Rainfall Prediction: A Deep Learning Approach

Emilcy Hernández[1(✉)], Victor Sanchez-Anguix[2], Vicente Julian[3], Javier Palanca[3], and Néstor Duque[4]

[1] Departamento de Ingeniería de la Organización, Universidad Nacional de Colombia, Carrera 80 No 65-223 - Núcleo Robledo, Medellín, Colombia
ejhernandezle@unal.edu.co

[2] School of Computing, Electronics and Maths, Coventry University, Gulson Road, Coventry CV1 2JH, UK
ac0872@coventry.ac.uk

[3] Departamento de Sistemas Informaticos y Computacion, Universitat Politècnica de València, Cami de Vera s/n, 46022 Valencia, Spain
{vinglada,jpalanca}@dsic.upv.es

[4] Departamento de Informática y Computación, Universidad Nacional de Colombia, Campus La Nubia, Bloque Q, Piso 2, Manizales, Colombia
ndduqueme@unal.edu.co

Abstract. Previous work has shown that the prediction of meteorological conditions through methods based on artificial intelligence can get satisfactory results. Forecasts of meteorological time series can help decision-making processes carried out by organizations responsible of disaster prevention. We introduce an architecture based on Deep Learning for the prediction of the accumulated daily precipitation for the next day. More specifically, it includes an autoencoder for reducing and capturing non-linear relationships between attributes, and a multilayer perceptron for the prediction task. This architecture is compared with other previous proposals and it demonstrates an improvement on the ability to predict the accumulated daily precipitation for the next day.

Keywords: Artificial neural networks · Deep learning · Meteorological data · Rainfall prediction

1 Introduction

Rainfall prediction remains a serious concern and has attracted the attention of governments, industries, risk management entities, as well as the scientific community. Rainfall is a climatic factor that affects many human activities like agricultural production, construction, power generation, forestry and tourism, among others [1]. To this extent, rainfall prediction is essential since this variable is the one with the highest correlation with adverse natural events such as landslides, flooding, mass movements and avalanches. These incidents have

F. Martínez-Álvarez et al. (Eds.): HAIS 2016, LNAI 9648, pp. 151–162, 2016.
DOI: 10.1007/978-3-319-32034-2_13

affected society for years [2]. Therefore, having an appropriate approach for rainfall prediction makes it possible to take preventive and mitigation measures for these natural phenomena [3].

Also these predictions facilitate the supervision of agriculture activities, construction, tourism, transport, and health, among others. For agencies responsible for disaster prevention, providing accurate meteorological predictions can help decision-making in the face of possible occurrence of natural events.

For achieving these predictions there are a number of methods, ranging from naive methods, to those that use more complex techniques such as artificial intelligence (AI), artificial neural networks (ANNs) being one of the most valuable and attractive methods for forecasting tasks. In prediction, ANNs, as opposed to traditional methods in meteorology, are based on self-adaptive mechanisms that learn from examples and capture functional relationships between data, even if the relationships are unknown or difficult to describe [4].

Over the last few years, Deep Learning has been used as a successful mechanism in ANN for solving complex problems [5]. Deep Learning is a general term used to refer to a series of multilayer architectures that are trained using unsupervised algorithms. The main improvement is learning a compact, valid, and non-linear representation of data via unsupervised methods, with the hope that the new data representation contributes to the prediction task at hand. This approach has been successfully applied to fields like computer vision, image recognition, natural language processing, and bioinformatics [6]. Deep learning has shown promise for modeling time-series data through techniques like Restricted Boltzmann Machine (RBM), Conditional RBM, Autoencoder, Recurrent neural network, Convolution and pooling, Hidden Markov Model [7].

In this experimental study we use data gathered from a meteorological station located in a central area of Manizales, Colombia. The data gathered comprises more than a decade of measurements taken in real time and stored by Instituto de Estudios Ambientales (IDEA) of Universad Nacional de Colombia, also located in the same city. In order to perform forecasts, a deep architecture combining the use of an autoencoder and a multilayer perceptron is used. For testing the validity of the proposed model, we optimized the parameters of the deep architecture and tested the resulting network via a series of error measurement criteria. The results show that the proposed architecture outperforms the state of the art in the task of predicting the accumulated daily precipitation for the next day.

The structure of the paper is as follows: Sect. 2 describes previous approaches for the prediction of meteorological variables, mainly precipitation; Sect. 3 contains a description our proposed architecture, the techniques used for forecasting, and details of the dataset employed as case of study; meanwhile Sect. 4 shows the experiments realized and the results obtained. Finally, in Sect. 5 the conclusions and future work are presented.

2 Related Work

With the influence of weather in social and economic activities, prediction of atmospheric phenomena like rainfall would contribute to the prevention of adverse events. In the last few years, several approaches have been proposed in order to deal with this goal. There is extensive research carried out in the area of rainfall prediction, and particularly, one can notice the efficacy of neural networks for the problem.

Sawale & Gupta [8] present an algorithm based on an ANN that predicts atmospheric conditions from a dataset that includes the variable temperature, humidity and wind speed. The authors employed a hybrid architecture composed by a Back Propagation Network (BPN) and a Hopfield Network (HN). The output from the BPN is fed into the HN, which is in charge of making the actual predictions. The approach proposed is able to determine the non-linear relationship that exists between the historical data of parameters: temperature, wind speed,humidity, etc., and from this make a prediction of what the weather. In the paper does not define the time horizon to which the forecasts are made.

Luk et al. [9] present a model which aims to identify the spatio-temporal data necessary for achieving more accurate and short-term (5 min 30 min) rainfall prediction for an urban catchment in Sydney, Australia. An ANN is used to predict rainfall based on historical rainfall patterns from a series of measurements carried out in a basin study. To achieve this, the authors compared the accuracy of the predictions made by ANN configured with different variables of delay and number of inputs. The study found that the network that used a lower lag obtained better performance.

Beltrn-Castro et al. [10] adopt a decomposition and ensemble principle for daily rainfall forecasting. The decomposition technique employed is the Ensemble Empirical Mode Decomposition (EEMD), wich divides the original data into a set of simple components. Moreover, a Feed Forward Neural Network (FNN) is used as a forecasting tool for model each component. The model was tested in a case study with rainfall data registered in a meteorological station from Manizales city, Colombia. In this case study only the accumulated precipitation variable from past days was used as an input for predictions of daily rainfall of next day.

Abhishek et al. [11] proposed the use of ANNs for predicting the monthly average rainfall in an area of India characterized by monsoon type climate. The presented case study used data of eight-month per year. In these months, there is certainty that rainfall events will be present. The authors use the average humidity and average wind speed as explanatory variables. The experiments were carried out with three types of different networks: Feed Forward Back Propagation, Layer Recurrent, and Cascaded Feed Forward Back Propagation. Then, the results obtained with each network are compared, finding that the type of network that obtained the best results was Feed Forward Back Propagation.

Liu et al. [12] propose an alternative over the previous model. They explore the use of genetic algorithms as a feature selection algorithm, an then Nave Bayes as the predictive algorithm. The problem is decomposed into two prediction problems: rainfall event (i.e., a binary prediction problem), and a categorization of rainfall in case that rainfall is present (i.e., light, moderate and strong rainfall). The adoption of genetic algorithms for the selection of inputs, shows that it is possible to reduce the complexity of the dataset obtaining similar or slightly better performance.

Liu et al. (2014) proposed a model based on Deep Learning (DL). In their research they apply Deep Neural Network (DNN) to process massive data involving datasets of almost 30 years (1-1-1983 to 31-12-2012) of environmental records provided by the Hong Kong Observatory (HKO). The data is used to predict the weather change in the next 24 hours, given four variables: temperature, dew point, Mean Sea Level Pressures (MSLP) and wind speed. The results obtained for authors show that the DNN provide a good feature space for weather datasets and a potential tool for the feature fusion of time series problems. However they do not predict with their model more difcult weather data, such as rainfall dataset [13].

Finally, Grover et al. (2015) presented a Model for Weather Forecasting which makes predictions by considering the joint influence of climatic essential variables. For the case of study, they took data of winds, temperature, pressure and dew point from 2009 to 2015 observed in 60 points in the United States. The proposal has a novel hybrid model with discriminative and generative components for spatio-temporal inferences about the variables mentioned above. The proposed architecture combines a bottom-up predictor for each individual variable with a top-down deep belief network that models the joint statistical relationships. Another key component in the framework is a data-driven kernel, which is given on a similarity function learned automatically from the data. The kernel is used to assign long-range dependencies through the space and to ensure that inferences respecting the natural laws. The model is called "Deep Hybrid Model" and the results obtained are compared with other model proposed in [14] and with the model used by the National Oceanic and Atmospheric Administration (NOAA) of the United States, showing a better performance than these [15]. Forecasts are made at 6, 12 and 24 hours; however, the rainfall is not predicted.

Summarizing the literature review concludes that the ANN are one of the intelligent systems that has been used more for studies of weather prediction. In general, neural networks are an appropriate mechanism when undertaking systems and phenomena with nonlinear dynamics, as in the case of meteorological phenomena. Likewise, the Deep Learning is seen as a new promising strategy for forecasting meteorological variables.

3 Data Preparation

The problem solved at this paper is that of predicting the accumulated rainfall precipitation for the next day, by using data gathered in the previous days. More specifically, we employed data from a meteorological station located in a central area of Manizales. The station makes transmissions every five minutes, which are grouped into a daily time series covering from 2002 to 2013.

The dataset used for model validation contains a total of forty-seven explanatory characteristics, including temperature, relative humidity, barometric pressure, sun brightness, speed and direction of wind, and other derivatives of the

Table 1. Details of dataset

Input column	Description	Observation
1-3	Rainfall in the last 3 days ago	
4	Rainfall average 5 days ago	Measurement unit: mm
5	Difference between the temperature at 4:00 and 24:00	
6-8	Temperature in the last 3 days ago	
9	Temperature average 5 days ago	Measurement unit: C
10	Difference between the temperature at 4:00 and 24:00	
11-13	Barometric pressure in the last 3 days ago	
14	Barometric pressure average 5 days ago	Measurement unit: hPa
15	Difference between the barometric pressure at 4:00 and 24:00	
16-18	Relative humidity in the last 3 days ago	Measurement unit: %
19	Relative humidity average 5 days ago	
20-22	Wind speed in the last 3 days ago	Measurement unit: m/s
23-25	Sun brightness in the last 3 days ago	Measurement unit: W/m2
26	Sun brightness average 5 days ago	
27	Dew point temperature	Measurement unit: C
28-39	Month value of the input records	12 components are used, i.e., February: 010000000000 May: 000010000010
40-47	Prevaling wind direction	North: 10000000 Northeast: 01000000 East: 00100000 Southeast: 00010000 South: 00001000 Southwest: 00000100 West: 00000010 Northwest: 00000001

main variables. The reader can observe a description of all of these variables at Table 1.

The data is taken from an environmental datawarehouse administered by IDEA. The data stored in the datawarehouse has been preprocessed. More specifically it has undergone an ETL (Extract, Transform and Load) [16] process in order to achieve data integrity and standardization [17].

The dataset is divided into training, validation and testing with a percentage of 70, 15 and 15 percent respectively. Therefore, from the total of 4216 samples conforming the dataset, 2952 were randomly selected as training. From the rest, 632 samples were selected for validation, and the remaining 632 samples were kept for testing.

Once the data has been extracted and divided into training, test, and validation, we normalized variables to a $[0, 1]$ interval in order to avoid effects of scale in our deep learning architecture. The normalization method employed for the dataset can be observed in the Eq. (1), where a_i represents a value to normalize for the i-th variable, min_{a_i} is the minimum valor registered for this variable in the training set and max_{a_i} is the maximum valor registered for this variable in the training set.

$$V_i = \frac{a_i - min_{a_i}}{max_{a_i} - min_{a_i}} \tag{1}$$

4 Proposed Architecture

In this section, we describe the general architecture of our proposed model. As mentioned throughout the paper, we employ a deep learning architecture to predict the accumulated rainfall for the next day. The architecture is composed of two networks: an autoencoder network and a multilayer perceptron network. The autoencoder network is responsible to feature selection and as mentioned in [7], the autoencoder is a deep learning technique promise for the feature treatment in time series. A multilayer perceptron network is responsible for classification, prediction task. Next we will detail each network.

The first element in our architecture is the autoencoder. An autoencoder is an unsupervised network that aims to extract non-linear features for a data input. Being more specific, an autoencoder is composed by three layers: the input layer, a hidden layer using the sigmoid activation function, and the output layer. Differently to classic neural networks, autoencoders are trained so that the output layer attempts to be as similar as possible to the input layer. This way, the hidden layer results in a non-linear compact representation of the input layer, achieved thanks to the sigmoid activation function. The rationale behind this transformation is that data will be more compact (i.e., less prone to overfitting) and hopefully some interesting non-linear relationships that improve the explanation of the output variable have been discovered. In our architecture, the type of autoencoder that we employed is a denoising autoencoder [18] provided by Theano [19], a Python GPU-based library for mathematical optimization.

The hidden layer of the autoencoder, the non-linear compact representation of the original input, is directly connected to a Multilayer perceptron. This network is the one responsible for making predictions in our problem, by taking the new problem representation as an input. The MLP consists of one hidden layer and uses the sigmoid activation function. Figure 1 presents our architecture, it shows in detail the input and output layers and the way in which the autoencoder connects to the MLP network.

Figure 2 presents a general overview of all the system. Data is extracted for the data storage by means of a query, and then the information is normalized in order to be used by the autoencoder. The autoencoder builds the compact non-linear representation of the data and then it is fed to the MLP, which is responsible for making the predictions. In Table 2 are presented in detail the values tested in each parameter.

5 Experiments

In order to evaluate the performance of the proposed criteria we use the Mean Square Error (MSE) and the Root Mean Square Error (RMSE) as measurement errors. Let $\hat{Y}_i$ be a vector of n predictions and Y_i be the vector of observed values corresponding to the expected output of the function which generates the prediction, then MSE and RMSE can be calculated according to the Eq. (2).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$
$$RMSE = \sqrt{MSE} \tag{2}$$

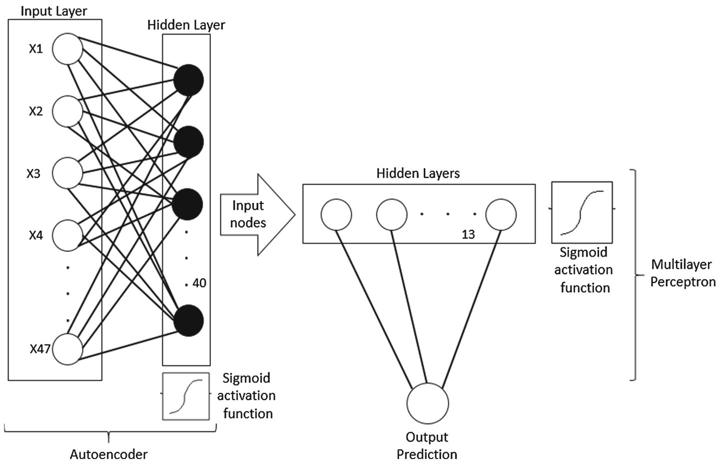

Fig. 1. Architecture: autoencoder and MLP.

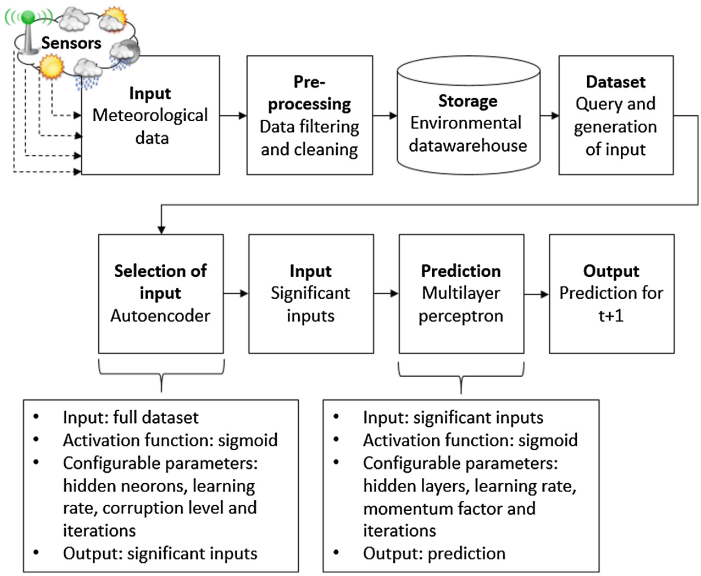

Fig. 2. Flow of the proposed method.

5.1 Optimizing the Proposed Network Architecture

Multiple network architectures were implemented with different configurations for the autoencoder and the MLP. The trainings and tests were carried out in parallel on a cluster consisting of 72 nodes. Initially, several combinations of value parameters were tested in order to look for the best possible architecture configuration. More specifically, we carried out a grid search for the following set of parameters, the Table 2 presents the approved values for each parameter.

Table 2. Approved values for each parameter

	Parameter	Values
Autoencoder	Hidden neurons	[11, 13, 19, 23, 29, 35, 40, 45]
	Learning rate	[0.1, 0.3, 0.9]
	Corruption level	[0.1, 0.3, 0.9]
	Iterations	[33, 100, 300, 1000, 3000, 5000, 9000]
MLP	Hidden layers	[10, 13, 19]
	Learning rate	[0.1, 0.3, 0.9]
	Momentum factor	[0.1, 0.3, 0.9]
	Iterations	[33, 100, 300, 1000, 3000, 5000, 9000]

Once carried out the tests, we found that the best configuration for the autoencoder was the following: 40 hidden neurons, 0.9 of learning rate, 0.1 for corruption level, and 5000 iterations of training. In the case of the MLP configuration, the best result was adjusted with the following combination of values: 13 hidden layers, 0.3 of learning rate, 0.9 of momentum factor and 1000 iterations of training.

5.2 Evaluation of the Proposed Network

Once optimized, the results of our network were compared with a wide range of methods: a naive approach consisting on predicting the accumulated rainfall of $t - 1$ for t (Naive 1), a naive approach consisting on always predicting the average accumulated rainfall (Naive 2), an optimized MLP whose parameters were optimized with the training and validation set, the method proposed in [11], and the method proposed in [10].

The two latter models [10,11] have been selected mainly for two reasons: they provide enough information for replication, and they attempt to predict the rainfall for the next day as our proposal. In [11], the authors test three types of neural networks: Back Propagation network (BP), Layer Recurrent network (LR), and Cascaded Back-Propagation (CBP). Similarly, we optimized the three types of networks proposed by [11] via a grid search on prospective parameters. In [10], the authors use Ensemble Empirical Mode Decomposition (EEMD) and

Feed-forward Neural Networks (FNN) to predict accumulated rainfall on the next day. Similarly to the authors, we configured the their network to take the last two days of data as input, and we set the EEMD to split the signal into nine different components which is fed into nine different FNN that form a prediction via summation of each prediction.

The results for this experiment can be observed in Table 3. The experiment shows the average MSE and RMSE for the best network configuration of each implemented method. The average is calculated by training each network 30 times and evaluating the MSE and RMSE against the test set. This allows us to capture statistical differences between the different networks. Error measures were calculated on the desnormalized data, that is, in the original unit of measurement.

Table 3. Results obtained by the different approaches

Method	MSE	RMSE
Autoencoder and MLP	40.11	6.33
MLP	42.34	6.51
Naive 1	132.82	11.52
Naive 2	88.43	9.40
Abhishek et al. (BP) [11]	93.51	9.67
Abhishek et al. (LR) [11]	81.72	9.04
Abhishek et al. (CBP) [11]	81.36	9.02
Beltran et al. [10]	98.73	9.94

It can be appreciated in the table above that our proposed method (Autoencoder and MLP) achieves a lower MSE and RMSE on average, with the single MLP being the next best algorithm. In order to assess whether both methods were statistically different, we carried out two one-tailed t-test ($\alpha = 0.01$) with the results obtained from both methods (one for the MSE and another for the RMSE). The null hypothesis for the test is that the average for both methods is equal, whereas the alternative hypothesis is that the difference between the average obtained by our method (Autoencoder and MLP) and the single MLP is less than zero. The first t-test was run with data from the MSE, and it obtained a p-value of $2.2E - 16$ supporting the alternative hypothesis and then assessing that our proposal was able to obtain a statistically lower average on the MSE than the single MLP. The second t-test was run for the RMSE and it obtained again a p-value of $2.2E - 16$ and, thus, concluding that our proposed method obtains lower RMSE than the single MLP. Hence, in both cases it is shown that our proposal is the best one for the case study.

We decided to explore further the results obtained by our proposal. For that matter, we compared the predictions obtained by our architecture with the real values for those data instances. Figure 3 shows the prediction with the proposed

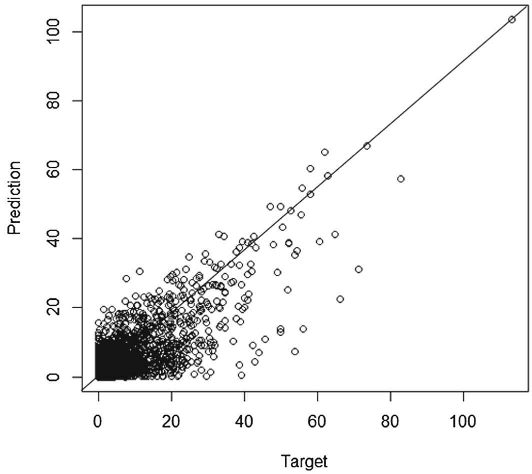

Fig. 3. Target versus prediction.

architecture versus the target data in the original unit of measurement. The horizontal axis represents the real value, whereas the vertical axis represents the prediction obtained by our algorithm. A perfect prediction would be place in a straight line with slope equal to one and intercept equal to zero. As it is possible to observe, data points tend to follow a straight line around the aforementioned straight line. This specially true for days with heavy rain, which are the focus of our study. For days with light rainfall (less than 20mm[1]), which are less interesting for our study as we aim to predict risk scenarios, the predictions tend to be less accurate and deviate from the target value. This indicates that there is room for improvement in the proposed architecture, and one of our goals in the future is improving the predictions for such cases.

6 Conclusions and Future Work

This paper has presented a deep learning approach based on the use of autoencoders and neural networks to predict the accumulated precipitation for the next day. The approach forecasts the daily accumulated rainfall in a specific meteorological station located in a central area of Manizales city (Colombia). The proposed architecture has been compared with other state of the art methods. The results suggest that our proposed architecture outperform other approaches in terms of the MSE and the RMSE.

As future work, we plan to improve our architecture for light rain scenarios. It should be highlighted that the focus of our study was precisely the opposite, heavy rain scenarios, as they are the scenarios that may lead to negative consequences (e.g., landslides). Additionally, we also plan to test other types of

[1] This threshold was obtained by talking with domain experts.

deep learning architecture like deep belief networks based on Restricted Boltzmann Machines [20] or multiple stacked denoising autoencoders. Moreover, we also consider the inclusion of new input features. More specifically, we plan to include data points of neighbor stations (considering neighborhood as stations with geographical proximity and similar altitude).

Acknowledgements. This work is partially supported by the MINECO/FEDER TIN2012-36586-C03-01 of the Spanish government.

The authors thank the Instituto de Estudios Ambientales - IDEA, of Universidad Nacional de Colombia - Sede Manizales for their help in obtaining the data and guidance regarding the operation of environmental monitoring networks and hydrometeorological stations.

References

1. World Health Organization: Climate Change and Human Health: Risks and Responses. World Health Organization, January 2003
2. Alcntara-Ayala, I.: Geomorphology, natural hazards, vulnerability and prevention of natural disasters in developing countries. Geomorphology **47**(24), 107–124 (2002)
3. Nicholls, N.: Atmospheric and climatic hazards: Improved monitoring and prediction for disaster mitigation. Natural Hazards **23**(2–3), 137–155 (2001)
4. Zhang, G., Eddy Patuwo, B., Hu, M.Y.: Forecasting with artificial neural networks: The state of the art. Int. J. Forecast. **14**(1), 35–62 (1998)
5. Bengio, Y.: Learning deep architectures for AI. Found. Trends Mach. Learn. **2**(1), 1–127 (2009)
6. Arel, I., Rose, D.C., Karnowski, T.P.: Deep machine learning - A new frontier in artificial intelligence research. IEEE Comput. Intell. Mag. **5**(4), 13–18 (2010)
7. Lngkvist, M., Karlsson, L., Loutfi, A.: A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recogn. Lett. **42**, 11–24 (2014)
8. Sawale, G.J., Gupta, S.R.: Use of artificial neural network in data mining for weather forecasting. Int. J. Comput. Sci. Appl. **6**(2), 383–387 (2013)
9. Luk, K.C., Ball, J.E., Sharma, A.: A study of optimal model lag and spatial inputs to artificial neural network for rainfall forecasting. J. Hydrol. **227**(14), 56–65 (2000)
10. Beltrán-Castro, J., Valencia-Aguirre, J., Orozco-Alzate, M., Castellanos-Domínguez, G., Travieso-González, C.M.: Rainfall forecasting based on ensemble empirical mode decomposition and neural networks. In: Joya, G., Gabestany, J., Rojas, I. (eds.) IWANN 2013, Part I. LNCS, vol. 7902, pp. 471–480. Springer, Heidelberg (2013)
11. Abhishek, K., Kumar, A., Ranjan, R., Kumar, S.: A rainfall prediction model using artificial neural network. In: 2012 IEEE Control and System Graduate Research Colloquium (ICSGRC), pp. 82–87, July 2012
12. Liu, J.N.K., Li, B.N.L., Dillon, T.S.: An improved naive Bayesian classifier technique coupled with a novel input solution method [rainfall prediction]. IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. **2**, 249–256 (2001)
13. James, N.K., Liu, Y.H., You, J.J., Chan, P.W.: Deep Neural Network Based Feature Representation for Weather Forecasting, 261–267 (2014)

14. Kapoor, A., Horvitz, Z., Laube, S., Horvitz, E.: Airplanes aloft as a sensor network for wind forecasting. In: Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, IPSN 2014, pp. 25–34. Piscataway, NJ, USA, IEEE Press (2014)
15. Grover, A., Kapoor, A., Horvitz, E.: A Deep Hybrid Model for Weather Forecasting, pp. 379–386. Sydney (2015)
16. Vassiliadis, P.: A survey of extracttransformload technology. Int. J. Data Warehouse Min. **5**(3), 1–27 (2009)
17. Duque-Mndez, N.D., Orozco-Alzate, M., Vlez, J.J.: Hydro-meteorological data analysis using OLAP techniques. DYNA **81**(185), 160 (2014)
18. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, ICML 2008, pp. 1096–1103. ACM, New York, NY, USA (2008)
19. LISA lab. Deep Learning Tutorials DeepLearning 0.1 documentation
20. Larochelle, H., Bengio, Y.: Classification using discriminative restricted boltzmann machines. In: Proceedings of the 25th International Conference on Machine Learning, ICML 2008, pp. 536–543. ACM, New York, NY, USA (2008)