Optimized Backpropagation Neural Network for Rainfall Forecasting

Vertika Shrivastava[1],Sanjeev Karmakar[2]

Email ID:mail2vertika@gmail.com, dr.karmakars@gmail.com

[1] Department of Computer Application, Maharaja Agrasen International College, Raipur, Chhattisgarh.

[2]Department of Computer Application, Bhilai Institute of Technology, Durg, Chhattisgarh.

Abstract

Deep learning has recently emerged as a viable method for solving complex problems and analyzing large amounts of data. The proposed method generated a precipitation forecasting model by analyzing precipitation data from India and predicting future precipitation using an optimized neural network. Rainfall prediction helps farmers to cultivate crops and improve the economy as well as the economy of the nation. This study aims to provide a comprehensive overview of current scientific studies for forecasting short-term precipitation based on area, month, and temperature on a geographic scale. This article provides an in-depth review and comparison of several neural network topologies used by experts to predict precipitation. The paper also discusses the difficulties of using different computational models to forecast annual/monthly rainfall. In addition, the article provides some accuracy metrics used by experts to evaluate ANN performance.

Keywords: BPNN, CPNN, ANN, MLP

DOI Number: 10.14704/nq.2022.20.13.NQ88116 **Neuro Quantology 2022; 20(13):909-915**

Introduction

Rainfall has the essential function in all types of meteorological events in terms of human existence. Rainfall has a massive impact on human civilization. Rainfall is a result of a natural phenomenon that is difficult to anticipate. Accurate rainfall data is critical for water resource planning and management, as well as reservoir operation and flood prevention. Effective rainfall prediction is critical for agriculture-dependent nations such as India. Rainfall forecast is critical for evaluating agricultural production, water resource usage, and water resource planning. Because of the dynamic character of climatic events, statistical rainfall forecasting methods only operate well for long-term rainfall prediction. This research aims to provide a thorough overview of current scientific studies for temporary warmth regions and month-based rainfall forecasts on a geographical scale. Some meteorological variables, such as impressive heaviness, precipitation, dew point temperature, solar radiation, vapor pressure, blur wrap, snowstorm, dampness, wind speed quickness, and storm way, have been shown in the literature to be possible predictors of future temperature and rainfall.

Systems for rainfall-runoff (RR) predictions that are efficient and reliable are critical for disaster risk reduction. However, most rainfall systems being used today are process-driven, which means they answer simplified empirical formulae. With the advancement of machine-learning methods, we may soon be able to simulate rainfall models using neural networks, for example the research article examines several ANN architectures to determine the best approach for such systems. However, it was seldom apparent that the most advantageous design of ANNs in terms of architectural parameters was required for the most beneficial performance. In this research, the architecture of ANNs such as BPNNs and linear regression will be investigated and improved for modeling of long-term monsoon rainfall across the Mahanadi River basin in Chhattisgarh, where skeletons of BPNNs will be evaluated for rainfall modeling. The following is how the paper is structured. Section

II provides a review of the literature on RF Prediction. Sections III and IV provide the background information as well as the suggested prediction. Section V explains the experimental findings and dataset details. Finally, Section VI, the proposed approach and its performance are summarized.

Literature Review

F. D. Diba et al. [1] examined rainfall monitoring findings from experimental measurements and deep learning methods, including artificial neural networks and long short-term memory. Experiment sets were in South Korea over terrestrial and satellite connections and in Ethiopia over terrestrial links for various frequency bands and link distances. N. Parashar et al. [2] concentrated on contemporary applications of machine learning and deep learning models such as Deep Echo State Networks (DESN), Recurrent Neural Networks (RNN), Convolution Recurrent Neural Networks (CRNN), Graph Convolution Networks, Auto encoders (GCNA), Multi-layer Perception (MlP), and Long Short-Term Memory (LTSM). The time gap between data recordings also significantly impacts forecast accuracy. Input data that was recorded more often performed better than data that was collected less frequently. N. Madhukumar et al. [3] developed a hybrid climate learning approach that combines the CM and deep learning models. A probabilistic multilayer perceptron network, in particular, assesses several predictions from the CM-generated forecasts and chooses the best one. The chosen prediction is then fed into a hybrid deep long short-term memory (HD-LSTM) network, which looks back and learns the connection between the chosen forecast and the associated rainfall and temperature data to create the next-day rainfall forecast. R. K. Grace et al. [4] present a rainfall prediction model for an Indian dataset using Multiple Linear Regression (MLR). The supplied data contains various meteorological characteristics that may be used to better precisely forecast rainfall. The parameters utilized to verify the proposed model are the Mean Square Error (MSE), accuracy, and correlation.

A. Samad et al. [5] use a Recurrent Neural Network (RNN) based on Long Short-Term Memory (LSTM) to forecast rainfall. The created model is trained and tested using a standard dataset. To improve prediction analysis, the time series rainfall data is pre-processed using Additive Seasonal Decomposition. The model is then given this pre-processed data. The LSTM model has been benchmarked using an Artificial Neural Network (ANN). Li, W., et al. [6] presented a data-driven RR model built using a sequence-to-sequence long-short-term-memory (LSTM) network. Because of the LSTM networks capacity to learn long-term relationships between its input and output, it was possible to simulate rainfall-runoff with great precision in time. Samantaray S. et al. [7] studied rainfall forecasting in Bolangir district, Odisha, India, utilizing data from the previous 48 years and different artificial neural network (ANN) techniques. As predicting inputs, several combinations of historical rainfall data were created to evaluate the effectiveness of ANN approximation. Rainfall analysis employs RNN, SVM, and ANFIS methods. Ouma et al. [8] compare the LSTM Neural Network (LSTMNN) with the Wavelet Neural Network (WNN) for the ratio-temporal prediction of rainfall and runoff time series trends in sparsely gauged hydrologic basins. The rainfall and runoff patterns in the Nzoia River basin they are projected using long term in situ observed data from 10 rain gauge sites and three discharge measuring stations spanning 30 years (1980–2009).

Background

The Mahanadi River Basin was chosen as the research region in our study; the Mahanadi River Basin is one of India's main river basins, with only the Ganga, The Godavari, and Krishna River basins having larger catchment areas. The Mahanadi delta area is the most vulnerable to flooding of any basin in India, and the Mahanadi's climate warming process is becoming more dangerous. Over 10 million people reside in the delta region, where population density is high. Three-quarters of Odisha's population is dependent on climate-sensitive natural resource-based livelihoods, focused on agriculture, forest, and fisheries, and deriving most of their income from agriculture. Mahanadi's atmosphere continues to change. The Mahanadi basin is

located between 80030' with 86050' East longitudes and 19020' and 23035' North latitudes.

The basin's entire catchment area is 1,41,600 km2. The drainage basin's typical height is nearly 426 meters with a high elevation of 877 meters along low elevation of 193 meters. The Mahanadi river's peninsula is a significant inter-state east-flowing river in peninsular India. It begins at a height of about 442 meters. It drains quite significant portions of Chhattisgarh and Orissa, as well as a smaller amount of Jharkhand and Maharashtra. The river's entire length starting from headwaters to its terminating point at the Bay of Bengal is approximately 851 km, 357 km of that length is in Chhattisgarh state while the rest of its length, i.e., 494 km in Orissa state. Many tributary rivers at the both sides of the main rivers as it flows. There are 14 main tributaries, 12 of which join upstream and two downstream of Hirakund reservoir. Upstream of the dam, about 65 percent of the basin is located. The yearly flow averages 1,895 m3/s, with a peak of 6,352 m3/s during the monsoon season. The lowest outflow is 759 m3/s and happens from October to June. The Mahanadi basin, like the rest of India, has a tropical monsoon climate. The months of July, August, and the first part of September often see the most precipitation. The basin's normal annual rainfall is 1360 mm (16 percent CV), of which about 86 percent, or 1170 mm, falls during the monsoon season (15 percent CV) during June to September (Rao,1993). The river flows through the tropical zone, where it is exposed to cyclonic storms and periodic rains. The average daily low temperature in the winter ranges from 4°C to 12°C. May is the warmest month, with mean daily maximum temperatures ranging from 42°C to 45.5°C.

Methodology

Network of Back Propagation (BPN) One of the most significant breakthroughs in neural networks is the back-propagation learning method [2]. This network remains the most often used and successful model for complicated, multi-layered networks. This learning method is used in multilayer feed-forward networks composed of processing units with continuously differentiable activation functions. Back-propagation of the network is related to the back-propagation learning method (VPNs). It is a technique of supervised learning. This method offers a mechanism for adjusting the weights in a BPN to properly categorize the supplied input sequence for a specified set of training input-output pairs. The fundamental idea behind this method is that it makes two runs across the network's various layers:

A fourth direction pass and a back-going pass. In the fourth direction pass, an input vector is imposed on the sensory nodes of the network, and the influence of it spreads across the various strips one by one. Finally, a collection of outputs is generated as the network's actual response. The synaptic weights of the networks are entirely set during the forward pass. During the reverse pass, however, all synaptic weights are changed in line with an error control rule. To generate an error signal, the network's actual response is deducted from the intended (target) response. This erroneous signal is subsequently transmitted back across the network, in the opposite way of the synaptic connection, giving rise to the term "error revert propagation. The synaptic loads are varied to bring the network's real response closer to the intended response in a statistical sense [1]. An input strip, an output strip, and minimum one hidden strip are common components of a back-propagation network. The counts of neuron in each strip, as well as the count of hidden strip, influence the network's capacity to produce correct outputs for a given data set. A number of researchers have utilized this network to forecast rainfall.

Optimized Backpropagation Neural Network(O-BPNN)

The Optimized Backpropagation network of the neuron is a kind of supervised neural forward feed network that belongs to the feed-forward neural network class. A generalized neural regression network (O-BPNN) was proposed, a method that utilizes neural networks to design and regulate non-linear systems and incorporates one-pass learning. The benefit of O-BPNN is that it "learns" in a single pass over the entire data and can evolve in standardized from the instance as soon as they are saved. It implies that using O-BPNN is beneficial because of its

ability to focus the standard underlying function of the data with few training samples. The O-BPNN is made up of four levels. Figure 1 shows the input strip, pattern strip, summation strip, and output strip. The neurons of the input in the input layer are simple distribution neurons that give entire measurement variables x to the complete neuron obtained in the second strip that is the second strip. The patterned strip is the O-BPNN's initial hidden strip. The neuron count in this strip equals the count of training samples in the training data set. The output of the pattern layer provides the distance between the input and stored patterns, while each neuron acts as a training pattern. The third layer is a summation layer with two neurons: one for denominator summation, which combines the weight values from each hidden neuron, and another for an overview of the numerator, which sums the importance of the weight multiplied to the actual values for every hidden neuron. The fourth layer is the output layer which takes both outputs from the hidden strip and distributes them to give the expected outcome.

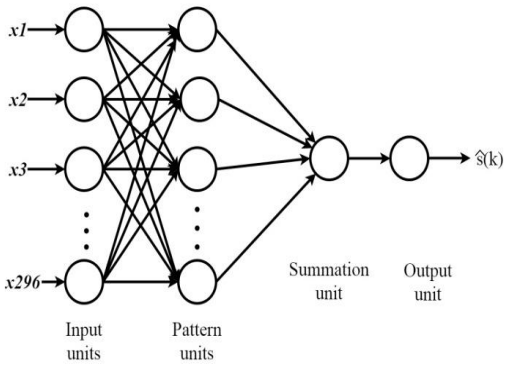


Figure:1 O-BPNN Architecture

The extracted characteristics are retrieved from the picture after pre-processing. These retrieved characteristics are utilized as input feature vectors to the neural network that is employed for tumor detection. Concatenation of all retrieved features is used for O-BPNN training. O-BPNN is trained on 40 samples from our database with a deadly head. For testing purposes, a trained model is utilized. The O-BPNN model consists of 100 input neurons, 100 neurons of the hidden strips, and one output neuron. For training purposes, the normalized radial basis function is employed. O-BPNN is a probability-based neural network often utilized in applications such as prediction, control, plant process modeling, and general mapping.

Because O-BPNN is a feed-forward back-propagation neural network variant that can train in a single run, it quickly converges to the subjacent regression area. It is well established in its capacity for quickly preparing to tackle any regression issues. It computes the distance between the patterns in the training set and a given sample pattern. As illustrated in Figure 1 b, the O-BPNN has four strips: an input strip, a pattern strip, a summation strip, and an output strip. The input strip receives the input signals, and the pattern strips conduct the required mapping across the applied data of input using the data set of the training. In the hidden layer of the O-BPNN, the count of the neuron typically equals the count of the pattern in the training set. The outcome of the pattern strip nodes are together at the summation layer with suitable connectivity weights before being multiplied. The output layer nodes provide the expected result according to the supplied dataset. The O-BPNN will estimate the value of M for any collection of N inputs in less time than the propagation time. The network's propagation time is the amount of time it takes to pass the input through it. M is supposed to represent the estimated (k) signals of interest (k). As shown in Fig.1b, N represents the values of the input feature vectors from the input nodes. The estimated M (N) (i.e.s (k)) is estimated using O-BPNN as follows.

$$M(N) = \frac{\sum_{i=1}^{n} M_i \exp\left(\frac{-D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{n} \exp\left(\frac{-D_i^2}{2\sigma^2}\right)} \qquad (1)$$

$$D_i^2 = (N - N_i)^T . (N - N_i) \qquad (2)$$

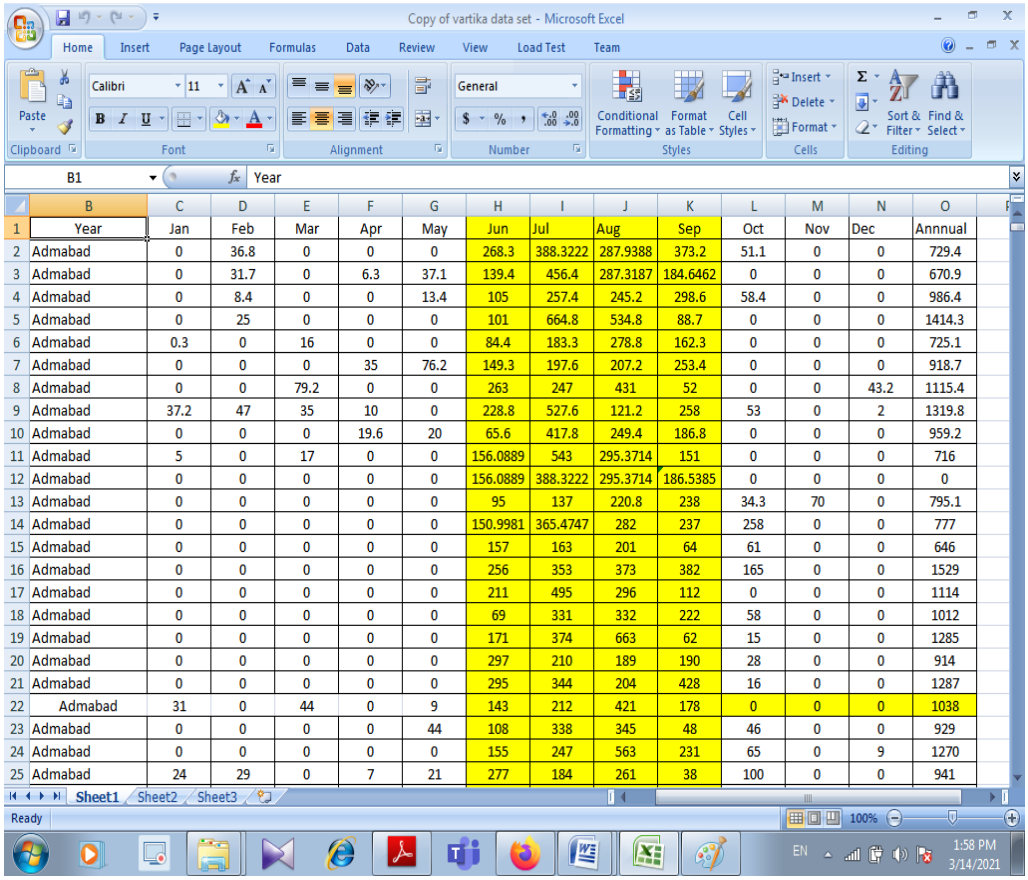
The smoothing factor for O-BPNN is, and the number of input data samples is n. Choosing a smoothing factor is critical. A high value increases the network's capacity to generalize, whereas a low value decreases the network's ability to generalize. (The higher the smooth value, the better the function approximation, and vice versa.) Estimate M (N) is a ratio of the total of all perceived values Mi, where each perceived value is measured to its exponential rate by its Euclidean distance from N. In this study, O-BPNN inputs are generated features from a breast

picture that serve as input features for the vectors x 1 to x m. The output of O-BPNN, on the other hand, depicts the tumor portion of picture s(k), as illustrated in Fig.

Dataset

Using chaotic data time series analysis, rainfall data may be utilized to forecast rain fall. However, the data is considered to consist of a structured prototype and unintentional clatter, which often makes the prototype challenging to classify. Time series patterns may be divided into two categories: trend and seasonality. Seasonality may have a formally similar temperament, and it may repeat itself in frequent intervals throughout time. We gathered meteorological data from the neighboring Mahanadi 138 station in our research.

For example, Bhagon, Bhatapara, Raipur, Bilaspur, Bhalukona, Bhanupratappur, Bhatapara, Raipur Bhilai, Bilpur, Bhatapara, Champa, Chandai, Charoda, Chhati, Gudhiyari, Dhoarg, Gandi, Gangral, Kanker, Kaki, Kasd



	Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Annnual
2	Admabad	0	36.8	0	0	0	268.3	388.3222	287.9388	373.2	51.1	0	0	729.4
3	Admabad	0	31.7	0	6.3	37.1	139.4	456.4	287.3187	184.6462	0	0	0	670.9
4	Admabad	0	8.4	0	0	13.4	105	257.4	245.2	298.6	58.4	0	0	986.4
5	Admabad	0	25	0	0	0	101	664.8	534.8	88.7	0	0	0	1414.3
6	Admabad	0.3	0	16	0	0	84.4	183.3	278.8	162.3	0	0	0	725.1
7	Admabad	0	0	0	35	76.2	149.3	197.6	207.2	253.4	0	0	0	918.7
8	Admabad	0	0	79.2	0	0	263	247	431	52	0	0	43.2	1115.4
9	Admabad	37.2	47	35	10	0	228.8	527.6	121.2	258	53	0	2	1319.8
10	Admabad	0	0	0	19.6	20	65.6	417.8	249.4	186.8	0	0	0	959.2
11	Admabad	5	0	17	0	0	156.0889	543	295.3714	151	0	0	0	716
12	Admabad	0	0	0	0	0	156.0889	388.3222	295.3714	186.5385	0	0	0	0
13	Admabad	0	0	0	0	0	95	137	220.8	238	34.3	70	0	795.1
14	Admabad	0	0	0	0	0	150.9981	365.4747	282	237	258	0	0	777
15	Admabad	0	0	0	0	0	157	163	201	64	61	0	0	646
16	Admabad	0	0	0	0	0	256	353	373	382	165	0	0	1529
17	Admabad	0	0	0	0	0	211	495	296	112	0	0	0	1114
18	Admabad	0	0	0	0	0	69	331	332	222	58	0	0	1012
19	Admabad	0	0	0	0	0	171	374	663	62	15	0	0	1285
20	Admabad	0	0	0	0	0	297	210	189	190	28	0	0	914
21	Admabad	0	0	0	0	0	295	344	204	428	16	0	0	1287
22	Admabad	31	0	44	0	9	143	212	421	178	0	0	0	1038
23	Admabad	0	0	0	0	44	108	338	345	48	46	0	0	929
24	Admabad	0	0	0	0	0	155	247	563	231	65	0	9	1270
25	Admabad	24	29	0	7	21	277	184	261	38	100	0	0	941

Figure.2. Screenshot of Dataset of different stations

Experimental Result

Rainfall	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Actual	24.55631	22.76	20.12852	7.94215	16.57848	187.1919	352.3904	349.9124	184.7734	61.49665	6.249006	12.30946
OBPNN	12.24818	38.91615	9.81219	33.02525	39.76464	196.2551	351.7922	344.6531	181.319	66.24022	19.48065	33.78198
MLP	21.21498	23.76601	19.80468	15.48618	71.90322	206.3965	387.21	340.8202	231.7689	81.73046	28.91663	37.82342
CPNN	78.61232	18.61102	51.4975	27.50246	43.3311	207.8043	421.6278	339.3677	329.7083	203.123	90.04907	87.35141

Table :1 Results

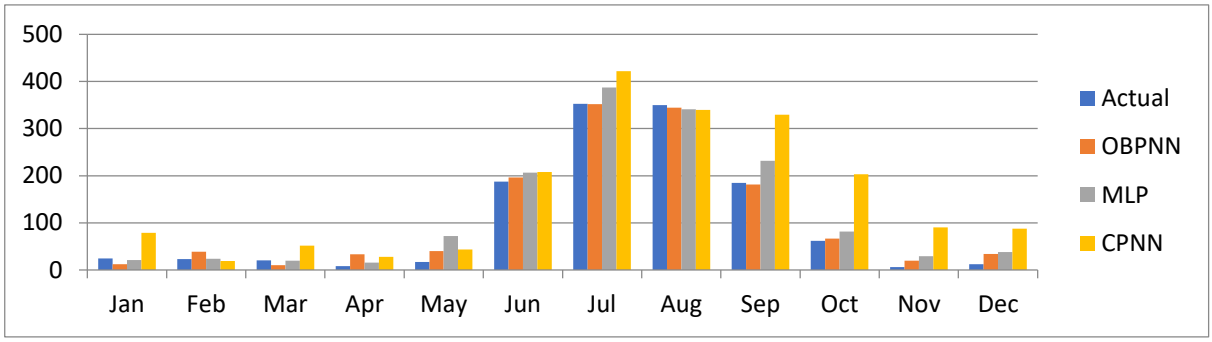


Figure 3: Comparative analysis of proposed technique with well-known methods (Month wise)

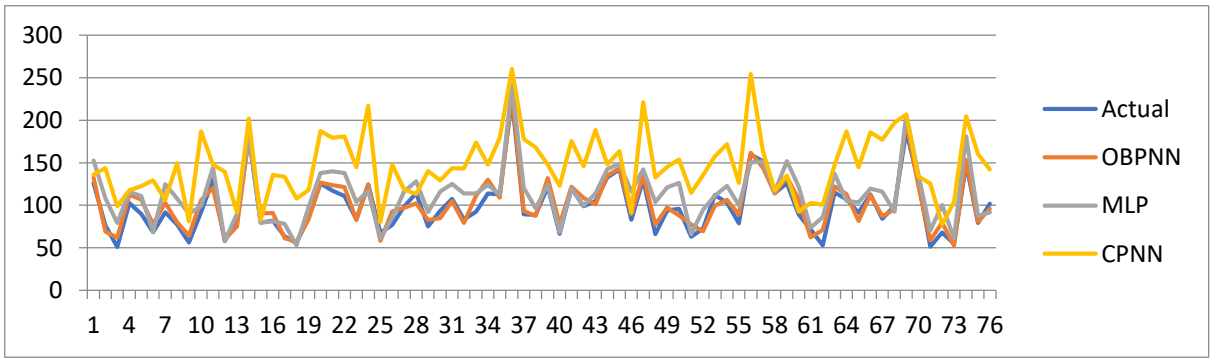


Figure 4: Comparative analysis of proposed technique with well-known methods (Citywide)

Conclusion

It is found that the suitability of ANN for Weather Forecasting has dramatically improved the performance and the result accuracy. It is also found that the different types of a network of neural packages training and investigating algorithms like Artificial Neural Networks (ANN), Back Propagation Neural Network, (BPN). Since weather forecasting is a dynamic and non-linear process, ANN can be used to predict the weather. The research also found that ANN is the best approach to Numerical and traditional methods.

REFERENCES

1] F D. Diba et.al IEEE Access, vol. 9, pp. 66769-66780, 2021, doi: 10.1109/ACCESS.2021.3076781.

2] N. Parashar et al., International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2021, pp. 742-746,2021 DOI: 10.1109/ICACITE51222.2021.9404767.

3] N. Madhukumar et al., in IEEE Internet of Things Journal, vol. 8, no. 9, pp. 7270-7278, 1 May 2021, DOI: 10.1109/JIOT.2020.3040736.

4] R. K. Grace et al., 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 227-229, DOI: 10.1109/ICACCS48705.2020.9074233.

5] A. Samad et al., IEEE 5th International Conference on Computing Communication and Automation (ICCCA), 2020, pp. 190-195, DOI: 10.1109/ICCCA49541.2020.9250809.

6] Li, W et.al. 33, 1261–1278 (2021). https://doi.org/10.1007/s00521-020-05010-6

7] Samantaray S. et. vol 159. Springer, Singapore. https://doi.org/10.1007/978-981-13-9282-5_74

8] Ouma et.al https://doi.org/10.1007/s40747-021-00365-2