International Journal of Engineering Research and Technology.
ISSN 0974-3154 Volume 11, Number 10 (2018), pp. 1511-1524

http://www.irphouse.com

Classification with Modified Deep Belief Network for Large Dataset

Azhagammal Algarsamy[1,*], Ruba Soundar Kathavarayan[2]
[1]Research Scholar, Department of Computer Science, Bharathiyar University, Tamilnadu, India.
E-mail: azhagammal37@gmail.com

[2]Department of Computer Science Engineering, P. S. R Engineering College, Sivakasi-626140, Tamilnadu, India.
rubasoundar@gmail.com

ABSTRACT

Due to the technology development, the data's are increased in their volume, velocity and variety, referred to as large data or Big Data. From this large data, we can gain knowledge by using techniques like classification, association rule learning, outlier detection, clustering, and regression. By definition, classification canvass the specific set of data, procreate a set of grouping rules which can be used to codify the future data. The aim of classification is to meticulously forecast the target class for each case in the data. The classification stint starts with a dataset in which the class assignments are known. Available methods for classifications are support vector machines, quadratic classifiers, k-nearest neighbor, boosting, decision trees, neural network, learning vector quantization. From the above list, we use Neural Network for classification in this paper, Neural network is one of the most effective method for classification which are relatively rude electronic networks of neuron based on the neural structure of the brain. Neural network technique is either supervised or unsupervised. The supervised method contains algorithms like Back Propagation, Feed Forward, and so many. Deep neural network comprises more than one hidden layers which actually improves the accuracy in classification. The Deep belief network is an unsupervised learning. It includes more than one trained RBMs. In this paper we implemented novel Modified Deep Belief Network (MDBN) on four datasets namely SPECT, SPECTF, Pest formation in crops, readmission rate of diabetic patients in medical dataset. In all our works we gain high accuracy while comparing to the normal Deep Belief Network by introducing learning rate decay to the weight updating process.

Keywords: Modified DBN, DBN, Deep learning, Deep Belief Network

---

[*] *Assistant Professor, Dept. of Computer Science, Ayya Nadar Janaki Ammal College, Sivakasi*

I. INTRODUCTION

Classification is a data mining approach to prognosticate group membership for record instances. The goal of classification is to foretell the target class for each case in the data. Many classification methods are used based on applications. There are two main classification techniques supervised and unsupervised. In supervised classification set of possible class is known in advance. In unsupervised classification technique, the set of possible class is unknown, after classification we can assign name to the class. Some classification algorithms are Logistic Regression, Navie Bayies classifier, SVM, Decision tree, Boosted Trees, Random Forest, Neural Network, Nearest Neighbor. The classification block diagram showed in Fig.1.

Fig 1: Classification Block Diagram

From the above methods we use neural network for classifying larger dataset. Neural network is composed of nodes, arranged in layers, which convert an input vector into some output. Each unit takes an input, applies a function to it and then passes the output on to the next layer. The standard neural network structure contains three layers. First one is input layer which is used to read the user dataset. The second layer is hidden layer, the activation function is implemented. The third layer is output layer which denoted the class. Each neuron in each layer is connected and weighted. Weightings are applied to the signals passing from one unit to another and during the training phase; the weights are adjusted based on the output. The standard neural network architecture demonstrated in Fig 2.

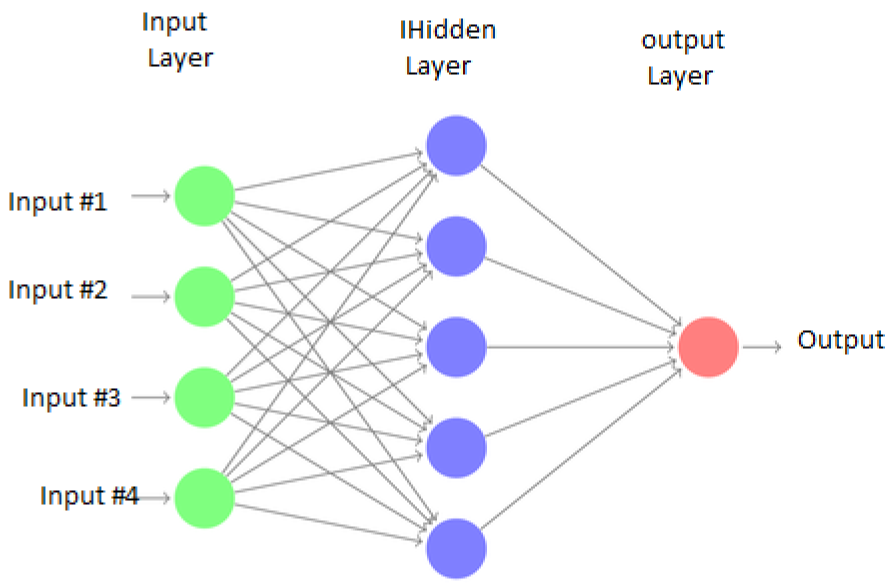

Fig 2: Neural Network Architecture

Modification in neural network structure known as deep learning is introduced by many researcher to improve the classification accuracy. Deep learning referred to as " Stacked Neural Network", that is the network is composed of more than one hidden layer, the number of node layers through which data passes in a multistep process of pattern recognition. Deep learning networks capable of handling very large, high dimensional datasets with billions of parameters that pass through non-linear functions. The deep learning architecture is presented in Fig 3.

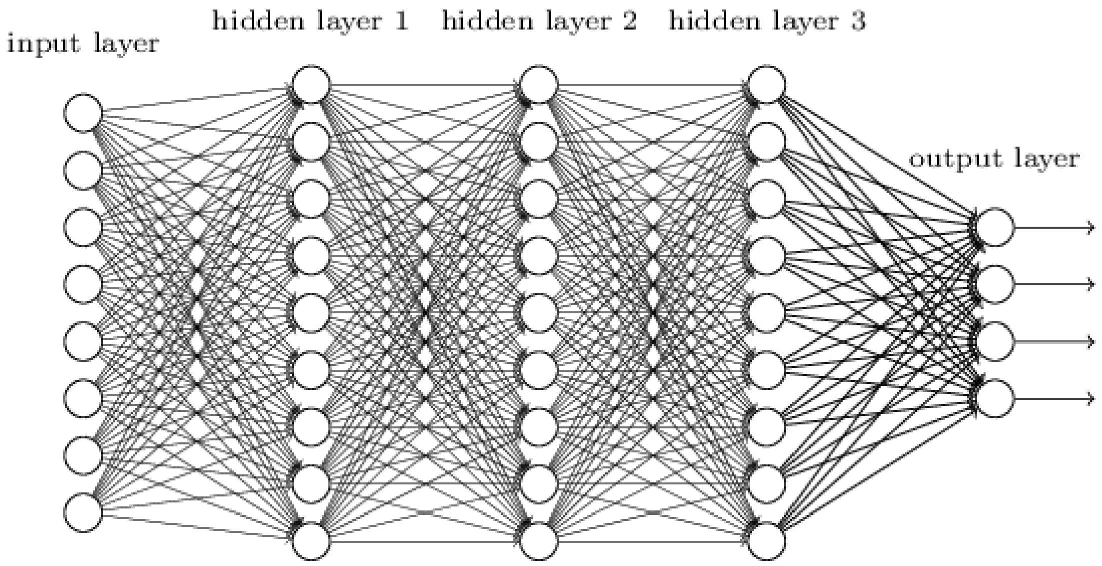

Fig 3: Deep Learning Architecture

One of the most common deep learning architecture is Deep Belief Network (DBN). DBN is a generative graphical model, composed of multiple hidden units. When DBN trained on a set of training sets with output class labels, it can learn to probabilistically reconstruct its input. The layers then act as feature detectors. After this learning step, a DBN can further trained with supervision by updating the weights to perform classification. In weight updating process the learning rate is used to diminish function of time, it is relevant to the error rate in each epoch, and also determines how much epoch is required to train the network based on the current values of the weights. In the proposed work Modified Deep Belief Network (MDBN), the learning rate decay is introduced in weight updating process which increases the accuracy in classification of large data. The learning rate decay limits the number of free parameters which eliminates the problem of over fitting, it also formalize the cost function which provides high efficiency than standard learning rate in classification problem. This paper is organized as follows. The section II describes the review of the literature, section III presents the proposed methodology, which explains the Modified Deep Belief Network, section IV describes the experimental results and concluded in section V.

II. LITERATURE REVIEW

Dahl, G. Hinton, [3] introduces the multiple layers of features in pre-training. In this method the networks are first pre-trained as a multilayer generative model which replace the Gaussian mixture models. The fine tuning is performed by using backpropagation which makes better at predicting a probability distribution over the states.

Kamada, Ichimura, [1] explained that the adaptive learning method can discover the optimal number of hidden neurons and weights according to the input space. This method determine the optimal number of layers during the learning.

Roux NLe, Y Bengio, [5] discussed that Restricted Boltzmann Machines are interesting because inference is easy in them and because they have successfully used as building blocks for training deeper models.

Y Bengio, P Lamblin, D Popovici etal, [8] proposed a DBN algorithm with many layers of hidden casual variables. This training strategy helps the optimization by initializing weights in a region near a good local minimum giving rise to internal distribution that brings better generalization.

M. A. Ranzato, M. Szummer [9] proposed an algorithm to learn text document representation based on semi-supervised autoencoders that are stacked to form a deep network.

G. Hinto , [7] explains how to train the Restricted Boltzmann Machine in a generalized manner, he also [10] presents a function that resembles negative free energy and show that the M step maximize this function with respect to the model parameters and E step maximize it with respect to the distribution over the unobserved variables, which justify an incremental variant. He also describes [11] conversion of high-dimensional data to low dimension by training a multilayer Neural Network . Gradient descent can be used for fine tuning the weights in networks.

III. PROPOSED METHOD

The proposed method describes the I) RBM architecture, II) training methodology, III) algorithm for training network, IV) algorithm for weight updating process.

III.I RBM Architecture: Several Restricted Boltzmann Machines (RBM) can be stacked and trained in a greedy manner to form Deep Belief Network architecture. RBM is an unsupervised learning method which contains two layers: input and hidden layers. Each and every layer was constructed with nodes. The input nodes process the user dataset. The hidden nodes extract the multilevel features of the dataset. The connection between the input and hidden layers weighted parameters denoted by W [2]. DBN are graphical models which gain to extract a deep hierarchical representation of the training data. The graphical DBN is showed in Fig. 4. The standard RBM is presented in Fig. 5[2].

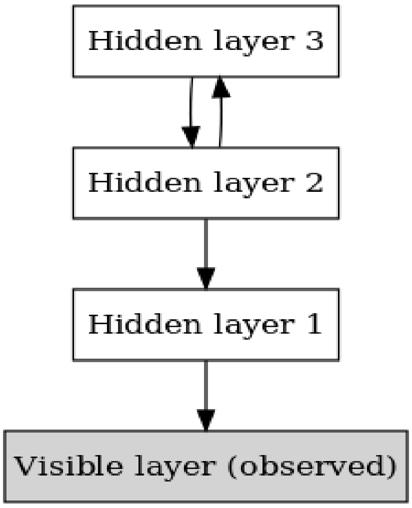

Fig 4: DBN Graphical Model

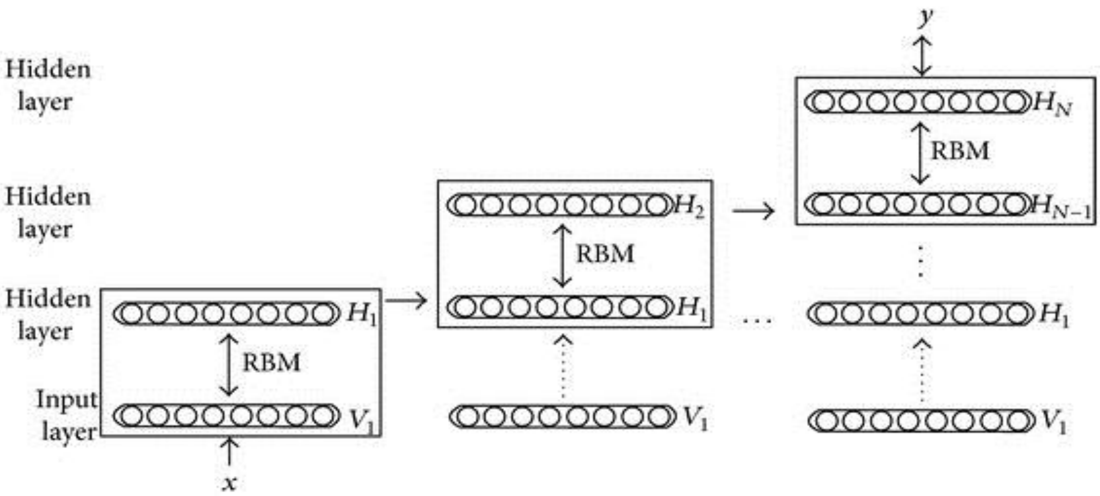

Fig 5: RBM Structure

III.II Training Methodology:

Let v_i(0≤i≤N) be a binary variable of input neurons, N is the number of input neurons, h_i(0≤j≤M) be a binary variable of hidden neuron, where M is the number of hidden neurons. The energy function $E(i, j)$ for input vector $i \in \{0,1\}^N$ and hidden vector $j \in \{0,1\}^M$ [2] is represented as

$$\boldsymbol{E(v,h)} = \sum_i \boldsymbol{b_i}\, \boldsymbol{v_i} - \sum_i \boldsymbol{c_j}\, \boldsymbol{h_j} - \sum_i \sum_j \boldsymbol{v_i W_{ij} h_j} \quad \textbf{(1)}$$

The joint probability distribution of v and h is represented as

$$\boldsymbol{p(v,h)} = \frac{1}{Z}\mathbf{exp(\text{-}E(v,h)), Z=}\sum_v \sum_h \mathbf{exp(\text{-}E(v,h))} \tag{2}$$

Where b_i and c_i are the coefficients for v_i and h_j respectively,

Weight w_{ij} is the parameter between v_i and h_j, Z is the partition function,

P(v,h) is a probability function, calculates sum over all possible pairs of visible and hidden vectors.

DBN copy the joint distribution between input vector v and the h hidden layers h^kas follows

$$\boldsymbol{P(a, h^1,\ldots.,h^h)= (\pi_{k=0}^{h-2}\ \mathrm{p}(h^k(h^{k+1}))\ \ P(1^{h-1},\ldots,h^h)} \tag{3}$$

Where $a = h^0$,

$$\boldsymbol{P(h^{k-1}|\ h^k)} \tag{4}$$

is a conditional distribution for the visible units conditioned on the hidden units of the RBM at level k, and $\boldsymbol{P(h^{h-1},h^b)}$ is the visible hidden joint distribution in the RBM[2]. Fig.6 and Fig.7 explains the procedure for training DBN and Weight updating process.

III.III Algorithm for Training DBN

Input: Dataset

Output: Trained network

Step 1: Train the first layer as RBM that models the input $a = h^{(0)}$ as its visible layer.

Step 2: By using the first layer obtain representation of the input that will be used as input for the next layer

$p(h^{(1)}=1|h^{(0)})$ or $p(h^{(1)}|h^{(0)})$

Step 3: Train the second layer as an RBM

Step 4: Repeat step 2 and step 3 for all the number of layers

Fig 6: Algorithm for Training Deep Belief Network

3.4 Steps to update the weight

Input: Random Weight initialized

Output: Wight updated based on error rate

Update the weight of Edge

$upd(w_{ij}$+η/2*(positive(E_{ij})-negative(E_{ij}))

Positive phase:

Compute positive statistics for edge E_{ij}

Positive (E_{ij})⇒ $p(H_J=1|v)$

The individual activation probabilities for hidden layer is

$p(H_J=1|v) = \sigma(B_j+\sum_{i=1}^{m} w_{ij}\, v_i)$

Negative Phase:

Compute negative statistics for edge E_{ij}

Negative (E_{ij})⇒ $p(v_i=1|H)$

The individual activation probabilities for visible layer is

$p(v_i=1|H) = \sigma(A_i+\sum_{j=1}^{n} w_{ij}\, H_j)$

Where σ is the activation function, η is the learning rate.

Fig 7: Algorithm for weight updation in Modified DBN

Training and Testing: Basically individual DBN is pre-trained using contrastive divergence. The pre-training process used to get faster convergence at the fine tuning stage. Testing process is used to determine how the network is trained based on the testing samples.

IV. RESULTS AND DISCUSSION

The proposed method has been implemented by R tool. This section will demonstrate the performance of the Modified DBN (MDBN) on several benchmark and real time datasets for various training and testing cases. We examine the accuracy. The datasets we tested are A) SPECT, B) SPECTF, C) PEST FORMATION IN CROPS, D) READMISSION RATE OF DIABETIC PATIENTS in medical dataset. The first two data sets are benchmark dataset which are downloaded from the UCI Machine Learning Repository [9] which is used to diagnose the heart diseases. The other two datasets are real time datasets. Pest Formation in crops are presented by agricultural department which is used to classify the percentage of crops affected by the pest, the medical datasets are generated by diabetology hospital which is used to classify the readmission rate of the diabetic patients. Above four datasets are divided into three test cases. In the first test case, the training process includes approximately 60% of instances and testing process includes 40% of an instance. In the second and third case, the training includes 70 %, 80%, and training includes 30%,20% of instance respectively. The test cases are listed in the Table 1.

A) SPECT dataset: This dataset describes diagnosing of Cardiac Single Proton Emission Computed Tomography Images. It classifies patients into two categories normal and abnormal. It has 267 instances and 23 binary attributes.. The accuracy for three test case are discussed on the Table 2. Fig.8 demonstrates the accuracy chart for SPECT dataset with three test cases. The test cases with 10 or more hidden layers indicate that too deep architecture brings opposite effect. The error rate for the training datasets are visualized in Fig.9 with learning rate of 0.04 and depth =9.

B) SPECTF dataset: This dataset describes diagnosing of Cardiac Single Proton Emission Computed Tomography Images. It classifies patients into two categories normal and abnormal. It has 267 instances and 45 binary attributes. The accuracy for three test cases are discussed on the Table 3. Fig.10 demonstrates the accuracy chart for SPECTF dataset with three test cases. The test cases with 10 or more hidden layers indicate that too deep architecture brings opposite effect. The error rate for the training datasets are visualized in Fig.11 with learning rate of 0.04 and depth =9.

C) Pest Formation in Crop dataset: This dataset used to classify the crops infected by pest based on color. It classifies crops color into three categories Green(normal),Yellow(mildly affected), white(fully affected). It has 50000 instances and 7 attributes. The accuracy for three test case are discussed on the Table 4. Fig.12 demonstrates the accuracy chart for Pest formation in crops dataset with three test cases. The test cases with 7 or more hidden layers indicate that too deep architecture brings opposite effect. The error rate for the training datasets are visualized in Fig.13 with learning rate of 0.04 and depth =6.

D) Readmission rate of diabetic patients in medical dataset: This dataset used to classify the diabetic patients readmission rate in the hospital. It classifies the patients readmission as NO,>30 and <30. It includes 500000 instance and 51 attributes. The accuracy for three test case are discussed on the Table 5. Fig.14 demonstrates the accuracy chart for medical dataset with three test cases. The test cases with 10 or more hidden layers indicate that too deep architecture brings opposite effect. The error rate for the training datasets are visualized in Fig.15 with learning rate of 0.04 and depth =3.

Table 1: Test Case Details

Dataset	Test Case 1		Test Case 2		Test Case 3	
	Training (instance)	**Testing (instance)**	**Training (instance)**	**Testing (instance)**	**Training (instance)**	**Testing (instance)**
SPECT	160	107	187	80	214	53
SPECTF	160	107	187	80	214	53
Pest Formation in Crops	3000	2000	3500	1500	4000	1000
Medical Dataset	300000	200000	350000	150000	400000	100000

Table 2: Accuracy for SPECT dataset

Number of depth	TEST CASE 1			TEST CASE 2			TEST CASE 3		
	Accuracy in DBN	Accuracy in MDBN	Training Time	Accuracy in DBN	Accuracy in MDBN	Training Time	Accuracy in DBN	Accuracy in MDBN	Training Time
1	65.2	68.9	651.8	68.3	72.6	658.2	69.3	73.4	660.4
2	69	71.6	655.9	69.9	73.7	659.1	72.8	74.4	661.7
3	72	73.2	664.3	72.8	73.7	667.3	73.7	75.3	668.67
4	74	76.9	668.9	74.8	77.9	670.5	76.8	79.4	679.8
5	75.3	77.6	676.1	76.8	78.5	678.7	77.4	80.7	680.5
6	78.4	80.1	678.37	79.7	81.5	679.21	79.7	82.8	680.84
7	79.9	82.5	686.1	83.4	81.36	689.5	84.7	85.3	690.15
8	82.6	84.7	689.7	83.9	84.8	690.9	85.4	86.6	692.5
9	83.4	86.8	704.3	85.3	87.9	710.9	86.5	88.7	728.9
10	81.4	83.9	709.78	83.21	85.43	715.34	85.68	86.04	730.87
11	80.29	82.43	715.34	81.5	83.5	720.65	82.86	83.18	747.54

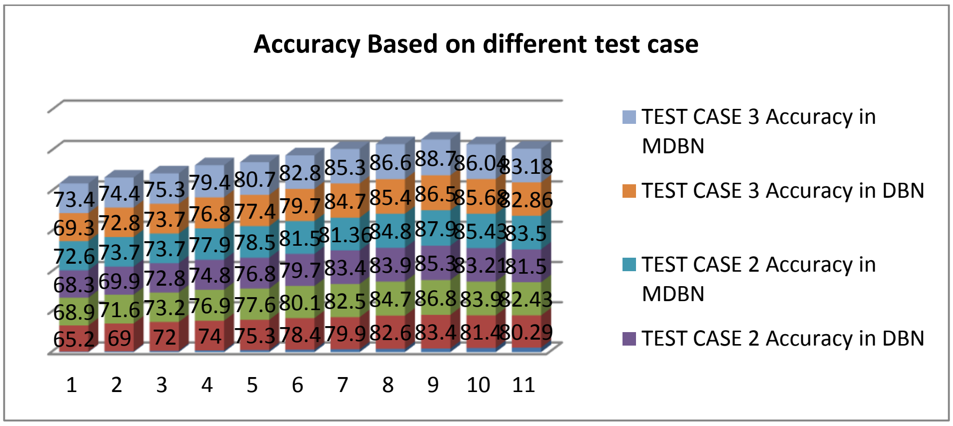

Fig 8: Accuracy for SPECT dataset on different test case and depth with learning rate=0.04

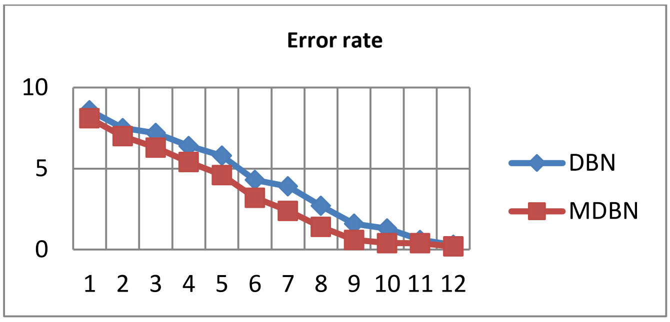

Fig 9: Error rate based on the learning rate=0.04 and depth=9

Table 3: Accuracy for SPECTF dataset

Number of depth	TEST CASE 1			TEST CASE 2			TEST CASE 3		
	Accuracy in DBN	Accuracy in MDBN	Training Time	Accuracy in DBN	Accuracy in MDBN	Training Time	Accuracy in DBN	Accuracy in MDBN	Training Time
1	63.2	66.58	650.9	67.8	71.6	656.2	68.3	72.4	665.4
2	67	69.4	654.6	68.1	72.7	657.1	71.8	73.4	666.7
3	70.4	71.2	661.4	71.5	72.7	664.3	72.7	74.3	670.67
4	72.67	74.74	665.0	73.7	76.9	668.5	75.8	77.4	675.8
5	73.94	75.97	672.7	75.34	77.5	680.7	75.4	79.7	679.5
6	74.76	79.91	679	78.72	80.5	681.21	78.7	80.8	682.84
7	76.321	81.38	690.54	81.6	80.36	689.5	82.7	83.3	690.15
8	80.74	82.20	695.30	82.874	82.8	690.9	83.4	84.6	692.5
9	81.5	84.17	705.85	84.5	86.9	718.9	84.5	85.7	730.9
10	79.0	81.54	711.1	81.0	81.43	725.34	83.68	83.04	735.87
11	76.317	81.43	718.4	80.1	81.0	729.65	80.86	81.18	738.54

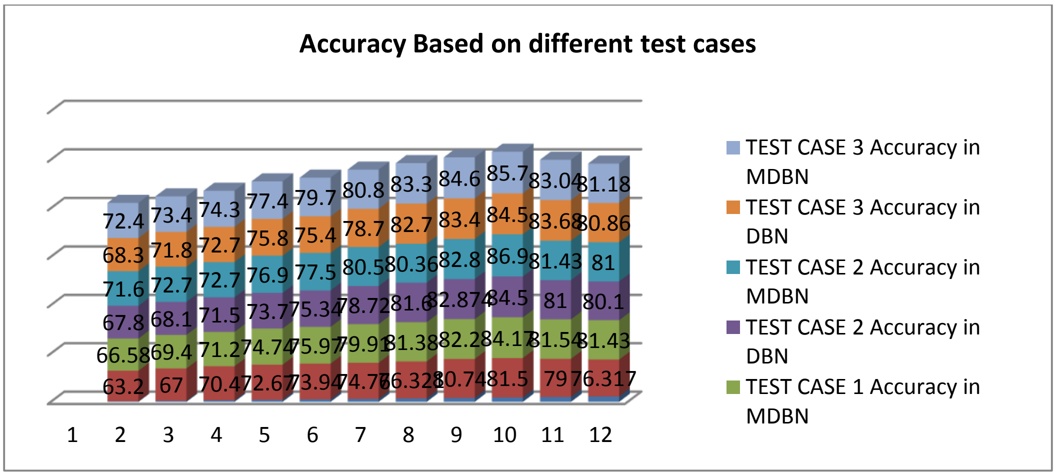

Fig 10: Accuracy based on different test cases with learning rate=0.04

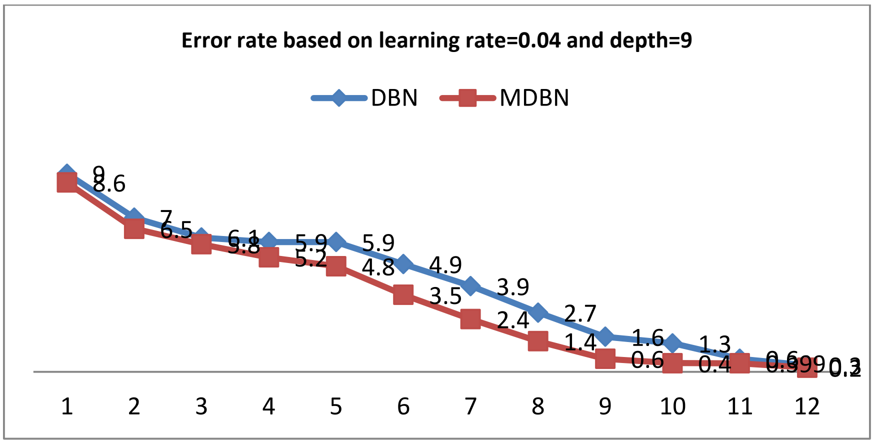

Fig 11: Error rate based on learning rate=0.04 and depth=9

Table 4: Accuracy for PEST Formation dataset

Number of depth	TEST CASE 1			TEST CASE 2			TEST CASE 3		
	Accuracy in DBN	Accuracy in MDBN	Training Time	Accuracy in DBN	Accuracy in MDBN	Training Time	Accuracy in DBN	Accuracy in MDBN	Training Time
1	60.74	62.34	817.4	61.57	63.0	819.5	62.7	64.27	821.4
2	61.2	63.54	823.15	62.71	64.61	824.80	63.861	65.76	825.74
3	62.9	65.64	825	63.49	65.49	826.1	65.49	67.58	830.95
4	64.28	66.83	829.1	65.84	66.82	831.71	66.4	68.63	832.4
5	66.47	68.45	832.48	66.963	68.980	834.5	68.73	70.76	834.889
6	69.54	70.736	836.57	71.64	73.94	836.69	73.8	74.69	837.5
7	67.25	68.51	839.54	69.48	71.56	838.4	71.7	72.0	839.1
8	64.73	67.26	841.48	68.24	69.5	839.7	70.1	71.2	842.7
9	62.16	66.47	842.75	67.41	68.721	841.0	68.9	69.87	843.9
10	59.25	64	843.86	67.0	67.3	842.864	67.28	68.4	846.97
11	56.75	62.1	845.71	66.48	66.57	844.5	66.47	67.2	848.427

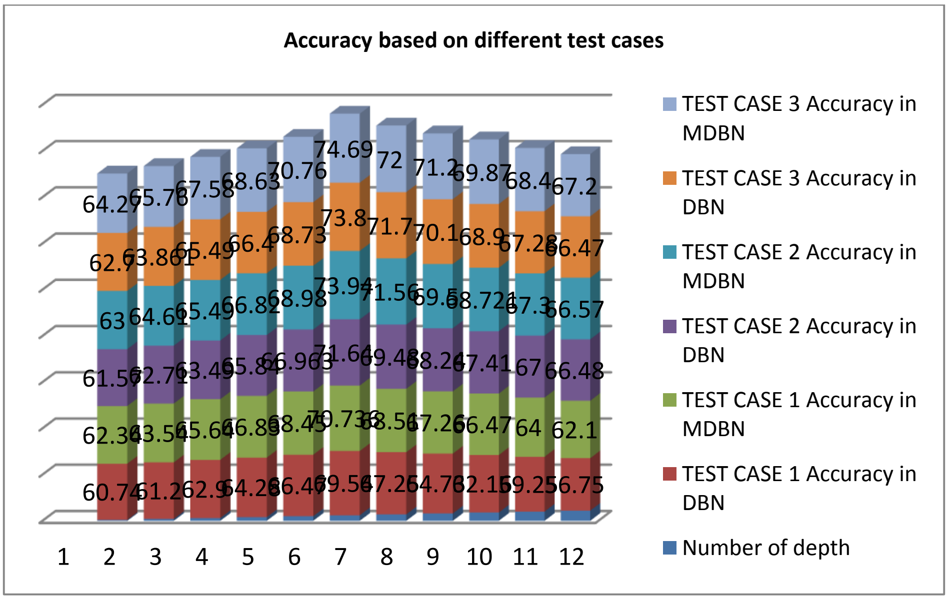

Fig 12: Accuracy based on different test cases with learning rate=0.04

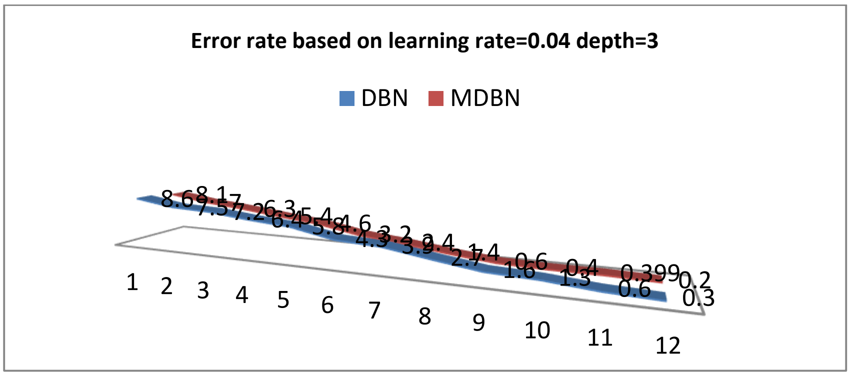

Fig 13: Error rate based on learning rate=0.04 depth=6

Table 5: Accuracy for medical dataset

Number of depth	TEST CASE 1			TEST CASE 2			TEST CASE 3		
	Accuracy in DBN	Accuracy in MDBN	Training Time	Accuracy in DBN	Accuracy in MDBN	Training Time	Accuracy in DBN	Accuracy in MDBN	Training Time
1	59.12	61.04	1007.8	62.09	64.5	1019.7	63.7	65.81	1027.5
2	61.72	63.89	1021.61	64.57	65.69	1024.1	65.98	67.47	1031.7
3	63.87	65.964	1029.4	66.71	67.89	1028.4	68.36	69.58	1039.5
4	62.75	63.58	1038.4	64.8	66.9	1032.48	66.8	67.49	1041.8
5	61.69	62.1	1045.91	63.7	64.28	1038.7	65.14	66.1	1057.9

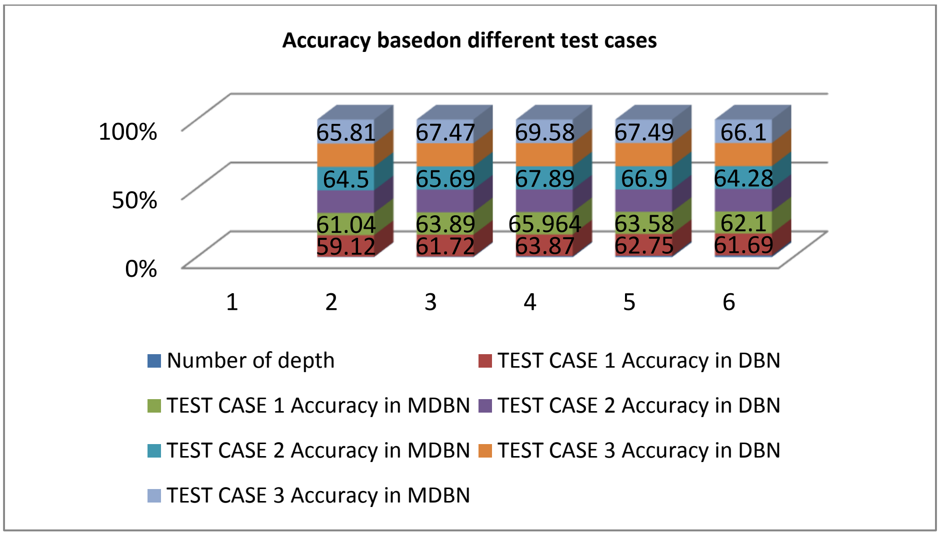

Fig 14: Accuracy based on different test cases with learning rate=0.04

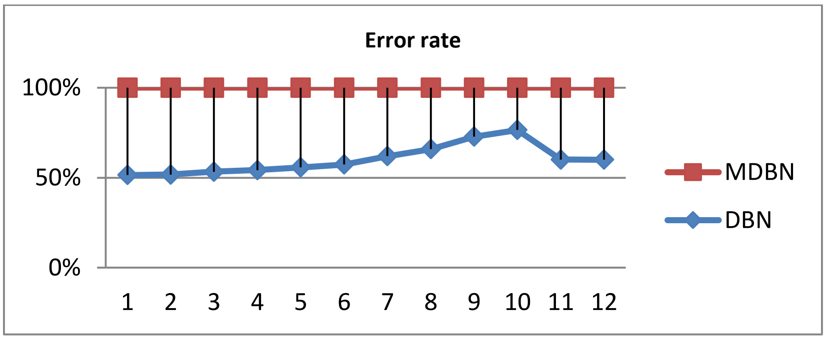

Fig 15: Error rate based on learning rate=0.04 depth=3

V. CONCLUSION

In this paper, a Modified Deep Belief Network is developed in deep neural architecture which is implemented in four datasets and its accuracy is tested against the normal Deep belief network. In the experimental results, our proposed system reached highest classification accuracy comparing with the existing DBN algorithm. For the first two dataset the depth=9 gives best accuracy, while for the third dataset depth=6 provides high classification accuracy comparing to other depths and for the fourth dataset depth=3 presents better accuracy than others. In future work, we introduce the map reduce process before training which reduces the training time and also increase the accuracy rate in proposed system.

REFERENCES

[1] Shin Kamada, Takumi Ichimura.: An adaptive learning method of Deep Belief Network by layer generation algorithm. IEEE Xplore. Nov 2016.

[2] Hinton. G. E.: A Practical Guide to Training Restricted Boltzmann Machines. *Neural Networks Tricks of the Trade Lecture Notes in Computer Science*. vol. 7700,.pp. 599-619, 2012.

[3] Mohamed A,Dahl G E, Hinton. G.: Acoustic modeling using deep belief networks[J]. *Audio Speech and Language Processing IEEE Transactions an*, vol. 20, no. 1, pp. 14-22, 2012.

[4] Boureau Y, Cun L. Y.: *Sparse feature learning for deep belief networks [C] //Advances in neural information processing systems*. pp. 1185-1192, 2008.

[5] Roux N Le, Y Bengio.: Representational power of restricted Boltzmann machines and deep belief networks[J]. *Neural Computation*, vol. 20, no. 6, pp. 1631-1649, 2008.

[6] UCI Machine Learning Repository: https://archive.ics.uci.edu.

[7] Hinton. G.: A practical guide to training restricted Boltzmann machines[J]. *Momentum*, vol. 9, no. 1, pp. 926, 2010.

[8] Bengio Y, Lamblin P, Popovici Det al.:Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, vol. 19, pp. 153, 2007.

[9] Ranzato A M, Szummer M.: Semi-supervised learning of compact document representations with deep networks[C]//Proceedings of the 25th international conference on Machine learning. *ACM*, pp. 792-799, 2008.

[10] Neal R M, Hinton G. E.: A view of the EM algorithm that justifies incremental sparse and other variants[M]//Learning in graphical models. pp. 355-368, 1998.

[11] Hinton G E, Salakhutdinov R. R.: Reducing the dimensionality of data with neural networks[J]. *Science*, vol. 313, no. 5786, pp. 504-507, 2006.