



Hybrid Contractive Auto-encoder with Restricted Boltzmann Machine For Multiclass Classification

Muhammad Aamir^{1,2} · Nazri Mohd Nawi^{1,3} · Fazli Wahid⁴ · Muhammad Sadiq Hasan Zada² · M. Z. Rehman¹ · Muhammad Zulqarnain¹

Received: 19 April 2020 / Accepted: 13 April 2021 / Published online: 23 June 2021
© King Fahd University of Petroleum & Minerals 2021

Abstract

Contractive auto-encoder (CAE) is a type of auto-encoders and a deep learning algorithm that is based on multilayer training approach. It is considered as one of the most powerful, efficient and robust classification techniques, more specifically feature reduction. The problem independence, easy implementation and intelligence of solving sophisticated problems make it distinct from other deep learning approaches. However, CAE fails in data dimensionality reduction that cause difficulty to capture the useful information within the features space. In order to resolve the issues of CAE, restricted Boltzmann machine (RBM) layers have been integrated with CAE to enhance the dimensionality reduction and a randomized factor for hidden layer parameters. The proposed model has been evaluated on four benchmark variant datasets of MNIST. The results have been compared with four well-known multiclass class classification approaches including standard CAE, RBM, AlexNet and artificial neural network. A considerable amount of improvement has been observed in the performance of proposed model as compared to other classification techniques. The proposed CAE–RBM showed an improvement of 2–4% on MNIST(basic), 9–12% for MNIST(rot), 7–12% for MNIST(bg-rand) and 7–10% for MNIST(bg-img) dataset in term of final accuracy.

Keywords Contractive auto-encoder · Restricted Boltzmann machine · Classification · Mnist variants

✉ Muhammad Aamir
aamir@uthm.edu.my; m.aamir@derby.ac.uk

✉ Fazli Wahid
fazli.wahid@uoh.edu.pk

Nazri Mohd Nawi
nazri@uthm.edu.my

Muhammad Sadiq Hasan Zada
m.hassanzada@derby.ac.uk

M. Z. Rehman
syedzubair@uthm.edu.my

Muhammad Zulqarnain
zulqarnainmalik321@gmail.com

¹ Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn, Parit Raja, Malaysia

² School of Electronics, Computing and Mathematics, University of Derby, Derby, UK

³ Soft Computing and Data Mining Center, Universiti Tun Hussein Onn, Parit Raja, Malaysia

⁴ Department of Information Technology, The University of Haripur, Haripur, Khyber Pakhtunkhwa, Pakistan

Abbreviations

AE	Auto-encoder
RBM	Restricted Boltzmann machine
CAE	Contractive auto-encoder
CAE–RBM	Hybrid contractive auto-encoder–restricted Boltzmann machine
ANN	Artificial neural network
SVM	Support vector machine
kNN	k-Nearest neighbor
CNN	Convolution neural network
DL	Deep learning
ML	Machine learning
ROC	Receiver operating characteristic
CM	Confusion matrix
MNIST	Modified National Institute of Standards and Technology (database)
rot	MNIST random rotation digits
bg-rand	Random noise background digits
bg-img	Random background digits



1 Introduction

In the field of data mining, classification algorithms are skilled for processing a huge amount of data. These algorithms are used to predict categorical classes for the test data based on training dataset and also used for classifying the new data instances. The field of classification covers many contexts in which a few decisions or forecasts are made on the basis of currently accessible information. The procedure of Classification is acknowledged as a method of frequently making conclusions in novel situations. Here, the assumptions are made for building a procedure that is applied to a continuous sequence of instances and in which every new instance is assigned to one of the groups of pre-defined classes based on observed features of existing data instances. A type of classification procedure in which the exact classes labels are known in training phase is the so-called supervised learning or pattern recognition [32]. The famous AI techniques for solving classification problem are: support vector machine (SVM) [30], k-nearest neighbor (kNN) [56] and artificial neural network (ANN) [39]. Furthermore, a group of techniques known as deep learning (DL) algorithms has been introduced recently for solving these sophisticated problems.

Deep learning is a subdomain of machine learning that learns high-level abstraction present in the data by using hierarchical learning. Deep learning algorithms are the advanced versions of traditional neural networks which possess some serious limitations. In traditional neural networks, if there are more layers and units, there will be a higher expressional power of the network which leads to more complexity of cost functions. The network learning will be difficult and cause overfitting. Overfitting can be reduced using several methods, for example keeping the network size smaller, maximizing the training part of the dataset and some other techniques. In order to overcome the limitations associated with traditional neural networks, deep learning algorithms have been introduced. It is an advanced approach and has been utilized in family of applications for example transfer learning [26], semantic parsing [44], computer vision [34], natural language processing [53] and many other complex applications. The reasons behind the usage of deep learning algorithms are the low cost of computing hardware, powerful processing capabilities and high level of advancement in the machine learning techniques. There are three famous DL algorithms in the most recent research, namely convolution neural network (CNN), restricted Boltzmann machine (RBM) and auto-encoder (AE). All of these deep learning algorithms have further many variants suitable for different types of applications.

Auto-encoders (AEs) are a kind of artificial neural network consist of three layers. These layers are input layer, hidden layer and output layer, which are used for learning of efficient encoding [50]. The AE reconstructs its own

inputs instead of predicting outputs from the inputs. In auto-encoder, the output vector has the same dimension as the inputs. During the reduction process in auto-encoder, the purpose is to minimize the reconstruction error and learned features are actually the code generated by the encoder [4]. A single layer cannot extract informative features from the raw data. Therefore, in recent studies on AEs, researchers have used multiple layers to extract the most useful features from the raw data. AE have many variants, e.g., sparse auto-encoder [51], denoising auto-encoder [11], Saturating auto-encoder [14], convolutional auto-encoder [20], zero-bias auto-encoder [21] and contractive auto-encoder [13].

Boltzmann machine (BM) falls under the category of deep learning models that is 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 [16]. Restricted Boltzmann machine (RBM) [23] is one of the famous variants of standard BM. The main purpose of RBM is reducing high-dimensional data into low-dimensional feature space. Since it is a probability-based approach, it is stochastic and generative in nature [3]. The internal architecture of the RBM is similar to other neural networks (NNs) having layers with neurons in each layer. Except that 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 [6]. 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.

In this paper, we integrate the RBM layers with the CAE in order to improve the dimensionality reduction capability of CAE more efficiently. This integration of RBM layers with CAE creates a learning approach with the two major properties including robustness toward small noise or changes in the input features vector and the capability of learning high-order statistical feature from high-dimensional features space. The proposed CAE–RBM has the capabilities of both CAE and RBM which proves the significance of the proposed model. However, this paper has three main research contributions:

1. This paper gives a brief introduction, literature study and applications of CAE and RBM in solving the multiclass classification problems in different domain.
2. This paper proposed CAE–RBM approach which improves the dimensionality reduction ability of conventional CAE and enhancing the learning of high-order statistical features by integrating the RBM layers during feature reduction phase.
3. This paper conducted an extensive experiments on the proposed CAE–RBM model, in addition with state-of-

the-art multiclass classification algorithms on benchmark handwritten digit images dataset.

The rest of the paper is organized as follows: Sect. 2 shows related work; Sect. 3 presents the conventional contractive auto-encoder; Sect. 4 shows the proposed CAE–RBM approach; Sect. 5 shows experimental results; Sect. 6 shows the comparative analysis of results; and Sect. 7 presents the conclusion and future work.

2 Related Work

In the field of multiclass classification, supervised behavior learning algorithms are aimed to assign a designated class to every input instance [43]. For an input 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 testing instances. Many models and algorithms have been proposed in the last few decades for solving the binary class classification problems, most of them are easy to extend for multiclass classification problem, while some of them use special formulation to solve multiclass classification problem [8,31]. Solving the multiclass classification problem is challenging [24]. In state-of-the-art classification approaches, the multiclass classification problem is solved using the splitting of problem into many independent binary classification sub-problems. Image classification is the famous application of classification in which images are being categorized on the basis of contents and object it consists [12]. Image classification is one of the challenging problems in the field of classification and decision making [1,41].

In the recent literature studies, generative adversarial networks (GANs) has been widely used in well-known image generation applications including image super-resolution [19,55], image-to-image translation [45] and text-to-image transformation [47]. The authors in [54] have proposed the self-attention GAN and named it SAGAN. Their proposed SAGAN has auto-attention mechanism for generating convolutions in convolutional GANs. This auto-attention helps the network in modeling the multilevel dependencies for long-range image regions the results in drawing the images with finer details in distant portion of image. Additionally, its discriminator can enforce complex geometric constraints more accurately on the image structure globally. In [29], the authors presented the idea of adding least square methods in GANs and formed an enhance network called LSGANs. The proved by their experiments that reducing the cost of objective function can ultimately yields the minimization of Pearson divergence. The tested their proposed LSGAN using LSUN and CIFAR-10 benchmark datasets based on the train-

ing stability and performance. They also applied it to Chinese handwritten characters datasets containing 3740 classes.

Moreover, the recent literature gives a new non-iterative approach for multiclass classification called a neural-like structure of SGTm (successive geometric transformation model). The SGTm neural-like structure of models is more effective on the processing time of large datasets in classification task [48]. However, the processing time can be dramatically reduced by the usage of non-iterative greedy-based training approach in high-speed SGTm neural-like structure. All the mathematical and statistical proof and description of the training procedure and formulation of activation functions along with its computation intelligence are presented in [17]. In [49], the authors have used the SGTm neural-like structure in a hybrid with Kolmogorov–Gabor polynomial for solving the regression problems on large-sized datasets. According to the deep literature study, [48] has provided enough evidence to prove the effectiveness and significance of the non-iterative SGTm neural-like structure for solving different types of regression, classification and prediction problems.

As the study of image processing and classification becomes vaster in the last few decades, the performance of many deep learning (DL) approaches is promising. One of the reasons of DL-based approaches is its subtype called convolutional neural network (CNN) specifically for solving image classification problems [15]. A typical CNN model is a feed-forward neural network that consists of different layers including convolutional layers, pooling layers and fully connected layers, which emphasis on multidimensional correlation by connecting the neuron of each adjacent layer. When the neuron of one layer is connected to all the neurons on the proceeding layer, the architecture is called deep CNN (DCNN) [40]. DCNN captures the quick interest of researchers for the solution of image classification and processing problem. In [22], the authors used a variant of DCNN called ImageNet for large scale visual recognition challenge in 2012. The winners of that challenge with new record breaking results, used DCNN for classification of approximately 1.2 million images in 1000 classes. Since then, the subsequent variants of DCNN got and interest of most researchers for image classification.

The authors in [33] proposed a dimensionality reduction approach for image classification. The proposed cross-attention mechanism and graph convolution integration algorithm boosts hyperspectral data classification efficiency. PCA is used to reduce the dimensions of hyperspectral images to achieve more expressive low-dimensional features. The model employs a cross-attention strategy to jointly assign weights based on its two strategies and then employs a graph convolution algorithm to establish directional relationships between the features. Deep features, as well as the relationships between them, are used to predict hyper-

spectral data. Three hyperspectral datasets show that the proposed algorithm outperforms other algorithms using various training set division methods. Additionally, [9] presented a new dimensionality reduction method for face recognition called Biomimetic uncorrelated locality discriminant projection (BULDP). It is based on unsupervised discriminant projection and two human bionic characteristics: homology continuity and heterogeneous similarity principles. A new representation has been proposed based on these two human bionic features that can collect category information between different samples and represents consistency between similar samples and the similarity between different samples. Besides, it also can transform the original space into an uncorrelated discriminant subspace using the new unsupervised discriminant projection [35]. Singular value decomposition provides a detailed solution to the BULDP. A nonlinear version is developed using kernel functions for BULDP that minimizes nonlinear dimensionality. Four publicly accessible face recognition benchmarks are compared to state-of-the-art methods. The experimental results prove that the BULDP achieves competitive recognition efficiency performance.

In the most recent literature, enormous number of researchers are working the improvements of image classification. From a deep overview of the existing research, it has been concluded that proper dimensionality reduction is one of the most important and compulsory steps in image classification [36]. Though the deep learning approaches are considered as one of the better approaches for solving a number of multiclass classification problems, still there are few weaknesses related to deep learning that encourages researchers to overcome these weaknesses. The first weakness related to these approaches is the failure in application of independent data dimensionality reduction based on preferences. This failure is the inability of model to capture the useful information from the data which defects the quality solution of the classification [7]. Therefore, to overcome the dimensionality reduction problem associated with these models, many researchers have improved and hybridized the standard version of CAE for solving multiclass classification problem. [25] and [27] proposed the stacked versions of CAE, in order to increase the dimensionality reduction capability of CAE. They overcome this issue to some extent according to their research requirements, but it led to an increase in the complexity of their proposed model.

3 Contractive Auto-encoder

[38] firstly proposed the idea of contractive auto-encoder (CAE). CAE is a variant of AE family that aims to show robustness and senseless toward minor variation in the training datasets during encoding process. CAE is based on an

additive penalty term called “Frobenius norm of the Jacobian matrix,” which is used to reduce the cost of objective function. The result of this modification to AE is to minimize learned representation with respect to the input which increases the robustness of CAE toward the training data sample [2]. CAE is commonly used as the same as other variants of AE nodes, particularly when there exist some noise in the input data where usually other encoding algorithms fail to classify the data points. The aim of denoising auto-encoder (DAE) and CAE is used to bring robustness in the model, but the working mechanism is different for both [10,52]. DAE injects noise in the data to bring robustness in the model, whereas the CAE adds analytic contractive penalty to error reconstruction function [25,27].

Although CAE is considered as one of the most powerful approaches used for solving various types of multiclass classification problems, some drawbacks are associated with this algorithm which needs proper attention to develop a technique that leads to a problem-independent and high-quality solution generation for solving these complex problems. Same as other classification approaches, CAE perform the classification task in three major stages, namely features extraction, feature reduction and classification. The major drawback associated with standard CAE is its failure in application independent data dimensionality reduction according to user satisfaction. The result of this failure is the incapability of the CAE model to capture the finer details possessing the useful information. Resultantly, it leads to low-quality solution of the classification problem. The work carried out in this research has targeted the issue associated with conventional CAE while solving multiclass classification problem in order to enhance its performance.

The whole processing inside CAE takes place in two parts, namely encoding and decoding.

3.1 Encoding

The process of mapping the input feature set to transform it to give as intermediate representation to the hidden layer is called encoding given by Eq. 1:

$$y = f(x) = s_e(w_x + b_h) \quad (1)$$

where $f(x)$ represents the outputs of the input layer which is given as inputs to the hidden layer, w represents weights given to each input and b represents the biasness value associated with input feature set.

3.2 Decoding

The process of mapping the output of the hidden layer back into the input feature set is called decoding and is given by Eq. 2:

$$r = g(y) = s_d(w_y + b_r) \tag{2}$$

where $g(y)$ represents the output of the hidden layer, w represents the weights given to the inputs of the hidden layer and b represents biasness of the inputs to the hidden layer. In both encoding and decoding functions, weights w and biasness b values are generated randomly based on the number of nodes in that layer.

The s_e and s_d are the encoding and decoding activation functions, respectively, and are given by Eq. 3 for nonlinear representation (sigmoid function), whereas by Eq. 4 for linear representation (hyperbolic tangent function):

$$\text{sigmoid}(x) = \frac{1}{(1 + e^{-x})} \tag{3}$$

$$\text{tanh}(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \tag{4}$$

The major aim of the reconstruction is to generate the outputs as much similar to the original inputs as possible by reducing the reconstruction error. The following parameter set is used to reconstruct the original inputs by reconstruction layer:

$$\theta = [W, b_h, b_r] \tag{5}$$

Suppose, we have the input feature set as $D_i = [x_1, x_2, x_3, \dots, x_n]$, then the reconstruction error is minimized by minimizing the following cost function presented in Eq. 6:

$$J_{AE}(\theta) = \sum_{x \in D_i} R(x, r) \tag{6}$$

where R is reconstruction error. In case of linear representation, it is the Euclidean distance, whereas in case of nonlinear representation, it is the cross-entropy loss. In order to avoid overfitting and penalizing the large weights rose from Eq. 6, the simplest form of Eq. 6 is Eq. 7:

$$J_{AE} - w_d(\theta) = \sum_{x \in D_i} R(x, r) + \frac{1}{2}(\lambda \|W\|) \frac{2}{2} \tag{7}$$

in which the relative importance of regularization is controlled by weight coefficient decay λ . From the inspiration of learning robust feature set, in [38] the authors proposed a CAE with an unconventional regularization yielding objective function presented in Eq. 8 based on Eqs. 6 and 7.

$$J_{CAE}(\theta) = \sum_{x \in D_i} R(x, r) + \frac{1}{2}(\lambda \|f(x)\|) \frac{2}{F} \tag{8}$$

In Eq. 8, the Jacobian matrix is $f(x) = \partial f(x)/\partial x$ of the encoder f at value x . The mapping of feature set in order to be contractive in local domain of training data is encouraged by

adding the penalty of Frobenius norm of encoder Jacobian, for example the intermediary representation of features that are robust to the minor variations or noise in the input data.

In [37], the authors additionally deliver an experimental proof that the trade-off between the CAEs regularization term and the reconstruction error which produce a representation that captures the local representation of variation verbalized by the data. It is frequently corresponding to a low-dimensional non-directed manifold, on the other hand, being additional invariant to the enormous majority of directions orthogonal to the manifold. The contractive additive term for a sigmoid encoder is easy to calculate.

$$J(x) = f(x)_j(1 - f(x)_j)W_j \tag{9}$$

$$\|J(x)\|_F^2 = \sum_{j=1}^{d_h} (f(x)_j(1 - f(x)_j))^2 \|W_j\|^2 \tag{10}$$

The computational complexity by using Eq. 10 is same as the computing cost of a linear reconstruction error.

$$R(x, r) = \|x - b_r - \sum_{i=1}^{d_h} f(x)_i W_i\|^2 \tag{11}$$

For example, the squared error is equal then calculating the gradient update and objective in CAE is just a double expensive as compared to and conventional AE. While the overall complexity of both is approximately equal, that is equal to $O(d_h d_x)$. The step-by-step working mechanism and architecture of conventional CAE are presented in Algorithm 1 and Fig. 1, respectively.

Algorithm 1 Conventional CAE

Parameters initialization

- No of hidden layers: h
- Input feature set: $[x_1, x_2, x_3, \dots, x_n]$
- Encoding activation function: EAF
- Decoding activation function: DAF
- Inputs weights: w_i
- Biasness values: b_i

Encoding

- Compute encoded inputs $f(w)$ by Multiplying x_n and w_i

Decoding

- Compute decoded outputs $f(w^o)$ by Multiplying x_n and w_i

Optimization

- Optimize value of equation 8
-

4 The Contractive Auto-encoder with Restricted Boltzmann Machine (CAE-RBM)

The proposed CAE-RBM model focuses on the solution of first problem associated with the standard CAE that relates to

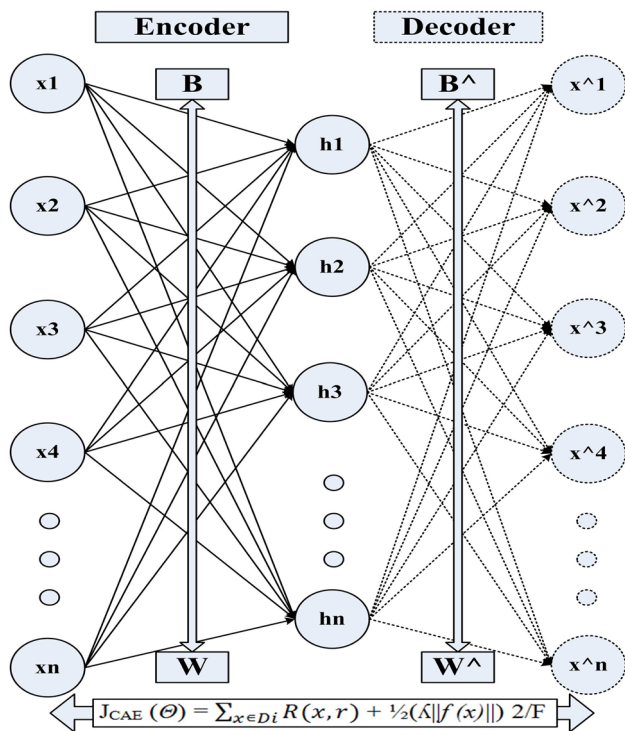


Fig. 1 Standard CAE internal architecture

the dimensionality reduction. We named the model as CAE–RBM because the enhancement has been made based on the addition of restricted Boltzmann machine hidden layers. It maps high-dimensional input data to a lower-dimensional representation, in which the encoder used to compress the input, whereas the decoder refers to the reconstruction of original input from the lower-dimensional feature representation. A cross-entropy loss function quantifies the information loss derived from the deviation between the original input and the reconstructed output. The goal of the training is to minimize the information loss of the reconstruction. Because target labels for the reconstruction are generated from the input data, the CAE–RBM is observed as self-supervised. In this model, RBM layers are added with CAE in order to enhance the dimensionality reduction property of CAE for solving multiclass classification problem efficiently. The addition of RBM layers with standard CAE designs a learning approach with the two major properties: a) Firstly, it is robust toward small noise or changes in the input features vector, and b) secondly, it has the capability of learning high-order statistical feature from high-dimensional features vector due to RBM layers. The proposed CAE–RBM has the advantages of both CAE and RBM. The basic operational procedure of the proposed model consists of two major stages, namely the training of the CAE layers and the RBM layers. In the first stage, all the parameters associated with the CAE and RBM are initialized. Some of the CAE parameters that have been considered include number of hidden layers, input fea-

ture set, encoding and decoding activation functions, input weights and bias values. The parameters that have been considered for the RBM include input feature set, weight matrix, visible layer and hidden layer bias vectors. After initializing these parameters, the CAE working mechanism starts in the second stage. The extracted features vector is given as input to the CAE input layer. The encoding process occurs from input to hidden layer of CAE and the decoding process happens during information transfer from hidden layer to the output layer of CAE along with reconstruction error. If the reconstruction error is below than the threshold value, the information is then propagating to the third stage. To find the best threshold value that provides better results, in every epoch the model keep updating threshold value in order to achieve the final accuracy, which did not increase or decrease further. The third stage of the proposed CAE–RBM is based on the working procedure of RBM, where the output from the CAE hidden layers is transformed to the RBM layer for learning the high-order statistical features. The hidden layer nodes for RBM are represented by H_i , and $i \in [1, 2, 3, \dots, x]$. Visible nodes are represented by V_j , such that $j \in [1, 2, 3, \dots, y]$. The weight connection between hidden and visible layers is represented by W_{ij} . The weights for RBM hidden layer nodes are also calculated randomly in a sequence based on its input from CAE. The final equation for energy of RBM is:

$$E_{rbm}(v, h|\phi) = - \sum_{i=1}^x v_i a_i - \sum_{j=1}^y h_j \times b_j - \sum_{i=1}^x \sum_{j=1}^y w_{ij} \times v_i \times h_j \tag{12}$$

In Eq. 12, $\phi = (W_{ij}, a_i, b_j)$, 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 Eq. 13, if ϕ is known.

$$P(v, h|\phi) = \frac{1}{z} \exp(-E(v, h|\phi)) \tag{13}$$

The normalization parameter $Z = \sum(v, h) \exp(-E(v, h|\pi))$. This visible and hidden layer’s activation functions can be formatted in Eqs. 14 and 15, respectively.

$$P(h_j = I|v) = \text{sigmoid} \left(\sum_j^x v_i \times w_{ij} + b_j \right) \tag{14}$$

$$P(v_i = I|v) = \text{sigmoid} \left(\sum_i^y h_j \times w_{ij} + a_i \right) \tag{15}$$

$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$. Probability that the first j th hidden node is activated is presented by Eq. 14, while the activation of first i th visible node is presented in Eq. 15. Finding the best suitable parameter set that is possible for ϕ is the main purpose of RBM training. The input data can be betterly fit by the trained model. Calculate the best value of ϕ using maximum likelihood function based on \log by giving training sample T_r , such that $T_r = [v_1, v_2, v_3, v_{T_r}]$. The outputs of the RBM hidden layers are then proceed to the fourth stage. In the fourth stage, the output feature vector from RBM layers is given as input to the softmax layer for final classification process. The complete structure of the proposed model is shown in Fig. 2 and explained in Algorithm 2, while the internal layered architecture is presented in Fig. 3.

5 Experimental Results

In this section, we conducted the experiments to evaluate the feature learning ability of the proposed and comparative models, toward better solution for multiclass classification problems. All the experiments are carried out on Intel core i5 CPU with 8GB of RAM having windows 10 operating system. Python 3.6 is used as compiler and language used for developing and testing these algorithms. For quick implementation of the proposed and comparative approaches, an efficient numeric computational open-source library Tensorflow [5] is used which allows a simple and fast development for both CPU and GPU support. Table 1 presents a summary of parameters setting for the proposed and comparative approaches.

We used four variants of benchmark MNIST dataset for evaluation and comparison. MNIST is a handwritten digit images dataset that contain 70,000 images from 0 to 9. Each image in this dataset has a size of 28x28 pixels. In the conducted experiments, 12,000 images are considered for evaluation and comparison with some comparative models. The experiments performed with 70:30 ratio of training and testing data. Six random images from each MNIST variant dataset are shown in Fig. 4.

5.1 CAE–RBM-Based Feature Reduction

The feature reduction capability of the CAE–RBM model for all the datasets is presented in this section. The main role of CAE in multiclass classification is the dimension reduction of feature space, while the RBM performs better in learning high-order statistical features. The purpose of merging both CAE and RBM in the developed CAE–RBM is to enrich the model with the advantages of both CAE and RBM. The output of the experiments for feature reduction of the developed CAE–RBM is presented in Fig. 5. The first row contain the original images, the second row shows the encoded features

Algorithm 2 CAE–RBM

Parameters initialization

CAE Parameters

- No of hidden layers: h
- Input feature set: $[x_1, x_2, x_3, \dots, x_n]$
- Encoding activation function: EAF
- Decoding activation function: DAF
- Inputs weights: W_i
- Biasness values: b_i
- The state of visible unit is set as training vector

RBM Parameters

- Input feature set: $[y_1, y_2, y_3, \dots, y_n]$
- Weight matrix: $[y_1, y_2, y_3, \dots, y_n]$
- Bias vectors of hidden layer: b_j
- Bias vectors of visible layer: b_i

CAE Operations

Encoding

- Compute encoded inputs $f(w)$ by Multiplying x_n and W_i
- Compute biased inputs $f(b)$ by adding b_i to encoded inputs
- Compute $f(x)$ using Equation 1 by applying $f(w)$ and $f(b)$

Decoding

- Compute decoded outputs $f(w^o)$ by Multiplying x_n and W_i
- Compute biased outputs $f(b^o)$ by adding b_j to decoded outputs
- Compute $g(y^o)$ using Equation 2 by applying $f(w^o)$ and $f(b^o)$

Optimization

- Optimize value of Equation 8

While (All layers trained)

RBM Operations

Assign $[x_1, x_2, x_3, \dots, x_n]$ to $[y_1, y_2, y_3, \dots, y_n]$

Visible units training

- Compute $P(v, h|\phi)$ using Equation 13
- Apply $P(v_i = 1|v)$ of Equation 15
- Perform Gibbs sampling from visible units to hidden unit using $v_{1i} \in [0, 1]$

Hidden units training

- Compute $P(v, h|\phi)$ using Equation 13
- Apply $P(h_j = 1|v)$ of Equation 14
- Perform Gibbs sampling from hidden units to visible units using $h_{1j} \in [0, 1]$

Update rule

- Update weight W : $W = W + W_n$ while $W_n =$ 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 12

While (All layers trained)

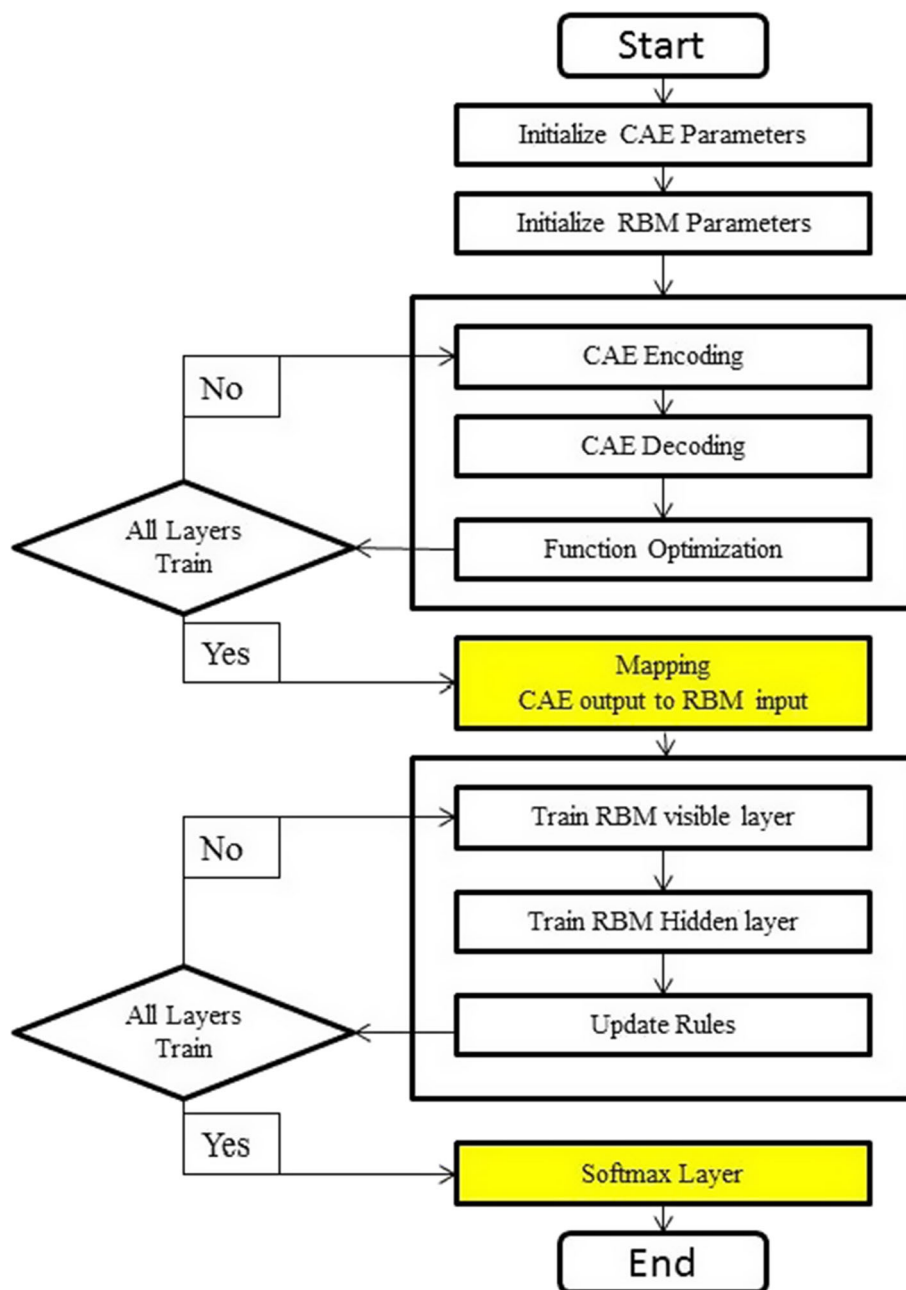
images, while the third row consists of the decoded images from reduced feature set. The information loss in dimensionality reduction is given in Table 2.

5.2 CAE–RBM Measures

This section shows the experimental results of the proposed CAE–RBM model with the reference of confusion matrix and ROC. Confusion matrix and ROC curve present the detailed results of the correctly classified and misclassified instances at class level. The output results of the proposed model on all of four aforementioned datasets are shown in Figs. 6, 7,



Fig. 2 CAE–RBM working flow



8 and 9. In these experiments, the core CAE–RBM model leverages a softmax classifier to find out the overall classification behavior on images data. All these figure presents the output confusion matrix and roc curve for 70:30 training–testing ratio. The highest accuracy is observed for class 1 followed by class 6 and class 0 because their features were more distinct as compared to other classes. On the other hand, the lowest accuracy can be seen for class 8 followed by class 9 and class 5, and mainly these classes were misclassified to one another because of high similarity in features.

6 Comparative Analysis

The comparative analysis of the proposed CAE–RBM with different state-of-the-art multiclass classification models including ANN, standard CAE and standard RBM is presented in here. In this comparison, accuracy and complexity are used as performance evaluation attributes. Testing accuracy and precision/recall are compared in order to compare the accuracy, whereas execution time with *big – O* notation is selected for complexity. However, in image processing, most of the time is consumed by the image representation learning. We reduced the high-dimensional features to low-

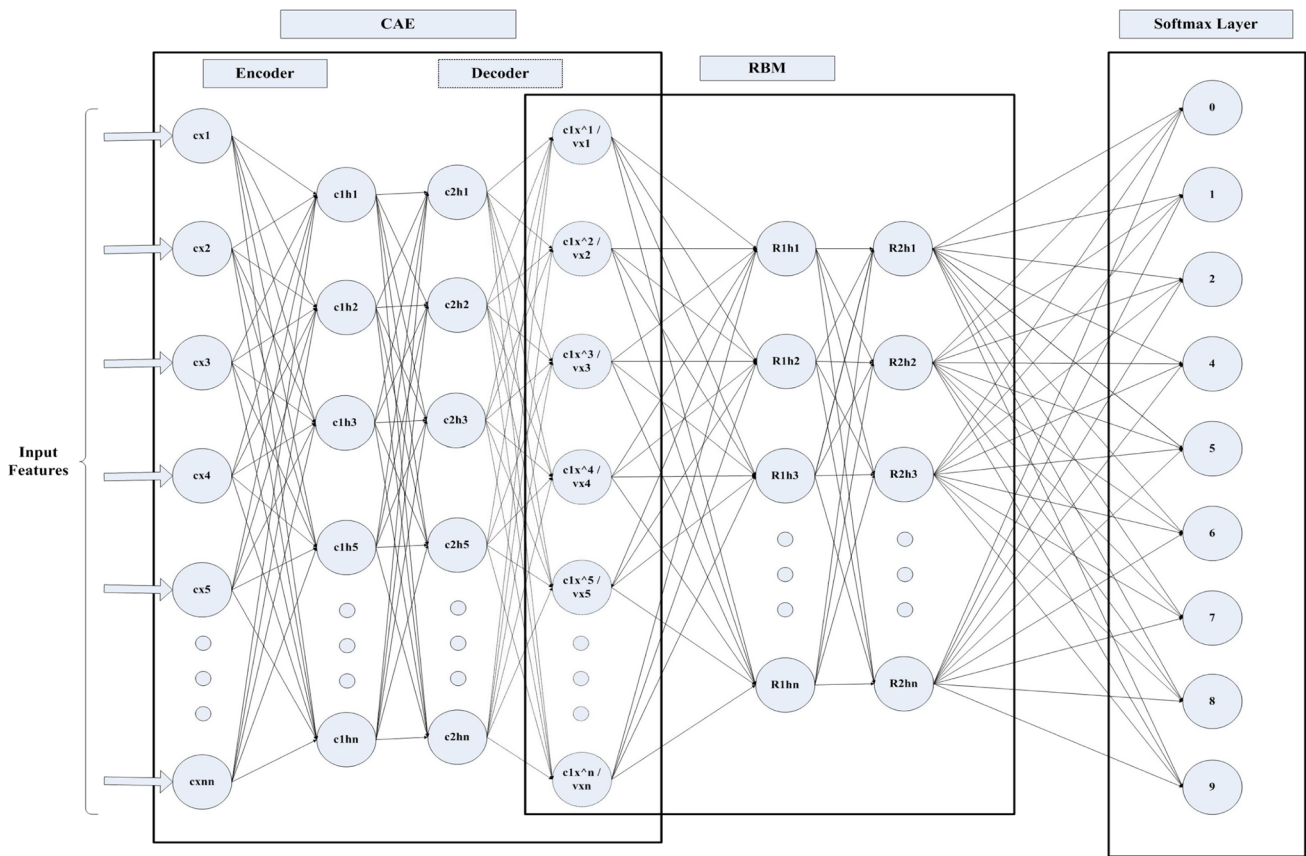


Fig. 3 Architecture of the proposed CAE–RBM model

Table 1 Parameter tuning of the proposed and comparative models

Models	Parameters	Tested values	Considered value
	Input Nodes	784	784
ANN [39]	Output nodes	10	10
CAE [28]	Hidden layers	2,4,6	4
RBM [18]	L1 hidden nodes	400,256	256
AlexNet [42]	L2 hidden nodes	256,64	64
CAE–RBM	Activation unit	Relu, Softmax	Softmax

Fig. 4 Sample images from different MNIST variant datasets

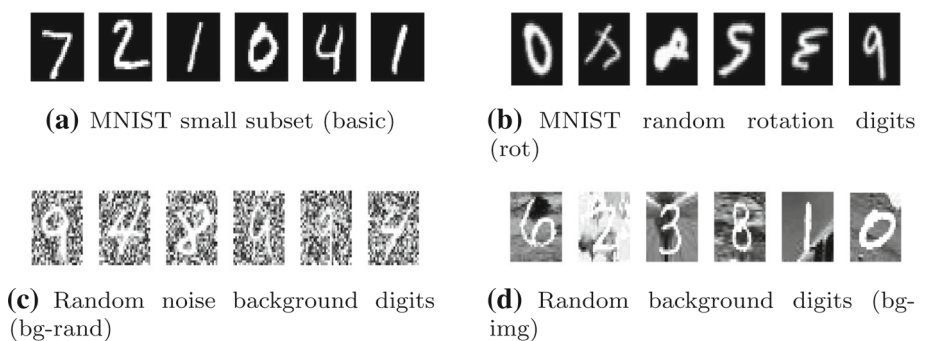
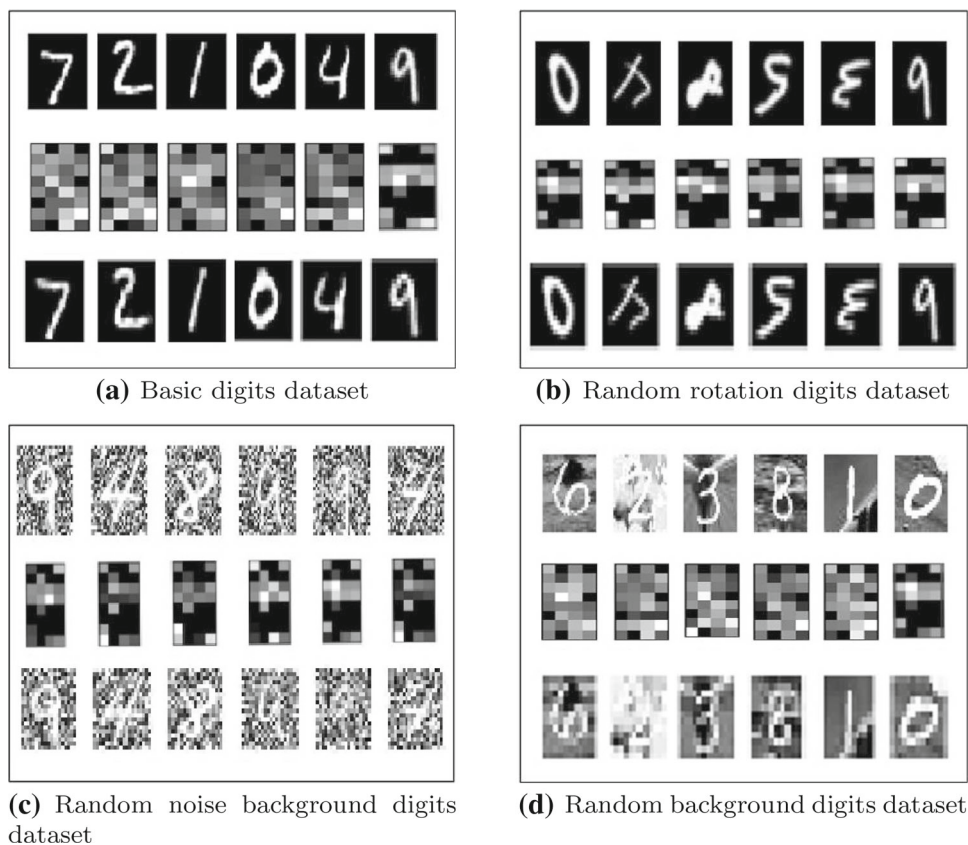


Fig. 5 CAE–RBM-based encoded and decoded sample images from different MNIST variant datasets



dimensional feature vectors for efficiently reduce the time complexity of the proposed CAE–RBM. It can be further reduce by increasing the memory capacity, because memory consumption is directly proportional to representation learning. Therefore, extending memory will enhance the representation learning speed.

6.1 Accuracy

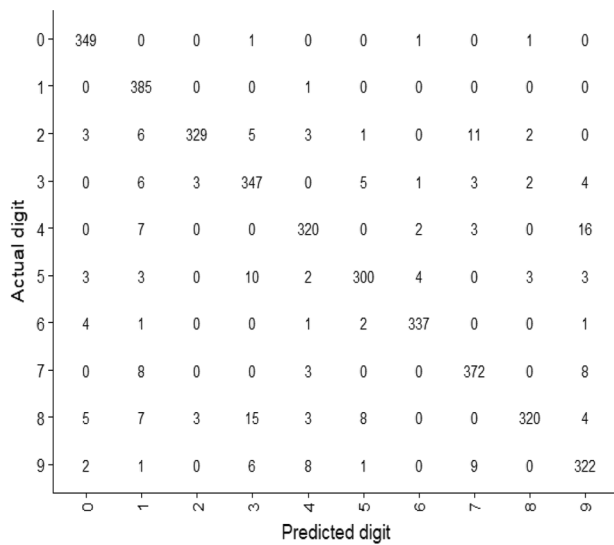
All of the experimented models are trained and tested using the same architecture for pre-classification steps. Table 3 lists all the models with their training accuracy rates. In order to conduct most fair results, the experiments for each of the model has been repeated for five times. All the testing accuracy results of the proposed CAE–RBM and state-of-the-art models discussed in Table 4 are the mean value these repeated experiments. Furthermore, with the training and testing accuracies, Table 5 provides more explanatory performance evaluation metric based on the precision and recall value the proposed and comparative models. The final results conclude that CAE–RBM with Softmax classifier outperforms ANN, CAE and RBM.

Table 2 Information loss during CAE–RBM dimensionality reduction

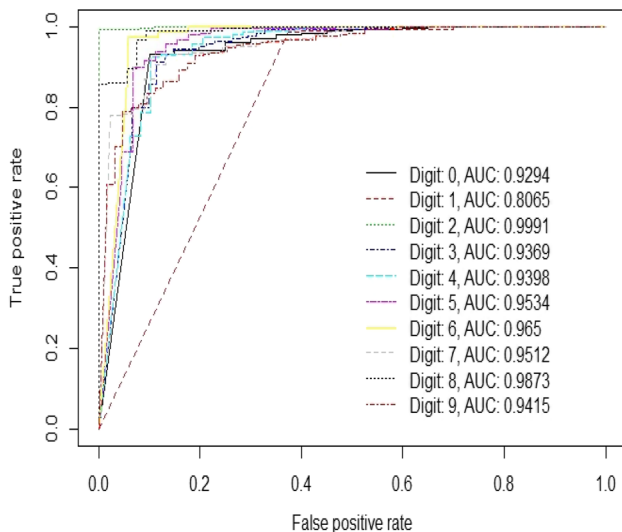
MNIST variants	Basic	Rot	bg-rand	bg-img
Loss	0.24	0.31	0.38	0.42

6.2 Time Complexity

Besides the different computational behavior of different algorithms, all experiments were carried out on same hardware and software architecture as mentioned earlier. The complexity of a neural network is based on its architecture, the number of layers and nodes per each layer. In the conducted experiments, a four hidden layered architecture has been considered. For computing the activation of all nodes in layer L requires $\mathcal{O}(L(n^2))$, where n is the number of features at layer L . The final complexity for ANN in conducted experiments requires $\mathcal{O}(4n^2)$ and AlexNet $\mathcal{O}(4n^3)$. The standard RBM has a runtime complexity of $\mathcal{O}(n^2)$ for a single hidden layer [46]. In the experiments performed followed the same architecture of four hidden layers, according to which the overall complexity of RBM in experiments become $\mathcal{O}(4n^2)$, where n is the number of features in each layer. In a conventional CAE, the complexity has been divided according to encoding and decoding functions. The complexity in calcu-



(a) Confusion Matrix

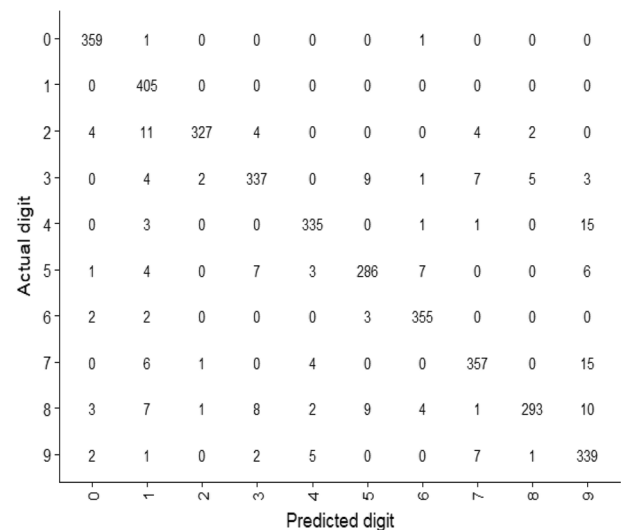


(b) ROC Curve

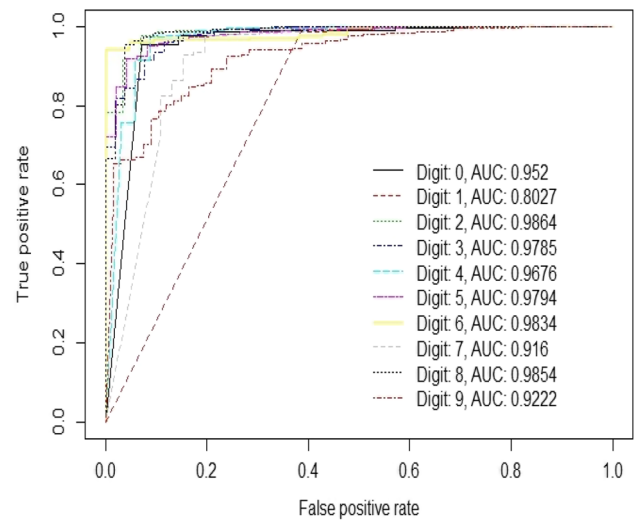
Fig. 6 CM and ROC for MNIST small subset (basic)

lating the compressed layer is $\mathcal{O}(xn)$ and decoding requires $\mathcal{O}(xh)$ runtime. In such case, the overall complexity of a four-layered CAE become $\mathcal{O}(4(xn + xh))$, where x is the number instances, n is the number of encoded features and h refers to number decoded features.

In the proposed model, the CAE and RBM advantages have been merged but as the internal architecture has been kept same for developed and comparative models so the complexity is not increased like other hybridized model. The computation runtime of CAE layers is $\mathcal{O}(4(xn + xh^2))$, where x is the number instances, n is the number of features in CAE hidden layer and h refers to the number of features in RBM hidden layer.



(a) Confusion Matrix



(b) ROC Curve

Fig. 7 CM and ROC for MNIST random rotation digits (rot)

Table 6 summarizes the runtime of different competitive models from the literature based on MNIST variant datasets. It shows a clear observation of the time taken by all of the considered models and algorithms, i.e., ANN, RBM, AlexNet and CAE with four hidden layers, which is greater than that of CAE–RBM. The time taken by CAE–RBM is less because of the randomization factor, which boostup the training but did not increase the computational complexity of the model. Although the difference between the time consumed by CAE–RBM and other approaches is very small, collectively with accuracy and time complexity CAE–RBM presents better performance. Nevertheless, in the overall experimental results the proposed model outperformed the state-of-the-art models with the same architecture and parameter tuning.

Table 4 Testing accuracy of different models with softmax classifier on MNIST variants

MNIST variants	Basic	Rot	bg-rand	bg-img
ANN	89.42	74.50	71.35	70.08
CAE+Softmax	87.60	77.45	76.30	73.60
RBM+Softmax	87.92	76.17	76.41	73.40
AlexNet	85.28	78.35	74.38	70.55
CAE–RBM+Softmax	90.33	86.42	83.61	80.43

Table 5 Precision and recall of different models on MNIST variants

MNIST variants	Basic		Rot		bg-rand		bg-img	
Evaluation metric	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
ANN	0.84	0.76	0.74	0.72	0.70	0.67	0.66	0.63
RBM + Softmax	0.84	0.78	0.77	0.74	0.73	0.70	0.67	0.66
CAE + Softmax	0.86	0.77	0.76	0.77	0.75	0.74	0.70	0.69
AlexNet	0.82	0.74	0.76	0.77	0.74	0.72	0.71	0.70
CAE–RBM + Softmax	0.89	0.84	0.81	0.81	0.79	0.80	0.73	0.71

Table 6 Execution time of different models on MNIST variant

Models	Complexity	Basic	Rot	bg-rand	bg-img
ANN	$\mathcal{O}(4n^2)$	18m 51s	22m 3s	30m 42s	36m 25s
RBM	$\mathcal{O}(4n^2)$	33m 49s	39m 40s	44m 32s	59m 51s
CAE	$\mathcal{O}(4(xn + xh))$	27m 45s	28m 53s	37m 40s	41m 57s
AlexNet	$\mathcal{O}(4n^3)$	38m 49s	41m 40s	44m 32s	60m 51s
CAE–RBM	$\mathcal{O}(4(xn + xh^2))$	19m 57s	20 52s	30m 56s	35m 12s

Table 3 Training accuracy of different models on MNIST variant

MNIST variants	Basic	Rot	bg-rand	bg-img
ANN	92.45	90.44	82.63	77.98
RBM	93.81	88.48	90.24	83.30
CAE	94.54	89.96	91.32	86.42
AlexNet	93.98	90.10	90.40	84.62
CAE–RBM	96.90	90.85	94.65	87.34

7 Conclusion

The main focus of this paper is to propose a better approach for multiclass classification. Here we present the CAE based on RBM (CAE–RBM) and evaluate the performance of proposed model. We use CAE for dimensionality reduction and RBM for learning high-order statistical feature. The output of the developed model is compared analytically with existing state-of-the-art classification algorithms including standard CAE, RBM, AlexNet and ANN. The results presented in different figures and tables are based on four benchmark datasets of MNIST. In these experiments, 12000 images are randomly selected from each of the five benchmark datasets.

The experiments performed with 70% training and 30% testing ratio of the each dataset. The performance is evaluated in term of accuracy and time complexity. In order to validate the performance of the proposed model, the class level classification results are presented in the form of confusion matrices and ROC curves. The experimental outputs project a minor decrease in the accuracy starting from MNIST basic dataset to the most complex MNIST random background digits dataset. This gradual increment in the complexity of dataset is directly related to the accuracy decrement. In the developed models, CAE–RBM outperformed all the comparative models based on accuracy and time complexity. On the other hand, the proposed models do not consider the space complexity of the algorithm for less complex and small-sized datasets. The CAE–RBM has been added the functionalities standard CAE from RBM, both of these algorithms have different working procedures for processing the information in order to perform the final classification phenomenon. So, merging the concepts of one algorithm in another algorithm involves many technical procedures. In CAE–RBM, the RBM layers are merged with the CAE layers and during this process, the complex computational steps of both CAE and RBM have not been removed which may increase the

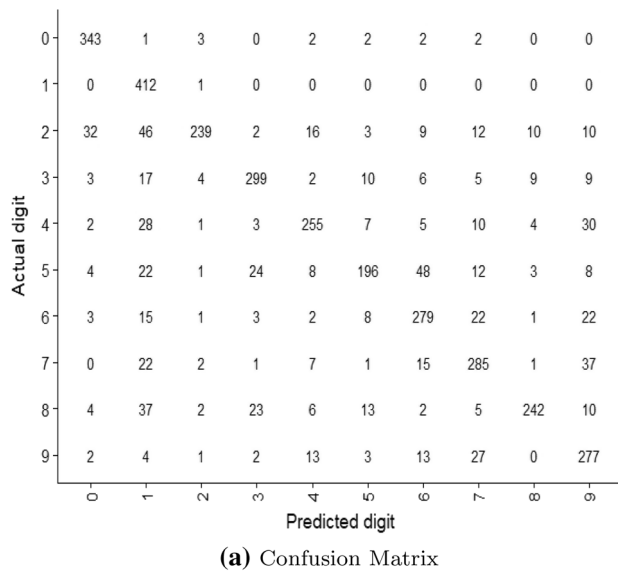


Fig. 8 CM and ROC for MNIST random noise background digits (bg-rand)

space complexity of the model. In future, we will proceed with the space complexity reduction of the proposed model and perform experiments on the Big Data application.

Acknowledgements The authors would like to thanks Ministry of Education Malaysia, Universiti Tun Hussein Onn Malaysia (UTHM) and University of Derby, United Kingdom, for carrying out this research activity.

References

1. Aamir, M.; Nawi, N.M.; Shahzad, A.; Mahdin, H.; Rehman, M.Z.: A new argumentative based reasoning framework with rough set for decision making. In: 2017 6th ICT International Student Project Conference (ICT-ISPC), pp. 1–4, May (2017)

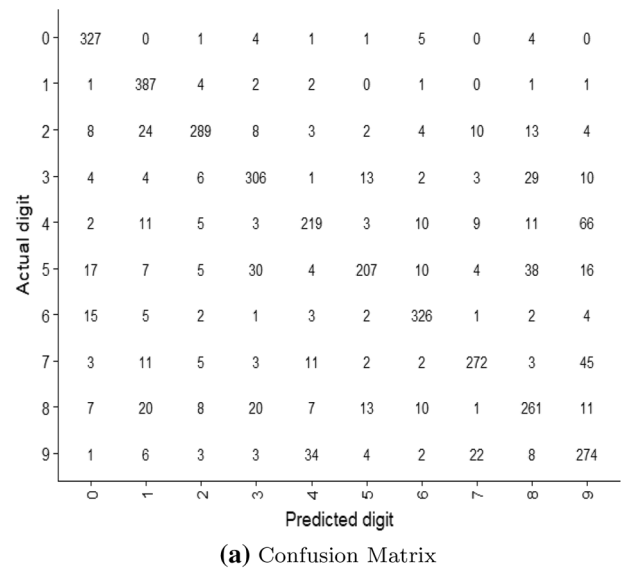


Fig. 9 CM and ROC for MNIST random background digits (bg-img)

2. Aamir, M.; Nawi, N.M.; Wahid, F.; Mahdin, H.: A deep contractive autoencoder for solving multiclass classification problems. In: Evolutionary Intelligence, pp. 1–15 (2020).
3. Aamir, M.; Wahid, F.; Mahdin, H.; Nawi, N.M.: An efficient normalized restricted boltzmann machine for solving multiclass classification problems. *Int. J. Adv. Comput. Sci. Appl.* **10**, 8 (2019)
4. Aamir, M.; Nawi, N.M.; Mahdin, H.B.; Naseem, R.; Zulqarnain, M.: Auto-encoder variants for solving handwritten digits classification problem. *Int. J. Fuzzy Logic Intell. Syst.* **20**(1), 8–16 (2020)
5. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M. et al.: Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), pp. 265–283 (2016).
6. Aldwairi, T.; Perera, D.; Novotny, M.A.: An evaluation of the performance of restricted boltzmann machines as a model for anomaly network intrusion detection. *Comput. Netw.* **144**, 111–119 (2018)

7. Ayesha, S.; Hanif, M.K.; Talib, R.: Overview and comparative study of dimensionality reduction techniques for high dimensional data. *Inf. Fusion* **59**, 44–58 (2020)
8. Buda, M.; Maki, A.; Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw.* **106**, 249–259 (2018)
9. Cai, W.; Wei, Z.: Remote sensing image classification based on a cross-attention mechanism and graph convolution. In: *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5 (2020)
10. Chen, M.; Weinberger, K.Q.; Xu, Z.; Sha, F.: Marginalizing stacked linear denoising autoencoders. *J. Mach. Learn. Res.* **16**(1), 3849–3875 (2015)
11. Dehban, A.; Jamone, L.; Kampff, A.R.; Santos-Victor, J.: Denoising auto-encoders for learning of objects and tools affordances in continuous space. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4866–4871, IEEE (2016)
12. Druzhkov, P.; Kustikova, V.: A survey of deep learning methods and software tools for image classification and object detection. *Pattern Recogn. Image Anal.* **26**(1), 9–15 (2016)
13. Goodfellow, I.; Bengio, Y.; Courville, A.: *Deep Learning*. MIT Press, New York (2016)
14. Goroshin, R.; LeCun, Y.: Saturating auto-encoders. *arXiv preprint arXiv:1301.3577* (2013).
15. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al.: Recent advances in convolutional neural networks. *Pattern Recogn.* **77**, 354–377 (2018)
16. Hinton, G.E.: A practical guide to training restricted boltzmann machines. In: *Neural Networks: Tricks of the Trade*, pp. 599–619. Springer (2012)
17. Izonin, I.; Tkachenko, R.; Kryvinska, N.; Tkachenko, P. et al.: Multiple linear regression based on coefficients identification using non-iterative sgtm neural-like structure. In: *International Work-Conference on Artificial Neural Networks*, pp. 467–479, Springer (2019).
18. Kang, T.; Chen, P.; Quackenbush, J.; Ding, W.: A novel deep learning model by stacking conditional restricted boltzmann machine and deep neural network. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1316–1324 (2020).
19. Karras, T.; Aittala, M.; Hellsten, J.; Laine, S.; Lehtinen, J.; Aila, T.: Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676* (2020).
20. Knyaz, V.A.; Vygolov, O.; Kniaz, V.V.; Vizilter, Y.; Gorbatshevich, V.; Luhmann, T.; Conen, N.: Deep learning of convolutional auto-encoder for image matching and 3d object reconstruction in the infrared range. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2155–2164 (2017)
21. Konda, K.; Memisevic, R.; Krueger, D.: Zero-bias autoencoders and the benefits of co-adapting features (2014). *arXiv preprint arXiv:1402.3337*.
22. Krizhevsky, A.; Sutskever, I.; Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2017)
23. Li, B.; Najafi, M.H.; Lilja, D.J.: 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*, pp. 36–41, ACM (2016)
24. Liu, L.: Hierarchical learning for large multi-class network classification. In: *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 2307–2312, IEEE (2016).
25. Liu, Y.; Feng, X.; Zhou, Z.: Multimodal video classification with stacked contractive autoencoders. *Signal Process.* **120**, 761–766 (2016)
26. Long, M.; Zhu, H.; Wang, J.; Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2208–2217, JMLR. org (2017)
27. Lv, N.; Chen, C.; Qiu, T.; Sangaiah, A.K.: Deep learning and superpixel feature extraction based on contractive autoencoder for change detection in sar images. *IEEE Trans. Ind. Inf.* **14**(12), 5530–5538 (2018)
28. Mahmoud, A.M.; Alrowais, F.; Karamti, H.: A hybrid deep contractive autoencoder and restricted boltzmann machine approach to differentiate representation of female brain disorder. *Proc. Comput. Sci.* **176**, 1033–1042 (2020)
29. Mao, X.; Li, Q.; Xie, H.; Lau, R.Y.; Wang, Z.; Paul Smolley, S.: Least squares generative adversarial networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802 (2017).
30. Meyer, D.; Wien, F.T.: Support vector machines. *The Interface to libsvm in package e1071*, p. 28 (2015)
31. Nasiri, J.A.; Charkari, N.M.; Jalili, S.: Least squares twin multi-class classification support vector machine. *Pattern Recogn.* **48**(3), 984–992 (2015)
32. Nawi, N.M.; Hamzah, F.; Hamid, N.A.; Rehman, M.Z.; Aamir, M.; Ramli, A.A.: An optimized back propagation learning algorithm with adaptive learning rate. *Learning* **500**, 2 (2017)
33. Ning, X.; Li, W.; Tang, B.; He, H.: Buldp: biomimetic uncorrelated locality discriminant projection for feature extraction in face recognition. *IEEE Trans. Image Process.* **27**(5), 2575–2586 (2018)
34. Oyedotun, O.K.; Khashman, A.: Deep learning in vision-based static hand gesture recognition. *Neural Comput. Appl.* **28**(12), 3941–3951 (2017)
35. Ran, R.; Ren, Y.; Zhang, S.; Fang, B.: A novel discriminant locality preserving projections method. In: *Journal of Mathematical Imaging and Vision*, pp. 1–14 (2021)
36. Reddy, G.T.; Reddy, M.P.K.; Lakshmana, K.; Kaluri, R.; Rajput, D.S.; Srivastava, G.; Baker, T.: Analysis of dimensionality reduction techniques on big data. *IEEE Access* **8**, 54776–54788 (2020)
37. Rifai, S.; Mesnil, G.; Vincent, P.; Muller, X.; Bengio, Y.; Dauphin, Y.; Glorot, X.: Higher order contractive auto-encoder. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 645–660, Springer (2011).
38. Rifai, S.; Vincent, P.; Muller, X.; Glorot, X.; Bengio, Y.: Contractive auto-encoders: explicit invariance during feature extraction. In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 833–840, Omnipress (2011).
39. Russell, S.J.; Norvig, P.: *Artificial Intelligence: A Modern Approach*. Pearson Education Limited, Malaysia (2016)
40. Sainath, T.N.; Mohamed, A.-R.; Kingsbury, B.; Ramabhadran, B.: Deep convolutional neural networks for lvsr. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8614–8618, IEEE (2013).
41. Senan, N.; Aamir, M.; Ibrahim, R.; Taujuddin, N.S.A.M.; Muda, W.W.: An efficient convolutional neural network for paddy leaf disease and pest classification. *Int. J. Adv. Comput. Sci. Appl.* **11**, 7 (2020)
42. Sengan, S.; Arokia Jesu Prabhu, L.; Ramachandran, V.; Priya, V.; Ravi, L.; Subramaniaswamy, V.: Images super-resolution by optimal deep alexnet architecture for medical application: a novel localn. *J. Intell. Fuzzy Syst., no. Preprint*, pp. 1–14 (2020).
43. Sharma, N.; Singh, M.: Modifying naive bayes classifier for multinomial text classification. In: *2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, pp. 1–7, IEEE (2016)
44. Sharma, A.; Tuzel, O.; Jacobs, D.W.: Deep hierarchical parsing for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 530–538 (2015)
45. Sur, C.: Relgan: generalization of consistency for gan with disjoint constraints and relative learning of generative processes for

- multiple transformation learning. arXiv preprint [arXiv:2006.07809](https://arxiv.org/abs/2006.07809) (2020).
46. Taherkhani, A.; Cosma, G.; McGinnity, T.M.: Deep-fs: a feature selection algorithm for deep boltzmann machines. *Neurocomputing* **322**, 22–37 (2018)
 47. Tao, M.; Tang, H.; Wu, S.; Sebe, N.; Wu, F.; Jing, X.-Y.: Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis. arXiv preprint [arXiv:2008.05865](https://arxiv.org/abs/2008.05865) (2020).
 48. Tkachenko, R.; Izonin, I.; Vitynskyi, P.; Lotoshynska, N.; Pavlyuk, O.: Development of the non-iterative supervised learning predictor based on the ito decomposition and sgtm neural-like structure for managing medical insurance costs. *Data* **3**(4), 46 (2018)
 49. Vitynskyi, P.; Tkachenko, R.; Izonin, I.; Kutucu, H.: Hybridization of the sgtm neural-like structure through inputs polynomial extension. In: 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), pp. 386–391, IEEE (2018)
 50. Wang, Y.; Yao, H.; Zhao, S.: Auto-encoder based dimensionality reduction. *Neurocomputing* **184**, 232–242 (2016)
 51. Wen, L.; Gao, L.; Li, X.: A new deep transfer learning based on sparse auto-encoder for fault diagnosis. *IEEE Trans. Syst. Man Cybernet. Syst.* **99**, 1–9 (2017)
 52. Xiong, P.; Wang, H.; Liu, M.; Lin, F.; Hou, Z.; Liu, X.: A stacked contractive denoising auto-encoder for ecg signal denoising. *Physiol. Measur.* **37**(12), 2214 (2016)
 53. Young, T.; Hazarika, D.; Poria, S.; Cambria, E.: Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag* **13**(3), 55–75 (2018)
 54. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A.: Self-attention generative adversarial networks. In: International Conference on Machine Learning, pp. 7354–7363, PMLR (2019)
 55. Zhang, W.; Liu, Y.; Dong, C.; Qiao, Y.: Ranksrgan: generative adversarial networks with ranker for image super-resolution. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3096–3105 (2019).
 56. Zhang, S.; Li, X.; Zong, M.; Zhu, X.; Wang, R.: Efficient knn classification with different numbers of nearest neighbors. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(5), 1774–1785 (2017)

