Boosted Hybrid Recurrent Neural Classifier for Text Document Classification on the Reuters News Text Corpus

Emmanuel Buabin, *Member, IEEE*

***Abstract*—The objective is multi–classed news text classification using hybrid neural techniques on the modapte version of the Reuters news text corpus. In particular, a neuroscience based hybrid neural classifier fully integrated with a novel boosting algorithm is examined for its potential in text document classification in a non-stationary environment. The novel boosting algorithm termed NeuroBoost is an Adaboost-like algorithm that computes and integrates boosted weights into neural network weights, using back-propagation approach. The main contribution of this paper is the provision of an obvious scientific basis for integrating boosted weights into hybrid neural network weights. Results attained in this experiment show impressive performance by the hybrid neural classifier even with minimal number of neurons in constituting structures. A minimal but appreciable increase is observed in performance if an appreciable number of neurons are added.**

***Index Terms*—Hybrid neural classifier, intelligent agent, natural language processing, recurrent neural network, text classification.**

I. Introduction

With each computer literate as potential author on the web, millions of domains have been registered and billions of files published on the internet daily. Today, Really Simple Syndication (RSS) feeds have been deployed to promote reader/viewership by publishing time based activities such as news. Due to the vast amount of information RSS feeds contain, researchers have investigated approaches by which contents of such platforms could be extracted for research purposes. With increasing complexity in customer needs, intelligent systems meant to work in such domains ought to be robust in both design and learning, so as to take effective decisions within the shortest possible time. In this paper, hybrid neural and boosting techniques are combined to design a hybrid recurrent neural classifier for a text classification task. The potential of the resulting hybrid classifier in the task is explored on the modapte version of the Reuters news text corpus. The hybrid neural classifier is however observed to be robust in learning and outperforms the standard SRN, even with minimal number of hidden layer neurons in constituting structures. Also, the weight integration mechanism makes it possible to update learned hybrid weights with boosted weights. The remainder of the paper is arranged as follows. Section II provides literature background and related research work in the area of neural based text classification. Section III describes the proposed system, whereas Sections IV, V and VI present experimental results, discussions and conclusions on the experiment.

II. Literature Review & Related Work

Recurrent neural networks (RNNs) with their stability, feedback loops, coupled with generalization capabilities, adaptiveness and fault tolerance tend to be good candidates for learning large scale real world task [1]-[13]. The Simple Recurrent Network (SRN) being the simplest amongst the recurrent family has the capability to learn reasonably short sequences and produce efficient outputs. Whereas the context unit of the SRN architecture writes activation values from the hidden units, the hidden units receive input from the context units. The hidden units are first updated before the context units copy hidden units' values and provide input to the hidden units. In effect, the context layer enables the output values of the network's hidden units to be stored and then re-used in the network by providing the hidden layer with the pattern of the previous activation state. In this experiment, SRN classifiers with input, output, hidden and context layers are trained with supervised learning techniques. Bias nodes were connected to the input and hidden layers to facilitate learning. In general, the input to a hidden layer $\mathbf{L_n}$ is constrained by the underlying layer $\mathbf{L_{n-1}}$ as well as the incremental context layer $\mathbf{C_n}$. The activation of a unit $\mathbf{L_{ni}(t)}$ at time step t is calculated based on the weighted activation of the units in the previous layer $\mathbf{L_{(n-1)}}(t)$ and the units in the current context of this layer $\mathbf{C_{ni}}$ (t). The resultant is limited by a logistic function. This provides a simple form of recurrence for training networks to perform sequential tasks over time.

$$C_{ni}(t) = (1-\Psi_n)\, L_{ni}(t-1) + \Psi_n\, C_{ni}(t-1) \tag{1}$$

$$L_{ni}(t) = f\,(\Sigma_k\, w_{ki}\, L_{(n-1)k}(t) \;+\; \Sigma_l\, w_{li}\, C_{(n)l}(t)) \tag{2}$$

where Ψ_n is the hysteresis value.

A. Hybrid Neural/Connectionist Architectures

Several scientific research works have reported that knowledge-based neural networks (1) train faster than standard neural networks mainly because the initial information introduced into the network gives a good starting point for network operation (2) generalize better to future examples than standard networks. [8], [14] and [15] provide motivation in the direction of knowledge based architectures. With a lot of work undertaken in the area of text classification, hybrid classifiers [16]-[22] and boosting enabled artificial network systems [23]-[27] have also been investigated to improve classification performance. Although all of the above-mentioned research works are relevant in their respective applications, they are limited from basic neuroscience perspective. From neuroscience theory, (1)

Manuscript received May 28, 2012; revised September 5, 2012.

E. Buabin is with the Methodist University College Ghana, Dansoman, Greater Accra, Ghana (e-mail: emmanuel.buabin@ieee.org).

10.7763/IJMLC.2012.V2.195

localized neurons are similar in structure and perform related function – [28] provides a detailed representation of the human brain. (2) minimal number of localized neurons are selected and put into an ensemble for information processing. (3) results are enhanced to attain high throughput. Aside the neuroscience contradictions, none of the boosting enabled systems provide clear integration ability for updating learned neural network weights after boosting. [29] confirms this by making the following statement: "…boosting is not quite a stand-alone algorithm: the classifiers need to consider the weights when they perform their classifications. It is not always obvious how to do this for a particular classifier, but we have seen methods of doing it for a few classifiers."

III. Proposed System

Motivated by the need to implement a hybrid neural based text classification system that (1) has similar localized constituents in an ensemble performing related function, (2) uses minimal number of hidden layer neurons in constituting structures (3) is fully integrated with a boosting algorithm harnessed with back-propagation techniques, (4) provides an obvious scientific basis for updating learned neural network weights and (5) is robust, adaptive, fault tolerant etc, the boosted Hybrid Recurrent Neural Classifier abbreviated Hy-RNC, was proposed. In this paper, the general and special cases of Hy-RNC are investigated for their potential in text document classification. The general Hy-RNC contains an ensemble of N simple recurrent networks (SRNs) and a boosting unit. The boosting unit contains an Adaboost-like algorithm termed NeuroBoost. NeuroBoost operates by accepting network outputs from the hybrid neural classifier environment and extracting weak hypothesis from the outputs. The algorithm computes temporary weights based on extracted (weak) hypothesis and updates learned neural network weights through back-propagation approach. Having integrated the algorithm's weights into the neural based weights, the new neural weights are copied into the hybrid environment for retraining or re-validating of exemplars. Hy-RNC is further trained and validated repeatedly on updated neural weights until an early stopping criterion is reached.

A. Hybrid Neural/Connectionist Architectures

For this experiment, the modapte version of the Reuters news text corpus was used. The news text documents in this corpus appeared in the Reuters newswire in 1987 and were made available by Reuters & CGI for academic research purposes. In this paper, eight (8) document categories are used namely, Money/Foreign Exchange (Money-fx), Shipping (Ship), Interest Rates (Interest), Economic, Currency, Corporate, Commodity and Energy. In all the number of unique files extracted are Train (891, 8.26%), Validation (160, 1.48%) and Test (9737, 90.26%). The first (unsorted) one hundred and twenty-five (125) news text document titles of each category (i.e. currency, economy etc) were extracted, pre-processed and fed into the hybrid classifier in sequential pattern. The extracted titles provide Hy-RNC, an opportunity to train on examples belonging to all classes. Abbreviations such as "nat'l", 'int'l' were replaced with full words ("National", "International"). For a lexicon, the three hundred and fifty thousand (350,000) simple English word-set from Infochimps.org was used. The lexicon base was automatically updated with new words encountered in the training set. For each word in the lexicon, the corresponding word frequencies in the training set were retrieved and normalized (along and across classes) as semantic vectors. The semantic vectors were then used to form a new training, validation and test set. The training set was fed sequentially into Hy-RNC and the classifier validated repeatedly on the validation set, till an early stopping criterion was reached. Having tested Hy-RNC on the test set to ascertain performance, the F-measure performance metric was used to evaluate performance. The rationale being that, F-Measure is neither biased towards Recall or Precision. Although this paper explores a real world text classification task based on neuroscience principles, engaged technique(s) could be adopted in other ergodic domains where both regular and irregular sequences are generated. For example, classifying sequential based performance measurements generated by a robot or intelligent agent in a telecommunications, automobile, medical, financial and military domain. To be specific, classifying sequential based performance data produced by (a) an assistive robot monitoring recovery process of an autistic patient; (b) an intelligent agent monitoring a radar system; (c) an intelligent agent monitoring the effectiveness of an adaptive missile defense system; (d) an intelligent agent monitoring stock performance of certain stock exchanges.

B. Hy-RNC

Hy-RNC is a fully connected hybrid network built on neuroscience principles. Hy-RNC can be subdivided into two (2) networks namely the general and special cases of Hy-RNC. In this paper, the general and special cases of Hy-RNC are investigated. In the general case, a number of SRN networks are trained and corresponding learned weights averaged. The averaged weights are then passed into the boosting unit for network output enhancement and weight integration. Newly integrated weights are fed into the hybrid classifier environment for further training and validation on examples. Each constituting SRN in the general case has *i* number of hidden neurons and *k* number of input/output units. Each SRN is fully connected and has bias connections at both Input and hidden layers. Whereas the general case trains N number of SRNs, the special case trains a single SRN and passes learned information directly to the boosting unit. Similarly, the learned information in the special case is boosted and fed into the hybrid classifier environment through the NeuroBoost algorithm. The constituting SRN in the special case has the same network structure as those in the general case (i.e. *k-i-k*). Hy-RNC basically operates by initializing parameters such as hidden layers, hidden layer neurons, input-output units with default values. The input, output, context and hidden layer weights are randomly selected and used to initialize the hybrid network. For each news title in an epoch, Hy-RNC creates and initializes layer activations (context and hidden) with default values. The Layer activations are updated after every time step until the end of the sequence is reached. Having ended the sequence, Hy-RNC updates the hybrid network weights and re-initializes layer activations to default values. Hy-RNC is trained and validated repeatedly until an early stopping criterion is reached. The early stopping technique employed in this experiment ensures that the learning process is not affected with noise in the dataset.

C. Novel Boosting Algorithm – NeuroBoost

NeuroBoost is an Adaboost-like algorithm that uses back-propagation approach to integrate boosting weights into neural network weights. The algorithm operates by (1) accepting neural network weights and outputs (2) Initializing temporary vectors such as weights, error, and alpha. In one time step (T), NeuroBoost retrieves weak hypothesis from N classified examples and computes corresponding vectors error: (1*T), alpha: (1*T), boosting weights: ones (N*T), deltah and deltao. The algorithm's weights (i.e. boosting weights) are integrated into the hybrid network weights through back-propagation approach and subsequently transferred into the hybrid neural classifier environment for re-training and re-validation on examples. The algorithm's ability to integrate boosting weights into hybrid neural weights makes it feasible to measure the deviation of network outputs on the Least squares error method. The integrating ability also provides an obvious scientific basis for connecting boosting algorithms to neural based systems.

D. Integrating NeuroBoost into Hy-RNC

The integration between NeuroBoost and Hy-RNC can be done in two ways. The first approach splits the boosting weight dimension (N*1) by the number of output units (*k*) followed by the number of hidden layer neurons (*i*), whilst the second splits by the number of hidden layer neurons (*i*) and subsequently number of output units (*k*).

Proof 1 – Split by Output Units and then by Hidden Layer neurons ("*k*" before "*i*")

Let (N*1) be the boosting weight dimension. For every generated Hypothesis (h) in the boosting unit, there exists a set of output units (k) that were employed in the computation of the boosting weights – i.e. N*1 = N*k | k*1. Secondly, for every generated output activation there exist a set of hidden layer activations (i.e. "i") that were employed in the computation of the output activations – i.e. N*1 = N*i | i*k | k*1. Let matrices **P** = N * 1, **Q** = N * i, **R** = i * k, **S** = k * 1. Using Matrix property **A'A** = **AA'** = **I**. **Q R S** = **P**. Making **R** the subject, multiple through by **Q'**. **Q'** (**Q R S**) = **Q' P** ; **I R S** = **Q' P.** Multiplying by **S', R S S'** = **Q' P S'** ; **R I** = **Q' P S'** ; **R** = **Q' P S'.** In the case of this split, "N*1" is the boosted weight dimension, "N*i" is the hidden layer activations dimension for instances. "i*k" is the output weight dimension and "k*1" is the average of network outputs across instances. The output weight is therefore updated as follows

$$x = \eta * R' + (\text{momentum} * w^n) \quad (3)$$

$$w^{n+1} = w^n + x \quad (4)$$

where **R'** is the new parameter introduced in the learning formula. All other variables retain their standard definitions in the learning formula.

Proof 2 - Split by Hidden Layer Neurons then by Output Units ("*i*" before "*k*")

Similarly, the split could be done as follows N*1 = N * k | k * i | i * 1. Let matrices **P** = N * 1, **Q** = N * k, **R** = k * i, **S** = i * 1. Using Matrix property **A'A** = **AA'** = I. Making R the subject, Multiplying **Q'**. **Q R S** = **P** ; **Q'** (**Q R S**) = **Q' P** ; **I R S** = **Q' P** . Multiplying through by **S'**, **R S S'** = **Q' P S'** → **R I** = **Q' P S'** → **R** = **Q' P S'.** In the case of this split, "N*1" is the boosted weight dimension, "N*k" is the output dimension of all instances, "k*i" is the output weight dimension and "i*1" is the average of hidden layer activations across instances. The output weight is therefore updated as follows

$$x = \eta * R + (\text{momentum} * w^n) \quad (5)$$

$$w^{n+1} = w^n + x \quad (6)$$

where **R** is the new parameter introduced in the learning formula. All other variables retain their standard definitions in the learning formula.

E. F-Measure

In evaluating performance of Hy-RNC, the F-Measure was used. The F-Measure metric by its nature is neither biased towards Recall or Precision. In data mining, Recall is defined as the number of True Positives over the sum of True Positives and False Negatives. Precision is also defined as the number of True Positives over the sum of True Positives and False Positives. The F-Measure is therefore expressed as

$$F = (2 * \text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}) \quad (7)$$

F. Confusion Matrix Analysis

For all instances in the test set, Hy-RNC generates a corresponding *k* (i.e. eight) bit data. A predicted network output of a class is deemed to have been classified (i.e. 1), if it is at least 0.50. Where a predicted network output is strictly less than 0.50, the resulting classification is set to 0. For example, a *k* bit output [0,0,1,1,0,0,0,1] of an instance with corresponding target [0,0,1,0,0,0,0,1], is represented in Hy-RNC as follows: [0,0] = 5, [0,1] = 1, [1,0] = 0, [1,1] = 2, where [*a*,*b*] is interpreted as "*a*" classified as "*b*". Similarly, the confusion matrix is used to extract weak hypothesis from predicted network outputs. A predicted output (i.e. *k* bit data) is deemed to be classified if the number of correctly classified classes or bits is at least 50% or 0.5. Furthermore the confusion matrix is used to communicate the overall correct (i.e. [0,0] and [1,1]) and incorrect (i.e. [0,1] and [1,0]) classifications on the test set. Although the experiment uses a bit-wise approach to represent classifications, the approach could be modified to generate classification results for each class across the dataset.

IV. Experiment

For each SRN classifier used in this experiment, a corresponding k-2-k and k-16-k network topologies were used. A k-2-k constituted Hy-RNC network was compared against a corresponding k-2-k standard SRN network. Similarly, a k-16-k constituted Hy-RNC network was compared against a k-16-k standard SRN network. Overall, three different kinds of classifiers were used – namely Hy-RNC (General Case), Hy-RNC (Special Case) and SRN. The general case consists of two (2) SRN classifiers. For every classifier type of a certain topology (k-i-k), three test runs were conducted. The experiments were conducted at fixed learning rate (0.01), momentum (0.9) and at an early stopping criterion ((old error 1- new error >0.0000001) or (old error 2- old error 1 >0.0000001)). The results are as follows – See Tables I and II.

TABLE I: SHOWS THE THREE KINDS OF CLASSIFIERS UNDER INVESTIGATION. CORRESPONDING PERFORMANCE ACROSS THREE TEST RUNS IS ILLUSTRATED AGAINST CLASSIFIER NAMES. THE CONSTITUTING SRN NETWORK TOPOLOGY UNDER CONSIDERATION IS K-2-K, WHERE K IS THE NUMBER OF INPUT / OUTPUT UNITS. HY-RNC (2GC) AND HY-RNC (SC) ARE BOOSTED HYBRID RECURRENT NEURAL NETWORKS WITH TWO (2) AND ONE (1) SRN CLASSIFIERS RESPECTIVELY. THE STANDARD SRN IS OF TYPE K-2-K

Classifier	Test Run 1	Test Run 2	Test Run 3
Hy-RNC(GC)	0.942235	0.950328	0.948067
Hy-RNC(SC)	0.946769	0.949775	0.950106
SRN	0.927516	0.927898	0.928362

TABLE II: SHOWS THE THREE KINDS OF CLASSIFIERS UNDER INVESTIGATION. CORRESPONDING PERFORMANCE ACROSS THREE TEST RUNS IS ILLUSTRATED AGAINST CLASSIFIER NAMES. THE CONSTITUTING SRN NETWORK TOPOLOGY UNDER CONSIDERATION IS K-16-K, WHERE K IS THE NUMBER OF INPUT/ OUTPUT UNITS. HY-RNC (2GC) AND HY-RNC (SC) ARE BOOSTED HYBRID RECURRENT NEURAL NETWORKS WITH TWO (2) AND ONE (1) SRN CLASSIFIERS RESPECTIVELY. THE STANDARD SRN IS OF TYPE K-16-K

Classifier	Test Run 1	Test Run 2	Test Run 3
Hy-RNC(GC)	0.958947	0.958324	0.957333
Hy-RNC(SC)	0.959169	0.958788	0.957947
SRN	0.958886	0.957879	0.957312

V. DISCUSSION

In all of the test runs, the hybrid networks performed better than the standard SRN. Although, all classifiers were trained with different initial weights, the networks generated results that reflected information learnt over time. Meaning that, the performance of the classifiers did not just occur by chance, but based on learned information. This however provides a basis for comparison amongst the classifiers under investigation. From experimental result 1 (Table I), Hy-RNC (2GC) performed better than the SRN by at least 1.471% and at-most 2.243%. The Hy-RNC (SC) performed at least 1.92% and at most 2.17% better than the standard SRN. From experimental result 2 (Table II), Hy-RNC(2GC) performed at least 0.006% and at most 0.044%, Hy-RNC(SC) performed at least 0.028% and at most 0.091% than the standard SRN. This means that, the performance of the hybrid classifiers generally appreciated when the number of hidden layer neurons was increased to sixteen (16), but although it still performed better than the SRN, the margin was not large enough to be considered significant. Unlike Table II, Table I recorded high performances by the hybrid classifiers as compared to the SRN.

VI. CONCLUSION

This paper demonstrated that (1) an ensemble of recurrent neural networks could used be in a real world news text classification task such as news routing (2) minimal number of hidden layer neurons could be used in constituting structures to attain high performance (3) network responses could be boosted to increase network performance (4) a connection between boosting algorithms and neural based hybrid models such Hy-RNC is feasible (5) the connection makes it possible for boosting weights to be integrated into neural based weights (6) back-propagation approach could help assist in integrating learned weights (7) newly integrated weights could be used on examples in the validation set for onward measurement of deviation using least squares method. Although the weight integration is explained for a classifier from the recurrent neural network family, the engaged technique could be adequately used to update weights in other neural based supervised learning networks such as support vector machines (SVM). The approach taken in this paper, is different from most of the related approaches for natural language processing, in that, a boosted hybrid recurrent neural network has been used and yields better classification results or performance. Owing to results attained by Hy-RNC, it is worth investigating novel approaches that have the possibility to improve classification results and widen performance margin over other networks. For this reason, a robust and rigorous training regime on the revised boosted Hy-RNC classifier is worth exploring for its potential in real world text document classification.

REFERENCES

[1] S. Garfield and S. Wermter, “Call Classification using Recurrent Neural Networks, Support Vector Machines and Finite State Automata”, *Knowledge and Information Systems: An International Journal*, .vol. 9, 2,131-156, doi=10.1.1.63.3387, 2006.

[2] S. Garfield and S. Wermter, “Comparing State Vector Machines, Recurrent Networks and Finite State Transducers for Classifying Spoken Utterances”. *International Conference on Artificial Neural Networks*.Istanbul.Turkey.646-653, doi=10.1.1.7.9464, 2003a.

[3] S. Garfield and S. Wermter, “Recurrent Neural Learning for Classifying Spoken Utterances”, *Expert Update, Special Issue on Neural Language Processing*, vol. 6, No. 3, 31-36, doi=10.1.1.64.1860, 2003b

[4] S. Wermter, “Hybrid Connectionist Natural Language Processing”. *Chapman and Hall Thomson International, London*, UK, 1995.

[5] S. Wermter “Preference Moore Machines for Neural Fuzzy Integration”. *Proc. International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 840-845, 1999.

[6] S. Wermter, G. Arevian and C. Panchev, “Towards Hybrid Neural Learning Internet Agents”. In: Wermter S., Sun R. (Ed.) *Hybrid Neural Systems*. 160-176. Springer, Heidelberg, Germany. doi=10.1.1.38.1800, 2000.a

[7] S. Wermter, G. Arevian and C. Panchev, “Network Analysis in a Neural Learning Internet Agents”. *Proc, International Conference on Computer Intelligence and Neuroscience*. Atlantic City. USA., 880-884, doi=10.1.1.41.4409, 2000b.

[8] S. Wermter and V. Weber, “SCREEN: Learning a Flat Syntactic and Semantic Spoken Language Analysis Using Artificial Neural Networks”, *Journal of Artificial Intelligence Research*. vol. 6, no. 1,35-85, doi=10.1.1.104.5361, 1997.

[9] S. Wermter , G. Arevian and C. Panchev, “Recurrent Neural Network Learning for Text Routing”. *Proc. International Conference on Artificial Networks*.p. 898-903, Edinburgh, UK, doi=10.1.1.120.4922, 1999.

[10] S. Wermter, C. Panchev and G. Arevian, “Hybrid Neural Plausibility Networks for News Agents”, *Proc. National Conference on Artificial Intelligence AAAI*. Orlando, USA. 93-98, 1999b.

[11] G. Arevian "Recurrent Neural Networks for Robust Real-World Text Classification" *Proc. of the 2007 IEEE/WIC/ACM International Conference on Web Intelligence*; Silicon Valley, USA, 2007

[12] G. Arevian, G. and C. Panchev ``Robust Text Classification using a Hysteresis-Driven Extended SRN", *Proc. 2007 International Conference on Artificial Neural Networks*, Porto, Portugal, 2007.

[13] G. Arevian and C. Panchev,"Optimizing the Hystereses of a Two Context Layer RNN for Text Classification", *Proc. International Joint Conference on Neural Networks*, Orlando, Florida, USA, 2007.

[14] J. W. Shavlik, "Extended Abstract: Combining Symbolic and Neural Learning", Machine Learning 14. *Kluwer Academic Publishers*. Boston. 321-331, 1994.

[15] R. Sun, "Introduction to connectionist symbolic integration", R. Sun and F. Alexandre, (eds.) *Connectionist-Symbolic Integration. Lawrence Erlbaum Associates*, 1997.

[16] D. W. Abbott, "Combining Models to Improve Classifier Accuracy and Robustness", *Proc. International Conference on Information Fusion - Fusion 1999*, Sunnyvale, CA, 1–7, 1999.

[17] T. G. Dietterich, "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms", *Neural Computation 10*, 1895–1923. doi=10.1.1.37.3325, 1998.

[18] G. Giacinto, F. Roli, and G. Fumera, "Design of Effective Multiple Classifier Systems by Clustering of Classifiers", *Proc. of ICPR 2000 15th International Conference on Pattern Recognition*, Barcelona, Spain, 160–163, doi=10.1.1.11.5328, 2000.

[19] T.G. Dietterich, "Ensemble Methods in Machine Learning", *Lecture Notes in Computer Science 1857*, Springer-Verlag, New York, 1-15. doi=10.1.1.34.4718, 2000.

[20] R. P. W. Duin, "The Combining Classifier: To Train or Not to Train", In: Kasturi, R., Laurendeau, D., and Suen, C. (eds) *Proc. of 16th International Conference on Pattern Recognition* vol. II, Quebec,765-770. doi=10.1.1.19.2114, 2002.

[21] K. Turner and J. Ghosh, "Analysis of Decision Boundaries in Linearly Combined Neural Classifiers", *Pattern Recognition*, 29(2):341–348, doi=10.1.1.94.954, 1996.

[22] S. Garfield, S. Wermter and S. Devlin, "Spoken Language Classification using Hybrid Classifier Combination". *International Journal of Hybrid Intelligent Systems* vol. 2, No.1,13-33, doi=10.1.1.62.7669, 2005.

[23] Lim, C. P. and W. Goh, "The Application of an Ensemble of Boosted Elman Networks to Time Series Prediction: A Benchmark Study".*International Journal of Computational Intelligence*, Volume 3 Number 2, 2005.

[24] R. Boné, M. Assaad and M. Crucianu, "Boosting Recurrent Neural Networks for Time Series Prediction", Proc. International Conference in Roanne, D.W. Pearson, , Steele, N. C., Albrecht, R. F.(eds).Springer.RFAI.*Artificial Neural Nets and Genetic Algorithms*. France.18-22. doi=10.1.1.125.6008, 2003

[25] A. J. Grubb and A. Bagnell, "Boosted Backpropagation Learning for Training Deep Modular Networks". In: *Proc. 27th International Conference on Machine Learning*. Haifa. Israel, doi=10.1.1.170.1257, 2010.

[26] H. Schwenk, Y. Bengio, "*Boosting Neural Networks*". Neural Computation 12. MIT. USA. pp1869–1887, 2000.

[27] D. W. McKee,"Boosting Evolved Artificial Neural Network to Improve Breast Cancer Classification Accuracy", 2001.

[28] S. Haykin,"Neural Networks and Learning Machines", *Pearson Prentice Hall*. 3rd edition. pp 40. 2008.

[29] S. Marsland, "Machine Learning: An Algorithmic Perspective", *Chapman & Hall/CRC*, pp. 157, 2009.

Emmanuel Buabin, is a Lecturer in the Department of Information Technology at the Methodist University College Ghana, Ghana. As part of his professional activities, Emmanuel has been an Executive Member on the IEEE Ghana Section since 2010. He has worked in the capacity as Software Engineer on both local and International projects. His main research interests among others include Machine Learning with Artificial Neural Networks, Natural Language Processing, Reinforcement learning and Cognitive Neuroscience. Emmanuel holds a BSc Honours degree in Computer Science & Statistics and an MEng degree in Electrical Engineering & Information Technology (Technical Computer Science option) from the University of Ghana and Deggendorf University of Applied Sciences respectively.