Rainfall Prediction for Khordha District in Odisha using NARX Neural Network

Prakhar Golchha[1], Palak Lunia[2], Muktikanta Sahu[3]
[1]*International Institute of Information Technology, Bhubaneswar*
[2]*International Institute of Information Technology, Bhubaneswar*
[3]*International Institute of Information Technology, Bhubaneswar*
(E-mail: prakhar686@gmail.com
palaklunia23@gmail.com
muktikanta@iiit-bh.ac.in)

***Abstract*—**Predicting future rainfall from past recorded data for a certain geographical region can be classified as a non-linear problem. To do an accurate prediction of rainfall, a non-linear prediction model can be built using artificial neural networks. Dynamic neural network such as Non-linear Autoregressive models with exogenous input (NARX) is used for prediction as well as nonlinear filtering of time series data. In this paper, the NARX model has been used to predict the yearly rainfall of Khordha district in Odisha which lies in the eastern coast of India by doing a time series analysis of the past rainfall data available for the same region. Outcomes of the proposed prediction model are measured in terms of Mean Square Error (MSE). The minimum MSE value that we were able to get is 0.000032.

***Keywords*—**Time series data; Dynamic Neural Networks; NARX model; Levenberg-Marquardt algorithm

I. INTRODUCTION

India is a land of agriculture and to have a healthy agriculture proper rainfall is required. Farming in India is mostly dependent on monsoon rains. The factors like average humidity level, atmospheric pressure, wind speed, direction and average temperature in a year affect the average rainfall for a particular year. Controlling these environmental factors are beyond the human capability. However, we can predict the rainfall on the basis of past data by applying some modern computational techniques. Statistical methods are more suitable for linear data to do any kind of prediction. As rainfall data is highly nonlinear in nature, the dynamic Neural Networks are more suitable for doing rainfall prediction through time series analysis. Non-linear Autoregressive models with exogenous input (NARX) are equally powerful as Turing Machines, since the NARX model have limited feedback that is generated from the output neuron only and not by any hidden neurons. NARX neural network converges faster and when applied with Levenberg-Marquardt algorithm gives the output in very less time and thus most suited for nonlinear problems like rainfall prediction. As Indian agriculture is largely dependent on monsoon rainfall so it becomes the major task of researchers to predict the rainfall in various parts of the country. Currently our focus is to predict the rainfall in Khordha district of Odisha state which lies in the eastern coast of India. Prediction accuracies of our proposed neural network model for the rainfall data are shown through MSE graphs.

II. LITERATURE REVIEW

In [1] a model is developed for prediction of rainfall intensity in time and space using neural network. One hour of lead time is taken for testing data. It was observed that the technique works well in most of the cases and performs better when more number of hidden nodes are used.

In the paper [2] the authors have proposed a neural network based method to predict rainfall using geostationary meteorological satellite images.

In [3] Zhang and Scofield used a neural network based method to do a heavy convective rainfall estimation and cloud merger recognition with the help of satellite data.

A multi-layered feed-forward neural networks is trained with the error-back-propagation (EBP) algorithm in [4] to predict the monsoon rainfall in India.

In [5] the researchers have proposed a technique where they have divided the locations in Switzerland into four regions and applied different neural networks technique in each of the regions. It was observed that rainfall prediction in larger regions, which were formed by RBF networks gave better predictions than those that were implemented by Orographic Effect using linear models.

The authors in [6] proposed the real time rainfall prediction models for the purpose of flood forecasting in Italy. Their analysis indicates that the ANN based techniques give the best prediction results. Three different neural networks models such as the multilayer feedforward neural networks, the partial recurrent neural networks, and the time delay neural networks were considered in [7] for forecasting rainfall over an urban catchment in western Sydney, Australia.

The authors in the paper [8] compared Multivariate Adaptive Regression Splines(MARS) with soft computing techniques for long-term rainfall predictions for the state of Kerela located in the Indian peninsula. It was observed that MARS performs better than the picked soft computing tools. In the paper [9] the researchers worked on self-organizing map, fuzzy rule systems and backpropagation neural networks to predict the rainfall of an unknown location by having a prior knowledge of nearby locations.

In [10] the researchers have used soft computing technique, Multilayer Perceptron (MLP) to predict the rainfall in India. The performance of the model is determined by computing the overall prediction error (PE). An optimal data-driven model is introduced in [11] to improve the accuracy of rainfall forecasting in India and China. Authors have concluded that the model works good when modular artificial neural network (MANN) is combined with singular spectrum analysis (SSA).

The researchers in paper [12] used monthly total rains (mm) and the respective rain days to determine rain intensity in Athens, Greece using artificial neural network.

Feed Forward Neural Network (FFNN) model and Multi Regression (MLR) were implemented for rainfall forecasting in [13] for city Alexandria, Egypt and it was observed that FFNN model performs better than MLR model due to its non-linear nature. Comparison of Autoregressive integrated moving average (ARIMA) and Adaptive Network Based Fuzzy Inference System (ANFIS) is done in [14].

The parameters considered are average temperature, air pressure, and wind-speed. It was observed that ANFIS model gives better results than ARIMA model as it is simple as well as reliable in time series prediction.

Karmakar et al. developed a probabilistic Artificial neural network and deterministic feed forward back propagation models in [15] for long-range seasonal rainfall prediction of Subdivision ‘EPMB’, Chhattisgarh .

Work done by Vamsidhar et al. in [16] predicts the rainfall in India by applying back-propagation neural network models with humidity, dew point and pressure as parameters.

Geetha et al. in [17] proposed an approach to increase the performance of monthly rainfall forecasting of Chennai city, using back-propagation as an ANN technique. The developed ANN model takes wind speed, relative humidity, mean temperature etc as its parameters.

Rainfall prediction of Delhi in [18] is done using back-propagation algorithm. The paper focuses on training and testing of data set along with studying the attributes of hidden neurons.

NARX neural network model in [19] is compared with classic Box-Jenkins (AR) model and other standard neural networks methods. The authors implemented all the models to forecast monthly rainfall precipitation in the city of Fortaleza, located at Brazil. It was observed that NARX model outperforms the rest of the models.

To provide an early warning to the Disaster Risk Reduction Management Office (DRRMO) community, back-propagation approach, Levenberg-Marquardt optimizing algorithm and feed forward architecture is used in [20] to develop an NARX model for floods affected areas of Philippines such as Masantol and Pampanga.

The data driven models like ANN having dynamic network i.e., which can handle non-stationary data, and ARIMA, which is a statistical technique and works well only for stationary data were compared to predict rainfall of Hyderabad, India in [21] and of Malaysia in [22] and the authors from both the research confirmed that the performance of ANN is far better than ARIMA.

A conjunction model in [23] is developed by the combination of ANN and singular spectrum analysis. The monthly precipitation data from rain gauge station, Ponel station, in Iran, are used for implementation. For analysing the results, root mean square error (RMSE), correlation coefficient (R) and coefficient of efficiency (CE) statistics are used.

In [24], multi-layer perceptron (MLP) network is trained using three meta-heuristic algorithms namely Centripetal accelerated particle swarm optimization (CAPSO), Imperialist competitive algorithm (ICA) and Gravitational search algorithm (GSA) to enhance the accuracy of rainfall forecasting in Johor State, Malaysia. And it was observed that CASPO algorithm provides the best results in terms of accuracy of rainfall forecasting, minimum error and classification accuracy.

In the paper [25] back-propagation Neural Network, Support Vector Machine and decision tree algorithm are used for rainfall prediction in India using parameters like temperature, pressure, wind speed, wind direction, relative humidity, cloud cover and precipitation.

Odisha is one of the most rain affected regions in India as it is located at the east coast of India and on the bank of Bay of Bengal. This plateau is one of the most prone areas affected by natural disasters like floods, cyclones and droughts. The Cyclone of 1989, 1999, 2011, 2013 [26] in Odisha had affected the land worst with huge loss of lives and property.

Rainfall is the major contributor of such disasters and it becomes necessary to develop a technology which can correctly predict the rainfall. Bhubaneswar is not only the capital of Odisha but also is the most urbanized city in Odisha which lies in Khordha district.

Thus, predicting rainfall for Khordha district would ultimately contribute the rainfall prediction for Bhubaneswar. As per our extensive research, there is no work available for rainfall prediction of Khordha district in Odisha. Hence the goal of this paper is to develop a model or a technique using ANN, that can precisely predict the rainfall for the coming year.

III. ARTIFICIAL NEURAL NETWORK

There had been a great research in development of intelligent systems and one such development is inspired by the study of brain and nervous system, whose most essential component are neurons. A neuron is a nerve cell and is the main component to process the information. Artificial Neural Networks (ANN) are massively parallel computing systems with huge number of interconnected simple processors that can solve a variety of complex computational problems by behaving like natural neurons in a human body. An ANN resembles weighted directed graph in which the nodes are artificial neurons and directed edges with weights are the connections between input and output neurons. ANN was discovered in early 1980s [27]. The major contributions in highlighting the effectiveness of ANN was by the approaches developed by Hopfield in [28].

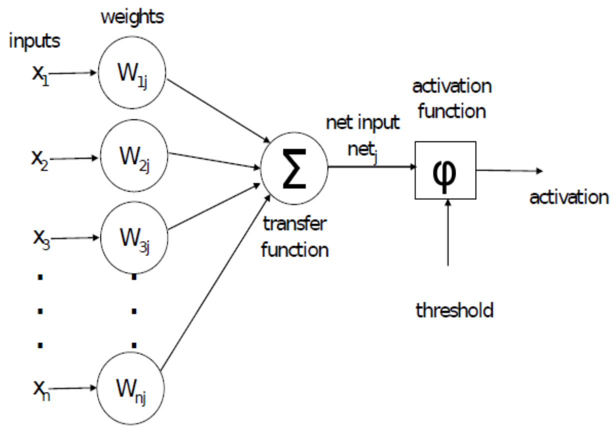

Fig 1: Basic structure of an ANN

Besides this, the work done on multilayer feedforward Networks by Werbos in [29] is equally appreciable. Later in 1986, it was made popular by Rumelhart et al. [30]. Further Anderson and Rosenfeld [31] provided a comprehensive study of ANN developments. A structural representation of an ANN is shown in Figure 1. ANNs is categorized into two groups as feedforward networks and recurrent networks. Feedforward neural networks are static in nature and are memory less i.e. they cannot store the previous state results and hence every network state is independent of each other. Whereas recurrent neural networks are dynamic in nature as the feedback paths are available here. The output from the feedback path sometimes act as an input neuron to the next network state [32]. ANN are widely used in function approximation, data fitting, classification and pattern recognition. In [33] neural networks are used for time series forecasting of seasonal and trend patterns. A hybrid approach of combining conventional and ANN techniques to deal with the non-linear complex time series systems is proposed in [27].

A. NARX Model

Nonlinear Auto Regressive models with exogenous inputs (NARX) is a recurrent neural network and is a significant part of discrete-time nonlinear systems [34]. Unlike other recurrent neural networks (RNN), NARX has a limited feedback which comes only from the output neuron and not from any hidden neurons. And thus these networks are proved to be universal computation devices such as Turing machines [35]. A schematic representation of a basic NARX model is shown in Figure 2.

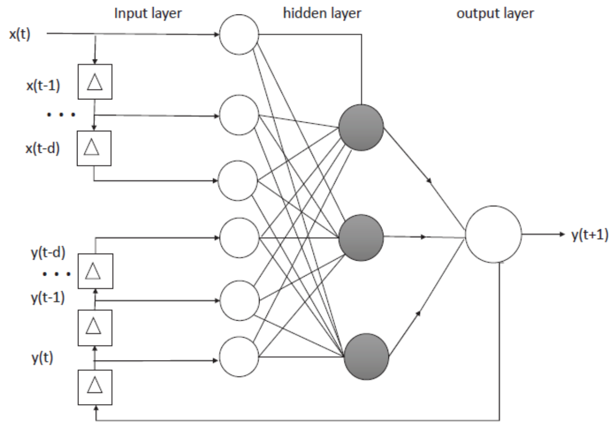

Fig 2: Schematic representation of NARX Model

NARX models works well for catalytic reforming systems in a petroleum refinery, heat exchangers, waste water treatment plants, nonlinear oscillations associated with multi-legged locomotion in biological systems and many such artificial nonlinear systems which can be found in [36], [37], [38], [39], [40], and, [41]. NARX models, compared to other recurrent neural networks converge faster, generalize much better and works without any computational loss [35]. NARX models are most effective and productive when combined with optimization algorithms such as gradient-descent learning and Levenberg Marquardt (LM).

IV. PROPOSED WORK

Levenberg and Marquardt developed a very classic algorithm that is used for solving non-linear problems [42]. It is a robust algorithm basically used for curve fitting problems but is able to find only the local minimum. Prediction refers to dynamic filtering, in which future values are predicted by feeding back the past values of one or more time series. The NARX network has a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. In open loop NARX architecture the actual output is available during the training of the network so the input to the feedforward network will be more precise. We have used NARX neural network along with Levenberg Marquardt optimization algorithm for training, testing, and validation to obtain the final model for rainfall prediction. The goal of our work is to develop the most appropriate model to predict the coming year rainfall of Khordha district of Odisha.

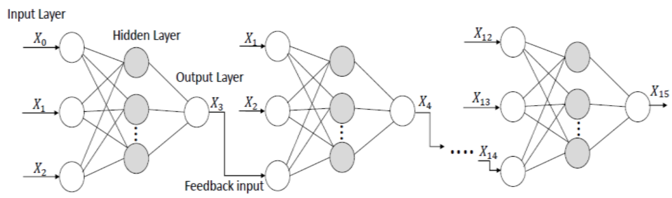

Fig 3: NARX model with 3 inputs and 1 feedback

Table 1: Sample Input and Target Rainfall Data pattern

Input			Target
x0	x1	x2	x3
x1	x2	x3	x4
x2	x3	x4	x5
x3	x4	x5	x6
x4	x5	x6	x7
x5	x6	x7	x8
x6	x7	x8	x9
x7	x8	x9	x10
x8	x9	x10	x11
x9	x10	x11	x12
x10	x11	x12	x13
x11	x12	x13	x14
x12	x13	x14	x15

Block-wise past record of rainfall data of Khordha district from the year 2000 to 2015 are available in the online site of the Government of Odisha namely "Odisha Rainfall Monitoring System" [43]. We have collected monthly rainfall data from this site for our proposed work. The available monthly rainfall data were summed up to get the rainfall data for a particular year. The input rainfall data is in multivariate form with each row having three consecutive years' rainfall data and the rainfall data of the very next year is its respective target. The values of the input data and target are normalized using a very popular normalization technique called min-max normalization. The sample input and output pattern for NARX model is shown in Table 1 and the model that corresponds to our work is displayed in Figure 3. Now the next input to the NARX model is the last two consecutive years of previous input along with the previous target as seen in Table 1. The amount of data used for training is 70% and for validation and testing is 15% each. Levenberg Marquardt algorithm is applied for training the network and Mean Square Error (MSE) is obtained as performance of the model. This MSE is the difference between the actual output and predicted output. Lesser is the MSE, the more accurate is our model. So, with the purpose of fetching the best model, a series of hit and try is done by varying number of hidden nodes from 3 to 6 and delay from 1 to 3 for each hidden node. The average MSE is calculated by considering 40 iterations for each combination of the number of hidden nodes and the number of delays in the NARX model. The NARX model for which the minimum MSE value is found has been chosen as the best model for our proposed work. A comparison of all the average MSE values in terms of graphs are shown through Figures 4 to 15 with MSE in Y axis and number of trainings in X-axis. The solid black colored circles in the respective graphs show the minimum of all 40 runs and the respective values are shown in Tables 2 to 5 for 3, 4, 5 and 6 hidden nodes respectively. After analyzing the graphs, it was observed that the minimum average MSE is achieved when, hidden nodes and delay are fixed to 3 each and thus this is the best model and further prediction of rainfall of next year can be done using this model.

V. RESULTS AND DISCUSSION

In this section the results of the experiments performed are discussed along with the properties of the final model deduced. To deal with the non-linear nature of time series data, ANN techniques are encouraged. The network opted in this paper is NARX neural network along with a well-known optimization algorithm Levenberg-Marquardt algorithm (LMA). With the motivation to develop the best possible model for prediction of rainfall of Khordha district of odisha a series of experiments are done. The main criteria to deduce the best model is based on minimum MSE which is also the performance of the network. The model showing the minimum MSE value is declared as the best model to predict the rainfall. A total of 16 years rainfall data from 2000 to 2015 of Khordha district of Odisha is taken. It is presented in a recursive format which is well described in Table 1. The next input to the model is the rainfall data of previous 2 years along with the previous target. For better handling of these data, they are normalized between -1 to 1. After normalizing the input and target data, a series of trials is performed by varying the number of hidden nodes and delays. The results of the experiments performed are shown in Tables 2 to 5. Each cell in the table shows the MSE value obtained after each time the rainfall data is trained. The minimum of all the average MSE calculated of each column is considered as the best model. Our results show that the model trained with 3 hidden nodes and 3 delays has the minimum MSE value. The graphs describing our experiments are shown in Figures 4 to 15. The black coloured circle in the graphs shows the minimum MSE value of all 40 runs in each case.

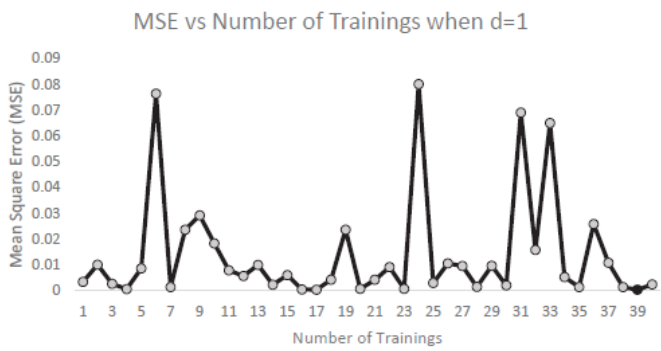

Fig 4: MSE graph with 3 hidden nodes and delay = 1

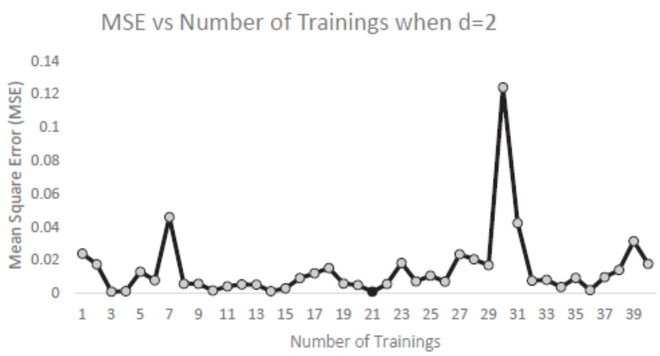

Fig 5: MSE graph with 3 hidden nodes and delay = 2

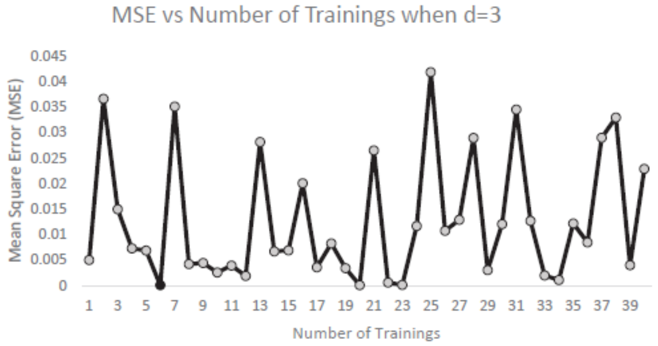

Fig 6: MSE graph with 3 hidden nodes and delay = 3

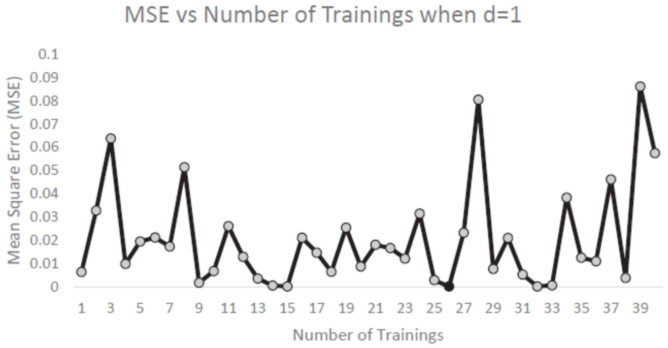

Fig 7: MSE graph with 4 hidden nodes and delay = 1

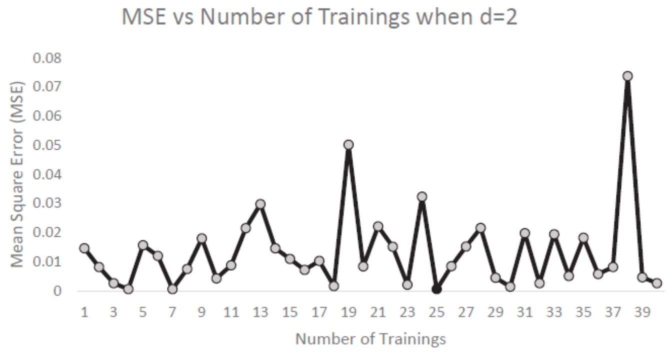

Fig 8: MSE graph with 4 hidden nodes and delay = 2

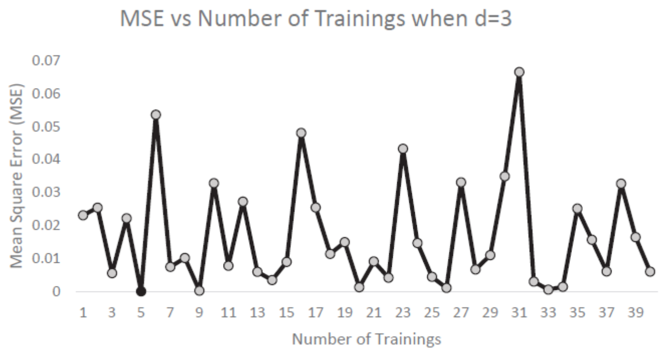

Fig 9: MSE graph with 4 hidden nodes and delay = 3

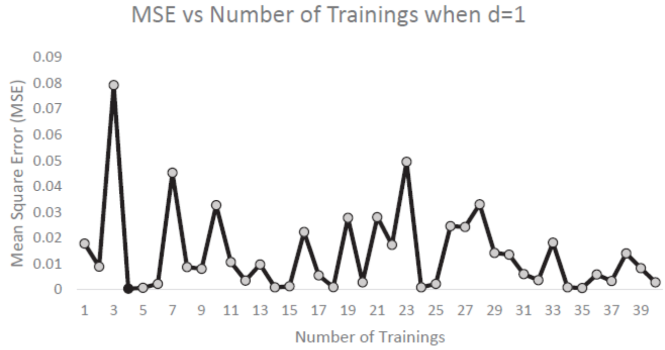

Fig 10: MSE graph with 5 hidden nodes and delay = 1

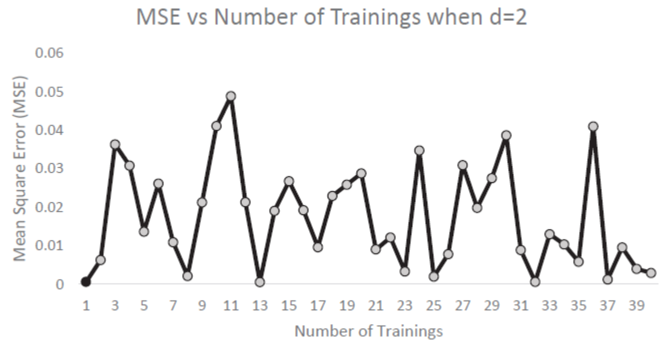

Fig 11: MSE graph with 5 hidden nodes and delay = 2

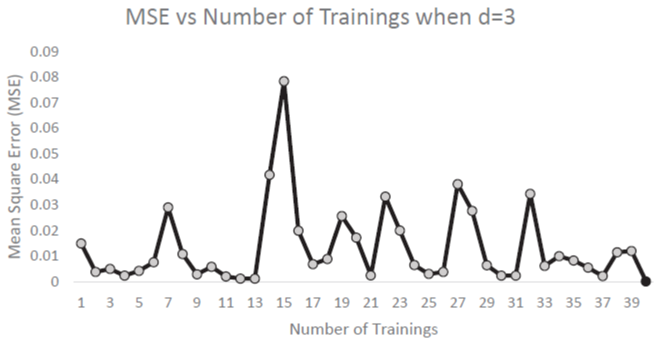

Fig 12: MSE graph with 5 hidden nodes and delay = 3

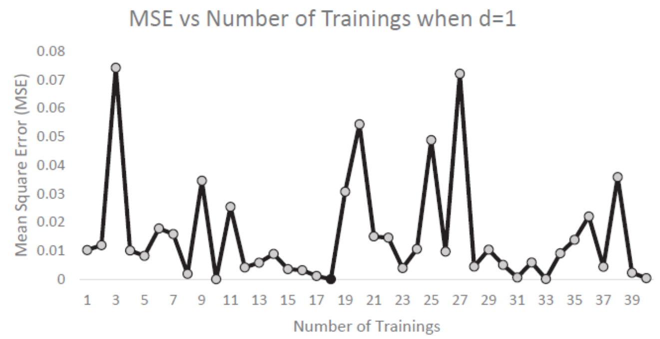

Fig 13: MSE graph with 6 hidden nodes and delay = 1

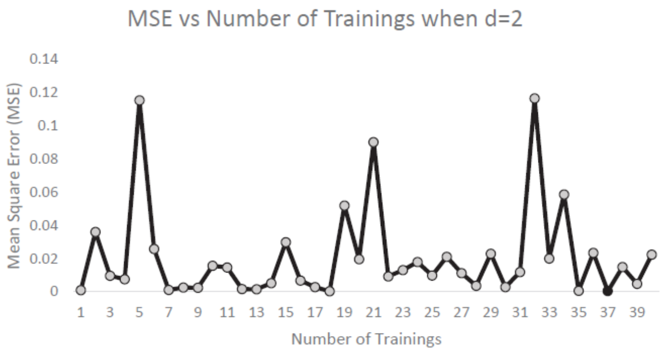

Fig 14: MSE graph with 6 hidden nodes and delay = 2

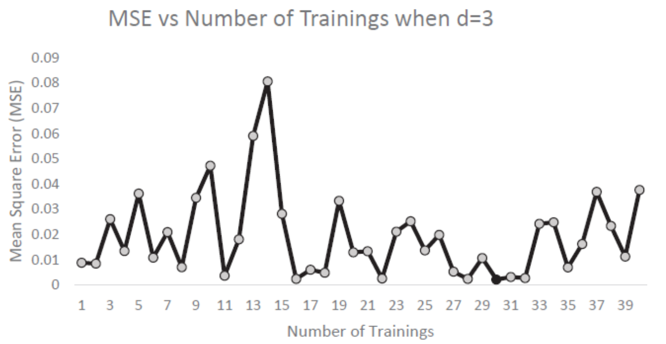

Fig 15: MSE graph with 6 hidden nodes and delay = 3

All the experiments were performed using Microsoft Windows10 64-bit operating system. The computing system was empowered with 8GB RAM and Intel(R) core(TM) i7-6500U CPU @2.50 GHz processor. The experiments were simulated using MatlabR2017b and the resulting MSE graphs were generated using MS Excel-2016.

Table 2: MSE values for 3 hidden nodes with delay =1,2,3

Delay=1	Delay=2	Delay=3
0.003380303	0.024041814	0.004998357
0.010036588	0.017569552	0.036656487
0.002622958	0.001151006	0.01495898
0.000608	0.001465933	0.007281896
0.008611907	0.013096457	0.0068622
0.07643901	0.008041468	0.000032
0.001365262	0.046007604	0.035145173
0.02361552	0.00589522	0.004220374
0.029133298	0.005911636	0.004398683
0.018267701	0.001789648	0.002569638
0.00780828	0.004261281	0.003926873
0.005621464	0.005516552	0.001869498
0.00991381	0.005327974	0.028156657
0.002228973	0.001425044	0.006709474
0.005968702	0.003188888	0.006915269
0.000437	0.009303913	0.020080673
0.000323	0.012137016	0.003580442
0.004178718	0.015306227	0.008249133
0.023643412	0.00601327	0.00339083
0.000647	0.005131476	0.0000521
0.004123312	0.001030564	0.026507367
0.009083646	0.005692441	0.000584
0.000657	0.018319934	0.000129
0.08014324	0.007284388	0.011667759
0.002870837	0.010722476	0.04187344
0.010554147	0.007244458	0.010739638
0.009504891	0.02359915	0.012908669
0.001432035	0.020566874	0.028960466
0.009556278	0.017046112	0.003012843
0.001976247	0.124139434	0.012060616
0.069093975	0.042375484	0.034545157
0.015733985	0.007768263	0.012670596
0.065003956	0.00828528	0.001988388
0.005244431	0.003855802	0.001078421
0.001356773	0.009431603	0.012179341
0.025793596	0.002108821	0.008471398
0.010816367	0.009821731	0.029006207
0.001306503	0.014205913	0.032972496
0.000272	0.031574223	0.004012101

Table 3: MSE values for 4 hidden nodes with delay =1,2,3

Delay=1	Delay=2	Delay=3
0.006587916	0.014569074	0.023068642
0.032811743	0.008136335	0.025356446
0.06378129	0.002639058	0.005569899
0.010015411	0.000593	0.022134792
0.019643492	0.01566359	0.00000375
0.021315383	0.011969622	0.053547545
0.01745598	0.000603	0.007462284
0.051412853	0.007447802	0.010145692
0.001965791	0.017956139	0.000241
0.006908574	0.004289543	0.032806704
0.026145278	0.00878752	0.007768695
0.013062856	0.021501863	0.027243382
0.003772677	0.029733731	0.005967687
0.000703	0.014569074	0.003444496
0.000345	0.010909138	0.00898926
0.021221968	0.007191437	0.04801767
0.014799499	0.010219046	0.025423217
0.006688928	0.001664909	0.011390572
0.025541515	0.050153086	0.014942624
0.008857069	0.008399756	0.001300388
0.018128369	0.022133286	0.009063733
0.016828525	0.015122778	0.004208237
0.012264337	0.002081244	0.043207975
0.031582124	0.032303599	0.014680471
0.003014695	0.000507	0.004404051
0.000337	0.00842038	0.001105342
0.023358063	0.015230934	0.033087913
0.080370502	0.021592534	0.006653219
0.00782375	0.004442397	0.010976162
0.021075087	0.001387793	0.034926604
0.005394538	0.019789389	0.06646083
0.00034	0.002591793	0.002970504
0.000847	0.019374262	0.000563
0.038427156	0.005053121	0.001437739
0.012674251	0.018186	0.025141932
0.011133069	0.005787216	0.01565199
0.046152695	0.008160035	0.006113239
0.003990897	0.073737412	0.032670072
0.086021638	0.004605864	0.016457835
0.057421473	0.002579185	0.006045099

Table 4: MSE values for 5 hidden nodes with delay =1,2,3

Delay=1	Delay=2	Delay=3
0.01789926	0.000563	0.014913674
0.008925113	0.006284865	0.003777564
0.07921627	0.036359205	0.004918987
0.000335	0.030815743	0.002300422
0.000769	0.01368231	0.004184838
0.002230754	0.026159205	0.007557575
0.045286371	0.01091951	0.028972417
0.008746174	0.002148636	0.010759007
0.008109608	0.021285698	0.002862025
0.032735732	0.041096921	0.005822577
0.010692593	0.048886883	0.002040174
0.003478716	0.021305494	0.00119849
0.009752421	0.000582	0.001150674
0.000915	0.019048068	0.041705683
0.001366297	0.026758097	0.078332408
0.022311414	0.019242135	0.019920714
0.005547426	0.009621128	0.006847388
0.000979	0.022961045	0.008846088
0.027819405	0.025850177	0.025587271
0.00282567	0.028814919	0.01722701
0.028044658	0.009019379	0.002412408
0.017371296	0.012145591	0.033180999
0.049471255	0.003341542	0.019946626
0.000866	0.03474778	0.006513215
0.002239246	0.001951018	0.002960026
0.024612232	0.007840064	0.003795056
0.02426925	0.030941951	0.038126925
0.033076151	0.019839627	0.027644562
0.014240128	0.027558608	0.006389997
0.013592646	0.038695786	0.002315235
0.006085468	0.008889971	0.002322754
0.00365002	0.000616	0.034298721
0.018165585	0.013	0.00613499
0.000926	0.010342223	0.009899828
0.000615	0.005854947	0.008241375
0.005882652	0.041004399	0.005469691
0.003383387	0.001235102	0.002162342
0.014085333	0.009567685	0.011443294
0.008332378	0.003992372	0.011935616
0.002807766	0.002942325	0.0000464

Table 5: MSE values for 6 hidden nodes with delay =1,2,3

Delay=1	Delay=2	Delay=3
0.01028343	0.000918	0.008754992
0.01197615	0.035931809	0.008408849
0.074191334	0.009613062	0.026058979
0.010099984	0.007445341	0.013355196
0.008263876	0.115242524	0.036177182
0.017806712	0.025661505	0.010734137
0.015906479	0.001021402	0.020883824
0.001922099	0.002473151	0.006962625
0.034540538	0.002367004	0.034468256
0.000077	0.015564576	0.047262859
0.025381582	0.014564742	0.003606055
0.004156961	0.001551336	0.018028745
0.005854501	0.00142961	0.059016682
0.00892779	0.005144308	0.080612021
0.003607567	0.029789495	0.028145446
0.003237456	0.006740289	0.002357581
0.001163174	0.002717646	0.006004097
0.0000361	0.000301	0.004759907
0.03071031	0.051890329	0.033313018
0.054353698	0.01952399	0.012892154
0.015004538	0.089990388	0.013266321
0.014618578	0.009212825	0.002497398
0.003989886	0.013028393	0.021048146
0.010606007	0.017894605	0.025236661
0.048792078	0.009666382	0.013644557
0.009695592	0.02105437	0.019784945
0.072071625	0.01109304	0.005221061
0.004485068	0.003468816	0.00231929
0.010419719	0.022837897	0.010571081
0.005064265	0.002763192	0.002070613
0.000625	0.011778376	0.003084721
0.005882543	0.116451941	0.002646935
0.000112	0.019930628	0.024199555
0.00912758	0.058558044	0.024798683
0.01382676	0.000514	0.006905301
0.022057191	0.023389303	0.016148492
0.004399436	0.000277	0.036815343
0.035851997	0.014863546	0.023335026
0.002325272	0.004670199	0.011155987
0.000422	0.022281245	0.03764894

VI. CONCLUSION

The proposed method with the respective experimental set up was able to do a rainfall prediction for Khordha district in Odisha of India with a minimum MSE value of 0.000032. Our proposed method used an NARX neural network to predict the rainfall of Khordha district of Odisha. A number of experiments were carried out to reduce the MSE value and achieve the best prediction model. It was observed that the model having a single layer of 3 hidden nodes with delays equal to 3 is evaluating the least MSE value. Hence, this model was used for accurate rainfall prediction of the given data set. Since ANN models tend to be trapped in local minimum problem, in future a better prediction model can be developed by using Support Vector Machines (SVM) as they are able to find the global minimum.

VII. REFERENCES

[1] Tao Chen And Mikio Takagi. Rainfall Prediction Of Geostationary Meteorological Satellite Images Using Artificial Neural Network in Geoscience And Remote Sensing Symposium, 1993.

[2] Igarss’93. Better Understanding Of Earth Environment., International, Pages 1247–1249. Ieee, 1993.

[3] Ming Zhang And Ra Scofield. Artificial Neural Network Techniques For Estimating Heavy Convective Rainfall And Recognizing Cloud Mergers From Satellite Data. Remote Sensing,15(16):3241–3261, 1994.

[4] C Venkatesan, Sd Raskar, Ss Tambe, B Dandkulkarni, And Rn Keshavamurty. Prediction Of All India Summer Monsoon Rainfall Using Error Backpropagation Neural Networks. Meteorology And Atmospheric Physics, 62(3-4):225–240, 1997.

[5] Sunyoung Lee, Sungzoon Cho, And Patrick Mwong. Rainfall Prediction Using Artificial Neural Networks. Journal Of Geographic Information And Decision Analysis, 2(2):233–242,1998.

[6] E Toth, A Brath, And A Montanari. Comparison Of Short-Term Rainfall Prediction Models For Real-Time Flood Forecasting. Journal Of Hydrology,239(1-4):132–147, 2000.

[7] Kin C Luk, James E Ball, And Ashish Sharma. An Application Of Artificial Neural Networks For Rainfall Forecasting. Mathematical And Computer Modelling, 33(6-7):683–693, 2001.

[8] Ajith Abraham, Dan Steinberg, And Ninansajeeth Philip. Rainfall Forecasting Using Soft Computing Models And Multivariate Adaptive Regression Splines. Ieee Smc Transactions, Special Issue On Fusion Of Soft Computing And Hard Computing In Industrial Applications,1:1–6, 2001.

[9] Kok Wai Wong, Patrick M Wong, Tam´As Dgedeon, And Chun Che Fung. Rainfall Prediction Model Using Soft Computing Technique. Soft Computing, 7(6):434–438, 2003.

[10] Surajit Chattopadhyay And Manojit Chattopadhyay. A Soft Computing Technique In Rainfall Forecasting. Arxiv Preprint Nlin/0703042,2007.
[11] Cl Wu, Kw Chau, And C Fan. Prediction Of Rainfall Time Series Using Modular Artificial Neural Networks Coupled With Data Pre-Processing Techniques. Journal Of Hydrology, 389(12):146–167, 2010.
[12] Pt Nastos, Kp Moustris, Ik Larissi, And Ag Paliatsos. Rain Intensity Forecast Using Artificial Neural Networks In Athens, Greece. Atmospheric Research, 119:153–160, 2013.
[13] Amr H El-Shafie, A El-Shafie, Hasan G El Mazoghi,A Shehata, And Mohd R Taha. Artificial Neural Network Technique For Rainfall Forecasting Applied To Alexandria, Egypt. International Journal Of Physical Sciences, 6(6):1306–1316,2011.
[14] Mehmet Tektas¸. Weather Forecasting Using Anfis And Arima Models. Environmental Research, Engineering And Management,51(1):5–10, 2010.
[15] S Karmakar, Mk Kowar, And P Guhathakurta. Long-Range Monsoon Rainfall Pattern Recognition And Prediction For The Subdivision'epmb'chhattisgar Husing Deterministic And Probabilistic Neural Network. In Advances In Pattern Recognition, 2009. Icapr'09. Seventh International Conference On, Pages367–370. Ieee, 2009.
[16] Enireddy Vamsidhar, Kvsrp Varma,P Sankara Rao, And R Satapati. Prediction Of Rainfall Using Backpropagation Neural Network Model. International Journal On Computer Science And Engineering, 2(4):1119–1121,2010.
[17] G Geetha And R Samuel Selvaraj. Prediction Of Monthly Rainfall In Chennai Using Back Propagation Neural Network Model. International Journal Of Engineering Science And Technology,3(1), 2011.
[18] Ankita Sharma And Geeta Nijhawan. Rainfall Prediction Using Neural Network. Ijcst,3(3):65–69, 2015.
[19] Jos´E Maria P J´Unior And Guilherme A Barreto. Multistep-Ahead Prediction Of Rainfall Precipitation Using The Narx Network. Estsp08, Page 87, 2008.
[20] Mary Anne M Sahagun, Jennifer C Delacruz, And Ramon G Garcia. Wireless Sensor Nodes For Flood Forecasting Using Artificial Neural Network. In Humanoid, Nanotechnology, Information Technology, Communication And Control, Environment And Management(Hnicem), 2017 Ieee 9th International Conference On, Pages 1–6. Ieee, 2017.
[21] Vk Somvanshi, Op Pandey, Pk Agrawal, Nv Kalanker, M Ravi Prakash, And Ramesh Chand. Modeling And Prediction Of Rainfall Using Artificial Neural Network And Arima Techniques. J. Ind. Geophys. Union, 10(2):141–151, 2006.
[22] Junaida Sulaiman And Siti Hajarwahab. Heavy Rainfall Forecasting Model Using Artificial Neural Network For Flood Prone Area. In It Convergence And Security 2017, Pages 68–76. Springer, 2018.
[23] Aman Mohammad Kalteh. Enhanced Monthly Precipitation Forecasting Using Artificial Neural Network And Singular Spectrum Analysis Conjunction Models. Inae Letters, 2(3):73–81,2017.
[24] Zahra Beheshti, Morteza Firouzi, Siti Mariyamshamsuddin, Masoumeh Zibarzani, And Zulkifliyusop. A New Rainfall Forecasting Model Using The Capso Algorithm And An Artificial Neural Network. Neural Computing And Applications,27(8):2551–2565, 2016.
[25] R Sukanya And K Prabha. Comparative Analysis For Prediction Of Rainfall Using Data Mining Techniques With Artificial Neural Network. Volume-5, Issue-6, Page, (288-292), 2017.
[26] Madhusmita Patra, Swarnamayee Tripathy, And Indramani Jena. Health Hazards By Sea Cyclones In Odisha, The Super Cyclone And The Ph Ail In. Odisha Review, 30, 2013.
[27] Ashu Jain And Avadhnam Madhav Kumar. Hybrid Neural Network Models For Hydrologic Timeseries Forecasting. Applied Soft Computing,7(2):585–592, 2007.
[28] John J Hopfield. Neural Networks And Physical Systems With Emergent Collective Computational Abilities. Proceedings Of The National Academy Of Sciences, 79(8):2554–2558, 1982.
[29] Paul Werbos. Beyond Regression: New Tools For Prediction And Analysis In The Behaviour Science. Unpublished Doctoral Dissertation, Harvard University, 1974.
[30] D Rumelhart. Mcclelland.(Eds.) Parallel Distributed Processing: Explorations In The Microstructure Of Cognition, 1986.
[31] Ja Anderson. Rosenfeld: Neurocomputing: Foundation Of Research, 1988.
[32] Anil K Jain, Jianchang Mao, And K Moidin Mohiuddin. Artificial Neural Networks: A Tutorial. Computer, 29(3):31–44, 1996.
[33] G Peter Zhang And Min Qi. Neural Network Forecasting For Seasonal And Trend Time Series. European Journal Of Operational Research,160(2):501–514, 2005.
[34] Ij Leontaritis And Stephen A Billings. Input Output Parametric Models For Non-Linear Systems Part I: Deterministic Non-Linear Systems. International Journal Of Control, 41(2):303–328, 1985.
[35] Hava T Siegelmann, Bill G Horne, And C Leegiles. Computational Capabilities Of Recurrent Narx Neural Networks. Ieee Transactions On Systems, Man, And Cybernetics, Part B (Cybernetics),27(2):208–215, 1997.
[36] Sheng Chen, Sa Billings, And Pm Grant. Non-Linear System Identification Using Neural Networks. International Journal Of Control,51(6):1191–1214, 1990.
[37] Kumpati S Narendra And Kannan Parthasarathy. Identification And Control Of Dynamical Systems Using Neural Networks. Ieee Transactions On Neural Networks,1(1):4–27, 1990.
[38] Hong-Te Su And Thomas J Mcavoy. Identification Of Chemical Processes Using Recurrent Networks. In

American Control Conference, 1991, Pages 2314–2319. Ieee, 1991.

[39] Hong Te Su, Thomas J Mcavoy, And Paulwerbos. Long-Term Predictions Of Chemical Processes Using Recurrent Neural Networks: A Parallel Training Approach. Industrial & Engineering Chemistry Research, 31(5):1338–1352,1992.

[40] St Venkataraman. On Encoding Nonlinear Oscillations In Neural Networks For Locomotion. In Proceedings Of The Eighth Yale Workshop On Adaptive And Learning Systems, Pages 14–20,1994.

[41] S-Z Qin, H-T Su, And Thomas J Mcavoy. Comparison Of Four Neural Net Learning Methods For Dynamic System Identification. Ieee Transaction Son Neural Networks, 3(1):122–130, 1992.

[42] Jorge J Mor´E. The Levenberg-Marquardt Algorithm: Implementation And Theory. In Numerical Analysis, Pages 105–116. Springer, 1978.

[43] http://rainfall/login.asp.

Prakhar Golchha has been doing his M.Tech. in Computer Science and Engineering at IIIT Bhubaneswar. He is expected to get his degree in June 2018. He has completed his B.E. in Computer Science and Engineering from Chhattisgarh Swami Vivekanand Technical University, India in 2015. His research interests include Neural Networks and Adhoc Network.

Palak Lunia has been doing her M.Tech. in Computer Science and Engineering at IIIT Bhubaneswar. She is expected to get her degree in June 2018. She has completed her B.E. in Computer Science and Engineering from Chhattisgarh Swami Vivekanand Technical University, India in 2015. Her research interests include Neural Networks and Parallel Computing.

Muktikanta Sahu is working as an Assistant Professor in the Department of Computer Science and Engineering at IIIT Bhubaneswar. He has completed his B.E. and M.Tech. in Computer Science and Engineering from Biju Pattanaik University of Technology, Odisha in 2003 and 2007 respectively. His research interests include areas of Machine Learning, Parallel, and Distributed Computing. His current research projects are on Parallel Computing and Process Mining. He has published his past research work in IEEE Transaction on Neural Networks.