IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 8 Issue 3, March 2021
ISSN (Online) 2348 – 7968 | Impact Factor (2020) – 6.72
www.ijiset.com

Forging A Scalable Spectral-Clustering Multi-Agent Hybrid Deep Learning Model To Predict Rainfall Runoff In Nigeria

Arnold Adimabua Ojugo[1], Obinna Nwankwo[2]

[1]Department of Computer Science, Federal University of Petroleum Resources Effurun, Delta State, Nigeria

ojugo.arnold@fupre.edu.ng, arnoldojugo@yahoo.com, arnoldojugo@gmail.com

[2]Department of Computer Science, Novena University Ogume, Delta State, Nigeria.

tuk2obinna@gmail.com, obinna.nwankwo@novena.edu.ng

Corresponding Author:

Arnold Adimabua Ojugo (PhD),
Department of Mathematics/Computer Science, Federal University of Petroleum Resources Effurun,
P.M.B 1221, Effurun, Warri, Delta State, Nigeria.
Email: ojugo.arnold@fupre.edu.ng, arnoldojugo@gmail.com, maryarnoldojugo@gmail.com

ABSTRACT

Study seeks to adopt a spatial-temporal deep learning model on dataset retrieved from Metrological center at Oshodi in Lagos State, Nigeria. Selected area is Benin-Owena River Basin for the period 1999 - 2019. Dataset split into: training (45%), retraining (25%) and testing (30%). Study advances a profile hidden Markov deep neural network, which was found to have a classification error of 1.09% – when compared to other models such as profile hidden Markov, deep neural network and memetic algorithm with 10.2%, 19.7% and 3.27% respectively. It is observed also that other models underperformed against the proposed model. Study agrees with Ojugo et al 2013, in that annual rainfall is an effect of variation cycle. Models will help simulate future flood and provide, lead time warning in flood management.

1. INTRODUCTION (10 PT)

Rainfall runoff predictions have since become a critical feat in Nigeria – especially with the tsunami and deluge of 2015 through 2018 that saw a general displacement of citizens worldwide and Nigeria – in particular. Its prediction also, is a critical component to planning, coordination and execution of farming directives. Such prediction are made possible via the employment of various mathematical models grouped into: knowledge and data driven models [1-3]. Rainfall forecasts have always employed and focuses on runoff quantification. Its increasing awareness as well as the dynamic nature of environmental issues – have given additional impetus to model in hydrology. These new models must meet – rising new requirements and challenges to deal with associated tasks such as erosion, land degradation, pollutant leaching, sustainable flood resource management, land-use possible consequence and climate changes – to mention a few [4-5].

Rainfall has significant influence in flooding and downstream hydrology with a range of implications as well as complications of erosion, water quality, and the design of engineering structures – which in turn impacts on the quality of life, agriculture, sewage system, and tourism etc [6-7]. For these amongst many other reasons, early warning of rainfall runoff situations is critical in the management of water resources [8-10]. The complex and chaotic nature of atmospheric processes that produce rainfall makes rainfall runoff modeling as well as its prediction – a tedious cum difficult task [2, 5, 11-13]. Thus, [14] noted that in spite of advances in weather forecasting, accurate rain forecasting is the most challenging in operational hydrology.

1.1. Literature Review

Scientists world over – have continued in their quest to develop stochastic (knowledge-driven) models in a bid to enhance accurate rainfall prediction. [14] used ANN to predict and investigate rainfall runoff in India; while, [13] did same using ANN procedure to predict rainfall in Cyprus. Recently, efforts are intensified to use time series autoregressive, moving average (ARMA) model with an exogenous variable denoted (ARMAX) to model hydrological data. The use of ARMA models is justified as a result of its theoretical base in hydrological studies. [9] used the three-function parameter value of AR (3) model which showed significant association with rainfall was established for relative humidity, cloud cover and temperature difference. Their study noted that sunshine was dropped as a possible predictor using impulse response functions; while, the four TF models were identified as TF (3, 2, 2, 2) model, which predicted rainfall with a root mean square percentage error of 2.3%, was adopted as the most appropriate model. The model also performed better than both multiple regression and univariate SARIMA $(1, 0, 1)^*(1, 0, 1)_{12}$ models.

Also, [2] compared a hybrid gravitational search algorithm trained neural network with historical data for the Chad River Basin in Nigeria using dataset from 1996 - 2007. Results showed high degree of accuracy with computed COE as 58, 24, 56 and 42% respectively for the various stations. Observed annual rainfall variations from long-term runoff, is an effect of variation cycle with significant correlation between rainfall and runoff (as indicative in the dataset used). The study implementation will create a synergy between Artificial Intelligence and other fields – which in this case, hydrology via the hybrid ANN models – so that the trained system will help simulate future flood and provide, lead time warning in flood management. 5] developed a GARCH model to forecast rainfall using historical data from National Metrological Centre at Oshodi for the period of 1996–2007 on rainfall, temperature difference, relative humidity, sunshine and cloud cover. Study established significant association with rainfall for humidity, temperature difference and cloud cover. Using impulse response functions GARCH(1,0,1) model was adopted for predicting rainfall with root mean square error of 2.3% as the most appropriate. When compared, we agree that model performs better than both multiple regression and univariate SARIMA (1,0,1)*(1,0,1) models.

1.2. Motivation / Statement of Problem

Our study is motivated thus [15-18]:

1. Efforts geared towards hydrological models are still faced with the fundamental problem of calibration and validation due to limited data availability and natural heterogeneity of rainfall runoff process – leading to the consequent rise in data-driven model, which are poised on the focal need to learn feats in time that are somewhat not possible via knowledge-driven models.
2. Also, many problems are related to model testing such that traditional tests like split-sample are often insufficient to evaluate a model's validity and assess its pros/cons of the different model approaches. The need for additional data has been emphasized, advent of more powerful tests required and different dimensionality of model adopted via data driven models that employ evolutionary method to yield such dimensionality.
3. The formulation of an optimization task problem requires careful planning and selection of a few design variables as possible – whose outcome procedure indicates whether or not to include more variables in a revised formulation and/or to replace some previously considered design variables with new design variables. Optimization tasks are quite intricate to selected feats, variables, constraints and parameters for a multi-objective function that ultimately yields an optimum solution. These integral components vary greatly with each problem domain.

To overcome these shortfalls, we adopt a deep learning profile hidden Markov modular neural network design for rainfall runoff prediction using the Benin-Owena River Basin Dataset retrieved from the National Metrological Centre at Oshodi in Lagos State Nigeria.

2. MATERIALS AND METHODS

2.1. Data Gathering

The selected area is the Benin-Owena River Basin Development Agency (BORDA) Nigeria with land mass of $22045km^2$, mean annual rain of 1023mm and perennial discharge of $3.8m/1.5m^3/s$ for dry and peak periods. Time plot for data collected within the period 1999-2019 is seen in figure 1, which yields Table 1. The dataset collected for the period (1999 – 2019) is split into 3-sets: ***training*** (45%), ***cross-validation*** (25%) and ***validation*** (30%). All fragment starts at period of constant low rainfall. Figure 2 shows the project location.

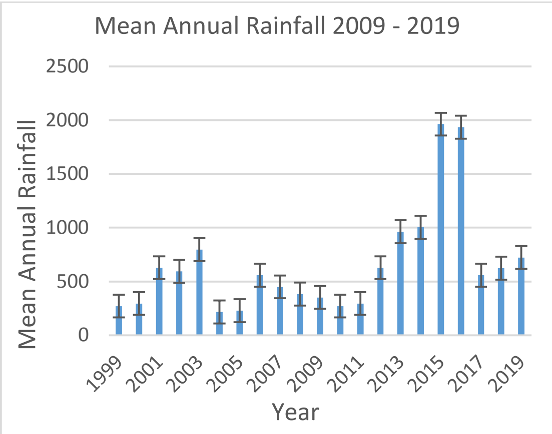

Figure 1. Graph of Mean Annual Rainfall

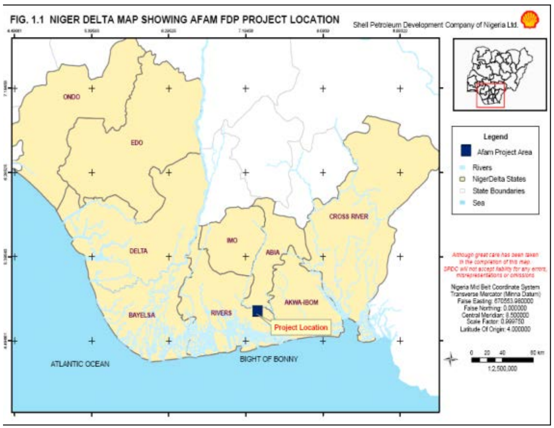

Figure 2. Map of Nigeria with Project Location

Table 1. Detailed Summary Sheet of Rainfall Features for 1999 – 2019

Year	Rainfall in mm	Temperature Differences		Mean Humidity	Mean Sunshine in Hours	Mean WindSpeed in mtrs/sec	Wind Direction
		TMax	TMin				
1999	271.4	31.567	22.909	78.901	3.256	2.902	SW
2000	295.1	32.092	23.405	76.902	3.761	3.508	S
2001	628.9	31.533	23.508	83.000	3.021	2.892	W
2002	594.4	32.017	23.817	85.200	2.994	2.858	SW
2003	795.7	31.575	23.703	83.134	5.012	2.917	W
2004	216.4	31.733	24.442	79.013	4.561	3.375	SW
2005	229.4	31.567	23.468	85.301	4.092	2.935	SW
2006	558.8	32.092	24.501	79.34	4.432	3.451	SW
2007	449.6	31.917	23.908	81.211	3.895	3.209	S
2008	383.4	32.042	24.091	83.120	4.501	3.021	S
2009	351.7	31.575	23.508	83.753	4.458	3.508	NE
2010	271.4	31.733	23.717	83.917	5.067	2.892	W
2011	295.1	31.567	23.700	83.751	4.433	2.858	SW
2012	628.0	32.092	24.042	83.667	3.850	2.917	S
2013	963.0	31.533	23.458	83.667	4.042	3.375	SW
2014	1005.0	32.017	23.183	83.583	3.883	3.733	SW
2015	1963.1	31.458	23.617	81.501	2.933	3.3	S
2016	1934.1	32.142	23.842	84.751	4.358	3.058	SW
2017	558.8	31.917	23.317	85.167	4.001	2.825	S
2018	623.9	32.042	24.825	83.001	4.158	2.983	S
2019	723.1	31.558	23.483	81.333	4.575	3.15	W

2.2. Hybrid Reinforcement Learning Ensemble

The hybrid reinforcement deep learning ensemble is as thus:

2.2.1. The Supervised Profile Hidden Markov Model

The profile HMM as a variant, avoids pitfalls in traditional HMM as it: (a) makes use of positional data contained in the observations, and (b) allows null transitions so that the model can match sequences that includes insert/delete [19]. For rainfall runoff, O is a candidate rule, T is for time, N is the number of rules, α is alphabet of model, M is the number of symbols in alphabet, π is initial state of transactions, A is state transition probability matrix, a_{ij} is probability of a rule from state i to j, B contains N-probability distributions for rules in knowledgebase; while, $HMM = (A, B, \pi)$ is denoted thus [20-24]. It should be noted that the general idea for parameters in HMM are still intact. The profile hidden Markov consists rules aligned as sequence with significant relations [25-28] as as in figure 3. Output sequence determines if an unknown rules is related to sequences belonging to a class. We use the profile HMM to score rules and make decision [19].

2.2.2. Deep Learning Neural Network Framework (DNN)

DNN uses deep learning to adapt useful select feats of interest as parameters, carefully constructing a multi-layer (MLP) net from vast amount of data. Its deep architecture at its input, hidden and output layers - helps to improve its prediction accuracy [. Its hidden layer transforms non-linearly from a previous layer to the next [28]. Proposed by Hinto et al [29], a DNN is trained via two phases: (a) pre-trained, and (b) fine-tuned processes. The Auto-Encoder is an unsupervised multi-layered neural network consisting both an encoder and a decoder network. Its encoder seeks to transform inputs data-points from a high unto a low-dimension [30] via an encoding function f_{encoder} as in Eq. 1 – where x^m is a data point, and h^m is the encoding vector obtained. Conversely, its decoder network seeks to reconstruct the function using $f_{decoder}$ as in Eq. 2 with x^m as decoding vector from h^m. Thus, reverts the operations of the encoder [31]. Erhan et al [32] in Gilrot and Bengio [33] details specific algorithms for encoding and decoding functions respectively.

$$h^m = f_{encoder}(X^m) \quad (1)$$

$$X^m = f_{decoder}(h^m) \quad (2)$$

At the pre-training phase, *N* auto-encoders can be stacked on to an N-hidden-layer so that with input accepted, the input layer and first hidden layer acts an encoders of the first auto-encoder. They are trained to minimize the reconstruction error. Training parameter(s) of the encoder are used to initialize first hidden layer before proceeding to second hidden layer. There, the first and second hidden layers are selected as encoder(s) and as in the earlier stage, the second hidden layer is initialized by the second trained auto-encoder. This process continues till the *N*th auto-encoder is trained and initializes the final hidden layer. With all hidden layers stacked in the auto-encoder at each training *N*-times, they are thus regarded as pre-trained. This feat has proven to be significantly better than random initialization. It also achieves better generalization [34].

Fine-tuning phase seeks to optimize a DNN's performance by retraining the network labeled training data. It computes errors as difference in real versus predicted values via back-propagated stochastic gradient descent (SGD), randomly selecting data, and iteratively updates gradient direction with the weight parameters. A merit of the SGD is that it converges faster and does not require the entire dataset. This makes it suitable for complex neural networks as given in Eq. 3 with E as loss function, *y* is label and *t* is output of the network [35]:

$$E = \frac{1}{2}\sum_{j=1} M(y_i - t_i)^2 \quad (3)$$

The gradient of the weight *w* is obtained as a derivative of the error equation – so that an updated SGD is given by Eq. 4 with ŋ is step-size, *h* is number of hidden layers [27, 29]:

$$W_{ij}^{new} = W_{ij}^{old} - ŋ.\left(y_j - t_j\right).y_j\left(1 - y_i\right).h_i \quad (4)$$

This process is optimized by the weight threshold of correctly labelled data. Thus, a DNN can learn accurately at its final output and direct thus, task all network parameters to perform correct classifications [30-31].

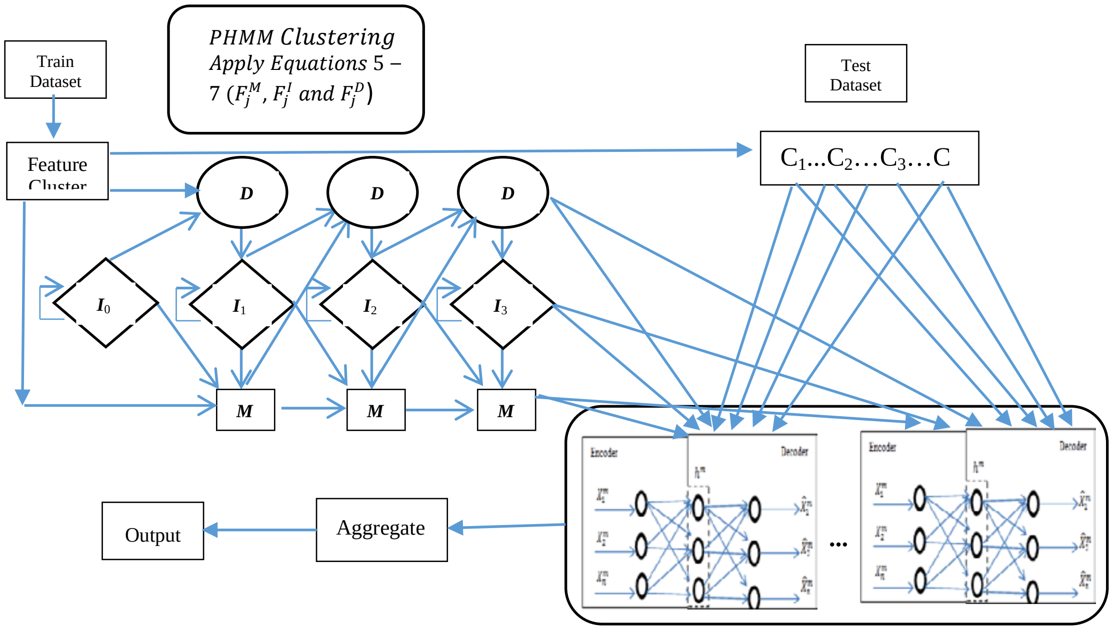

Figure 3. Reinforcement Cluster Deep Learning Neural Network Ensemble

2.3. Proposed Experimental Hybrid Ensemble

We adopt the hybrid ensemble [34] with selected training dataset to form cluster feats to go through the PHMM. With selected cluster features, circles are ***delete*** state for unclassified transaction rules, ***diamonds*** are insert states for gaps in transaction rules to update knowledgebase for false-positives and true-negative errors. The ***rectangles*** are matched states that accurately classifies rules into class types. Match and insert are emission observations as PHMM passes via the states with probabilities corresponding to B in HMM; And, computed from frequency of symbols [35-38] denoting the amount of rules emitted at particular state, which is position-dependent Finally, ***delete*** states allows PHMM to go through the gaps in the network to reach other emission states. These gaps helps model avoid over-fitting. We use the forward algorithm to recursively computes probabilities via reusing scores for partial sequences via Equation 5-7 as thus [39-40]:

$$F_j^M = Log\frac{eM_j(x_i)}{qx_i} + \log(aM_{j-1}M_j \exp\left(F_{j-1}^M(i-1)\right) + aI_{j-1}M_j \exp\left(F_{j-1}^I(i-1)\right) + aD_{j-1}M_j \exp\left(F_{j-1}^D(i-1)\right) \quad (5)$$

$$F_j^I = Log\frac{eI_j(x_i)}{qx_i} + \log(aM_jI_j \exp\left(F_j^M(i-1)\right) + aI_jI_j \exp\left(F_j^I(i-1)\right) + aD_jI_j \quad (6)$$

$$F_j^D = \log(aM_{j-1}D_j \exp\left(F_{j-1}^M(i)\right) + aI_{j-1}D_j \exp\left(F_{j-1}^I(i)\right) + aD_{j-1}D_j \exp\left(F_{j-1}^D(i)\right) \quad (7)$$

After this, classification is handed over to the DNN, which seeks to resolves tasks as thus: (a) divides training data into clusters, computing center point of each cluster, (b) each of the cluster is trained and scaled so that each DNN learns all attributes of each of the subset, (c) test data applies previous cluster centers to detect outlier(s) by the pre-trained DNNs, and (d) output of each DNN is aggregated for the final result data/outliers. Proposed solution is divided into 3-steps namely:

1. **Step 1** divides data into train and test clusters or partitions. DNN stores computed cluster centers, used as initialization center(s) to generate test datasets. Dataset attributes are formatted as data-points for selected parameters, and the data-points in the training dataset are aligned into groups of same class. To improve the performance of the DNN, model revises cluster numbers (to between 2 to 6) and sigma values (i.e. 0.1 to 1.0). Minimum distance from a data point to each cluster center is measured, and a data-point's nearness to a cluster, assigns it to that cluster-class. Training sets generated by clusters are taken up as input to DNNs. For training, the number of DNNs equals number of clusters. DNN has five layers: an input, two-hidden, a softmax and output layer respectively. The hidden layers learn feats from each training subset, and the top layer is a five-dimensional output vector. Each training set generated from the *k*th cluster center is regarded as input data to feed into *k*th DNN respectively. Trained sub-DNN models are marked sub-DNN 1 to *k*.
2. **Step 2** uses test dataset to generate *k*-datasets with previous cluster center obtained from clusters in Step 1. The test sub-dataset are denoted as Test 1 through Test *k*.
3. **Step 3**: *k*-test data subsets are fed into *k* sub-DNNs, and were completed by the k training data subsets in Step 1. Output of each sub-DNN is integrated as final output and employed to analyse positive detection rates.

3. FINDINGS AND RESULT DISCUSSION

3.1. Model Evaluation

For performance evaluation, we compute classification rate and improvement percentages for the proposed model measured on the backdrop of benchmark models in both training and test dataset(s) – denoted by Equations 8 and 9 respectively; While, tables 2 and 3 yields summary of obtained values as thus:

$$Classification\ Rate\ (MR) = \frac{No.of\ Incorrect\ class}{No.of\ Sample\ set} \quad (8)$$

$$Improvement\ Percentage = \frac{MR(A) - MR(B)}{MR(A)}\ x\ 100 \quad (9)$$

Table 2. Summary Sheet For Benchmark Model(s) Versus Proposed Deep Learning

	Classification Errors		Improvement Percentage	
Model	*Training Data*	*Testing Data*	*Training Data*	*Testing Data*
Rule-Based (Pre-processor)	25.6%	21.1%	4.09%	4.79%
PHMM	13.7%	10.2%	7.52%	8.45%
GANN	21.3%	19.7%	6.03%	6.46%
DNN	6.78%	3.27%	7.89%	9.01%
PHMM-DNN	1.29%	1.09%	9.21%	9.83%

Results in table 2 indicates that the hybrid PHMM-DNN outperforms both the rule-based, PHMM, GANN and DNN heuristics. This can be attributed to the fact that PHMM was used as a pre-processor rule support for the hybrid ensemble. The PHMM-DNN has a classification error rate of 1.09% (that is, the model's capability to capture false-positives and true-negatives error rate); while classification error values for PHMM, GANN and DNN stood at 10.2%, 19.7% and 3.27% respectively. However, the various benchmark models of PHMM, GANN and DNN had promising improvement of about 8.45%, 6.46% and 9.01% respectively; while, our proposed model PHMM-DNN has an improvement percentage of 9.83%. Though, it was observed that DNN outperforms PHMM, which in turn outperforms GANN – they all however, underperformed against our experimental model PHMM-DNN. This is in agreement with [41-43].

3.2. Discussion of Findings

Hybrids are often difficult to implement as it generates its own range of complications that are due to the following: (a) encoding the data and its encoding conversion from one heuristics to another, (b) resolving conflicts of the data feats of interest from the underlying probability scores generated for each candidate solution, and lastly, (c) resolving the structural dependencies that are imposed on the heuristics by other features within the dataset that may or may not be contained therein from the outset. Thus, these conflicts must be adequately resolved in other for us to effectively and efficiently harness the benefits therein the model, exploit the numeric data and explore the domain problem space to yield an optimal solution [44-45]. Modelers must select the requisite parameter(s) to avoid model over-fitting. Encoded through the model's structured learning, this will help address issues of statistical dependencies between the various heuristics used, highlight implications of such a multi-agent populated model as well as resolve conflicts in data feats of interest. Thus, as agents create/enforce their own behavioral rules on the dataset, hybridization should be able to curb this.

4. CONCLUSION

Models are useful fictions and representation of reality as their primary value is to serve as educational tools for insight to help us better understand and reflect upon reality. They can also serve as means to compile existing knowledge and be employed as the new language to communicate hypotheses. In model development, the study of its sensitivity analysis will further help modelers to reflect on theories for varying systems. Thus, we require that a detailed model should be reasonable and applicable – on a larger scale. For rainfall runoff, it is used to examine hypotheses about a catchment, and to investigate which parameter input are most crucial to be estimated accurately. We thus have a keen interest on a model's application as a feedback mechanism; rather than its accuracy of the numeric agreement between its predictions and observations. Consequently, hybrid ensembles are valuable in comprehending such RR processes, and may not necessarily be suitable tool for concrete predictions. Only understood and manageable models are fully explored. There must be a balance for complexity and simplicity, which is crucial for studying RR processes. Thus, these recommendations are made:

1. Parameters are a major source of uncertainty. Model should have input ranges as computed via Monte-Carlo Integral methods.
2. Multi-criteria training with adequate datasets can help to reduce parameter uncertainty.
3. Prediction is of limited practical use, without clear data about reliability and accuracy.

REFERENCES (10 PT)

[1] R.E. Yoro., A.A. Ojugo., **An intelligent model to predict relationship of weather conditions for fish farming production yield in Nigeria**, Ame. J. Modeling & Opti., 7(2): pp35-41, doi: 10.12691/jcn-7-2-1, 2019

[2] A.A. Ojugo., J. Emudianughe., R.E. Yoro., E. Okonta., A.O Eboka., **A hybrid neural network gravitational search algorithm for rainfall runoff modeling and simulation in hydrology**, Progress in Intelligence Computing and Applications, 2(1): 22-33, doi: 10.4156/pica.vol2.issue1.2, 2013

[3] N.J. De Vos, T. Rientjes, L. Pfister, **Groundwater levels as indicator in rainfall-runoff modeling using Artificial Neural Networks**, In Proceedings of National Council for Rainfall, Australia, 2005.

[4] E. Gaume, R. Gosset, **Overparameterization, obstacles of artificial neural networks in hydrology?**, J. of Hydrology, Earth System Science, vol.7, pp.693–706, SRef-ID: 1607-7938/hess/2003-7-69, 2003

[5] A.A. Ojugo., R. Abere., A.O. Eboka., M. Yerokun., R. Yoro., C. Onochie., D.A. Oyemade., **Hybrid neural network models for rainfall runoffs: comparative study**, Advances in Sci. & Engineering Res., 1(2): 47-55, 2013

[6] A.A. Ojugo., I.P. Okobah., **Computational solution for modeling rainfall runoff using intelligent stochastic model: a case of Warri in Delta State**, Digital Inno. & Cont. Res. In Sci., Engr. & Tech., 5(4): pp45-58, 2017

[7] K.J. Beven, **How far can we go in distributed hydrological modeling?**, *Hydrological Earth System and Science*, vol. 5, pp.1 – 12, SRef-ID: 1607-7938/hess/2001-5-1, 2001

[8] C. Dawson, R. Wilby, **Comparison of neural networks in river flow forecasting**, *J. of Hydrology and Earth Science*, SRef-ID: 1607-7938/hess/2001-3-529, 2001

[9] D. Eni., E.I. Ibiang., J. Atsu., I.O. Ewona., **Three-input transfer function modeling of rainfall in Calabar, Cross River State of Nigeria,** *J. of Hydrology and Earth Science*, SRef-ID: 1607-7938/hess/2001-3-529, 2013.

[10] M.N. French, W.F. Krajewski, R.R Cuykendall, **Rainfall forecasting in space and time using neural network**. *Journal of Hydrology*, 137, pp1–31, 1992

[11] B. Sivakumar, **Chaos theory in hydrology: important issues and interpretations**. *Journal of Hydrology*, 227, pp1–20, 2000.

[12] B. Men, Z. Xiejing, C. Liang, **Chaotic Analysis on Monthly Precipitation on Hills Region in Middle Sichuan of China**, *Nature and Science*, 2, pp.45-51, 2004

[13] K. Gwangseob, P.B. Ana, **Quantitative flood forecasting using multi-sensor data and neural networks**, *Journal of Hydrology*, 246, pp.45–62, 2001

[14] P. Guhathakurta, **Long-range monsoon rainfall prediction of 2005 for the districts and sub-division Kerala with artificial neural network**. *Current Science*, 90, pp.773-779, 2006

[15] S.A. Kalogirou, C.N. Constantinos, S.C.. Michaelides, C.N. Schizas, **A time series construction of precipitation records using Artificial Neural Networks**. *In Proc. of EUFIT '97*, 2409-2413, September 8-11, 1997

[16] F.S. Chang, Y.C. Chen, **A Counter Propagatio Fuzy-Neural Network Modeling Approach to Real Time Streamflow Prediction**. *Journal of Hydrology*. 245, 153-164, 2001

[17] T. Farahmand, S.W. Fleming, J. Edward, **Detection and visualization of storm hydrograph changes under urbanization: An impulse response approach**. *Elsevier Journal of Environmental Management*, 85, 93–100, 2007

[18] N.Q. Hung, M.S. Babel, S. Weesakul, N.K. Tripathi, **A rainfall forecast model using Artificial Neural Network,** *.Hydrol.& Earth Syst. Science. Discuss,* 5, pp.183–218, 2008

[19] A.A. Ojugo., E. Ben-Iwhiwhu, O.D. Kekeje., M. Yerokun., et al., **Malware propagation on time varying networks: comparative study**, Int. J. Modern Edu. Comp. Sci., 6(8), pp.25-33, doi: 10.5815/ijmecs.2014.08.04, 2014

[20] A.A. Ojugo, D.O. Otakore., **Improved early detection of gestational diabetes via intelligent classification models: a case of Niger Delta**, J. of Computer Sci. & Application, Vol. 6, No. 2, pp. 82-90, doi: 10.12691/jcsa-6-2-5, 2018

[21] M. Al-Qatf., Y. Lasheng et al, **Deep learning approach combining sparse auto-encoder with SVM for network intrusion detection**. *IEEE Access*, Vol. 6: 52843-52856, 2018

[22] Y. Zhang., P. Li., X. Wang., **Intrusion detection for IoT based on improved genetic algorithm and deep belief network**. *IEEE Access*, 7: 31711-31722, 2019

[23] G. Mageswary, M. Karthikeyan., **Intrusion Detection Using Data Mining Techniques**, Int. J. Engineering Sci. Invention, Pp. 2319 -6726, 2018

[24] X.F. Wang, T. Sandholm, *Reinforcement learning to play an optimal Nash Equilibrium, in team Markov games*, Advances in Neural Info. Processing Systems, 1603-1610, 2017

[25] X.F. Wang, M.K Reiter, **Defending against denial of service attacks with puzzle auctions**, *2003 Symposium on Security and Privacy*, 78-92, 2003

[26] A.A. Ojugo., A. Eboka., **Empirical evaluation on comparative study of machine learning in detection of distributed denial of service attack**, J. Appl. Sci. Eng. Tech. & Edu., 2(1): pp18–27, 2020, doi: 10.35877/454RI.asci2192

[27] A.A. Ojugo., O.D. Otakore., **Forging optimized Bayesian network model with selected parameter for detection of Coronavirus in Delta State**, J. App. Sci. Eng. Tech. Edu., 3(1): pp37–45, 2021, doi: 10.35877/454RI.asci2163

[28] A.A. Ojugo., A.O. Eboka., **Modeling the computational solution of market basket associative rule mining approaches using deep neural network**, *Digital Technologies*, 3(1): pp1–8, 2018a, doi: 10.12691/dt-3-1-1, [web]: www.sciepub.com/dt/content/3/1

[29] G. Hinton, L. Deng, D. Yu, G. Dahl, A.R. Mohamed et al., **Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups**. *IEEE Signal Process. Mag*., Vol. 29, pp82–97, 2012.

[30] Y. Bengio., A. Courville., P. Vincent, **Representation Learning: A Review and New Perspectives**. *IEEE Transactions on Pattern Analysis and Machine Intelligence , 35* (8), 1798-1828, 2013

[31] T. Ma, F. Weng, J. Cheng, Y. Yu, X. Chen, **A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks**, Sensors, 16: 1701, doi: 10.3390/s16101701, 2016

[32] D. Erhan, Y. Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P and Bengio, S., **Why does unsupervised pre-training help deep learning?**, Journal of Machine Learning Res., Vol. 11, pp625–660, 2010.

[33] X. Glorot, X and Bengio, Y., **Understanding the difficulty of training deep feedforward neural networks**, *Proc. of 13th Int Conf. on Artificial Intelligence and Statistics*, Sardinia, Italy, 13–15 May 2010; pp. 249–256.

[34] A.A. Ojugo., E. Ekurume., **Towards a more satisfied user framework through a dependable secured hybrid deep learning ensemble for detection of credit-card fraud**, Technical Report 234-560291, pp37–45, 2021

[35] A. Jain, S. Srinivasulu, **Development of effective, efficient rainfall-runoff models using integration of deterministic, real-coded GA and ANN**, Water Resources, 40(4), pp. 23 – 45, 2004

[36] S. Nishimura, T. Kojiri, *Real-time rainfall prediction using neural network and genetic algorithm with weather radar data*, In Proc. 10th. Congress of Asia and Pacific Division of Int. Association for Hydraulic Research, 204-211, 1996

[37] W. Openshaw, **Rainfall runoff processes: workbook for online module**, 2010. Available online at [web]:www.engineering.usu.edu/dtarb/rrp.html, last accessed Feb 2013.

[38] M.P. Rajurkar, U. Kothyari, U. Chaube, **Modeling of daily rainfall-runoff relationship with artificial neural network**", Journal of Hydrology, Vol. 28, 96–113, 2004

[39] P. Reggiani, T. Rientjes, **Flux parameterization in the representative elementary watershed approach: Application to basin**, Water Resources, 41(4), pp 18-27, 2005

[40] T. Rientjes, **Inverse modeling of the rainfall-runoff relation: a multi objective model calibration approach**, published Ph.D thesis, Delft University of Technology, Delft, Netherlands, 2004

[41] J. Seibert, **Conceptual runoff models - fiction or representation of reality?**" *Acta University publications*, A

comprehensive summaries of Uppsala Dissertations from Faculty of Science and Technology 436, Uppsala. ISBN 91-554-4402-4.

[42] D.G. Tarboton, **Rainfall runoff processes: A workbook for online module**, 2003, Available online at [web]: www.engineering.usu.edu/dtarb/rrp.html, last retrieved: 16-8-2012.

[43] A.S. Tokar, P.A. Johnson, **Rainfall-runoff modeling using neural networks**, J. of Hydrology and Engineering, 4(3), pp.232–239, 1999

[44] R. Ursem, T. Krink, M Jensen, Z. Michalewicz, **Analysis and modeling of controls in dynamic systems**. Transaction on Memetic Systems and Evolutionary Computing, 6(4), pp.378-389, 2002

[45] P. Varoonchotikul, **Flood forecasting using artificial neural networks**, published Ph.D. thesis, Swets & Zeitlinger, Lisse, The Netherlands, 2003

BIOGRAPHIES OF AUTHORS (10 PT)

	Arnold Adimabua Ojugo received BSc Computer Science from the Imo State University Owerri in 2000, MSc in Computer Science from the Nnamdi Azikiwe University Awka in 2005, and PhD in Computer Science from the Ebonyi State University Abakiliki in 2013. He currently lectures with the Department of Computer Science at the Federal University of Petroleum Resources Effurun, Delta State, Nigeria. His research interests: Intelligent Systems / Machine Learning, Data Security/IoT, and Graph Theory. He is an Editor with Progress for Intelligent Computation and Application, SceincePG etc. He is a member of: The Nigerian Computer Society, Computer Professionals of Nigeria and The International Association of Engineers. He is married to Prisca with five children.