Joint Learning for Cyclone Track Nowcasting

Abhijit Mukherjee[1] and Pabitra Mitra[2]

[1] Centre of Excellence in Artificial Intelligence. Indian Institute of Technology, Kharagpur. abhijit_m@iitkgp.ac.in

[2] Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur. pabitra@cse.iitkgp.ac.in

Abstract. Data driven approaches have been used extensively to predict extreme weather condition and used in conjunction with classical Numerical Weather Prediction models for a computationally inexpensive yet precise short-term forecasts. With machine learning advances and increase in computational power and efficiency, robust and precise short-term forecasting methods can be developed that finds wide range of usage from marine warnings to early warnings for extreme weather events. In this work we address the problem of *nowcasting* of a cyclone that has been known to be one of the deadliest extreme weather events. Few previous research from machine learning perspective have adequately addressed this issue. Accompanied by strong devastating winds and heavy rain, they are known to cause heavy damage upon landfall. Cyclone track prediction is important for mitigation of damages and early warnings. We propose a Joint Learning model that simultaneously predicts the distance and direction of a cyclone 6 hours in advance, given the past track data.

Keywords: Joint Learning · Cyclone Nowcasting · Deep Learning

1 Introduction

With the effects of climate change becoming more prevalent, machine learning plays a pivotal role in not only predicting extreme weather events but also working in conjunction with other fields [19] to both mitigate and adapt. While climate modelling remains expensive, deployment of satellites and simulated models have produced massive amount of data and with increase in computational efficiency along with improvements in GPUs and TPUs, climate scientists have adapted machine learning assisted techniques to improve the state-of-the-art forecasting methods. Cyclones are known to be a major devastating form of extreme weather event and data-driven methods have proven to successfully predict the cyclone tracks and intensity.

Cyclones are a strong system of wind rotating inwards towards a low pressure zone and are sometimes also referred to as *hurricanes* or *typhoons*. Studies by NOAA (National Oceanic and Atmospheric Administration) and researchers [5], [22] have predicted that owing to global warming, in the coming years we might experience an increase in storm intensity by 1-10%. They mainly form in the warm ocean waters near the equator due to strong Coriolis Force and affects countries in the tropical regions worldwide. With major destructive elements of the cyclone like high-pressure gradients and consequent strong winds resulting in storm surges and heavy rainfall (especially in coastal areas with shallow bathymetry), cyclones remain a ubiquitous form of hydro-meteorological disaster with massive destructive power, causing loss of lives and economy.

Thus prediction of cyclone track remains of paramount importance for early disaster warnings and mitigation purposes. But mercurial traits with the likes of sudden weakening, rapid intensification or phenomenons such as Fujiwhara Effect[6] and Trochoidal Effect cause significant change in cyclone trajectory, making the prediction of cyclone tracks quite arduous.

A crucial task for forecasting such high-impact weather conditions is short-term prediction (also referred as *nowcasting*) for extreme consequences of cyclones like heavy rainfall and flash floods to optimize protective measures. *Nowcasting*, as defined by World Metereological Society is "the detailed description of the current weather along with forecasts obtained by extrapolation for a period of 0–6 h ahead". Thus these predictions needs to be computationally efficient and fast. Our work addresses nowcasting of a cyclone track by predicting the **distance** and **direction** from it's last known position, 6 hours in advance by learning from the past track data of the storm.

Our primary contributions in this work can be summarized as:

- We propose a novel approach addressing the Nowcasting problem of determining the distance and direction of a cyclone from it's past location,
- The proposed model is lightweight and easily adaptable to output other required features like intensity, windspeed etc.

1.1 Related Work

Evolving from single-station approaches to methods involving meteorological tools, cyclone prediction models have come a long way. But while many cyclone prediction models like dynamical, statistical, statistical-dynamical and ensemble models have been used by NOAA, they remain either computationally too demanding or fail to capture the time-sequential dynamics between variables of natural events. For example, classic statistical models like BCD5 uses only 0D features like longitude, latitude, windspeed, J-day predictor, etc.

While in most cases statistical model driven approach is used for this purpose, recent disemmination of cyclone data collected from various meteorological sources coupled with an increased computational power of GPUs and TPUs facilitates the usage of a more data-driven approach in tackling this issue. Over the past few years machine learning models have been used to address the cyclone track prediction problem.

[11] designed Globenet that takes 3D satellite imagery as inputs and has *Complex CNN* (Convolutional Neural Network and Inception Units) to predict the location of a single typhoon center. [14] used Artificial Neural Networks on satellite images of a cyclone. [4] focused on cyclone eye detection based on

satellite imagery and used PCA for this purpose. [12] used Convolutional LSTM based model to predict Cyclone track. [20] used Generative Adverserial Network on satellite images of typhoons in Korean peninsula to predict the track. [8] used fusion of 3 different Neural Networks : the Wind CNN and the Pressure CNN take atmospheric fields as input while Past track + meta NN takes 0D features as input. Each network first learns its parameters independently and then they are combined and retrained again. Grid-Based RNN was proposed by [1] that employed stacked LSTM over a gridded plane derived from superimposing finer grids over latitude and longitude. A study using sparse RNN on trajectory data was tested on 6h and 12h forecast for 4 hurricanes [15]. Another research used storm tracks and reanalysis maps for a hybrid ConvNet-LSTM network for learning the coordinates and output 6h forecast result [16].

While numerous aforementioned methods have been employed on cyclone track data for forecasting, the number of models deployed to address the problem of *nowcasting* remains limited. Also the prediction of distance and direction of a storm simultaneously, thereby giving the precise location of a storm from it's last known position has not been addressed directly. The proposed approach thus aims to address *Cyclone nowcasting* by predicting the distance and direction of a cyclone given its past track data. A demand for higher precision and resolution as compared to traditional forecasting models makes the problem of *nowcasting* challenging. To the best of our knowledge this work presents a method to learn the precise location of a cyclone while learning from it's past data by a Joint Learning approach for the first time.

1.2 Overview of proposed approach

While Convolutional LSTMs have been used previously [12], our proposed approach takes the Joint Learning approach to address the *Nowcasting* issue with high precision, predicting the **distance** and **direction** of a storm 6 hours in advance. With the shared parameters in the initial layers, our method employs two different paths involving LSTMs along with Fully Connected layers to learn the distance and direction of a cyclone simultaneously. The joint learning approach lets us not only address the problem of predicting the distance and direction of a cyclone with high accuracy but also helps us utilize the intricacies of dynamic behaviour of the cyclones embedded in the data.

2 Dataset

The dataset is taken from [7] and in it's raw form comprises of more than 3000 tropical and extra-tropical storm tracks from both North and South hemispheres recorded since 1979, collected from NOAA database IBTrACS (Fig. 1). Each storm track is sampled at a frequency of 6 hourly centre location in terms of latitude and longitude. The number of records per storm varies from 2 to 120 time steps, containing more than 90000 time steps overall. Some 0D features are added to the data and comprises of latitude, longitude, maximum windspeed

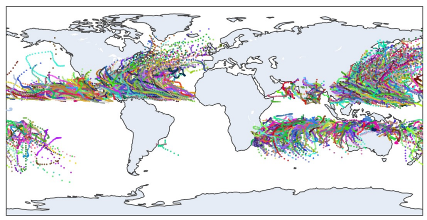

Fig. 1. Global distribution of cyclones in dataset

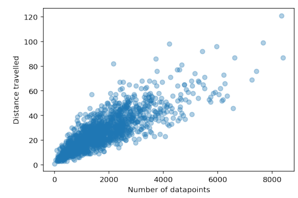

Fig. 2. Correlation between *Number of Datapoints* vs *Distance travelled* for cyclones

change in the last 12 hours, Jday predictor (Gaussian function of (Julian day of storm init - peak day of the hurricane season)), basin of the current location and distance to land.

The reanalysis data is extracted based on the storm centre's present location for every time step t, consisting of: 3 maps at 700hPa level with resolution of 25x25 degrees subsampled to 11x11 pixels where 1 pixel = 2 degrees (altitude z, u-wind, v-wind), sea surface temperature, sea level pressure, humidity and vorticity at 700hPa with resolution of 11x11 degrees where 1 pixel = 1 degree.

Predicting the distance and direction of a cyclone can prove to be of immense help for averting the losses incurred from cyclone landfall. Thus for each storm we have augmented the distance traveled and direction (angle of travelled path), as stated in [1].

We observe a correlation between distance travelled and the number of datapoints for each storm (Fig. 2), which although not explicitly evident from the data, motivated us in utilizing recurrent units to predict direction and distance of travel.

3 Methodology

3.1 Preprocessing

Besides improving the data quality, preprocessing raw data also helps in removing outliers and deal with missing values. The dataset comprises of latitude and longitude but it is easier for a neural network to learn from distance vectors since these values do not have negative values and also allows the network to learn from relative rather than absolute parameters. Thus we have added two new columns in our data with the angle of travel and the distance[1]. This has in turn also helped us to remove some outliers to give the data a normal distribution, which has been shown to help RNNs to learn faster, generalize better and converge significantly faster[3].

We have standardized the data by removing the mean and scaling it to unit variance, giving it a normal distribution. *Standard Scaler* [18] from sklearn has been used for this process. We prefer *Standard Scaler* over *MinMax Scaler* so as not to suppress the effects of outliers in the data.

The two major components used in this model are as follows:

3.2 Convolutional Neural Network

At its heart is the *convolution* operation which is a linear operation of multiplying a set of weights with the input. Using a filter size smaller than the input data, detection of a specific type of pattern across the input becomes easy. This operation produces an abstraction of the input called a *feature map*.

Convolution has the advantage of sparse interactions, parameter sharing and equivalent representation. The network can efficiently learn complicated interactions between variables, making it more efficient than matrix multiplication and learn different features appearing in the input from a timeline, produced by convolution [9]. We thus utilize this feature extracting property of Convolutional Layer to act as a mediator for shared parameters in two separate networks to learn distance and direction.

3.3 LSTM / Recurrent element

Operating over a sequence of vectors, recurrent units have the property of conserving information within states and has proven to be conducive in modelling temporal dynamic behaviour for time sequences by utilizing their memory (internal states). But vanilla RNNs tend to work well for shorter sequences or dependencies and are unable to connect information over long term dependencies. Hence LSTMs come to the rescue with it's inherent design to handle long-term dependencies [10] while avoiding vanishing and exploding gradients. The major advantage of LSTM is the memory cell c_t that accumulates state information. It comprises of: Forget Gate : Essentially a sigmoid layer, this decides on the information to be conserved from previous states

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{1}$$

Input Gate : This step decides what new information is to be stored in the cell state. It comprises of two parts : first a *sigmoid* layer, also called *input gate layer* gives the values to be updated. This is followed by a *tanh* layer that creates a vector $\tilde{C}_t$ of a new candidate value that is to be added to the next state.

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{2}$$

$$\tilde{C}_t = tanh\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{3}$$

Update stage : This step updates the old state $[C_{t-1}]$ into the new state $[C_t]$.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

Output stage : We obtain the output based on our cell state.

$$o_t = \sigma\left(W_0 \cdot [h_{t-1}, x_t] + b_0\right) \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

LSTMs have found widespread application in tasks involving sequence learning. In our model, this forms an essential element in learning the trajectory from the past records of a given storm.

3.4 Joint Learning

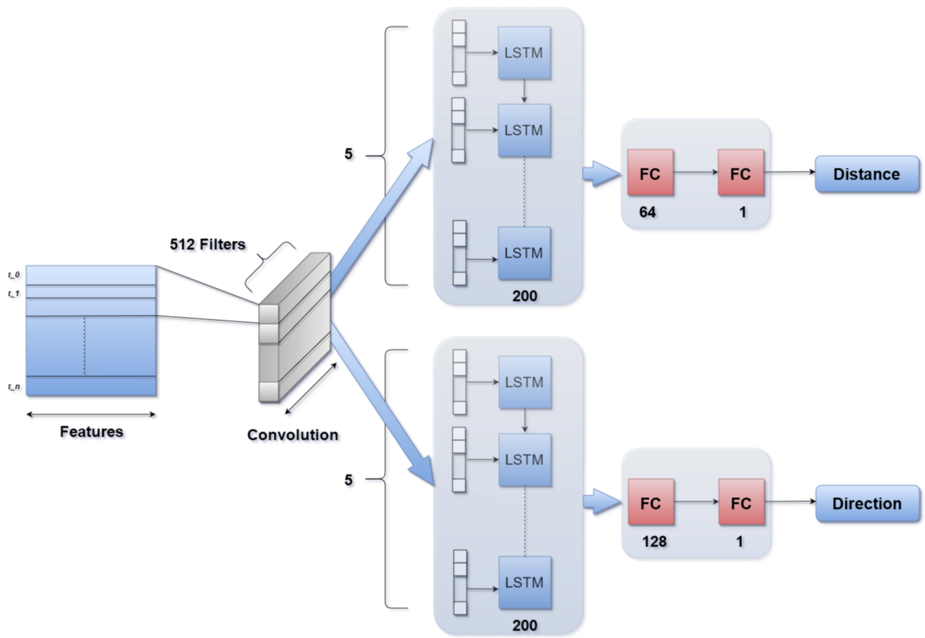

Fig. 3. Joint Learning Model for Cyclone track Nowcasting by predicting *Distance* and *Direction* of a Cyclone together

While handling complicated tasks, deep architectures perform much better when compared to shallow networks[2] because of multiple levels of non-linear operations.

LSTMs performs temporal modelling on the input features directly. It has been shown by [17] that LSTM learns better temporal structure from a higher level of abstraction. The major drawback of Fully Connected-LSTM while dealing with spatiotemporal data is the usage of full connections in input-to-state and state-to-state transitions in which no spatial information is encoded [23].

Therefore to reduce variance in frequency as stated in [21], we feed the data into CNN before passing it to LSTM, thus enabling a better modelling of the spatiotemporal relation. The proposed model has a Convolutional Layer for feature extraction from the input data while LSTM handles the sequence prediction task across time steps. Sharing a part of the model across tasks constrains that part towards better values and thus a better generalization, as stated in [9].

The model can be segmented primarily into two parts: Task specific parameters (*LSTM + Dense Layer*) and Generic parameters, shared across all the tasks (*Convolution Layer*).

The input data contains storm track of 1481 storms (filtered out of 3000 storm tracks) and reshaped to contain 6193 samples, 6 time steps and 856 features.

The proposed model starts by passing Input data (x) into CNN. The Convolution layer has 512 filters (with each filter of size 5), kernel_size of 2 and a stride of 1. The *activation* is defaulted to *linear* as it allows multiple outputs when compared to other types of activation in the model and works more as a linear regressor. This provides us a generic embedding of the input features.

This output C_{out} is then fed into two separate LSTMs (R_1 and R_2), each with 200 cell states. The *activation* is kept as *tanh*(hyperbolic tangent), that bounds the output within (-1, 1). LSTM instead of working on raw time steps is now operating on a higher level of abstraction of input time steps. These LSTM cells are now able to structure the temporal behaviour of the data.

$$x_1 = \sigma\left(W_0 \circledast [h_{t-1}, C_{out}] + b_0\right) \tag{7}$$

$$x_2 = \sigma\left(W_0 \circledast [h_{t-1}, C_{out}] + b_0\right) \tag{8}$$

. $\circledast$ denotes the Convolution operator.

Finally we keep two Fully Connected layers, with size 64 and 128 respectively followed by two separate Dense Layers of size 1 for the output.

We finally obtain the *Distance* and *Direction* from the two separate paths respectively.

$$Distance = tanh(w_{1_2}(tanh(w_{1_1}x_1 + b_{1_1})) + b_{1_2}) \tag{9}$$

$$Direction = tanh(w_{2_2}(tanh(w_{2_1}x_2 + b_{2_1})) + b_{2_2}) \tag{10}$$

We feed the network data from previous 5 time steps to forecast the next location of the cyclone.

From the same set of weights and shared parameters, a model cannot be expected to perform well for predicting two separate outputs. That is why we keep two separate networks to give a generalized embedding which helps in predicting two different outputs i.e. distance and direction.

4 Results

4.1 Experimental Setup

For realizing the proposed method we have used Keras, that is an API integrating lower-level languages like Tensorflow. With over 250000 users as of mid-2018, Keras has been used in research and industry alike and is been used in Uber, Netflix, Yelp, etc including large organizations with the likes of CERN and NASA. By using Keras, the designed model is pretty quick to train thus saving a lot of time for hyperparameter tuning.

The learning rate has been kept as 0.0001. Adam optimizer is used for the model; as stated in [13], it is well suited for non-stationary objectives and computationally efficient. This model was trained on a machine with 8 GB RAM and 4 GB NVIDIA Ge1050Ti and has taken 119.639 seconds to train successfully.

Huber Loss is being used to evaluate the model's performance. We have taken both Mean Squared Error and Minimum Absolute Error as a performance metric to evaluate our model. The dataset was split into 75% for training purpose while 25% data was kept for validation purpose. The test dataset comprised of 739 storm track data. The number of trainable parameters are 2,056,642.

4.2 Comparison

Since the joint prediction of distance and direction of a storm has not been adequately addressed before, we primarily keep our comparisons to a few models we tried out. We have also used the Stacked LSTM concept proposed in [1] in the essence that the outputs are modified to have 2 outputs simultaneously (thereby avoiding the error inherent from resolution of 1x1 degrees latitude by longitude).

We show a table of the aforementioned methods in comparison to ours.

Table 1. Model Comparison

	Distance		*Direction*	
Methods	*MSE*	*MAE*	*MSE*	*MAE*
Conv + Stacked GRU	6.982	2.454	2.539	1.181
Conv + Stacked LSTM	10.355	2.816	4.494	1.832
Stacked LSTM	0.550	0.580	1.663	0.978
Joint Learning	**0.270**	**0.390**	**1.561**	**0.928**

Our model can be observed to give significant better performance when compared to other methods from Table 1.

As an adage it can also be seen that stacking multiple recurrent units require more trainable parameters and are more susceptible to vanishing gradient, thus unable to produce good result over this dataset. The proposed model is computationally more efficient as compared to other models.

5 Conclusion

In this work we focused on the *nowcasting* problem for a cyclone track. The model can be integrated with present consensus methods for providing a more accurate yet computationally efficient short-term forecasting method. With some

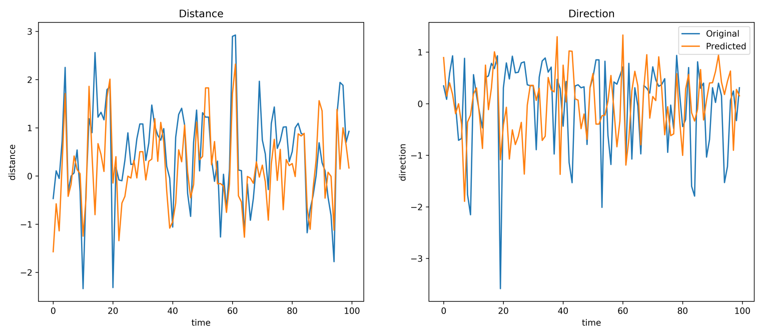

Fig. 4. Distance and Direction MAE obtained

significant improvement over some past methods used in predicting the cyclone track, the proposed model is computationally less expensive and shows more promising result in predicting the precise location of a cyclone 6 hours ahead, given it's past trajectory record.

While our method takes a joint learning approach, further enhancements to the method might be possible with physics guided model that are consistent with physical laws.

We would like to extend our gratitude to Department of Science and Technology (DST), India for sponsoring this work as part of "Machine Learning for Climate Change Analysis" project undertaken by Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur.

Also we would like to thank Sophie Giffard Roisin for providing the dataset used in this study.

References

1. Alemany, S., Beltran, J., Perez, A., Ganzfried, S.: Predicting hurricane trajectories using a recurrent neural network. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 468–475 (2019)
2. Bengio, Y., et al.: Learning deep architectures for ai. Foundations and trends® in Machine Learning **2**(1), 1–127 (2009)
3. Cooijmans, T., Ballas, N., Laurent, C., Gülçehre, Ç., Courville, A.: Recurrent batch normalization. arXiv preprint arXiv:1603.09025 (2016)
4. DeMaria, R., Chirokova, G., Knaff, J., Dostalek, J.: P189 machine learning algorithms for tropical cyclone center fixing and eye detection
5. Emanuel, K.: Will global warming make hurricane forecasting more difficult? Bulletin of the American Meteorological Society **98**(3), 495–501 (2017)
6. Fujiwhara, S.: The natural tendency towards symmetry of motion and its application as a principle in meteorology. Quarterly Journal of the Royal Meteorological Society **47**(200), 287–292 (1921)

7. Giffard-Roisin, S., Gagne, D., Boucaud, A., Kégl, B., Yang, M., Charpiat, G., Monteleoni, C.: The 2018 climate informatics hackathon: Hurricane intensity forecast (2018)
8. Giffard-Roisin, S., Yang, M., Charpiat, G., Kégl, B., Monteleoni, C.: Deep learning for hurricane track forecasting from aligned spatio-temporal climate datasets (2018)
9. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
10. Hochreiter, S., Schmidhuber, J.: Lstm can solve hard long time lag problems. In: Advances in neural information processing systems. pp. 473–479 (1997)
11. Hong, S., Kim, S., Joh, M., Song, S.k.: Globenet: Convolutional neural networks for typhoon eye tracking from remote sensing imagery. arXiv preprint arXiv:1708.03417 (2017)
12. Kim, S., Kim, H., Lee, J., Yoon, S., Kahou, S.E., Kashinath, K., Prabhat, M.: Deep-hurricane-tracker: Tracking and forecasting extreme climate events. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1761–1769. IEEE (2019)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
14. Kovordányi, R., Roy, C.: Cyclone track forecasting based on satellite images using artificial neural networks. ISPRS Journal of Photogrammetry and Remote Sensing **64**(6), 513–521 (2009)
15. Moradi Kordmahalleh, M., Gorji Sefidmazgi, M., Homaifar, A.: A sparse recurrent neural network for trajectory prediction of atlantic hurricanes. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016. pp. 957–964 (2016)
16. Mudigonda, M., Kim, S., Mahesh, A., Kahou, S., Kashinath, K., Williams, D., Michalski, V., O'Brien, T., Prabhat, M.: Segmenting and tracking extreme climate events using neural networks. In: Deep Learning for Physical Sciences (DLPS) Workshop, held with NIPS Conference (2017)
17. Pascanu, R., Gulcehre, C., Cho, K., Bengio, Y.: How to construct deep recurrent neural networks. arXiv preprint arXiv:1312.6026 (2013)
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
19. Rolnick, D., Donti, P.L., Kaack, L.H., Kochanski, K., Lacoste, A., Sankaran, K., Ross, A.S., Milojevic-Dupont, N., Jaques, N., Waldman-Brown, A., et al.: Tackling climate change with machine learning. arXiv preprint arXiv:1906.05433 (2019)
20. Rüttgers, M., Lee, S., Jeon, S., You, D.: Prediction of a typhoon track using a generative adversarial network and satellite images. Scientific reports **9**(1), 1–15 (2019)
21. Sainath, T.N., Vinyals, O., Senior, A., Sak, H.: Convolutional, long short-term memory, fully connected deep neural networks. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4580–4584. IEEE (2015)
22. Schulthess, T.C., Bauer, P., Wedi, N., Fuhrer, O., Hoefler, T., Schär, C.: Reflecting on the goal and baseline for exascale computing: a roadmap based on weather and climate simulations. Computing in Science & Engineering **21**(1), 30–41 (2018)
23. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems. pp. 802–810 (2015)