

Early Prediction of Extreme Rainfall Events: A Deep Learning Approach

Sulagna Gope¹(✉), Sudeshna Sarkar¹, Pabitra Mitra¹, and Subimal Ghosh²

¹ Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India

sulagna.student12@gmail.com, {sudeshna,pabitra}@cse.iitkgp.ernet.in

² Department of Civil Engineering, Indian Institute of Technology, Bombay, India
subimal@civil.iitb.ac.in

Abstract. Prediction of heavy rainfall is an extremely important problem in the field of meteorology as it has a great impact on the life and economy of people. Every year many people in different parts of the world suffer from the severe consequences of heavy rainfall like flood, spread of diseases, etc. We have proposed a model based on deep neural network to predict extreme rainfall from the previous climatic parameters. Our model comprising of a stacked auto-encoder has been tested for Mumbai and Kolkata, India, and found to be capable of predicting heavy rainfall events over both these regions. The model is able to predict extreme rainfall events 6 to 48 h before their occurrence. However it also predicts several false positives. We compare our results with other methods and find our method doing much better than the other methods used in literature. Predicting heavy rainfall 1 to 2 days earlier is a difficult task and such an early prediction can help in avoiding a lot of damages. This is where we find that our model can give a promising solution. Compared to the conventional methods used, our method reduces the number of false alarms; on further analysis of our results we find that in many cases false alarm has been raised when there has been rainfall in the surrounding regions. Thus our model generates warning for heavy rain in surrounding regions as well.

Keywords: Machine learning · Deep learning · Stacked auto-encoder

1 Introduction

Early prediction of heavy rainfall has always been a challenge in the field of weather forecasting. Early rainfall alert helps in relocating the population which could be affected, operating the flood control systems effectively, preparing the disaster mitigation team, etc. which minimizes the social and economic losses. This problem has become even more challenging with changing climatic patterns. According to [7] extreme rainfall events are expected to increase in changing climate. Therefore a proper scientific understanding of the rainfall extremes has become very important for correct prediction. Every year some metropolitan

cities in India specially Mumbai and Kolkata experience very heavy rainfall during monsoon which brings life to a standstill in these places. Both these regions are urbanised and have high population of people living here, this makes it extremely difficult to take preparatory measures like relocation, rainfall alert broadcasting, etc. for high rainfall in a short notice like in 6 h or even less.

Currently, weather prediction is mainly based on numerical weather prediction (NWP) models. This in turn requires more detailed study of the physical processes responsible for heavy rainfall and simulate them, which is a computationally heavy process. Instead of doing that, we intend to use a data-centric approach and apply machine learning and data mining techniques to understand and predict rainfall.

2 Literature Survey

NWP models make use of a number of differential equations based on the laws of physics, fluid motion, atmospheric science, etc. The present weather conditions are fed to the models to get the prediction. Though these models perform well in predicting other weather conditions, they have not been efficient in predicting heavy rainfall events well in advance [14, 17]. Statistical and probabilistic models have also been used to overcome the shortcomings of the NWP models. Numerous works on precipitation prediction in India are available [22, 23], most of which have tried to relate extreme rainfall with anomalous weather behaviour. Though these models could predict rainfall in general, they again failed to predict extreme rainfall events exclusively, in advance.

Later on [21] used a clustering technique to identify the atmospheric parameters and the regions undergoing significant changes during extreme events. These parameters and regions act as fingerprints of extreme events which can be used for further classification into extremes or non-extremes. Nayak and Ghosh [19] modified the above method by using anomaly frequency method (AFM) of feature extraction and support vector machines (SVM) for classification. Munir's method could reduce the huge number of false alarms generated by the fingerprinting technique, but still the false alarms remained high. Several other machine learning techniques are available in literature for weather prediction, which make use of artificial neural network, support vector machines and bayesian networks. Though deep learning has become extremely popular in some fields like image processing, natural language processing, speech, etc., it has not yet been used much in the field of weather forecasting. There are only a very few work on application of deep learning in meteorology. Liu et al. [18] developed a deep neural network model to predict temperature, dew point, mean sea level pressure and wind speed in the next few hours. A deep hybrid model using deep neural network and probabilistic graphical model was proposed for forecasting weather parameters like temperature, wind, etc [9]. Recently a recurrent neural network model namely convolutional LSTM has been proposed [25] to address the precipitation nowcasting problem using radar echo data set. However none of these methods have dealt with extreme rainfall prediction problem in particular,

which is an anomalous weather event, and has a direct impact on the lives of people. Here we have made an attempt to predict extreme rainfall events much ahead of time, compared to the state of art methods. We have used a stacked auto-encoder model for feature learning and reduction. The reduced features have been used for classification.

The AFM-SVM Method: Munir et al. used the anomaly frequency method (AFM) for extracting features from the entire set of weather parameters. In this method a positive and negative anomaly threshold has been defined as follows: $\delta^+ = \bar{X} + 1.25\bar{SD}$ and $\delta^- = \bar{X} - 1.25\bar{SD}$, where \bar{X} denotes the climatological mean of a weather variable and \bar{SD} denotes the climatological standard deviation of the variable in a particular grid at a specific time instant. Weather parameters whose values exceed the positive threshold are said to have positive anomalous behaviour and parameters whose values are below the negative anomaly threshold are said to have negative anomalous behaviour. The features which consistently show anomalous behaviour during extreme rainfall events are extracted from the entire set of features, considering the spatial and temporal extent of the features and the remaining weather features are ignored. The extracted features are then used for classification. A two-phase SVM is trained to predict night and day extremes separately. This method performs much better than the fingerprinting technique [21] but still generates a large number of false alarms.

3 Objective

The available methods for heavy rainfall prediction are able to predict only 6 h prior to the event. We want to predict these extreme (heavy) rainfall events much earlier that is about 1 or 2 days before with greater precision. This will ensure that least damage is caused by heavy rainfall events. All the weather variables data that are needed for prediction are collected for the entire Indian sub-continent whose latitude ranges from 5 degrees to 40 degrees north and longitude ranges from 65 degrees to 100 degrees east. The region has been shown in Fig. 1 which is sub-divided into grids. In total there are 21 weather variables used, for over the entire region shown in figure, collected on a daily and six-hourly basis. Munir's method showed that heavy rainfall can occur due to some anomalous weather features prevalent in regions that are far away from the region of interest. We have thus used the weather features over the entire Indian subcontinent. This will help in capturing the non-homogeneity in land-sea interaction, weather system and topography, over entire India, which affects rainfall in all parts of the country. The weather data has been obtained from the National Centers for Environmental Prediction/National Center for Atmospheric Research (NCEP/NCAR) reanalysis data. The region is divided into 225 grids. There are a total of $21 * 225 = 4725$ variables available for each day (if daily data is taken). This is further increased if the 24 h and 48 h prior combined features are used, making it a total of 9450 features. Such huge set of features if used for training a machine learning model may lead to overfitting. This calls for feature

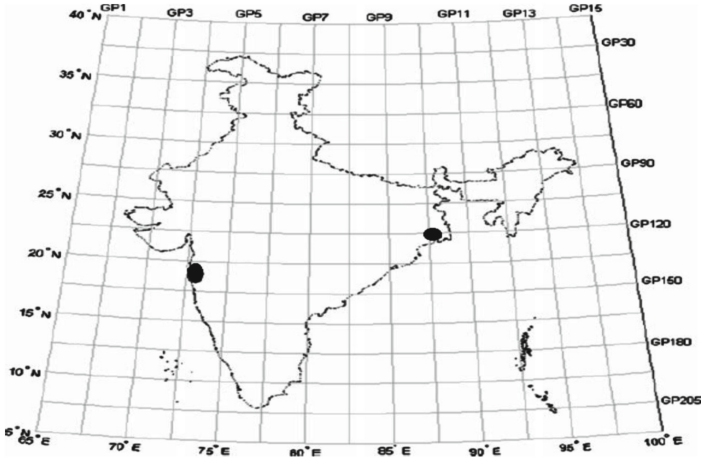


Fig. 1. Indian subcontinent and the regions of interest

reduction or feature extraction from the original set of features, before going into classification. Instead of relying solely on the anomalous features and extracting them greedily, we have used deep learning approach of learning and subsequently reducing the feature set. Weather condition is a combination of many complex interwind structures which need to be decoded to fully understand the weather system and the processes governing it. We believe that the different complex structures comprising the weather system can be learned and understood if they are unveiled in multiple layers just like images. We have thus used a deep neural network model for the purpose. Though deep learning has not yet been used much in weather prediction, we have tried to explore its ability in this domain. Here we have used a deep neural network architecture, namely the stacked auto-encoder for feature reduction. The reduced features are then used for further classification.

4 Problem Statement

In this work we have addressed the problem of predicting the occurrence of heavy rainfall events in Mumbai and Kolkata in monsoon season (months of June, July, August and September), when there is maximum number of heavy rainfall events in these regions. The prediction is to be done based on the weather conditions over the region and its surroundings in the past 48 to 6 h. Using historical data of some the important weather variables and rainfall over the mentioned regions, our model is trained. It is then tested with a new set of feature values of the present time and used to predict whether heavy rainfall will take place or not in the next 6–48 h.

Data Used: The weather data has been obtained from NCEP/NCAR website. The weather variables used are observed at surface level and at 850-, 600- and 400-hPa levels. The surface level parameters are temperature, mean sea level pressure, precipitable water, relative humidity, U-wind and V-wind. The 850-, 600-, 400-hPa level variables are air temperature, vertical wind velocity (omega), relative humidity, u-wind and v-wind. These parameters are generally found to influence heavy rainfall events. The reanalysis data has a spatial grid resolution of 2.5 degrees * 2.5 degrees. These data has been collected for the months of June, July, August and September for the years 1969–2008 for Mumbai and the years 1980–2000 for Kolkata. The rainfall data is also collected for the same time period from India Meteorological Department (IMD). The Mumbai data is the same as used in Munir’s paper.

5 Stacked Autoencoder Feature Learning

Hierarchical feature learning techniques help in extracting non-linear characteristics from the input in multiple layers. Deep belief network (DBN) was introduced by Hinton and Salakhutdinov [12], which solved the vanishing gradient problem of very deep neural architecture. After that a number of variants of deep neural network were proposed by others. Bengio et al. [6] and Ranzato et al. [20] trained the DNN by layerwise unsupervised pre-training followed by finetuning over the entire network. This approach is also known as the greedy layerwise unsupervised pre-training. In this study, we want to learn the weather attributes which are mainly in the form of real numbered values and thus have chosen stacked auto-encoder architecture of deep learning for the purpose.

A simple auto-encoder is an unsupervised one layered neural network where the input $X = x_1, x_2, x_3, \dots, x_n$ is a n dimensional feature vector. The output is given by

$$h_{W,b}(X) = f(W^T X) = f\left(\sum_{i=1}^n W_i x_i + b\right) \quad (1)$$

where $f : \mathbb{R} \mapsto \mathbb{R}$ is a non-linear transformation function and W and b are the weights and bias of the network respectively. The objective is to make $h_{W,b}(X) \approx X$, that is to learn the feature set and regenerate it. The hidden layer gets to learn a compressed representation of the input, such that the original input can be regenerated from it. The loss function of an autoencoder with a single hidden layer is given by,

$$J(W, b) = \left[\frac{1}{m} \sum_{i=1}^m \frac{1}{2} \|h_{W,b}(x_i) - x_i\|^2 \right] + \frac{\lambda}{2} \sum_i^n \sum_j^{nhid} (W_{j,i})^2 \quad (2)$$

where m is the number of training examples, nhid is the number of units in hidden layer, considering only one hidden layer. The second term in Eq. (2) is the regularization term and λ is the weight decay parameter. The autoencoder tries to minimize Eq. (2) by gradient descent. An autoencoder can also have

hidden layer whose size is greater than the size of input layer. In that case a sparsity constraint is imposed on the hidden units. The autoencoder is still able to discover interesting patterns in the input set. A hidden unit is said to be active or firing if its output is close to 1 and inactive if its output is close to 0.

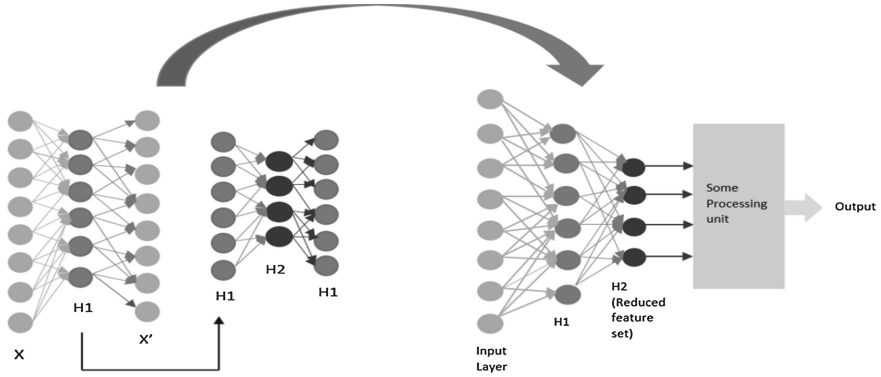


Fig. 2. Layer-wise pre-training in SAE followed by stacking of hidden layers for further processing.

A denoising autoencoder (DAE) is an improvement of the autoencoder, which is designed to learn more robust features and prevent the autoencoder from simply learning the identity. A stacked autoencoder is formed by stacking multiple such denoising autoencoders one on top of the other to form a deep neural network. Each layer is trained separately to adjust the parameters of that layer. After all the layers are pre-trained in this fashion, they are combined together. Finally finetuning is done over the entire network so that the output of the SAE becomes a good approximation of the input. An example of layer-wise pre-training followed by hidden layer stacking is shown in Fig. 2. The reduced feature set obtained after stacking the hidden layers can be used as input features for further processing.

6 Outline of Proposed Approach

6.1 Proposed Model

Our prediction model has two phases: Feature Learning and feature compression followed by classification. SAE has been used to get a compressed representation of the feature set. For classification we have used support vector machines (SVM) and also neural network. The configuration of the SAE used has been discussed in the later sections. The classifier parameters have to be adjusted to get the best performance. The inherent problem of biased data set has also been dealt with effectively to get the best performance. For this purpose a cost-sensitive SVM has been used.

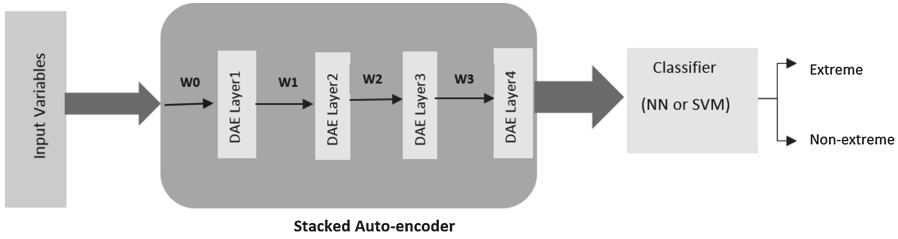


Fig. 3. Block diagram of the model

The AFM extracts only the anomalous features leaving out the rest. This may not be able to capture the entire weather picture and can lead to wrong prediction. Extreme events may not just occur due to the anomalous features but due to the combination of all the features, both anomalous and non-anomalous. We have thus used the total weather picture over entire India as our feature set and reduced it further with SAE. A block diagram of our model has been shown in Fig. 3.

6.2 Methodology for Extreme Rainfall Prediction

We have focused and tested our model for prediction mainly in two cities in India, namely Mumbai and Kolkata. Unlike data in other domains, weather data has some peculiar characteristics that should be kept in mind while dealing with them. For instance, there is a season-to-season and also year-to-year variation in the weather parameters. Sometimes there are significant changes in the weather patterns of consecutive months. Due to this reason we have focused our study on the monsoon months when there is maximum amount of heavy rainfall events.

7 Experimental Results

Our model has been used on the weather parameter data obtained for over the entire Indian subcontinent, to predict rainfall in Mumbai and Kolkata. Prediction is performed 6, 24, 36 and 48 h before the extreme event. The entire dataset has been divided into training and test periods. For Mumbai the training set consists of the rainfall events from 1969 to 1999 and the test set consists of rainfall events from 2000 to 2008 and for Kolkata the training period ranges from 1980–1995 and the test period ranges from 1996–2000. Total rainfall amount (throughout the day) exceeding 75 mm has been considered to be an extreme rainfall event for Mumbai and total rainfall amount exceeding 60 mm has been considered to be an extreme event for Kolkata. The thresholds considered are slightly on the higher side due to the fact that both Mumbai and Kolkata receive high intensity rainfall but in short durations.

7.1 Stacked Auto-encoder Configuration

There are 21 weather parameters (as mentioned above) used in this study. In literature these parameters have been found to be important factors influencing rainfall. We have the values of all the parameters for each of the grid points at different time steps. This results in a huge feature set, which is difficult to process and use. We have tried to reduce the feature space size by using a stacked auto-encoder, which reduces the feature space in such a way that the original feature set can be rebuilt from the reduced set. Thereby it does not eliminate any feature but captures all in a compact form.

Here we have built a 4 layer SAE model. The entire set of features is fed as input to the model. The first hidden layer has 2500 units, the second layer has 1000 units, the third layer has 500 units and the innermost layer has 200 units. The number of units in each layer has been set by trial and error method. For example we found that 2500 units are good enough for learning in the first layer compared to 3000 units. In order to reduce the size of feature space further, we have gradually learned the features in multiple layers and have obtained a compact representation of the feature set. Batch learning has been followed during layer-wise pre-training. The sparsity parameter value has been adjusted to tune the model. Dropout is used to prevent over-fitting. All these parameters are adjusted by trials such that the outputs of the network generated are good approximations of the input. The innermost hidden layer activation (output) which becomes the reduced feature set can now be used to carry out further classification.

7.2 Evaluation Methodology

The AFM-SVM method predicts rainfall using 6 to 48 h prior weather data. Thus the results can be obtained only 6 h prior to the event. This time may not be enough for evacuation, relocation or preparation of disaster mitigation team. Here we intend to predict atleast 24 h before or 48 h before. We have also done prediction in 6–48 h lead time just like the AFM-SVM method. The weather parameters are fed to the stacked autoencoder and the reduced feature space is obtained for further classification into extreme and non-extreme events. However there are a very few extreme rainfall cases compared to normal or no rainfall cases, that is the data set is biased. This naturally gives rise to a large number of false positives and false negatives during classification. This problem has been tackled using SMOTE (Nitesh et al. 2002) [5] for oversampling and Wilsons [26,27] edited Nearest Neighbor rule for under-sampling. For classification, we have used cost sensitive SVM [1,16] and neural network. In our method we have used only one classifier for prediction of both night and day extremes unlike Munir’s method.

7.3 Results and Comparison

Extreme rainfall prediction has been carried out with AFM-SVM method as well as our model. The results are as follows.

7.3.1 6 Hours Prior Prediction

Table 1 shows the results obtained with Munir’s method and SAE based methods for Mumbai. Here the raw features are taken 48, 36, 24 and 6h prior to the occurrence of heavy rainfall. The results using Munir’s method shown in the table is as available in the Munir’s paper, with the false alarms (for night-SVM and day-SVM) added up. The AFM threshold and the SVM classifier used for each of the experiments have been mentioned along with the actual extremes, the model is able to detect. The number of false alarms have also been shown. There were 15 extreme events (both day and night combined) in Mumbai during the period considered. The best result obtained from AFM based method predicts all the 15 extremes and generates 149 false alarm cases (combining the day and night cases together). Though in the AFM-SVM method day and night extremes have been predicted separately, here we present the aggregate results for better comparison. For instance, the AFM-SVM technique with night frequency threshold 29 and SVM with rbf kernel, predicts 12 (out of 12) night extremes correctly.

Table 1. Prediction 6 h prior for Mumbai (Actual number of extremes for Mumbai = 15)

Model	Extreme detected correctly	Extreme detected on previous day	Total true positives	Actual number of false alarms
SAE+SVM(quadratic kernel)	13	1	14	200
SAE+SVM(3rd order polynomial kernel)	13	2	15	128
SAE+Neural Network Classifier	12	2	14	224
AFM(night freq.=28, day freq.=14)+SVM (RBF, quadratic)	14	-	14	190
AFM(night freq.=29, day freq.=14)+SVM (quadratic)	15	-	15	149
AFM(night freq.=30, day freq.=14)+SVM(RBF, quadratic)	13	-	14	292
PCA + SVM(Quadratic)	7	1	8	260
Fisher LDA	3	3	6	205

With day frequency threshold 14 and SVM with quadratic kernel, it is able to detect 4(out of 4) day extremes correctly. The same raw data has been fed to our model. Firstly the data has been reduced with a SAE and then classifier is applied. We find that most of the extremes were correctly predicted for the day on which the extreme event occurred. However in a few cases our model has not been able to predict the extreme on the exact day when it occurred but one day before. This is still a good estimation because if high rainfall is predicted one day before the actual day when it occurs, there will still be enough preparedness to ensure least damage. Warning can be issued much ahead of time, and situation can be skillfully handled even if disastrous condition occurs the next day. The best result with SAE is obtained using SVM of polynomial kernel. It predicts 13 extremes correctly and 2 extremes on the previous day. There are 128 false alarms which is still less than that obtained with the AFM based method. However, we have achieved only a slight improvement with our method

over the conventional one, with the data of the past 48-6 h. The results obtained with some other conventional methods like principal component analysis and Fisher linear discriminant analysis are also shown.

7.3.2 24 Hours Prior Prediction

We have then fed only the weather parameters of the previous day, 36 and 24 h prior, to both the SAE based model and AFM based model and tried to predict high rainfall on the following day. Here we have reproduced Munir’s method with 36 and 24 h prior features as input. The results have been compared in the Table 2 for rainfall in Mumbai. We find that in this case our model works much better than AFM model. In our model the neural network classifier could predict all the 15 extreme events correctly. It generated 430 false positive cases. However further analysis of the false positives revealed that in some of those cases it rained heavily elsewhere in the surroundings of Mumbai. Therefore these alarms could also be treated as meaningful. Only in 180 instances our model generated false alarms when it did not rain heavily in Mumbai or the surroundings. SVM classifier with quadratic kernel has also been used and the results have been shown. SVM with quadratic kernel detects 10 extreme events correctly and the 5 remaining actual extremes were also predicted but one day before they actually occurred. There were 191 false alarms in total out of which in 37 instances there was heavy rain in the surrounding regions and the rest 154 were actual false alarms. It is found that the our model can predict rainfall in the surrounding regions as well. Thus it can be said that the features obtained from the SAE give a good prediction of rainfall not just for the region concerned but also its

Table 2. Some results for Mumbai. (Actual number of extremes for Mumbai is 15)

Raw Features	Model	Extremes predicted correctly	Extremes predicted on previous day	Total extremes predicted	Total false alarms	False alarms but rain in surroundings	Actual number of false alarms
24 hours prior prediction							
36, 24 hours prior features	<i>SAE+SVM (quadratic kernel)</i>	10	5	15	191	37	154
	SAE+ neural network classifier	15	-	15	430	250	180
	Munir’s model (freq threshold=29)	7	3	10	233	40	193
	Munir’s model(freq. threshold=28)	6	3	9	235	33	201
	PCA+SVM(quadratic kernel)	7	-	7	225	33	192
	Fisher LDA	3	-	3	121	12	109
36 hours prior prediction							
48, 36 hours prior features	<i>SAE+SVM (quadratic kernel)</i>	12	1	13	223	30	193
	SAE+ neural network classifier	12	1	13	398	180	218
	Munir’s method (freq. threshold=29)	6	2	8	256	35	221
	Munir’s method (freq. threshold=28)	5	3	8	244	32	212
	PCA+SVM(quadratic kernel)	7	1	8	256	32	224
	Fisher LDA	3	1	4	140	12	128
48 hours prior prediction							
48 hours prior features	<i>SAE+SVM (quadratic kernel)</i>	11	1	12	152	39	113
	SAE+ neural network classifier	12	1	13	413	158	255
	Munir’s method (freq. threshold=29)	5	3	8	270	38	232
	Munir’s method (freq. threshold=28)	7	1	8	254	40	214
	PCA+SVM(quadratic kernel)	6	1	7	268	45	223
	Fisher LDA	2	0	2	122	7	115

surroundings. Whereas the AFM based method in the best case detects 10 total extremes and generates 193 false alarms, leaving out those cases where it rained in the surroundings.

7.3.3 36 and 48 Hours Prior Prediction

Table 2 also shows the prediction of rainfall for Mumbai and its surroundings with weather features taken 48 and 36 h before. In the best case, SAE followed by SVM classifier of quadratic kernel is able to detect 13 out of 15 extremes correctly, out of which only one is detected on the previous day and generates 193 false alarm cases. Munir’s method, reproduced with 48 and 36 h prior features, detects 8 total extremes in the best case and generates 212 false positive cases. The weather features 48 h before rainfall have also been used as raw features that are fed to SAE. SAE followed by SVM classifier (quadratic kernel) predicts 12 total extreme cases and generates 152 total false positives out of which in 39 cases there was rainfall elsewhere. SAE followed by neural network detects 13 extremes, 48 h before and generates 255 false alarms. Munir’s method is able to detect only 8 out of 15 extremes correctly and generates 214 false alarms in the best case. PCA and LDA performs poorly in these tasks.

Table 3. Results for Kolkata (Actual number of extremes for Kolkata = 9)

Raw Features	Model	Extremes predicted correctly	Extremes predicted on previous day	Total extremes predicted	Total false alarms	False alarms but rain in surroundings	Actual number of false alarms
6 hours prior prediction							
48, 36, 24 and 6 hours prior features	SAE+SVM (quadratic kernel)	5	2	7	460	134	326
	SAE+neural network	4	2	6	263	56	207
	Munir’s method	5	2	7	600	151	449
	PCA+SVM(quadratic kernel)	3	0	3	231	51	180
	Fisher LDA	1	3	4	210	21	189
24 hours prior prediction							
36, 24 hours prior features	SAE+SVM (quadratic kernel)	8	-	8	399	231	168
	SAE+ neural network classifier	4	1	5	220	30	190
	Munir’s model	5	-	5	497	112	385
	PCA+SVM(quadratic kernel)	2	3	5	228	12	216
	Fisher LDA	2	3	5	228	12	216
36 hours prior prediction							
48, 36 hours prior features	SAE+SVM (quadratic kernel)	6	1	7	411	200	211
	SAE+ neural network classifier	5	1	6	300	25	275
	Munir’s method	4	1	5	505	124	381
	PCA+SVM(quadratic kernel)	2	3	5	234	101	133
	Fisher LDA	1	1	2	140	11	129
48 hours prior prediction							
48 hours prior features	SAE+SVM (quadratic kernel)	5	2	7	420	190	230
	SAE+neural network classifier	4	3	7	310	43	267
	Munir’s method	4	1	5	512	101	411
	PCA+SVM(quadratic kernel)	5	0	5	227	23	204
	Fisher LDA	1	1	2	149	19	130

7.3.4 Results for Kolkata

We have duplicated Munir’s method and have applied it for Kolkata. PCA and LDA techniques have also been used for comparison. We have also used the SAE based method for Kolkata and the results have been compared in Table 3, where the labels have their usual meaning. The raw features are taken in time periods 48, 36, 24 and 6 h before extreme events; 36 and 24 h before extreme event which gives 24 h prior prediction; 48 and 36 h before extreme event, which gives 36 h prior prediction; features 48 h before rainfall which gives 48 h prior prediction. Here also we find that our model is able to predict more extreme events in the test period and generates less false positive cases compared to Munir’s method. There are total of 9 extreme events for Kolkata. Using our method, 7 out of 9 could be detected 6 h before, 8 could be detected 24 h before, 7 could be detected 36 h before and again 7 could be detected 48 h before. The raw features and the false alarms generated in each of the cases is mentioned in the table. However the results for Kolkata is not very satisfactory which is because of the fact that the number of training examples are very few for Kolkata. With more training instances the results could improve further. The sensitivity and specificity values for both the models for the performed experiments are shown in Table 4.

Table 4. Sensitivity and Specificity of the models

Prediction Task	Mumbai				Kolkata			
	<i>Munir’s method</i>		<i>Our method</i>		<i>Munir’s method</i>		<i>Our method</i>	
	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity
6 hrs prior	1	0.845	1	0.867	0.778	0.253	0.778	0.458
24 hrs prior	0.67	0.799	1	0.84	0.556	0.358	0.89	0.72
36 hrs prior	0.53	0.78	0.867	0.799	0.556	0.365	0.778	0.648
48 hrs prior	0.53	0.777	0.8	0.88	0.556	0.314	0.778	0.616

8 Conclusion and Future Work

In this work we have explored a machine learning technique namely deep learning with SAE to learn and represent weather features and use them to predict extreme rainfall events. We found that though our method gives results that are quite similar to the baseline case when prediction is done in 6 h before the event, it gives significantly better performance when prediction is done 24 h and 48 h before the events. The AFM based method does not perform well when it comes to such early prediction. Our experiments show that these intelligently learnt features can improve the performance of the classical approaches. One of the reason why this model works better is that, here we include all the features and try to understand underlying patterns and dependencies unlike other approaches which rely on feature extraction or selective feature reduction. Our experiment shows that Deep Learning can be quite promising in the field of weather forecasting, just like in field of image recognition, speech processing, NLP, etc.

However there is still much scope of improvement. Firstly more and higher resolution data should be used for learning, since more the data available better becomes the learning. Thus higher resolution data with large number of training instances should be used in deep learning. As mentioned earlier weather data has many specific characteristics which depend on time and spatial location. This should be taken into consideration to understand and learn the system properly. This calls for more sophisticated learning methods. We would like to explore the data considering it's temporal and spatial behaviour. Here we have solved only a classification problem where we are only able to predict whether there will be heavy rainfall or not. In future we would also like to predict the amount of rainfall as well with the improved methods.

Acknowledgements. This research was supported and funded by Indian Institute of Technology, Kharagpur, India and MHRD, India under the project named “Feature Extraction and Data Mining from Climate Data (FAD)”. We would like to thank IIT Kharagpur, MHRD and also the IMD(India Meteorological Society) for their helpful suggestions and support. Without their help this work would not have been completed.

References

1. Akbani, R., Kwek, S.S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 39–50. Springer, Heidelberg (2004)
2. Bengio, Y.: Learning deep architectures for AI. *Found. Trends Mach. Learn.* **2**(1), 1–127 (2009)
3. Bengio, Y.: Deep learning of representations for unsupervised and transfer learning. *Unsupervised Transf. Learn. Chall. Mach. Learn.* **7**, 19 (2012)
4. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152. ACM (1992)
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002)
6. Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P., Bengio, S.: Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.* **11**, 625–660 (2010)
7. Goswami, B.N., Venugopal, V., Sengupta, D., Madhusoodanan, M., Xavier, P.K.: Increasing trend of extreme rain events over india in a warming environment. *Science* **314**(5804), 1442–1445 (2006)
8. Graves, A.: Generating sequences with recurrent neural networks (2013). arXiv preprint [arXiv:1308.0850](https://arxiv.org/abs/1308.0850)
9. Grover, A., Kapoor, A., Horvitz, E.: A deep hybrid model for weather forecasting. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 379–386. ACM (2015)
10. Haykin, S., Network, N.: A comprehensive foundation. *Neural Netw.* **2**(2004) (2004)
11. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. *Neural Comput.* **18**(7), 1527–1554 (2006)

12. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. *Science* **313**(5786), 504–507 (2006)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
14. Hong, S.Y., Lee, J.W.: Assessment of the WRF model in reproducing a flash-flood heavy rainfall event over Korea. *Atmos. Res.* **93**(4), 818–831 (2009)
15. Hong, Y.: Precipitation estimation from remotely sensed information using artificial neural network-cloud classification system (2003)
16. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. *Intell. Data Anal.* **6**(5), 429–449 (2002)
17. Khaladkar, R., Narkhedkar, S., Mahajan, P.: Performance of NCMRWF Models in Predicting High Rainfall Spells During SW Monsoon Season: A Study for Some Cases in July 2004. Indian Institute of Tropical Meteorology (2007)
18. Liu, J.N., Hu, Y., You, J.J., Chan, P.W.: Deep neural network based feature representation for weather forecasting. In: Proceedings on the International Conference on Artificial Intelligence (ICAI), p. 1. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp) (2014)
19. Nayak, M.A., Ghosh, S.: Prediction of extreme rainfall event using weather pattern recognition and support vector machine classifier. *Theoret. Appl. Climatol.* **114**(3–4), 583–603 (2013)
20. Ranzato, M.A., Huang, F.J., Boureau, Y.L., LeCun, Y.: Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, 2007, CVPR 2007, pp. 1–8. IEEE (2007)
21. Root, B., Knight, P., Young, G., Greybush, S., Grumm, R., Holmes, R., Ross, J.: A fingerprinting technique for major weather events. *J. Appl. Meteorol. Climatol.* **46**(7), 1053–1066 (2007)
22. Bhowmik, S.R., Durai, V.: Application of multimodel ensemble techniques for real time district level rainfall forecasts in short range time scale over indian region. *Meteorol. Atmos. Phys.* **106**(1), 19–35 (2010)
23. Sahai, A., Soman, M., Satyan, V.: All india summer monsoon rainfall prediction using an artificial neural network. *Clim. Dyn.* **16**(4), 291–302 (2000)
24. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2002)
25. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.C.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting (2015). arXivpreprint [arXiv:1506.04214](https://arxiv.org/abs/1506.04214)
26. Wilson, D.R., Martinez, T.R.: Instance pruning techniques. In: ICML, vol. 97, pp. 403–411 (1997)
27. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans. Syst. Man Cybern.* **3**, 408–421 (1972)