Weather Prediction Using Deep Learning Techniques

Meera Suthar, Yash Thesia, Vidhey Oza

Department of Computer Engineering, Nirma University, Ahmedabad, Gujarat, India

15bce120@nirmauni.ac.in, 15bce126@nirmauni.ac.in, 15bce130@nirmauni.ac.in

***Abstract* – This term paper surveys various research papers aimed at forecasting weather. We explore cutting-edge research towards deep learning algorithms and technology. Convolutional networks help preserve information of data points in physical or logical proximity when placed together on a matrix, and help reduce the computation size using the concept of 2-D filters, for example precipitation levels of a location based on levels of neighboring cities or towns. Recurrent networks help preserve information gained with past iterations of training, and are thus much more flexible when trying to analyze time-series data, like today's precipitation levels based on past precipitation levels. And autoencoders help overcome overfitting by extracting useful features from the given data, which also helps reduce the vanishing gradient problem.**

Introduction

In the past several decades, various statistical models have been used to forecast weather and climate. But as each year passes by, the meteorological community receives tens of terabytes of data every year, from various satellites and ground-based sensors. With this data being generated for locations around the world, analysis of this data is important not only from a local, but also from a global perspective. And with the looming threat of global warming, it is of alarming importance for governments as well as corporations to make sense of such vast volumes of data, and use it for several purposes from optimisation of energy consumption to efficient evacuation strategies for extreme weather forecasts.

With increasing interest towards machine learning and deep learning techniques to analyse data of all kinds, we focus our research in surveying methods proposed by researchers around the world to accurately predict various aspects of weather and climate, from nowcasting to extreme weather forecasting. Our analysis consists of two factors: data processing, and prediction algorithms.

Generally, the data used in many of these research papers are spatio-temporal data [1,2,5,6,7,8,12], meaning they are data either extracted from a matrix of sensors plotted on an image, or images generated by satellites. These datasets can be used to forecast various factors like wind speed and direction (for wind energy harvesting), but our main goal is to forecast precipitation levels for the next 1-5 days.

Also, several research papers lead us towards using algorithms like moving averages [10,11,12] or recurrent neural networks [2,4,10,12,15] for optimal time-series analysis.

Data Processing

Generally, the dataset used in research papers was generated by satellites in latitude-longitude matrix images [1,2,5,6,7,8,12]. These datasets contain values of meteorological variables like relative humidity, precipitation levels, wind speed, and minimum and maximum temperatures, received from sensors installed at various locations that can be mapped on a lat-long matrix, along with their elevation levels.

For any machine learning algorithm, it is of utmost importance to use data that is curated for the purpose of the prediction. Some of the important factors are: filling missing values, effectively managing outlier data and normalising data for avoiding bias towards a particular feature or set of features. And when generating images from numerical data, the lat-long positioning and elevation levels should be taken into account, for seamless mapping of the variables on the image matrix.

In [4], the authors used 15-year data collected by Wunderground API for Morocco, Tangier. They faced the problem of missing values as well as outlier data in multiple data points of their dataset. They tackled this by replacing the missing or noisy values by "forward filling", i.e. using the last recorded values in time. Then they normalised the data using mean normalization, a variation of rescaling (*Eq. 2*).

In [3], the data used by authors was taken from Meteorological Terminal Aviation Routine (METAR) weather reports of 57 stations in east coast including Massachusetts, Connecticut, New York, and New Hampshire. It was hourly data generated from January 6 to February 20, 2014. The dataset had missing values between every time step. For example, for a time-step of 6 hours, data is available for every 6 hours, but for every hour between those time-steps, data is not available for one node. For this, they forecast data using previous time-steps.

In [10], data from 2012 to 2016 was procured from Wunderground API from all over the world, making a total dataset of over 40,000 time-series data points. The authors used Z-score normalization *(Eq. 1)* and rescaling *(Eq. 2)*, and data smoothing using moving averages *(Eq. 3)*.

$$x_t = \frac{x_t - \bar{\bar{x}}}{s_x} \tag{1}$$

$$x_t' = \frac{x_t - xmin}{xmax - xmin} \tag{2}$$

$$x_t'' = \frac{1}{9}(x_t' + x_{t-1}' + \cdots + x_{t-8}') \tag{3}$$

For prediction of extreme weather conditions, research generally also includes data created by simulations. In our analysis, it seems like using data from simulations adds on to the learning process by using thresholds created by meteorological experts. In [7], the dataset was labelled using threshold-based criteria implemented by Toolkit for Extreme Climate Analysis (TECA), and also manual labelling by domain experts.

For using image data from satellites or from numerical data, one of the more important questions that arise is the idea for padding the images to maintain the spatial resolution while convolving. The authors of [2] defend their choice of zero-padding throughout their convolutional network by using the analogy of a closed box: a ball bouncing in a closed box can detect the walls by colliding with them, which cannot be seen otherwise.

The concept of moving averages is further explored to a great extent in [11]. Apart from simple MA algorithm, they discuss Centered MA, Weighted MA and Modified Weighted MA, which are all used for data smoothing using different intuitive algorithms that build on one another.

Prediction Algorithms

1. Convolutional Networks

Convolutional neural networks (CNNs) are generally used for predictions involving image data, but mainly include 2D data of any kind with local dependencies. CNNs were developed as a solution to many problems faced while implementing vanilla networks for image processing. Its salient features include:

1. **Feature Engineering**: The idea of feature extractors or filters was introduced, which was essentially a set of weights in a matrix that would convolve over the input image to generate an output image with features detected by weights in local dependencies.

2. **Preservation of Structural Information**: As the filters are 2D, there is no loss of structural information in the image.

3. **Pooling Layers**: Apart from basic convolutional layers, special pooling layers are used to reduce the dimensionality of the image while also maintaining the information from the original image.

4. **Dropout**: To reduce the Vanishing Gradient Descent problem and getting more general features from the image, we randomly switch off neurons in a particular layer based on probability P during learning, and use all the neurons during validation with the same probability P.

In [5], the authors used multiple models for predicting the classes of weather based on values of variables like temperature and humidity. They used CNN proposed by LeCun, simple RNN, and Conditional RBM, used majority vote for prediction and compared the results using K-fold cross validation. As their problem was solved with unsupervised learning, they used Predictive Sparse Decomposition Algorithm, a common method for the same in CNNs.

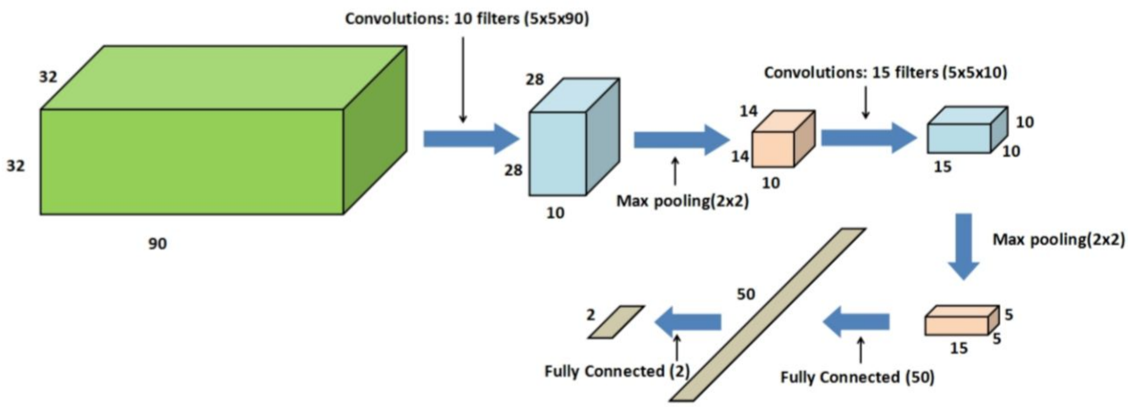

Fig. 1: CNN architecture as designed in [6].

Research documented in [6] is based on prediction of extreme precipitation cluster using a spatio-temporal dataset. Their network is neatly described in *Fig. 1*. The network consists of 2 convolution layers (28x28x10 and 5x5x10) each followed by a 2x2 max pooling layer, which preserve the patterns of the convolutional layers while reducing the image to a quarter of the original size. The network has two fully connected layers in the end, the last being 2 softmax outputs of extreme vs. non-extreme precipitation cluster probabilities.

Dynamic Convolution Layer:

A dynamic convolution layer was presented in [1], in which filters will vary according to input. The authors try to predict the weather variables in short-range. After using dynamic convolution layer, increase in results is observed. In this technique, the function to be learned is input to filters. Radar images of rain and snow are used as an input. Next predicted radar image of rain and snow is being output of this architecture. For this prediction short term data is used.

The dynamic convolution layer has two different inputs for applying kernel on feature maps. It takes kernel as an input for applying as a filter on image from a different sub-network. A sub-network which consists of different convolution layers take image as input and gives filter as an output for that corresponding input image. This is the reason why filters are changed for different inputs. The second input given to dynamic convolution layer is feature maps obtained from previous layer. So, the second input is same as traditional convolution layer. The only change is that filters are taken as an input. The kernel given as an input is applied on the input feature map and output feature map is obtained with traditional method. In *Fig. 2*, network B indicates the sub-network which gives filter as an output.

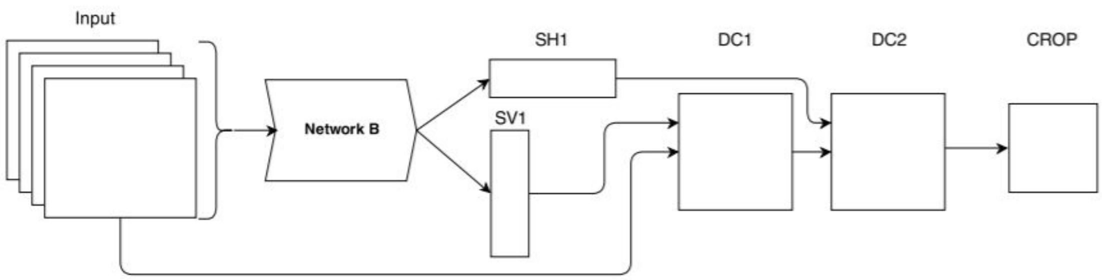

Fig. 2: CNN with Dynamic Convolutional Layer, as in [1]. Three convolution layers and three max pooling layers; followed by two fully connected layers are used for converting output feature map from pooling layer into 1D horizontal vector and 1D vertical vector.

Traditional convolutional layer:
Forward pass: Output feature map is computed by applying kernels on input during forward pass of training.
Backward pass: In backward pass, loss function is calculated at output layer and then that loss is propagated to previous layer. Generally gradient descent approach is used in traditional convolution network to propagate loss and to update weights and biases of previous layer during training.

Dynamic convolutional layer:
Forward pass: Two inputs are given as mentioned during forward pass. And corresponding to that input output feature map is computed.

Backward pass: As mentioned, fixed filters are not used for our layer. Filters will be changed for every image. So during back propagation, there is no need of updating parameters of filters. Instead the output of the loss function will be propagated to the sub-network which computes the filter as an output based on input. So, that sub-network will learn new parameters during training. So the weights and biases of sub-network will be updated.

Accuracy of the model is increased as compared to conventional CNN. All the filters used were totally based on input. So the filter values will be different for each and every input and those values are more input oriented. That implies increase in accuracy.

Research towards extreme weather forecasting usually use features like wind speed on the image matrix to predict either that these conditions can lead to extreme weather or not [7], or they are presently classifiable as extreme[6,7]. In [2,8,12], sequence of images were added to explore new directions and wind flow with forecasting weather and through that they predict weather value(s).

2. Time-Series Analysis

In predicting the weather, two types of information is important. First is short-term data which is taken into consideration to predict the temperature of next hour or next day. And long term data is useful when we want to find or want to analyse the general trend of every year.

Recurrent neural networks (RNNs) are useful when we are dealing with time-series data. An RNN remembers the important information which was fed earlier in the network. So these types of network are perfectly suitable for time series data. But the drawbacks of RNN are gradient vanishing and exploding. So, it cannot handle data effectively when our network is deep. This drawback can be avoided using LSTM (Long Short-Term Memory).

The two aspects of information discussed above can be used simultaneously using LSTM network. And change in accuracy can be observed. LSTM can remember important information during long time in network. Which information is important and which information is not can be decided by gates in LSTM. It has mainly three gates; forget gate, input gate and output gate. Sigmoid function is applied on each gate in which 0 indicates not important and 1 indicates information to keep. And by controlling all these gates, decision is taken whether information will be kept or not.

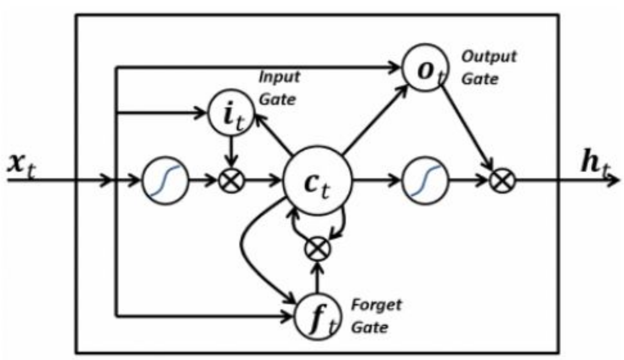

Fig. 3: Abstract architecture of an LSTM node.

LSTM based-approach for weather prediction is introduced [4]. As mentioned both short and long term information is useful for predicting the temperature. Here, the model consists of mainly three layers; two LSTM and one fully connected layer consisting 100 neurons. And the activation function chosen is rectifier function. So the model takes input of last 24 hours data and predicts the output for next 24 hours. Two LSTM layers are used and one dense layer is placed in between. The optimizer function used was "RMSprop", And mean squared error (MSE) was used as the loss function.

In [10], the authors used LSTM and have taken 100-frame time window for prediction. The network architecture is LSTM with subsequent layers having 200, 100, 90, and 50 hidden node. The last part of the model is a fully connected and having single output node. and survey paper[12], they have done comparative analysis of time series models like ensemble models, ARIMA and RNN networks and done comparison between their accuracy.

A novel approach is presented in [2,15] as Convolutional LSTM. ConvLSTM identifies the next state of current state. It takes two inputs, the first one is actual input and the second is input from the previous states of its neighbours. So, the advantages of convolutional layers (preservation of structural information) and LSTM layers (preservation of past information) are both used together. Their idea is first used on a synthetic Moving-MNIST dataset, where position of digits after 10 frames is predicted with input as previous 10 frames. The architecture of a single ConvLSTM node is shown in *Fig. 4*. This idea is also explored for other domain applications in [14].

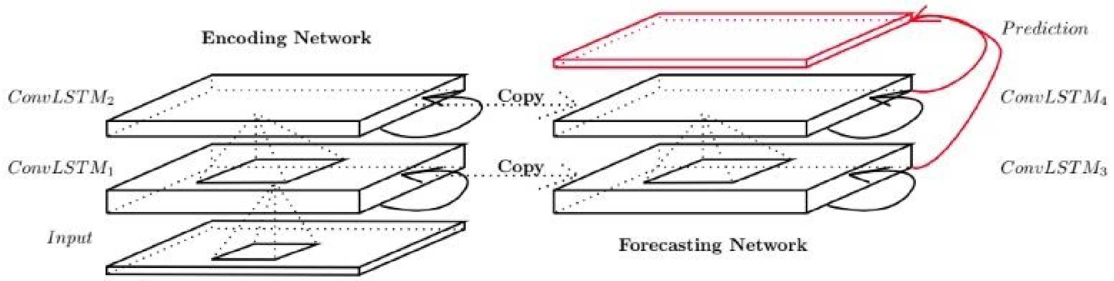

Fig. 4: Groundbreaking architecture of ConvLSTM node, as presented in [2].

3. Feature Extraction using Autoencoders

The Autoencoder is an unsupervised neural network which reconstructs the input at output layer. It consists of three layers: input layer, hidden layer (encoding layer) and output layer (decoding layer). It also belongs to the family of PCA. But because of its architecture it can also be considered as a neural network.

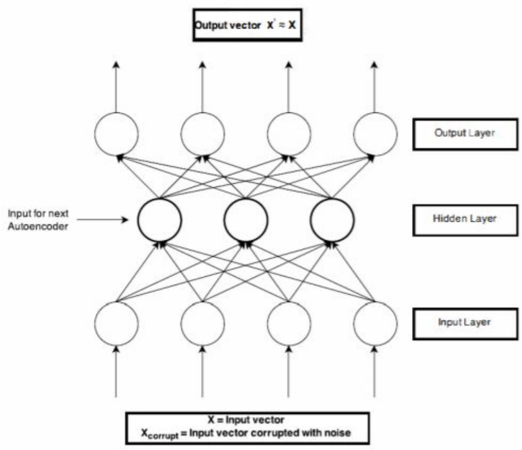

Fig. 5: Abstract architecture of an autoencoder network.

Here, input is reconstructed at output layer so the task of the hidden layer is to learn the code of given input. Number of neurons at hidden layer is lesser than previous layer. Autoencoder learns to approximate the identity function at the training. Denoising autoencoders are used to predict the weather with the help of given previous data [9].

Weather prediction is a challenging task because weather changes dynamically. In this type of autoencoder, inputs are given randomly with a little noise added into it. Corrupted inputs are given because we want auto-encoders to learn more robust features. So, autoencoders are trained to reconstruct the input from corrupted input. There exists different ways such as Gaussian noise, masking noise, etc to corrupt the input data.

The final step of training is to fine-tune the network with supervised learning. Stacked denoising autoencoders are used to predict the weather [9]. Weather is predicted using multiple parameters like air pressure, temperature, humidity, wind direction, etc. Last 24 hours data of temperature is taken and a feature matrix is formed. First 24 columns represents the data collected of 24 hours and 25th column represents the predicted value of 25th hour for each entry. They also took some other variables like barometric pressure, wind speed and relative humidity into consideration to create another feature matrix [9]. All the values are normalised using (*Eq. 2)*. Values of four variables of past 24 hours are considered, so the input size is 24 x 4. And based on that, the temperature of 25th hour is predicted.

The results show that stacked denoising autoencoders gives us higher accuracy as compared to simple artificial neural network.

Conclusion

Weather prediction is very crucial, will stay so for as long as we are willing to live and thrive on earth. And now, with unprecedented volumes of data being available to us every year, it is no more a distant future where weather can predicted on a short-term scale of hours as well as long term scale of months. Today, with the amount of data available and the value of research done everyday on machine learning and deep learning for pattern recognition, all that is left for us to do is explore them, and come up with as many different ways to predict weather as we can.

This term paper surveys various research papers aimed at doing the same. Leaving behind basic and traditional machine learning algorithms like SVM or Random Forest, we explore cutting-edge research towards deep learning algorithms and technology. Convolutional networks help preserve information of data points in physical or logical proximity when placed together on a matrix, and help reduce the computation size using the concept of 2-D filters, for example precipitation levels of a location based on levels of neighboring cities or towns. Recurrent networks help preserve information gained with past iterations of training, and are thus much more flexible when trying to analyze time-series data, like today's precipitation levels based on past precipitation levels. And autoencoders help overcome overfitting by extracting useful features from the given data, which also helps reduce the vanishing gradient problem.

Future Work

From the previous weather data like humidity, max. and min. temperature and wind speed, solar radiation, we can draw multiple graphs and find the relation between them. We observe that for the fixed area, there is very high probability that the precipitation value is going to repeat for next year. Fig. 6. shows an example of how this can be done for precipitation and max. temperature, and Fig. 7 shows how this can be done for location and precipitation. So, proper pre-processing of the features leads to high confidence value.

In socio-templeral matrix sensor data, we can combine all data for each date and make one matrix which contains all extracted feature, and apply time series prediction model like ConvLSTM, or a similar RCNN architecture which maintains the location image information to find the movement of the weather which will lead to good accuracy.

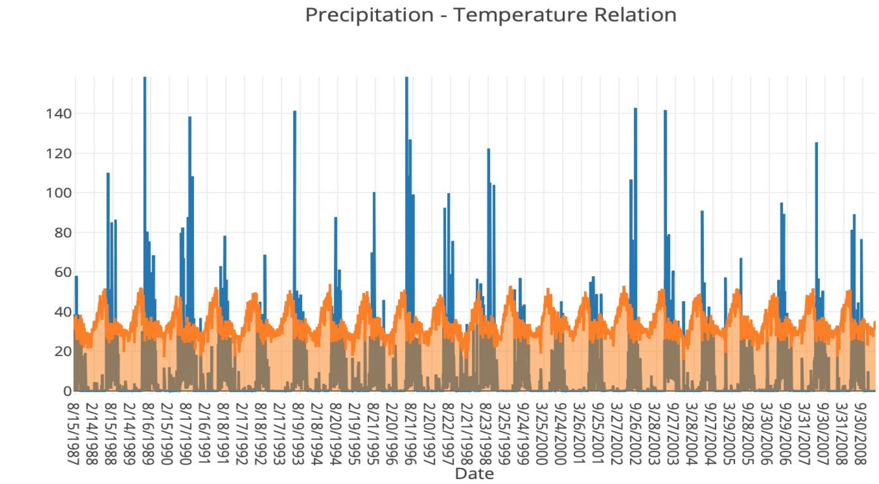

Fig. 6: Precipitation in mm (blue) and Temperature in degrees Celsius (orange). It is apparent that with each spike in temperature there is an immediate or next-to-immediate spike in precipitation levels.

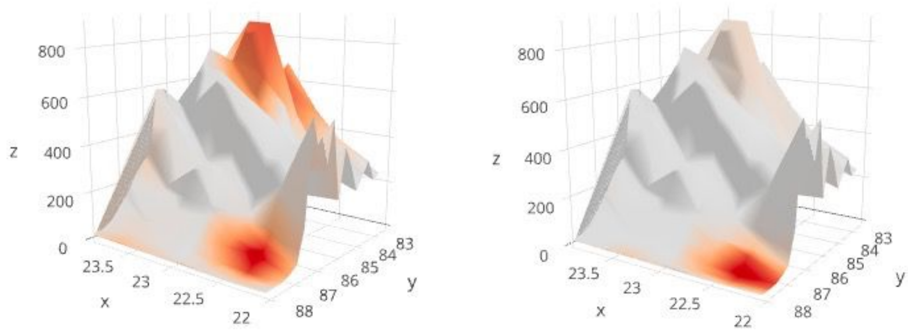

Fig. 7: Precipitation clusters near a hill on 2nd and 3rd July 1979 respectively. The intensity in red denotes the amount of precipitation in that area.

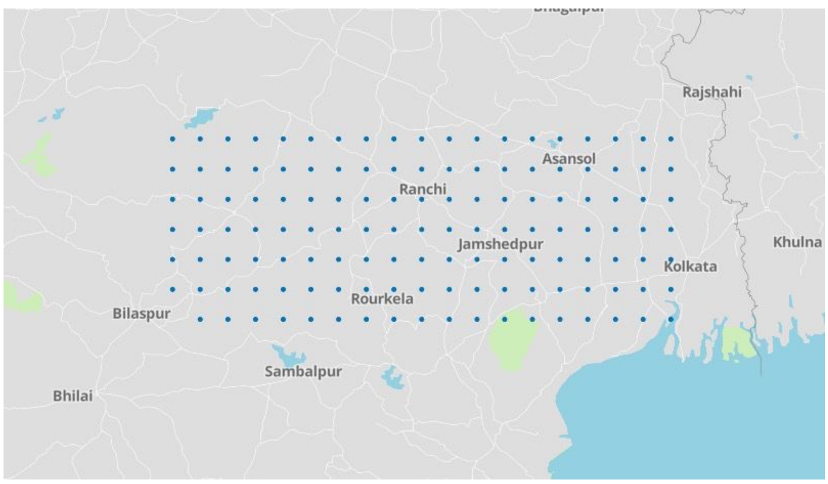

Fig. 8: The locations of sensors used to detect various meteorological variables for generating the data (blue) shown in Fig. 7 and 8. Obtained from ISRO with help of Dr. Sanjay Garg.

References

1. B. Klein, L. Wolf and Y. Afek, "A Dynamic Convolutional Layer for short rangeweather prediction," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 4840-4848.
2. Xingjian, S. H. I., et al. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." *Advances in neural information processing systems*. 2015.
3. Ghaderi, Amir, Borhan M. Sanandaji, and Faezeh Ghaderi. "Deep forecast: deep learning-based spatio-temporal forecasting." *arXiv preprint arXiv:1707.08110* (2017).
4. Zaytar, Mohamed Akram, and Chaker El Amrani. "Sequence to sequence weather forecasting with long short term memory recurrent neural networks." *Int J Comput Appl* 143.11 (2016).
5. Salman, Afan Galih, Bayu Kanigoro, and Yaya Heryadi. "Weather forecasting using deep learning techniques." *Advanced Computer Science and Information Systems (ICACSIS), 2015 International Conference on*. IEEE, 2015.
6. Zhuang, W. Y., and Wei Ding. "Long-lead prediction of extreme precipitation cluster via a spatiotemporal convolutional neural network." *Proceedings of the 6th International Workshop on Climate Informatics: CI*. 2016.
7. Liu, Yunjie, et al. "Application of deep convolutional neural networks for detecting extreme weather in climate datasets." *arXiv preprint arXiv:1605.01156* (2016).
8. Grover, Aditya, Ashish Kapoor, and Eric Horvitz. "A deep hybrid model for weather forecasting." *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015.
9. Hossain, Moinul, et al. "Forecasting the weather of Nevada: A deep learning approach." *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015.
10. Salman, Afan Galih, et al. "Weather Forecasting Using Merged Long Short-term Memory Model." *Bulletin of Electrical Engineering and Informatics* 7.3 (2018): 377-385.
11. Nhita, Fhira, Deni Saepudin, and Untari Novia Wisesty. "Comparative Study of Moving Average on Rainfall Time Series Data for Rainfall Forecasting Based on Evolving Neural Network Classifier." *Computational and Business Intelligence (ISCBI), 2015 3rd International Symposium on*. IEEE, 2015.
12. P. Darji, Mohini & Dabhi, Vipul & Prajapati, Harshadkumar. (2015). Rainfall forecasting using neural network: A survey. 10.1109/ICACEA.2015.7164782.
13. Zhang, Xike, et al. "A Novel Hybrid Data-Driven Model for Daily Land Surface Temperature Forecasting Using Long Short-Term Memory Neural Network Based on Ensemble Empirical Mode Decomposition." *International journal of environmental research and public health* 15.5 (2018): 1032.
14. Karim, Fazle, et al. "LSTM fully convolutional networks for time series classification." *IEEE Access* 6 (2018): 1662-1669.
15. Kim, Seongchan, et al. "DeepRain: ConvLSTM Network for Precipitation Prediction using Multichannel Radar Data." *arXiv preprint arXiv:1711.02316* (2017).