

An Efficient Normalized Restricted Boltzmann Machine for Solving Multiclass Classification Problems

Muhammad Aamir¹, Fazli Wahid³, Hairulnizam Mahdin⁴
Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn, Malaysia

Nazri Mohd Nawi²
Soft Computing and Data Mining Centre (SMC)
Universiti Tun Hussein Onn, Malaysia

Abstract—Multiclass classification based on unlabeled images using computer vision and image processing is currently an important issue. In this research, we focused on the phenomena of constructing high-level features detector for class-driven unlabeled data. We proposed a normalized restricted Boltzmann machine (NRBM) to form a robust network model. The proposed NRBM is developed to achieve the goal of dimensionality reduction and provide better feature extraction with enhancement in learning more appropriate features of the data. For increment in learning convergence rate and reduction in complexity of the NRBM, we add Polyak Averaging method when training update parameters. We train the proposed NRBM network model on five variants of Modified National Institute of Standards and Technology database (MNIST) benchmark dataset. The conducted experiments showed that the proposed NRBM is more robust to noisy data as compared to state-of-art approaches.

Keywords—Multiclass classification; restricted Boltzmann machine; Polyak averaging; image classification; Modified National Institute of Standards and Technology Datasets

I. INTRODUCTION

Classification refers to the procedure of data taxonomy and data categorization in different forms, types and other distinctive classes. The aim of classification refers to the categorization, assembling, organization and differentiation of data into classes or groups. In term of statistics and machine learning, computer programs that are based on classification, supervisory learn from existing data that is input and categories it into different classes on behalf on learn data [1]. There are a few classification based problems, such as: bio metric classification, text classification, audio classification, and video classification [2], [3]. The phenomenon of generating results from classification or categorization is carried out either in two classes or more, which is called binary classification or multiclass classification respectively. Among classification algorithms, some are multiclass classifier by nature that allows the categorization of data in more than two classes. The algorithms with binary classification nature can also be used as multinomial classification using different approaches [4].

In multiclass classification, the aim of supervised behavior learning algorithms is to assign a desired class label for every input instance. For a given dataset of (a_j, b_j) , where $b_j \in R_n$ is the j^{th} instance and $b_j \in 1, 2, 3, \dots, k$ is the j^{th} class label, the main purpose is to find a model for learning H , while $H(a_j) = b_j$ for all of the unknown testing instances. In the recent studies, many models and algorithms

have been proposed for solving the two class classification problems, some of them are easy to extend for multiclass classification problems [5], [6], while some of them are based on special formulation in order to be able in solving multiclass classification problems. In these algorithms, there is a category of models in which the algorithms use different type of approaches for transforming multiclass classification into a bunch of binary classification problems in order to affectively solve it using binary classification approach [7], [8]. Another way of multiclass classification approaches is based on a try to pose a tree hierarchy on the output, additionally with the available class labels, in order to achieve a series of tests for new class labels detection.

Solving the multiclass classification problem is still exists as a challenging issue. In famous classifications approach, the multiclass classification problem is solved using the splitting of problem into many independent binary classification sub-problems. Generally this type of solution is known as binarization, in which the classifier pretends to solve multiclass classification problem on the basis of binary classification [9]. The recent literature surveys shows that among all the multiclass classification problems, image classification attracted most of the current researchers interest.

Image classification based on unlabeled class oriented images data is a challenging problem in the field of computer vision. There is a need of classification model that is trained in an unsupervised behavior with less number of feature spaces. As there are various methods and techniques for image classification but still, it is an unneglectable challenge to design a classification model on unlabeled data. In this research, we are focusing to design and develop a normalized network based on restricted Boltzmann machine (RBM) [10] in addition with Polyak Averaging method [11] in training while updating parameters called NRBM. Since the learned features can be acquired economically at high scale yet comprises of temporal gestures because the images data contain temporally comprehensible frames.

In our conducted experiments, the proposed NRBM model has initiated the learning strategy from unlabeled data. We also showed that the proposed model can extract more useful features for final classification. Furthermore we conduct some experiments on MNIST variants benchmark datasets to validate our model. The results showed that the proposed NRBM model is able to learn the features that can be applied noisy and clear

images. Moreover, a promising guide model for extracting and learning features to classify in a better way is designed. Managing and classification of huge amount of images by training the proposed NRBM model using unsupervised manner on the basis of low level features, is the main contribution of our research in this paper. The rest of paper is organized the following structure. Section 2 explores literature work on image classification models. Section 3 briefly describes the conventional restricted Boltzmann machine. Section 4 explains our proposed NRBM. In Section 5 we described our experiments to examine our proposed model and Section 6 presents the comparative analysis of NRBM with state-of-art models. Finally Section 7 concludes our research work and future directions. We believe our research will have a positive impact in many computer vision applications, e.g. Image recognition systems, video analyzers and image processing approaches.

II. RESTRICTED BOLTZMANN MACHINE

Boltzmann Machine (BM) falls under the category of Artificial Neural Network (ANN) based on probability distribution for machine learning. Restricted Boltzmann Machine (RBM) is one of the famous variants of standard BM which was first created by Geoff Hinton [12]. The main purpose of RBM is reducing high dimensional data into low dimensional feature space. As it is a probability based approach that is why it is stochastic and generative in nature. The internal architecture of the RBM is similar to other Neural Networks (NNs) having layers with neurons in each layer. But in RBM there are only two layers. The first layer is referred to the input layer of the network, while the second layer is the hidden layer that is output from the input layer. There is a neural connection between the neurons of input layer and hidden layer. In standard BM, there exist connections between the neurons of the same layer but in RBM, there is a restriction that none of the neurons in same layer can communicate with the neurons between them.

Fig. 1 shows the internal architecture of RBM, in hidden layer there are x hidden nodes, while in visual layer has y number of visible nodes. The hidden nodes are represented by H_i and $i1, 2, 3, \dots, x$. While visible nodes are represented by V_j , such that $j1, 2, 3, \dots, y$. The weights connection between hidden and visible layers is represented by W_{ij} . The final equation for energy of RBM is:

$$E_{rbm}(v, h|\phi) = - \sum_{i=1}^x v_i a_i - \sum_{j=1}^y h_j \times b_j - \sum_{i=1}^x \sum_{j=1}^y w_{ij} \times v_i \times h_j \quad (1)$$

In Equation 1, $\phi = (W_{ij}, a_i, b_j)$, all these are real numbers. The bias value of hidden layer is denoted by b_j and the b_i represents the bias value of visible layer. The likelihood function that is also called the joint probability distribution of visible and hidden layer is formulated in Equation 2, if ϕ is known.

$$P(v, h|\phi) = \frac{1}{Z} \exp(-E(v, h|\phi)) \quad (2)$$

The normalization parameter $Z = \sum_{(v, h)} \exp(-E(v, h|\phi))$. This visible and hidden layer's activation functions can be formatted in Equation 3 and Equation 4.

$$P(h_j, I|v) = \text{sigmoid}\left(\sum_j v_i \times w_{ij} + b_j\right) \quad (3)$$

$$P(v_i, I|v) = \text{sigmoid}\left(\sum_i h_j \times w_{ij} + a_i\right) \quad (4)$$

While $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$. Probability that the first j th hidden node is activated is presented by Equation 3 while the activation of first i th visible node is presented in Equation 4. Finding the best suitable parameter set that is possible for ϕ is the main purpose of RBM training. The input data can be better fit by the trained model. For calculating the best value of ϕ using maximum likelihood function, we use the likelihood function based in log by giving training sample T_r , such that $T_r = v1, v2, v3, \dots, vTr$. The value of ϕ is:

$$\phi_* = \text{argmax} L(\phi) = \text{argmax} \sum_{t_r=1}^{T_r} \log P(v^{tr}|\phi) \quad (5)$$

The step to step processing of RBM is given in the following Algorithm 1 and the conventional RBM architecture is shown in Fig. 1.

Pseudocode Conventional RBM

1: **Parameters initialization**

- Input feature set: $[x_1, x_2, x_3, \dots, x_n]$
- Weight Matrix: W
- Bias vector of hidden layer: b_j
- Bias vector of visible layer: b_i
- The state of visible unit is set as training vector

Repeat

2: **Visible units training**

- Compute $P(v, h|\phi)$ using Equation 2
- Apply $P(v_i, I|v)$ of Equation 4

3: **Hidden units training**

- Compute $P(v, h|\phi)$ using Equation 2
- Apply $P(h_j, I|v)$ of Equation 3
- Perform Gibbs sampling from hidden unit to visible unit using $h_{1i} \in [0, 1]$

4: **Update rule**

- Update weight W : $W = W + Wn$ while $Wn =$ Weight update factor
- Update Bias b_i : $b_i = b_i + b_n$ while $b_n =$ Bias update factor for visible units
- Update Bias b_j : $b_j = b_j + b_{jn}$ while $b_{jn} =$ Bias update factor for hidden units
- Update energy function using Equation 1

While (All Layers Trained)

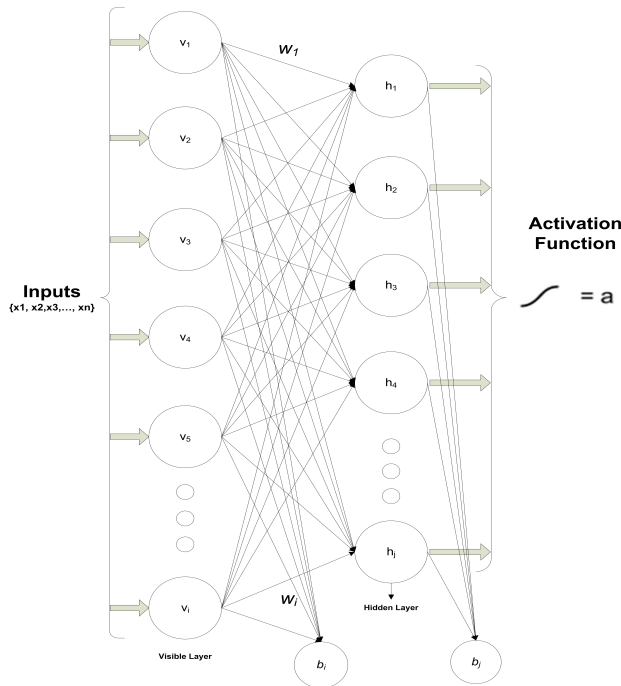


Fig. 1. Standard RBM internal architecture

III. RBM FOR SOLVING MULTICLASS CLASSIFICATION PROBLEMS

In the most recent research work RBM is widely used for different type of recognition, prediction and classification applications. For example [13] used RBM for face detection and recognition and get better results as compared to state-of-art methods. Particularly [14] conclude that confidently simple functions of stacked Restricted Boltzmann Machine (sRBM) can form a deep architecture for learning robust feature. Furthermore they trained convolutional Deep Boltzmann Machine (DBMs) [15] on faces image dataset (i.e., Caltech 101 images) and were impressive but unluckily their face detector can learn homogenous, aligned and a single category features.

In literature, BMs are been used more on prediction [16] and semantic based knowledge learning [17], [18] due the difficulty of generating output using low level representation, e.g. [19] split video frame sequence in chunks that describe actions in a discrete order. Author in [20] extract objects by removing background using the same manner in single domain boundary while [21] predict trajectories of human movement in a supervised way. Moreover there is also some found in order to model and forecast human to object [22] and human to human [23] connections based on supervised learning. In [24], authors have proposed Quantum Boltzmann Machine (QBM) inspired from the classical Boltzmann machines based one stochastic gradient descent in its training phase. They used a non-trivial training process for QBM and introduce bonds for the probabilities for quantum. The results have been verified by showing some examples that outperform conventional Boltzmann Machine based on training QBM with and without bounds using static diagonalizing.

In [25], the authors proposed a stochastic architecture based RBM digital images classification. They used four types of

stochastic bit streams in their experiments, valued 512, 1024, 2048, 4096 of bit streams in order to achieve the balance between error rate and computation time. Stochastic number generator has been used for conversion of deterministic input data values to stochastic data values stream. In [26], the authors evaluate the ML models, specifically RBM for Anomaly Network Intrusion Detection Systems (A-NIDS). They applied hyperparameter tuning in RBM and report a few observations on the accuracy along with true positive and true negative rate. A balanced subset of ISCX benchmark dataset was used for training, validation and testing the proposed RBM model. Resultantly, RBM was found to be the better network for identification of attacks pattern in A-NIDS.

Gou C. presented hybrid discriminative restricted Boltzmann machines (HDRBMs) in [27] for car license plate recognition. He claim that HDRBM is the first hybrid model based on RBM that used for license number detection and recognition in such a wide view. They used two data subsets with high resolution images from car number plate dataset. The conducted experiment evaluated that their proposed HDRBM based on RBM outperform many of the state-of-art model for number plate detection and recognition. The claim that their model can be used efficiently in other countries as well but the layout for characters must predefine. In [28], the author presents another model known as discriminative Restricted Boltzmann Machine (discRBM) for learning discriminative feature set based on class labels. However their results conclude that discRBM performed well as compared to conventional RBM, deep RBM and SVM based on classification accuracy but still their model is limited to dataset size and complexity.

IV. NORMALIZED RESTRICTED BOLTZMANN MACHINE

Our proposed Normalized Restricted Boltzmann Machine NRBM is inspired by some state-of-art techniques and algorithm that are based on deep learning and unsupervised feature learning. The main source of inspiration is Restricted Boltzmann Machine (RBM) that is one of the famous variant of basic Boltzmann Machine (BM). The proposed model train on class label-less images for the purpose of extracting some amount of temporal informative features contain in the huge number test images data. To achieve this goal we studied some variants of RBM [29], [30], [31] and other classification algorithms [32], [33], [34]. After deep literature study we extended the restricted Boltzmann machine (RBM) from [10]. We present a two-phase training model that can easily extract background and foreground of the image separately. This foreground and background separation helps the model to learn the object feature inside the image easily and fast.

The main focus of this research model is tackling the multiclass classification problem that is caused by the high dimensional features space of large amount of data. We add Polyak averaging method with RBM in order to increase the speed of parameter convergence rate. We did not use traditional feature reduction techniques because the traditional techniques do not fulfill the user requirements for better feature extraction. In this paper, we proposed a deep learning approach based on normalized RBM called NRBM. The reason of using RBM is that, RBM uses high order structured statistical feature in training phase. Also the low dimension features is easily distinguishable from the features learnt by NRBM.

Taking the advantage of RBM network, the proposed model efficiently reduces the feature dimension space of images which is resultantly fall in increasing the accuracy of multiclass classification. In training phase of NRBM, we add Polyak Averaging to the first layer of RBM in order to get the normalized parameter set for next layer. We repeat the same procedure for each iteration in the proposed model until final classification. Equation 6, 7, 8 are the Polyak Averaging formulas for parameter updates.

$$w^n = w^n - 1 + \eta w^n \quad (6)$$

$$\bar{w}^n = \bar{w}^n - 1 - \frac{1}{n}(\bar{w}^n - 1 - w^n) \quad (7)$$

$$w^n = \bar{w}^n \quad (8)$$

Where, η is the scalar learning rate and $\bar{w}^n = \frac{1}{n} \sum_{i=1}^x w^i$.

Polyak Averaging increase the parameter convergence rate and also it does not disrupt the calculation training cost of the model because it comprises of only two simple addition functions. The sudden incline or decline in difference between two neighboring parameters for each layer is abolished by using averaging method. The set of feature obtained in result of the NRBM is given to final layer for final classification. Softmax classifier is considered as the final layer in the proposed NRBM model. The detailed step to step pseudocode of NRBM is given in the following Algorithm 2 and the network architecture is shown in Fig. 2. In the proposed model W is calculated using Equation 6 while \bar{w}^n is solved using Equation 7.

V. IMAGE CLASSIFICATION STEPS USING NRBM

Image classification is the famous application of classification in which images are being categorized on the basis of contents and object it consists [35]. Image classification is one of the famous problems in the field of classification. Like other classification approaches, the whole image classification phenomenon is also carried out in a few phases. Following are the common steps of image classification.

A. Image Pre-Processing

Image preprocessing refers to the procedures of processing both the input and output of the image on a very low level of abstraction [36]. The main purpose of image preprocessing is to enhance the image representation in order to cover the unwanted distortion and improves the quality of informative features for advance image processing. The whole image preprocessing is carried in a few steps, first the image data loading, secondly is the image resizing in order to fix all the image in a single standard size. The third step refers to the noise removal from the images and forth one is segmentation for the purpose of separating foreground and background objects.

Pseudocode NRBM

1: Parameters initialization

- Input feature set: $[x_1, x_2, x_3, \dots, x_n]$
- Weight Matrix: W
- Bias vector of hidden layer: b_j
- Bias vector of visible layer: b_i
- The state of visible unit is set as training vector

Repeat

2: Visible units training

- Compute $P(v, h|\phi)$ using Equation 2
- Apply $P(v_i, I|v)$ of Equation 4

3: Hidden units training

- Compute $P(v, h|\phi)$ using Equation 2
- Apply $P(h_j, I|v)$ of Equation 3
- Perform Gibbs sampling from hidden unit to visible unit using $h_{1i} \in [0, 1]$

4: Update rule

- Update weight using Equation 6 where W is Weight update factor for visible units
- Update Bias b_i : $b_i = b_i + b_n$ while $b_n =$ Bias update factor for visible units
- Update weight using Equation 7 where \bar{w} is Weight update factor for hidden units
- Update Bias b_j : $b_j = b_j + b_j n$ while $b_j n =$ Bias update factor for hidden units
- Update energy function using Equation 1

While (All Layers Trained)

VI. FEATURES EXTRACTION

In the feature extraction phase, the image is divided into segments and then each segment is processed in order to convert the pixels to their equal numerical values [37]. From these multi-dimensional values, important features are extracted from each segment one by one. Finally all the important features of the whole image which holds the complete representation of the entire image are extracted. The extracted features are the input of feature reduction module. Better feature extraction affects the feature reduction module positively.

A. Features Reduction

In the feature reduction phase of image processing, the features dimension of the image data is reduced from high dimension feature space. The main purpose of features reduction is to reduce the processing speed because the processing of high amount data for a classifier is time consuming task [38]. The proposed model mainly concern in effective feature reduction that ultimately results in the better image classification.

B. Classification

The last main stage of image classification is the use of a classifier for the final classification of image data. In the area machine learning, the problem of recognizing to which of a set of groups a new observation fits, based on a training dataset having observations or samples whose group association is known, referred as classification [39]. Different types of classifiers are used for classification. In the recent research studies, deep learning techniques are vastly used specifically for image classification.

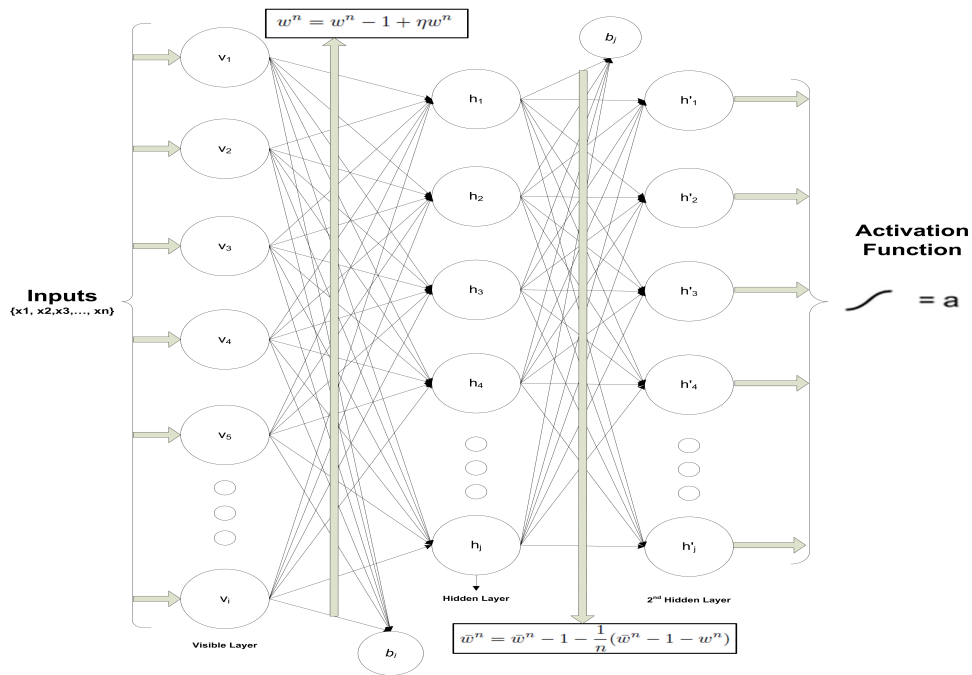


Fig. 2. NRBM internal architecture

The detailed factorial description of the proposed approach has been highlighted in Fig. 3.



Fig. 4. MNIST small subset (mnist-basic)



Fig. 5. MNIST random rotation digits (mnist-rot)



Fig. 6. MNIST random noise background digits (mnist-bg-rand)

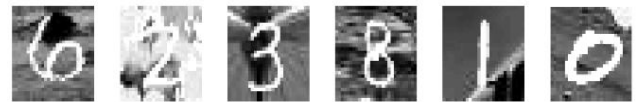


Fig. 7. MNIST random background digits (mnist-bg-img)



Fig. 8. MNIST rotation and image background digits (mnist-bg-img-rot)

VII. EXPERIMENTAL RESULTS

We performed several experiments to verify the proposed NRBM model and evaluate the feature learning ability, which is the main improvement towards better classification. In this research, all the experiments were carried out on Intel core i5 cpu with 8GB of RAM having windows 10 operating system. The compiler and language used for developing and testing these algorithms is python3.6. For quick implementation of the proposed NRBM approach, an efficient numeric computational open source library Tensorflow [40] is used. It allows a simple and fast development for both CPU and GPU support. The dataset used for evaluating the proposed model are the

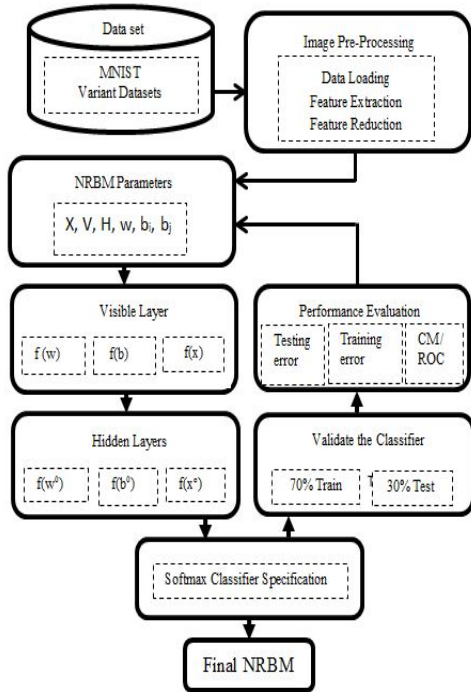


Fig. 3. Flow of image classification using NRBM

five variant subsets of benchmark data MNIST [41]. MNIST dataset contains 70,000 images of handwritten digits for 0 to 9. The size of each image is 28x28 pixels. In MNIST the digits have been centered and the size of each digit is fixed to specific area inside the image. The variation subsets of MNIST are basic (small), random rotation digits (rot), random noise background digits (bg-rand), random background digits (bg-img) and rotation and image background digits (bg-img-rot). Fig. 4 to Fig. 8 are the randomly selected images from each of the variant subset, respectively. We select 5000 random images from each MNIST subset. The experiments were performed with two different testing and training ratios in order to verify the performance of NRBM. In first phase of experiments we select 50% of data for training and 50% for testing while in second phase we increase the training data ratio from 50% to 70% and reduce the testing data ratio from 50% to 30% of the whole dataset.

In these experiments, backpropagation approach is used in learning phase. There is no pre-processing or pre-training performed. For the proposed NRBM, the number of input neurons is 784. Learning rate is set to 0.03 for the hidden layer and we adjust the threshold for reconfiguration error to 0.3 after performing experiments with different thresholds rates.

After the backpropagation parameter adjustment we reduce the reconfiguration error threshold to 0.003. The number of output nodes from the final layer is 10 and the fine count for backpropagation is finally set to 500. In this research, the criteria for classification evaluation performance are Confusion Matrix (CM) and Receiver operating characteristic (ROC) curve. CM and ROC curve presents detail results of NRBM accuracy of correctly classified and misclassified instances at class level.

In order to evaluate the proposed NRBM model and its classification performance, Softmax classifier is used in the last layer of the network. Fig. 9 presents results of NRBM for MNIST basic subset. Fig. 9(a) and 9(b) are representing CM and ROC curve for NRBM based on 70:30 training-testing dataset ratios while Fig. 9(c) and 9(d) for 50:50 of training-testing ratio. It is concluded easily that the results based on 70:30 ratio is are better than 50:50 training-testing ratio. Both of the CMs show that the classes with same pattern and similar features are mostly misclassified to each other e.g. class 4 and class 9, also 5 and 8. Fig. 10 represents the output of NRBM based on MNIST rot variant subset. The accuracy per class is similar to MNIST basic subset, but the overall accuracy is decreased until 5% because the images are randomly rotated in this dataset which resultantly increased the complexity of the datasets.

Fig. 11 briefly describe the output of the proposed model for MNIST bg-rand variant subset. The accuracy per class is better besides class 4 and class 2 that is misclassified to class 9 and class 7, respectively. This is because their features are quite similar. The ROC curve shows that there is a slowly paced decrease in the area under curve (AUC) for class 0 and 1 as compared to MNIST basic and rot.

Similarly, Fig. 12 shows the output of the NRBM model for MNIST bg-img variant subset. The overall accuracy is improved for 70:30 training-testing ratio data, but at level, there is present some misclassification i.e. class 4 and class 9 that is misclassified to class 9 and class 7. The quite similar features are the reason of misclassification. Even with naked eye, handwritten 4 and 9 looks quite similar. The ROC shows that the AUC for class 1 and class 9 is lower as compared to other classes. Fig. 13 represents the results on MNIST bg-img-rot dataset. The accuracy per class is improved which is verified by the ROC in Fig. 13(b) and 13(d). From analysis point of view for all the experiment concluded from Fig. 9 to 13, there is gradual decrease can be seen in the overall classification accuracy of NRBM starting for MNIST basic subset until MNIST random background digits. This decrease is the reason of gradual increase in the complexity and noisiness in the datasets.

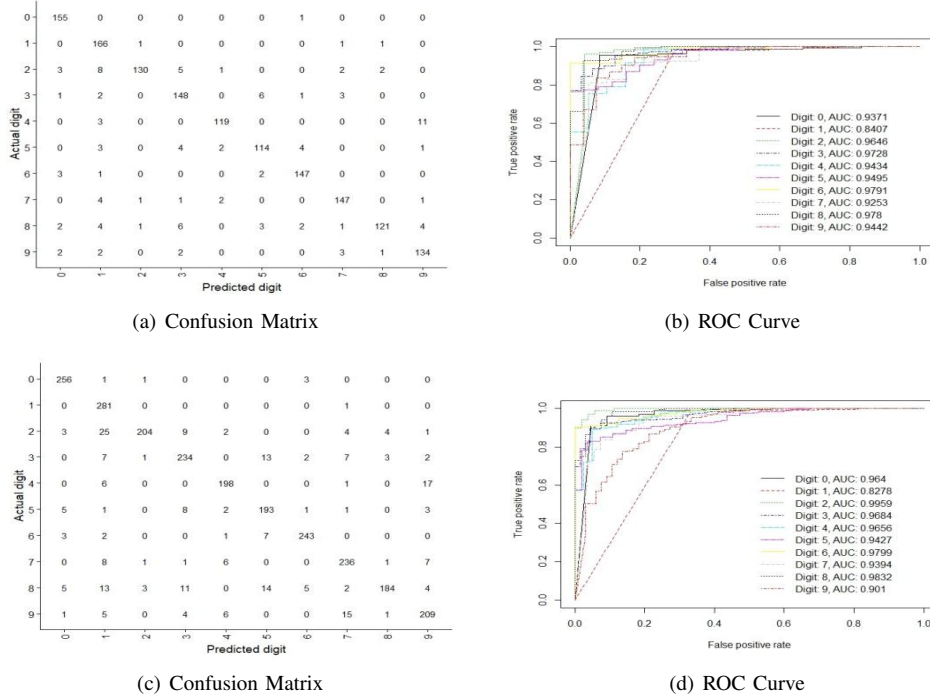


Fig. 9. CM and ROC curve for 70:30 and 50:50 of training-testing ratio on MNIST small subset (mnist-basic)

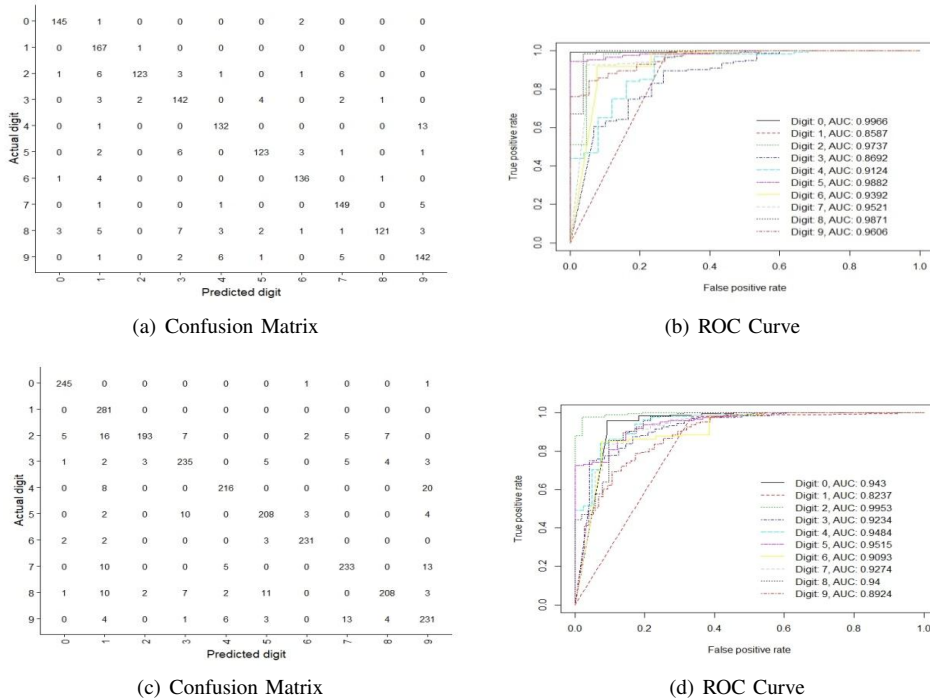
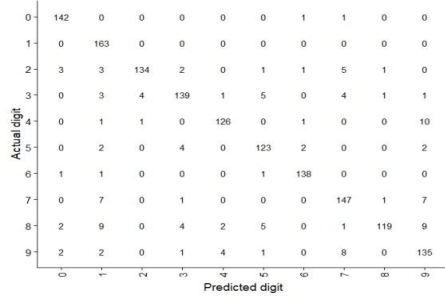
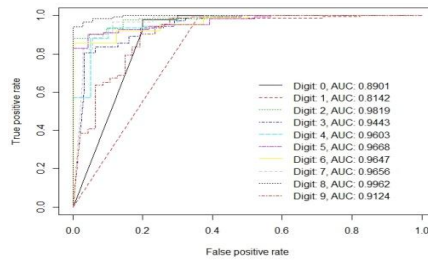


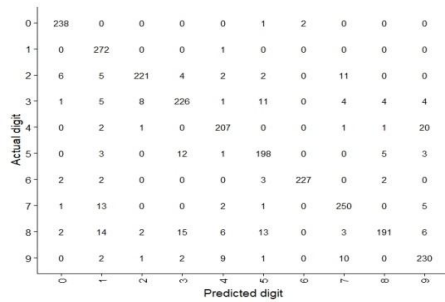
Fig. 10. CM and ROC curve for 70:30 and 50:50 of training-testing ratio on MNIST random rotation digits (mnist-rot)



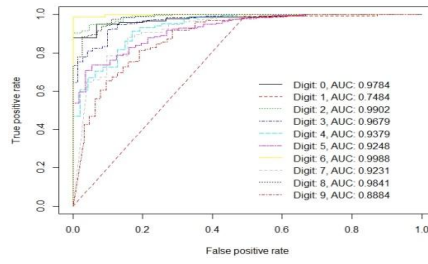
(a) Confusion Matrix



(b) ROC Curve

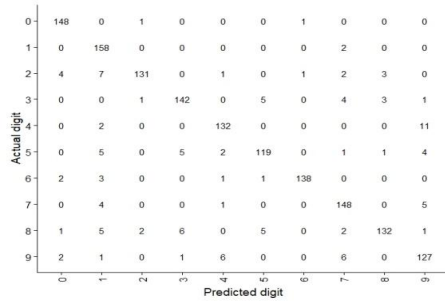


(c) Confusion Matrix

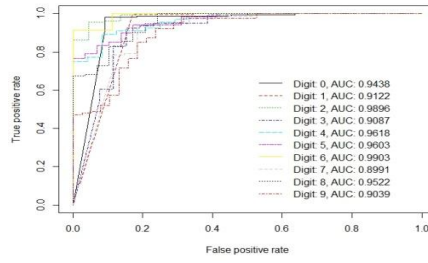


(d) ROC Curve

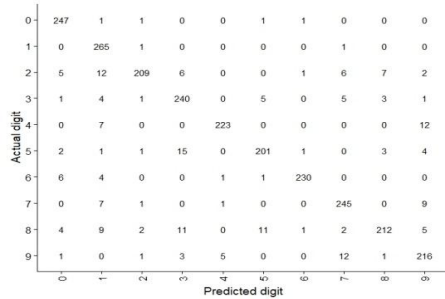
Fig. 11. CM and ROC curve for 70:30 and 50:50 of training-testing ratio on MNIST random noise background digits (mnist-bg-rand)



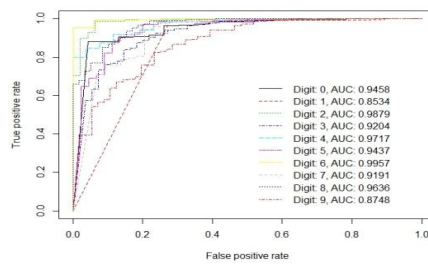
(a) Confusion Matrix



(b) ROC Curve



(c) Confusion Matrix



(d) ROC Curve

Fig. 12. CM and ROC curve for 70:30 and 50:50 of training-testing ratio on MNIST random background digits (mnist-bg-img)

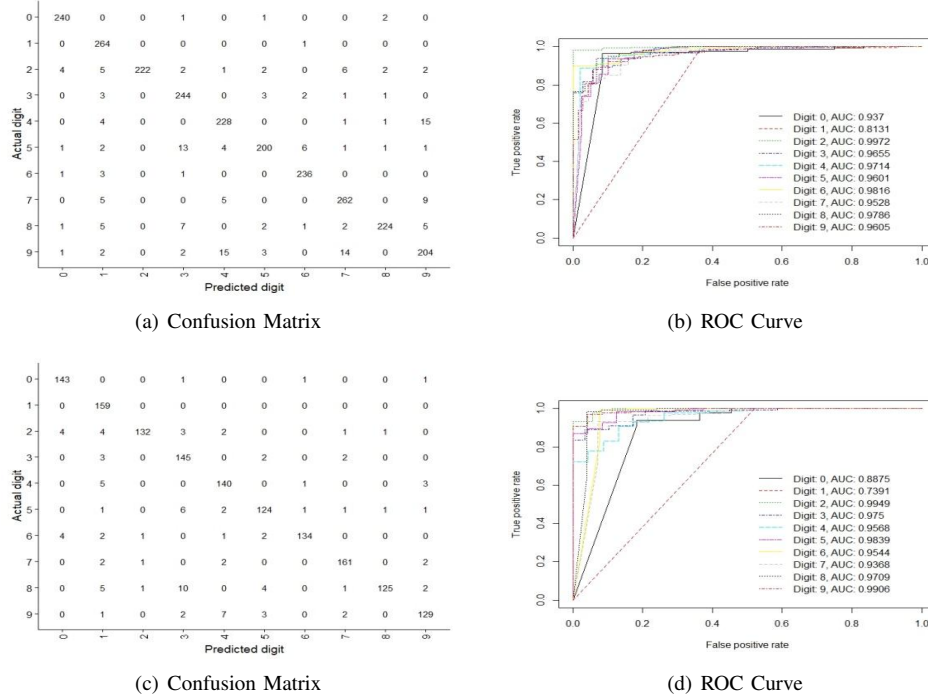


Fig. 13. CM and ROC curve for 70:30 and 50:50 of training-testing ratio on MNIST random background digits (mnist-bg-img-rot)

VIII. COMPARATIVE ANALYSIS

A. Execution Time Comparison

The computational performance of all algorithms depends on the hardware, software and compilers. In our research, we used the same hardware and software combination that is mentioned in SECTION 5. Results in Table I show the execution time comparison of the proposed NRBM model, Deep Belief Network (DBN) [42], Gated RBM (GRBM) [43], standard RBM [44] model. The overall performance of first two dataset between RBM GBM and NRBM is quite close. In MNIST basic, the execution time performance of RBM and GBM is better because the data is not complex relatively which causes feature extraction and feature reduction simpler and faster as compared to other datasets. As the data is more and noisy and complex in MNIST sub-bg-rand, sub-bg-img and sub-bg-img-rot, the execution time performance of NRBM is better as compared to DBN, GBM, and standard RBM.

B. Training Error Rate Comparison

The most crucial part in any AI model is in the training phase. In the conducted experiments, the same parameters tuning is been followed as mentioned in Section 5. 5000 random images from each dataset are selected for training the comparative models. The back propagation behavior is applied in training phase. Table II presents the training error rate comparative analysis for NRBM with some state-of-art models. In the concluded training error rates, firstly the DBN performed well for MNIST basic dataset. In sub-rot dataset, RBM is better for 70/30 ratio while NRBM proved to be better for 50/50 ratio of training-testing data. In the rest datasets experiments, RBM and NRBM are performed relatively close

to each other but in the overall results NRBM outperformed the comparative models.

C. Testing Error Rate Comparison

The experiments conducting for test error rate analysis are given in Table III. The classification accuracy and performance depends on the internal architecture of the model and the classifier used for final classification. In order to conduct fair experiments, we used the same parameter settings and followed architecture for all the models with Softmax layer for the final classification. Like training comparative analysis, we also test all the models with a two training-testing data ratio. The results in Table III conclude that the testing error of DBN is lower for MNIST basic subset and RBM for sub-rot subset because the dataset complexity is very low in these two subsets. Though the error rate of RBM is close to NRBM but still in sub-bg-rand, sub-bg-img and bg-img-rot dataset, the proposed NRBM outperformed DBN, GBM and RBM.

IX. CONCLUSION

RBM is one of the most usable models in the area of deep learning. This paper proposed a NRBM that is based on Polyak Averaging method in training, in order to improve the weights and bias update parameters function. This research is carried out with three main contributions. Firstly using NRBM to learn features and reduce feature dimension efficiently. Secondly we use Softmax classifier in the last layer of the proposed model for final classification. Lastly we conduct sufficient experiments for comparative analysis to verify the significance and effectiveness of the proposed NRBM. The normalization function in RBM makes the model faster in learning since Polyak averaging method is composed of simple addition

TABLE I. EXECUTION TIME COMPARISON OF DIFFERENT MODELS ON MNIST VARIANT-DATASETS

Datasets	Sub-Basic		Sub-Rot		Sub-Bg-rand		Sub-Bg-img		Sub-Bg-img-rot	
	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30
Training/Testing	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30
DBN	15m 55s	18m 23s	20m 26s	23m 42s	22m 56s	27m 02s	26m 36s	32m 43s	33m 55s	41m 24s
GBM	12m 32s	16m 46s	20m 50s	22m 28s	20m 10s	24m 05s	-	-	-	-
RBM	12m 15s	14m 52s	17m 44s	21m 36s	18m 26s	22m 12s	24m 57s	29m 03s	29m 45s	34m 28s
NRBM	14m 54s	15m 35s	17m 35s	19m 25s	19m 02s	21m 36s	21m 10s	23m 48s	25m 28s	28m 33s

TABLE II. TRAINING ERROR RATE COMPARISON OF DIFFERENT MODELS ON MNIST VARIANT-DATASETS

Datasets	Sub-Basic		Sub-Rot		Sub-Bg-rand		Sub-Bg-img		Sub-Bg-img-rot	
	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30
Training/Testing	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30
DBN	2.96	3.63	7.68	10.25	11.53	15.85	17.52	21.12	23.40	28.65
GBM	8.78	9.31	13.85	19.12	19.95	23.45	-	-	-	-
RBM	6.64	7.20	7.94	8.98	10.05	10.82	14.94	17.87	28.32	36.52
NRBM	3.10	3.78	6.05	9.06	9.20	10.10	15.10	17.02	21.60	24.85

TABLE III. TESTING ERROR RATE COMPARISON OF DIFFERENT MODELS ON MNIST VARIANT-DATASETS

Datasets	Sub-Basic		Sub-Rot		Sub-Bg-rand		Sub-Bg-img		Sub-Bg-img-rot	
	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30
Training/Testing	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30
DBN	9.58	7.43	14.96	12.10	18.14	22.74	21.87	23.70	27.25	33.85
GBM	15.54	13.31	18.10	19.12	24.68	26.34	-	-	-	-
RBM	14.78	10.08	14.17	11.19	15.85	17.70	18.63	22.12	42.32	38.12
NRBM	12.96	9.45	16.32	13.40	15.21	16.50	18.92	20.78	24.48	29.86

function. Python based efficient computational library named Tensorflow, is used for implementation. The experiments are carried out on five MNIST variant datasets. As a conclusion the overall result revealed that the proposed model performed better when the complexity of dataset increased. Our future goals is to move forward our research by incorporating more optimization and normalization methods to the standard RBM and also applying the proposed model to more complex and big datasets.

ACKNOWLEDGMENT

The authors would like to thank Ministry of Higher Education (MOHE) Malaysia and Universiti Tun Hussein Onn Malaysia (UTHM) for financially supporting this Research under Trans-disciplinary Research Grant Scheme (TRGS) vote no. T003.

REFERENCES

[1] R. Kune, P. K. Konugurthi, A. Agarwal, R. R. Chillarige, and R. Buyya, "The anatomy of big data computing," *Software: Practice and Experience*, vol. 46, no. 1, pp. 79–105, 2016.

[2] Y. Sun, M. Zhang, Z. Sun, and T. Tan, "Demographic analysis from biometric data: Achievements, challenges, and new frontiers," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 2, pp. 332–351, 2018.

[3] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," *arXiv preprint arXiv:1606.01781*, 2016.

[4] M. Liu, D. Zhang, S. Chen, and H. Xue, "Joint binary classifier learning for ecoc-based multi-class classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2335–2341, 2016.

[5] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[6] J. A. Nasiri, N. M. Charkari, and S. Jalili, "Least squares twin multi-class classification support vector machine," *Pattern Recognition*, vol. 48, no. 3, pp. 984–992, 2015.

[7] P. Xu, F. Davoine, H. Zha, and T. Denoeux, "Evidential calibration of binary svm classifiers," *International Journal of Approximate Reasoning*, vol. 72, pp. 55–70, 2016.

[8] E. Vissol-Gaudin, A. Kotsialos, C. Groves, C. Pearson, D. A. Zeze, and M. C. Petty, "Computing based on material training: Application to binary classification problems," in *2017 IEEE International Conference on Rebooting Computing (ICRC)*. IEEE, 2017, pp. 1–8.

[9] A. Sen, M. M. Islam, K. Murase, and X. Yao, "Binarization with boosting and oversampling for multiclass classification," *IEEE transactions on cybernetics*, vol. 46, no. 5, pp. 1078–1091, 2016.

- [10] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 791–798.
- [11] W. Sun, D. H. Poot, X. Yang, W. J. Niessen, and S. Klein, "Averaged stochastic optimization for medical image registration based on variance reduction," in *International Workshop on Biomedical Image Registration*. Springer, 2018, pp. 69–79.
- [12] G. E. Hinton, "A practical guide to training restricted boltzmann machines," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 599–619.
- [13] T. D. Nguyen, T. Tran, D. Phung, and S. Venkatesh, "Tensor-variate restricted boltzmann machines," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
- [14] J. Chu, H. Wang, H. Meng, P. Jin, and T. Li, "Restricted boltzmann machines with gaussian visible units guided by pairwise constraints," *IEEE transactions on cybernetics*, no. 99, pp. 1–14, 2018.
- [15] Z. Gan, R. Henaou, D. Carlson, and L. Carin, "Learning deep sigmoid belief networks with data augmentation," in *Artificial Intelligence and Statistics*, 2015, pp. 268–276.
- [16] R. Hrasko, A. G. Pacheco, and R. A. Krohling, "Time series prediction using restricted boltzmann machines and backpropagation," *Procedia Computer Science*, vol. 55, pp. 990–999, 2015.
- [17] S. Tsogkas, I. Kokkinos, G. Papandreou, and A. Vedaldi, "Semantic part segmentation with deep learning," *arXiv preprint arXiv:1505.02438*, vol. 3, no. 7, 2015.
- [18] N. Kaur, G. Kunapuli, T. Khot, K. Kersting, W. Cohen, and S. Natarajan, "Relational restricted boltzmann machines: A probabilistic logic learning approach," in *International Conference on Inductive Logic Programming*. Springer, 2017, pp. 94–111.
- [19] H. Vu, T. D. Nguyen, and D. Phung, "Detection of unknown anomalies in streaming videos with generative energy-based boltzmann models," *arXiv preprint arXiv:1805.01090*, 2018.
- [20] S. Lee and D. Kim, "Background subtraction using the factored 3-way restricted boltzmann machines," *arXiv preprint arXiv:1802.01522*, 2018.
- [21] M. Raj, V. B. Semwal, and G. Nandi, "Bidirectional association of joint angle trajectories for humanoid locomotion: the restricted boltzmann machine approach," *Neural Computing and Applications*, vol. 30, no. 6, pp. 1747–1755, 2018.
- [22] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4346–4354.
- [23] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom, "Deep representation learning for human motion prediction and classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6158–6166.
- [24] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchitsky, and R. Melko, "Quantum boltzmann machine," *Physical Review X*, vol. 8, no. 2, p. 021050, 2018.
- [25] B. Li, M. H. Najafi, and D. J. Lilja, "Using stochastic computing to reduce the hardware requirements for a restricted boltzmann machine classifier," in *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2016, pp. 36–41.
- [26] T. Aldwairi, D. Perera, and M. A. Novotny, "An evaluation of the performance of restricted boltzmann machines as a model for anomaly network intrusion detection," *Computer Networks*, vol. 144, pp. 111–119, 2018.
- [27] C. Gou, K. Wang, Y. Yao, and Z. Li, "Vehicle license plate recognition based on extremal regions and restricted boltzmann machines," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1096–1107, 2016.
- [28] H. Larochelle and Y. Bengio, "Classification using discriminative restricted boltzmann machines," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 536–543.
- [29] S. Nie, Z. Wang, and Q. Ji, "A generative restricted boltzmann machine based method for high-dimensional motion data modeling," *Computer Vision and Image Understanding*, vol. 136, pp. 14–22, 2015.
- [30] K. Cho, A. Ilin, and T. Raiko, "Improved learning of gaussian-bernoulli restricted boltzmann machines," in *International conference on artificial neural networks*. Springer, 2011, pp. 10–17.
- [31] V. Mnih, H. Larochelle, and G. E. Hinton, "Conditional restricted boltzmann machines for structured output prediction," *arXiv preprint arXiv:1202.3748*, 2012.
- [32] Z. Zhang, Zhang, and Khelifi, *Multivariate Time Series Analysis in Climate and Environmental Research*. Springer, 2018.
- [33] J. Tanha, M. van Someren, and H. Afsarmanesh, "Semi-supervised self-training for decision tree classifiers," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 1, pp. 355–370, 2017.
- [34] L. Gao, J. Li, M. Khodadadzadeh, A. Plaza, B. Zhang, Z. He, and H. Yan, "Subspace-based support vector machines for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 2, pp. 349–353, 2015.
- [35] P. Druzhkov and V. Kustikova, "A survey of deep learning methods and software tools for image classification and object detection," *Pattern Recognition and Image Analysis*, vol. 26, no. 1, pp. 9–15, 2016.
- [36] F. Albu and A. Zamfir, "Image processing method and apparatus," Mar. 24 2015, uS Patent 8,989,516.
- [37] Z. Fan, D. Bi, L. He, M. Shiping, S. Gao, and C. Li, "Low-level structure feature extraction for image processing via stacked sparse denoising autoencoder," *Neurocomputing*, vol. 243, pp. 12–20, 2017.
- [38] S. De, M. Singha, K. Kumari, R. Selot, and A. Gupta, "Dimension reduction using image transform for content-based feature extraction," in *Feature Dimension Reduction for Content-Based Image Identification*. IGI Global, 2018, pp. 26–40.
- [39] L. Breiman, *Classification and regression trees*. Routledge, 2017.
- [40] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.
- [41] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 473–480.
- [42] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.
- [43] X. Jin, T. He, C. Wan, L. Yi, G. Ding, and D. Shen, "Automatic gating of attributes in deep structure," in *IJCAI*, 2018, pp. 2305–2311.
- [44] M.-A. Côté and H. Larochelle, "An infinite restricted boltzmann machine," *Neural computation*, vol. 28, no. 7, pp. 1265–1288, 2016.