by GCM. After going through the literature about the work done on finding similarities between different climate variables, I found the below ones, the most common and useful.

- Pressure
- Humidity
- 3 Wind components (u, v & ω)

3.2 Proposition 2: Solving problem of Concept Drift

The projections of various climatic variable have both spatial as well as temporal dependencies. It was evident from **Figure. 2**. However, DeepSD[8] focussed only on utilizing the spatial component associated with the data. The complication with this implementation is the problem of *Concept drift*. The problem of *Concept Drift* arises when there is an assumption that the data is static and doesn't change over time. However, this is not true when dealing with projections from GCM. Climatic data, especially precipitation is spatiotemporal in nature and it changes over time.

My proposition is that, if the architecture is designed such that it utilizes both the spatial and temporal features, then the Model might be able to generate fine resolution projections over regions more precisely. However, the architecture should be such that it preserves both the counterparts when being processed, for example, while using the vanilla LSTMs, although it considered the temporal dependency involved but the spatial features were getting lost.

3.3 Attempt 1: Parallel Convolutional Network

My first approach was Convolution based which took into account both the spatial and temporal nature of the data. Convolutions are known for processing the data which has spatial features. However in-order to incorporate the temporal dependency, parallel Convolutional layer were stacked together, each parallel network responsible for each timestamp. A visual representation of this architecture is shown in **Figure.4**.

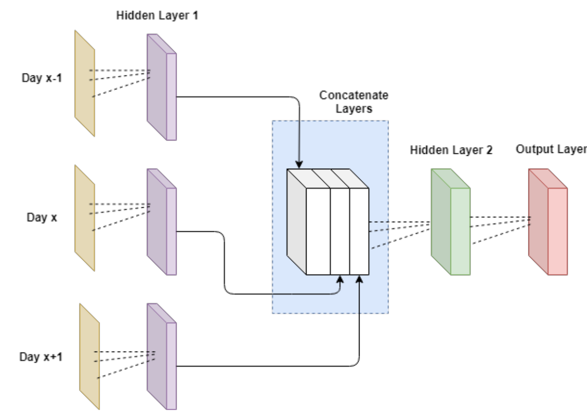

Figure 4: Attempt 1: Parallel Convolutional Network

My training data was a 7 channel input having the GCM precipitation data, elevation, and the other auxiliary variables described in the **Proposition 1**. Each of the training instance consisted of data for 3 consecutive days. Since, the data involved for each training instace consisted of 3 days, 3 parallel convolutional layers were there in the architecture.

RMSE Loss	67.3 mm

Table 1: RMSE Loss after 3500 epochs

However, despite the fact that this model took into account both the spatial and temporal features into account, but still the accuracy of the model was very low. **Table 1** shows the exact *RMSE* loss values of the model after sufficient training of 3500 epochs. The reason for this failure was due to one anomaly present in the data which is addressed in the **section** 3.4

3.4 Distribution anomaly in the data

Ideally, if a model is having a very low accuracy then it should perform poorly on most of the data. However that was not the case with the previous model. The model described in **Section 3.3** was performing well on approximately half of the year and very poor on the rest.

From the **Figure. 5** which shows the plot of *RMSE* (Root mean square error) between the predicted and the observed fine-resolution projections, It is clearly evident that the model has performed very well till the months of April and then the predicted projections differed largely from that of the observed one. But from the months of September to December, the model again starts predicting very well.

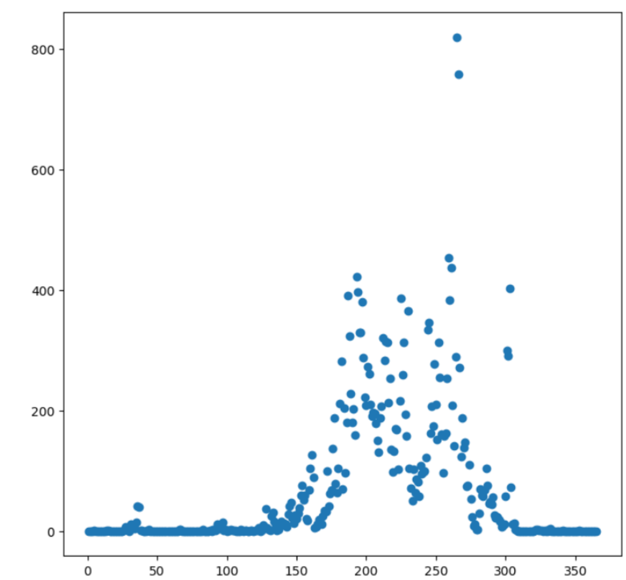

Figure 5: Plot of RMSE (between predicted and observed projections) over the days of the Year 1999

The reason behind this anomaly is that precipitation is not uniformly distributed over the year. The months (May-September) during which the model has a strange behaviour is the monsoon period in India. And during monsoon, the precipitation distribution is very different from the rest of the period both in terms of the peak values and regions covered by it. So, This period is actually creating a bias towards model which learned the mapping of projections in non-monsoon periods and vice-versa.

3.5 Proposition 3: Using Two-Model approach

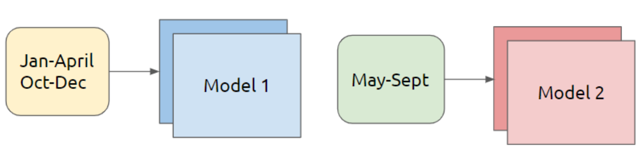

Figure 6: Proposed two model approach

To solve the above addressed problem of distribution anomaly, I propose to use different models on months having Low-Mid and Mid-High precipitation values. Using a single model would lead to an increased inconsistency in the predicted projections due to the distribution anomaly. But using two different models for these different distributions would reduce the burden and the effective mapping between the different resolutions would be much precise.

4 FINAL PROPOSED ARCHITECTURE

Each of the components of my final proposed model is built on the propositions I have made and attempts to solve the problems that I have addressed.

4.1 Model Description

I propose a Recurrent Convolutional LSTM based model which has the ability to capture and preserve both the spatial and the temporal features associated with the data. Recurrent Long-Short term memory(LSTM) are known for their application in sequential data and they are much preferred over the other Recurrent networks as they attempts to somewhat solve the problem of *Vanishing Gradients*. However, Basic LSTMs deal with single-dimensional data, hence, works poorly on the images.

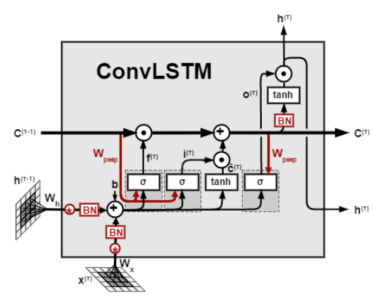

Figure 7: Convolutional LSTM block[7]

However, ConvLSTMs(Convolutional LSTMs)[7] tries to solve this by replacing the matrix multiplications with Convolutional operations. **Figure.7** shows the visual representation of ConvLSTM.So, when these cells are given the 2 dimensional inputs, the convolutional operations preserve the spatial dependencies of the data.

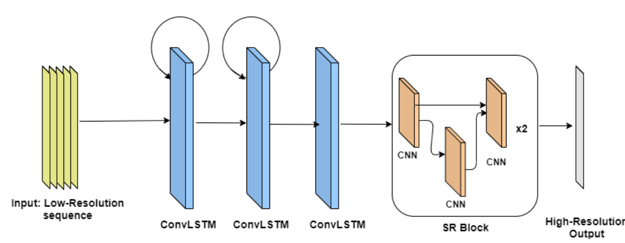

Figure 8: Proposed Convolutional LSTM based model

The proposed architecture consists of 3 ConvLSTM layers stacked together and followed by a SR block. **Figure.8** shows the visual representation of the proposed architecture. The primary job of this SR block is to increase the resolution of high dimensional feature data obtained from the previous ConvLSTM layer. This SR block consist of a 6 stacked deep Convolutional layers with skip connections in between them. Successive convolutions tend to lose some information by a factor which might be small but can lead to a poor performance when used in case of increasing the resolution. skip connections can play an important role here by passing on the features present at a previous step to forwards which provides successive Convolutional layers more information to work on.

4.2 Data

Based on my **Proposition 1.**, My training data consists of 7 channels. Each one are as follows:

- GCM coarse Projection (Precipitation)
- Elevation
- Pressure
- Humidity
- 3 wind Compnents (u, v, and ω)

Due to the distribution anomaly and based on my **Proposition 3.**, I have divided the data into 2 parts: one consisting of low-rainfall (from months Jan-April, and Oct-Dec) and other one consisting of high-rainfall (from months May-Sept). Each of the channels except the Precipitation projection has been normalized between zero to one, whereas the Precipitation has been normalized between zero to fifty in order to make it's effect more dominant then the other auxiliary climatic variables. The observation data has also been normalized between zero to fifty to ease the process of training.

Data was split for training and testing, the training data consisted of years from $1948 - 2010$ and the test data consisted of years $2010 - 2017$. the data was then made sequential (as per required by ConvLSTMs) according to it's required sequence length, where each such sequence was mapped to a single fine-resolution data.

4.3 Model Training

Since, the model consist of Recurrent ConvLSTMs, so it takes a fixed length sequence of 7-channel data, where each member of sequence represents climatic data of one particular day. The proposed ConvLSTM model has been trained on such sequences of length 5, which means that the model would take into consider the information from different climatic variables of past days 4 days along with the current one while generating a fine-resolution projection of a single day. The effective batch size for training is 20, so, the model would first predict the 20 days and then back-propagate it's losses to learn the parameters.

Inorder to avoid the case of overfitting, Recurrent dropouts of 0.2 were added to the stacked ConvLSTM layers along with a dropout of 0.1 in between these layer. Also, regularization with weight decay of value 0.02 were incorporated in the Convolutional layers.

Based on **Proposition 3.**, the described model was duplicated with exact same parameters. Both the models were trained for a total of 750 epochs on their respective data as split in **section 4.2**. The 2 models were trained parallely on 2 *Nvidia RTX 2080-Ti* and the total training time was approximately 37 hours for each model.

5 RESULTS

5.1 Generated Projections

Figure.9 shows one of the predicted projections of precipitation during the months of non-monsoon. It can be clearly observed that despite GCM having false projections over East-part of India, the model generates projections over the southern-part of India along with maintaining the extreme values.

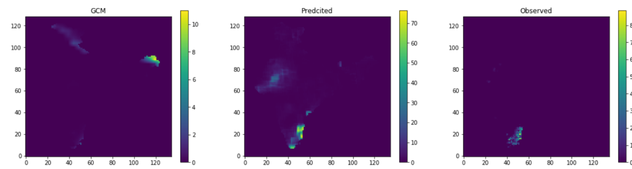

Figure 9: GCM vs Predicted vs Observed Precipitation of non-monsoon period

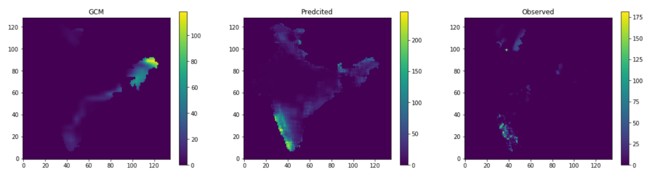

Figure 10: GCM vs Predicted vs Observed Precipitation of monsoon period

Similarly, **Figure.10** shows one of the predicted projection of precipitation which belongs to the monsoon period. Again, it can be seen that the actual rainfall is happening over the western ghats of India, similar to what was predicted by our model despite the GCM having projections over the Eastern part. However, it is also noticeable the slight shift in the extreme values.

5.2 Evaluation

RMSE Loss (mm)	Train	Test
Non-Monsoon Period	1.03	1.83
Monsoon Period	2.76	3.22

Table 2: Predictive Performance

The predictive performance of the model in terms of root-mean-square-error(RMSE) values are shown in **Table.2**. The model has performed remarkably well in the Non-monsoon periods with test accuracy of $1.83mm$ and even with Monsoon period, model has performed well enough with test RMSE of $3.22mm$ despite the period having more extremes over the southern region with large offset betweeen the GCM and the observed data.

6 CONCLUSION

My proposed model shows that inculcating both the spatial and the temporal dependencies, makes the projections more accurate in terms of predictive performance. However, the model still suffers from the problem of dealing with the extreme values while generating projections. Overall, the proposed model has very good predictive capability as observed by the losses. Still, current state-of-the-art DeepSD[8] needs to be scaled and need to be trained for Indian region for a comparison between both models. However, some crucial factors like having more auxiliary climatic variables, utilizing the temporal dependencies between those climatic variables and effective division between models for prediction on monsoon and non-monsoon makes my proposed model both reliable and have the potential to beat DeepSD in terms of performance. As far as scalability for other regions is concerned, the proposed architecture is very much scalable except the part of it's data distribution. Since, different regions would have different periods of monsoon, therefore, the data needs to be split accordingly. One other factor on which work needs to be done is inculcating the factor of climate change over longer periods. Due to a high computational cost associated with ConvLSTMs as compared to models with just CNNs, the effective sequence length(number of days as each training instance) that could be passed is short. So, the model does not take into account the long-term dependency such as the affect of precipitation of a year to the years ahead of it. Due to changing land usage patterns and climate change leading to changing distributions every decade[5], Inculcating this factor is important in-order for the model to get more generalized and adapted for the future years.

REFERENCES

[1] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2014. Image Super-Resolution Using Deep Convolutional Networks. arXiv:cs.CV/1501.00092

[2] Subimal Ghosh. 2012. SVM-PGSL coupled approach for statistical downscaling to predict rainfall from GCM output. *Journal of Geophysical Research* Volume 115 (2012). https://agupubs.onlinelibrary.wiley.com/doi/epdf/10.1029/2009JD013548

[3] Ethan D. Gutmann. 2012. A Comparison of Statistical and Dynamical Downscaling of Winter Precipitation over Complex Terrain. *Research Applications Laboratory, and Advanced Studies Program, National Center for Atmospheric Research* Volume 25 (2012). https://opensky.ucar.edu/islandora/object/articles%3A17101/datastream/PDF/download/citation.pdf

[4] Saptarshi Misra, Sudeshna Sarkar, and Pabitra Mitra. 2017. Statistical downscaling of precipitation using long short-term memory recurrent neural networks. *Theoretical and Applied Climatology* 134 (11 2017), 1–18. https://doi.org/10.1007/s00704-017-2307-2

[5] Kaustubh Salvi, Subimal Ghosh, and Auroop Ratan Ganguly. 2015. Credibility of statistical downscaling under nonstationary climate. *Climate Dynamics* 46 (2015), 1991–2023.

[6] Alex Sherstinsky. 2018. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. arXiv:cs.LG/1808.03314

[7] 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. arXiv:cs.CV/1506.04214

[8] Evan Kodra Ramakrishna Nemani Sangram Ganguly Auroop R Ganguly Thomas Vandal, Andrew Michaelis. 2017. DeepSD: Generating High Resolution Climate Change Projections through Single Image Super-Resolution. *In proceedings of* 23rd *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), ACM*, 1663–1672. https://doi.org/10.1145/3097983.3098004

[9] Kevin E. Trenberth. 2012. Framing the way to relate climate extremes to climate change. *Climatic Change* Volume 115, Issue 2 (2012). https://link.springer.com/article/10.1007/s10584-012-0441-5