INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 6.078
(Volume 6, Issue 3)
Available online at: www.ijariit.com

Rainfall prediction – A deep learning approach

Natarajan Iyer
thenkiyer8@gmail.com
Don Bosco Institute of Technology, Bangalore, Karnataka

Ranjeet Kumar
ranjeetcsedbit@gmail.com
Don Bosco Institute of Technology, Bangalore, Karnataka

Rohini Chamling
rohinichamling1697@gmail.com
Don Bosco Institute of Technology, Bangalore, Karnataka

Smruti Mishra
smruti.mishra175@gmail.com
Don Bosco Institute of Technology, Bangalore, Karnataka

Sourabh Singh
sourabh1007@gmail.com
Don Bosco Institute of Technology, Bangalore, Karnataka

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 accumulated daily precipitation for the next day. More specifically, it includes an auto encoder 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: ***Deep Learning, Auto encoder, Multilayer Perceptron***

1. INTRODUCTION

Rainfall is a climatic factor that affects many human activities like agricultural production, construction, power generation, forestry and tourism, among others. 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 affected society for years. Therefore, having an appropriate approach for rainfall prediction makes it possible to take preventive and mitigation measures for these natural phenomena. Over the last few years, Deep Learning has been used as a successful mechanism in ANN for solving complex problems. 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.

2. LITERATURE SURVEY

Sawale & Gupta 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).

Luk et al. presents 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.

Beltrn-Castro et al. adopt a decomposition and ensemble principle for daily rainfall forecasting. The decomposition technique employed is the Ensemble Empirical Mode Decomposition (EEMD), which 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. 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.

Finally, ***Grover et al.*** proposed has a novel hybrid model with discriminative and generative components for spatio-temporal inferences about the variables mentioned above. The model is called ”Deep Hybrid Model” and the results obtained are compared with other model proposed with the model used by the National Oceanic and Atmospheric Administration (NOAA) of the United States, showing a better performance than these. Forecasts are made at 6, 12 and 24 hours; however, the rainfall is not predicted.

3. PROPOSED SYSTEM

3.1 Prediction using random forest algorithm:

To predict consuming the trained random forest algorithm, the following pseudo code can be utilized:

1. Exhausting provided test features and using the rules by each randomly generated decision tree, so as to infer the outcome and storing these predicted outcomes (ie,targets).
2. Calculating the votes for every predicted target.
3. Considering highly voted inferred target as final prediction from random forest algorithm.

Assume that 100 random decision trees are created to form a random forest. Now, each of these random forests will foresee different target (or outcome) for same test feature. The target votes are then computed, considering every predicted target votes.

3.2 XGBoost Features:

The present library is totally focused on the computational speed and performance of the model, as there are some frills.

The three major methods of the gradient boosting that are supported are the following:

- Gradient Boosting method or algorithm, which is also known as gradient boosting machine together with learning rate.
- Stochastic Gradient Boosting using sub sampling at column, row, and the column per-split level.
- Regularized Gradient Boosting, which is using both the L1 and L2 regularization.

The two reasons for using XGBoost are also two goals of project:

- Execution speed of XGBoost, which is faster when matched too their implementations using gradient boosting.
- Model performance, which is better than most of the algorithms using gradient boosting as XGBoost dominates the tabular or structured datasets on regression and classification prediction modeling problems. The data was preprocessed before using, which contains removing of outliers and missing values.'\

3.3 Artificial neural network (ANN) :

ANNs are popular machine learning algorithms that are in a wide use in recent years. Multilayer Perception (MLP) is the basic form of ANN that updates the weights through back propagation during the training. There are other variations in neural networks, which are recently, became popular in texture classification Probabilistic Neural Network (PNN): It is derived from Radial Basis Function (RBF) network and it has parallel distributed processor that has a natural tendency for storing experiential knowledge.

3.4 Back propagation:

A typical BP network consists of three parts: input layer, hidden layer and output layer. Three parts in turn connect through the collection weight value between nodes.

The largest characteristic of BP network is that network weight value reach expectations through the sum of error squares between the network output and the sample output, and then it continuously adjusted network structure's weight

3.5 Support vector machine :

SVM is a non-linear classifier, and is a newer trend in machine learning algorithm. SVM is popularly used in many pattern recognition problems including texture classification. SVM is designed to work with only two classes. This is done by maximizing the margin from the hyper plane. The samples closest to the margin that were selected to determine the hyper plane is known as support vectors. Multiclass classification is applicable and basically built up by various two class SVMs to solve the problem, either by using one-versus-all or one.

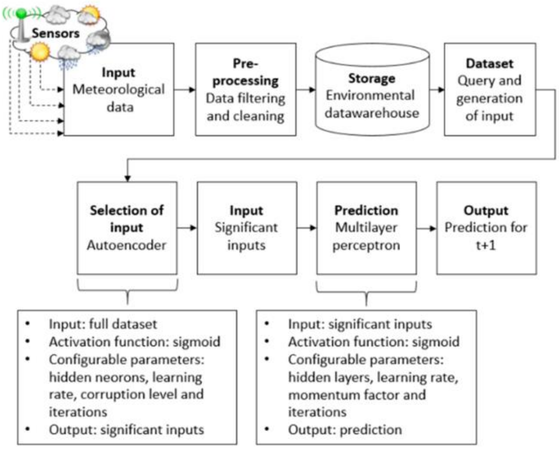

Chart 1 – System architecture

4. METHODOLOGY

4.1 Data preparation :

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

4.2 Feature extraction :

The main tasks of this step include determining the structure of MLP using a greedy algorithm and determining MLP parameters using SGD with momentum. We first explain the main idea and the process of building MLP. To predict rainfall in a certain area, the model establishes links between the forecasting area and surrounding areas. This connection refers to the relationship between the thirteen factors and the rainfall in the forecasting area. Since MLP has a strong learning ability, DRCF uses MLP to learn this connection. The input of MLP is the data preprocessed by PCA, and the output is the forecasting area.

4.3 Rainfall prediction :

In this step, we use multiple MLPs to forecast rainfall, and these MLPs have been trained in the previous step. The main work of this step is to determine the appropriate number of MLPs to be used. A single MLP may predict with large errors. To improve prediction accuracy, several MLPs are used to make decisions together. These MLPs must forecast rainfall from different aspects so that they can make up for each other. The perception process mentioned in the previous section can solve this problem. Taking the forecast area as the center area, we can establish an MLP with all the surrounding areas. The number of surrounding areas directly determines how many MLPs can be established. These MLPs can be used to forecast rainfall in the area.

4.4 Multi-layer perceptron:

Formally, a one-hidden-layer MLP is a function $f: R^D \rightarrow R^L$, where D is the size of input vector x and L is the size of the output vector $f(x)$, such that, in matrix notation:

$$f(x) = G(b^{(2)} + W^{(2)}(s(b^{(1)} + W^{(1)}x)))$$

with bias vectors $b^{(1)}$, $b^{(2)}$; weight matrices $W^{(1)}$, $W^{(2)}$ and activation functions G and s.

The vector $h(x) = \Phi(x) = s(b^{(1)} + W^{(1)}x)$ constitutes the hidden layer. $W^{(1)} \in R^{D \times D_h}$ is the weight matrix connecting the input vector to the hidden layer.

Each column $W^{(1)}_{\cdot i}$ represents the weights from the input units to the i-th hidden unit. Typical choices for s include $tanh$, with $tanh(a) = (e^a - e^{-a})/(e^a + e^{-a})$, or the logistic $sigmoid$ function, with $sigmoid(a) = 1/(1 + e^{-a})$. We will be using $tanh$ in this tutorial because it typically yields to faster training (and sometimes also to better local minima).

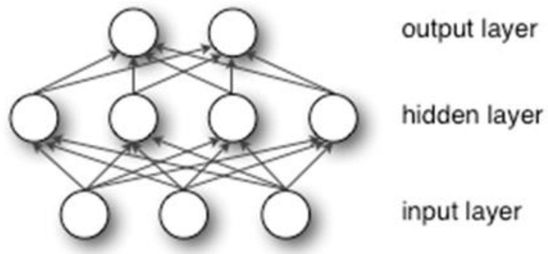

Chart 2- Multi-layer perceptron

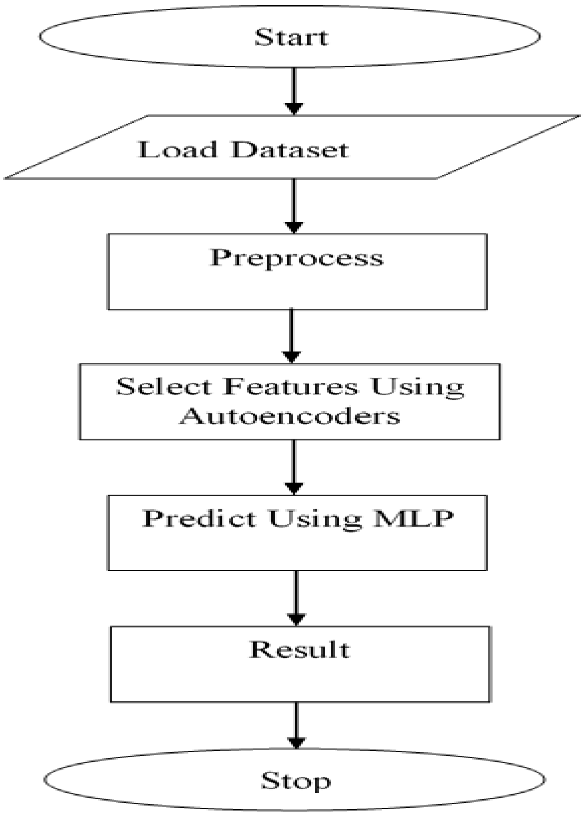

Chart 3 – Implementation of project

4.5 Data flow diagrams (DFD):

4.5.1 Level 0: Level 0 Describes the overall process of this project. We are passing metrological data as a input the system will predict the rainfall level using MLP classifier

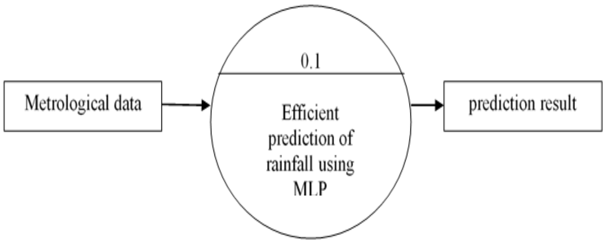

Chart 4- Level 0 DFD

4.5.2 Level 1: Level 1 Describes the first stage process of this project. We are passing Metrological data as a input the system will preprocess and extract the features using auto encoder.

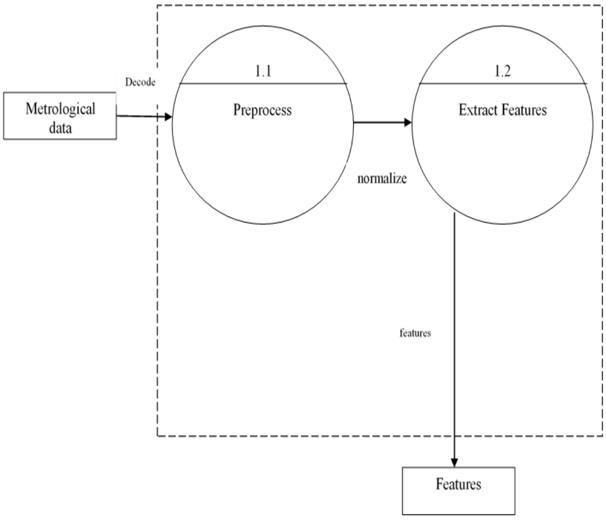

Chart 5 - Level 1 DFD

4.5.3 Level 2: Level 2 Describes the final stage process of this project. We are passing extracted features from level as input the system will predict the rainfall level using CNN-MLP classifier.

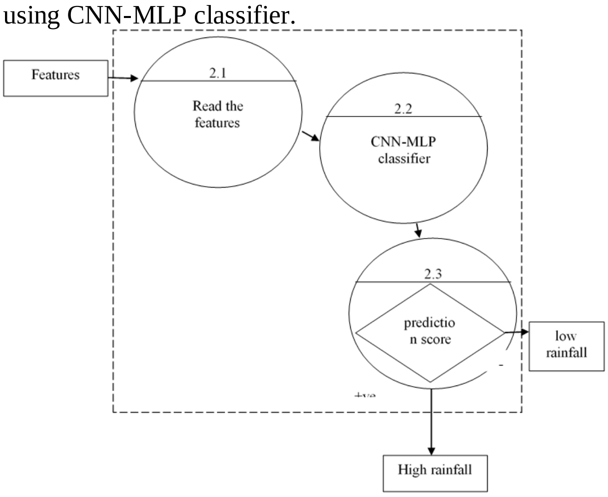

Chart 6 – Level 2 DFD

5. RESULTS

The result is a python based application that takes .csv files containing necessary metrics data as input and predicts rainfall according to the feature that has been extracted in the previous step.

5.1 Screen-shots of application:

5.1.1:

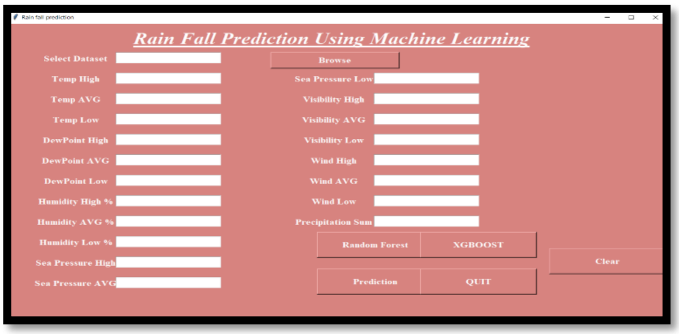

Chart 7- Initial stage

5.1.2:

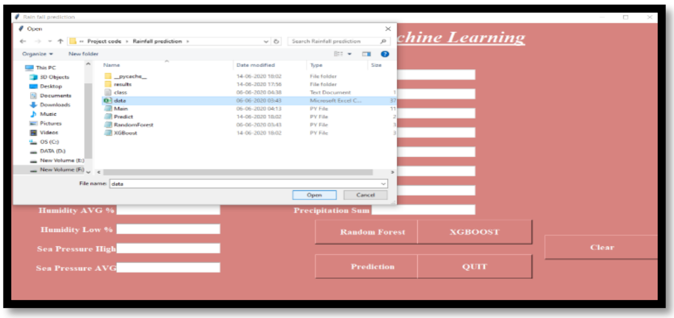

Chart 8 – Loading Dataset

5.1.3:

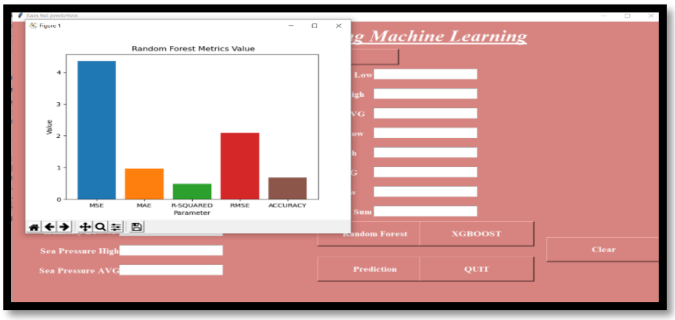

Chart 9 – Random forest metrics values

5.1.4:

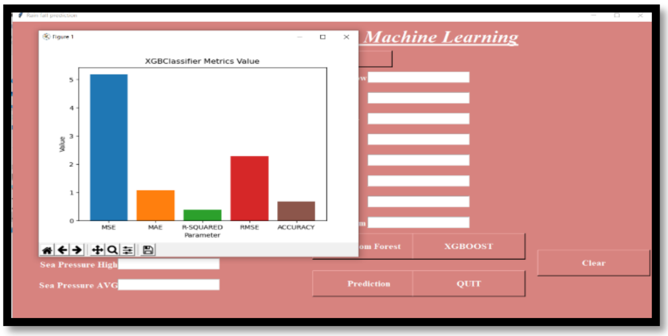

Chart 10 – Xgboost metrics values

5.1.5:

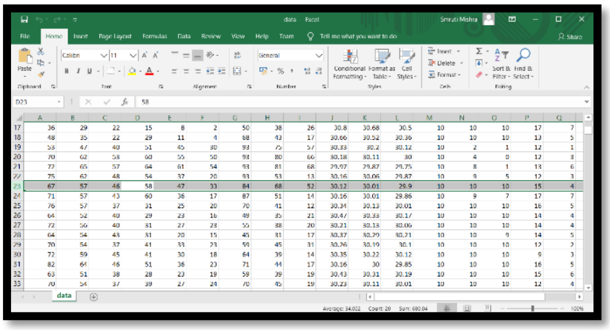

Chart 11 – Selection of data

5.1.6:

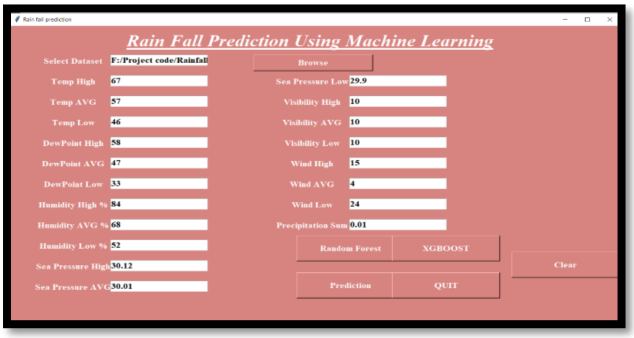

Chart 12- Input data

5.1.7:

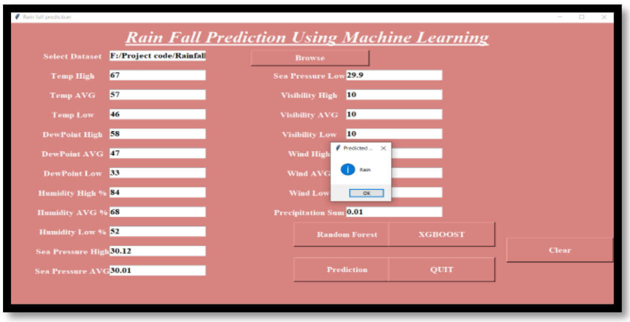

Chart 13 – Prediction result

6. CONCLUSION

This project has represented 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.

A result is the final consequence of actions or events expressed qualitatively or quantitatively. Performance analysis is an operational analysis, is a set of basic quantitative relationship between the performance quantities.

7. REFERENCES

[1] A.Kala, Dr. S .Ganesh Vaidyanathan, Prediction of Rainfall using Artificial Neural Network. IEEE Xplore Compliant Part Number:CFP18N67-ART; ISBN:978-1-5386-2456-2, (2018) IEEE.

[2] Choujun Zhan1, Fujian Wu1, Zhengdong Wu1 and Chi K. Tse. Daily Rainfall Data Construction and Application to Weather Prediction (2019) IEEE

[3] Aditya Grover, Ashish Kapoor, and Eric Horvitz. A Deep Hybrid Model for Weather Forecasting. pages 379–386, Sydney, 2015

[4] .Martin Lngkvist, Lars Karlsson, and Amy Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recognition Letters, 42:11–24, June 2014.

[5] Ashish Kapoor, Zachary Horvitz, Spencer Laube, and Eric Horvitz. Airplanes Aloft As a Sensor Network for Wind Forecasting. In Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, IPSN '14, pages 25–34, Piscataway, NJ, USA, 2014. IEEE Press.

[6] Nstor Dario Duque-Mndez, Mauricio Orozco-Alzate, and Jorge Julin Vlez. Hydrometeorological data analysis using OLAP techniques. DYNA, 81(185):160, June 2014.

[7] Gaurav J. Sawale and Sunil R. Gupta. Use of Artificial Neural Network in Data Mining For Weather Forecasting. International Journal Of Computer Science And Applications, 6(2):383–387, 2013.

[8] Juan Beltrn-Castro, Juliana Valencia-Aguirre, Mauricio Orozco-Alzate, Germn Castellanos-Domnguez, and Carlos M. Travieso-Gonzlez. Rainfall Forecasting Based on Ensemble Empirical Mode Decomposition and Neural Networks. In Ignacio Rojas, Gonzalo Joya, and Joan Gabestany, editors, Advances in Computational Intelligence, number 7902 in Lecture Notes in Computer Science, pages 471–480. Springer Berlin Heidelberg, 2013.

