IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 12, 2021, accepted April 7, 2021, date of publication April 13, 2021, date of current version April 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3073090

Seismic Data Compression Using Deep Learning

EMAD B. HELAL[1], OMAR M. SAAD[1,2], ALI G. HAFEZ[1,3,4], YANGKANG CHEN[2], (Member, IEEE), AND GAMAL M. DOUSOKY[3,5,6], (Senior Member, IEEE)

[1]Department of Seismology, National Research Institute of Astronomy and Geophysics (NRIAG), Helwan 11421, Egypt
[2]School of Earth Sciences, Zhejiang University, Hangzhou 310027, China
[3]Department of Communication and Computer Engineering, Faculty of Engineering, Nahda University, Beni Suef 65211, Egypt
[4]Research and Development Division, LTLab Inc., Fukuoka 814-0155, Japan
[5]Department of Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan
[6]Electrical Engineering Department, Minia University, Minia 61517, Egypt

Corresponding author: Emad B. Helal (emad.helal@nriag.sci.eg)

This work was supported by the Egyptian Science and Technology Development Fund (STDF) under Project 25681.

ABSTRACT The exponential growth of the size of seismic data recorded in seismic surveys and real time data monitoring makes seismic data compression strongly demanded. Furthermore, compression will lead to an efficient use of the bandwidth assigned for the communication link between the seismic stations and the main center. In this paper, two convolutional autoencoders (CAEs) are proposed for seismic data compression. The two algorithms are mainly based on the convolutional neural network (CNN), which has the capability to compress the seismic data into feature representations, thereby allowing the decoder to perfectly reconstruct the input seismic data. The results show that the first model is efficient at low compression ratios (CRs), while the second model improves the signal-to-noise ratio (SNR) from about 3 dB to 12 dB compared to the other benchmark algorithms at moderate and high CRs.

INDEX TERMS Convolutional autoencoders (CAE), deep learning, seismic data compression.

I. INTRODUCTION

Seismic data are collected for many purposes, e.g., crustal earth structure studies, earthquake parameter calculations, and oil and gas explorations. The huge amount of data records require a large storage and bandwidth for both site archiving and transmission to the main database center. Thus, it is beneficial to compress seismic data to save a huge amount of resources that are reserved for transmission and storage of data.

Seismic data compression methods are categorized into two groups: lossy and lossless compression. In the latter group of methods, there are no losses or noise added to the original data after reconstruction, which, is only applicable at low compression ratios (CRs). While in lossy compression, reasonable losses exist in the reconstructed data, thus allowing high data compression. In this work, we only consider the lossy compression. Among transformation techniques, discrete cosine transform (DCT) achieves robust performance, especially when the original input data rate was decreased by one-third [1]. Wavelet transform approaches [2]–[4] are considered popular transformation methods for compressing seismic data because of their effective data representation and direct data reconstruction. However, the common cosine and wavelet methods are non adaptive and cannot cope with the high oscillation nature of seismic waves [5]. In order to solve these issues, an improvement of wavelet transform method is proposed by using M-channel uniform filter banks [6]. Moreover, a hybrid compression method between wavelet and the local cosine method is proposed to handle the oscillatory behavior of seismic wave [7].

Numerous methods of data compression have been proposed recently. Liu *et al.* propose a distributed principal component analysis (DPCA) [8] to compress seismic traces by collecting seismic traces from different sensors and produces orthogonal global PCs based on a combination of probability density functions. Despite the low computational cost, the DPCA method has relatively low performance at high compression ratios. Compressive Sensing (CS) has recently been used in data compression. Contrary to the traditional compression methods, CS theory assumes that compression can be implemented before data digitization [9]. Thus, it is no longer limited by Shannon's sampling theory. CS criterion proves that from a small number of samples, the signal

The associate editor coordinating the review of this manuscript and approving it for publication was Weiping Ding.

can be recovered if and only if it achieves two conditions: (1) sparseness, and (2) incoherence. Regarding seismic data compression, CS achieved high compression ratios while ensuring high accuracy of the reconstructed data [10], [11]. However, seismic compression based on CS methods mainly depends on theoretical analysis and mathematical tests which is time consuming and still faces several limitations. Learning from data or dictionary learning (DL) methods has recently been used in seismic data compression. In Payani *et al.* [12] two approaches are proposed: (1) sparse-incremental online DL (SIODL) is faster than the ordinary DL method, and (2) rate-optimized DL achieves good performance. However, these algorithms suffer from high communication cost in data acquisition. Recently, multiscale sparse dictionary learning has been proposed for data compression [13]. In this method the dictionary learning process is integrated with wavelet transform. This method achieves acceptable performance with the existence of noise. However, the calculation process of sub-band weights needs to be modified to achieve better results. Deep learning methods are widely used to solve the seismic compression problem. Restricted Boltzmann machine (RBM) [14] with single layer neural network is proposed in Nuha *et al.* [15]. Although being computationally efficient, a single layer cannot capture all data features and the reconstruction error is high. Using multiple hidden layers, Nuha *et al.* [16] integrates the extreme learning machine technique [17] with deep neural networks autoencoder. This method is considered a fast method due to the analytically calculated encoder/decoder weights without any iterations. However, this method is not a generalized method because the feature representation is not generated from different attributes of input traces. A 3D deep learning technique is proposed in Schiavon *et al.* [18] to compress seismic data with low bit rate. This technique is less sensitive to noise and can reconstruct the seismic data with high quality. However, this model concentrates on high compression rather than high bit rates.

In contrary to the most machine learning methods, deep learning deals with the raw data directly without the need of extra processing operations of the input data [19]–[21]. In addition, deep learning extracts the significant features by dividing the input data via multiple abstraction levels in unsupervised manners [21]. One of the most powerful deep learning architectures is the autoencoder which encodes the input data using hidden layers then enables the decoder to reconstruct the input data efficiently [14]. AE is widely used in dimensionality reduction applications, e.g., image processing [22], feature extraction [23], and semantic hashing [24]. In the seismology field, Saad *et al.* use the autoencoder in automatic earthquake arrival time detection [25] and denoising [26].

Convolutional autoencoder (CAE) is one of the most robust AE architectures. CAE is based on convolutional neural network (CNN), which is a powerful tool in classification and feature extraction applications. Unlike other deep AE methods, CNN uses convolutional filters rather than neurons to extract the required feature map. Recently, CAE is utilized in many applications, e.g., radar-based activity classification [27], denoising of speech signals [28], and fault detection in aircraft engine [29]. In the geophysical community, CAE solves enormous problems, such as, lithology prediction [30], arrival picking [31], seismic data interpolation [32], simultaneous-source separation [33], earthquake parameters classification [34], and waveform-based source-location imaging [35]. Although CAE has been used for compression in many scientific domains such as biomedical context [36], [37], to the best of our knowledge, it is *the first time* to be used for 1D seismic data compression. In this paper, two CAE models have been proposed to characterize the high oscillation nature of seismic waves. Applications to real earthquake data show that the proposed CAE models can compress the input samples at different CRs. Then, the original data is reconstructed successfully with a high signal-to-noise ratio (SNR). Both models introduce an integrated solution for low and high data compression. Model-II achieves a higher SNR than other state-of-the-art methods at high and moderate compression ratios, while model-I achieves reasonable performance at low compression ratios. The proposed models are robust and have a strong generalization ability.

II. PROPOSED COMPRESSION ALGORITHMS

Seismic data compression is important for all components of the seismic network: from the size of the memory of the in-site transmitter, to the bandwidth of the communication link, to traffic at the central node and finally to the storage media at the data center. Therefore, it is significant to compress the transmitted records as small as possible, while preserving the important information required for reconstruction purpose. Unlike handwritten digits in image processing, seismic data have different wave characteristics that should not be changed after reconstruction. We propose two deep learning models based on the convolutional neural network (CNN). Note that, the first model (model-I) and the second model (model-II) complement each other. The first model is preferable for low CRs, while the second model is more suitable for high and moderate CRs.

A. DATASET

The dataset from the East Texas, USA, has been used in this work [38]. For 594 seismic traces, 70% of the data are utilized for training, 10% are used for validation, and the remaining (20%) are used for testing. The record length for each trace is 3 seconds and the sampling rate is 500 Hz. Thus, the number of acquired samples is 1500 samples. The magnitude of the data is normalized in the range [0-1]. The CNN network models are built using Keras with TensorFlow backend [39]. The performance of the deep learning models is executed using the GPU enabled by Google COLAB.

B. PROPOSED MODEL-I

The first convolutional autoencoder (CAE) (model-I) depends on the dimensional reduction of the input data to

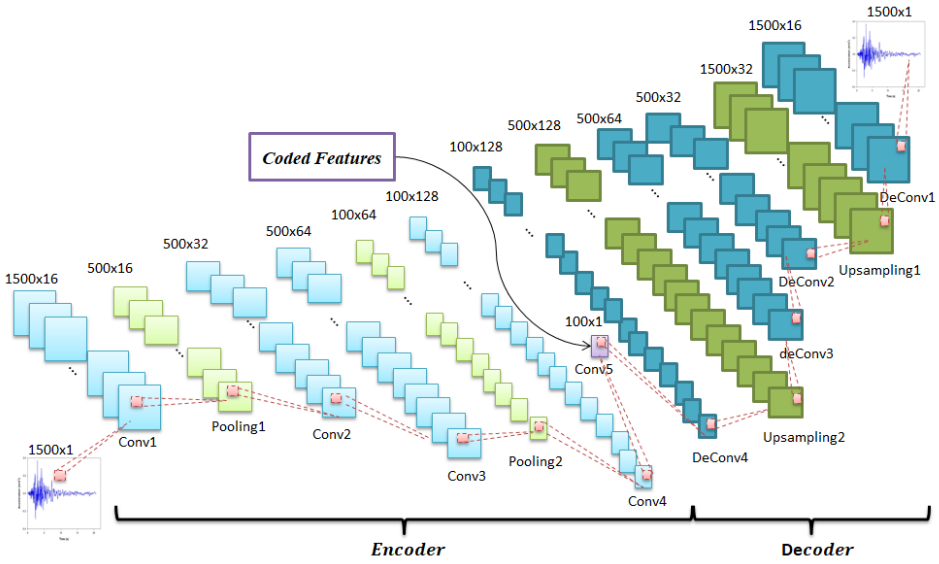


FIGURE 1. The proposed model-I example when CR = 15.

compress the seismic data during the transmission process. The CAE model consists of two stages: encoder and decoder. The encoder extracts the significant features from the input data and compresses the input data into smaller dimensions. The encoder stage consists of several layers, e.g., 1D convolutional and pooling layers to encode the input data. The last encoder layer (coded layer or bottleneck layer) contains the least feature map to represent the input data. The required CR is determined by the number of features transmitted in the coded layer. The decoder reconstructs the input data using the extracted features from the coded layer. The decoder utilizes successive deconvolutional and upsampling layers. Assuming that the encoder part will be executed at the seismic station and the decoder will be executed at the main center.

The encoder extracts several feature maps using the convolutional layer with several kernels. For an input data x, the output of the $i-th$ kernel, h^i, can be determined as follows:

$$h^i = \sigma(x * K^i + b^i), \tag{1}$$

where $*$ refers to 1D convolution, K is the ith convolution filter, b^i is a bias of ith feature map, and σ is the activation function. The activation function in the convolutional layers is the exponential linear unit (ELU) with $\alpha > 0$ shown as follows:

$$\sigma = \begin{cases} z, & \text{for } z > 0 \\ \alpha \cdot (e^z - 1), & \text{for } z \leq 0, \end{cases} \tag{2}$$

where, α, controls the scale of the negative inputs. Afterward, the maxpooling layer is used to reduce the size of the data by a rate of p, where p is the downsampling rate [40]. The output of the coded layer has a size of $\frac{l_i}{CR}$, where l_i is the input data size.

Then, the original value of the input samples are reconstructed using the symmetrical decoding architecture but with inverse operations to the encoder. The reconstruction is determined using the following formula:

$$y = \sigma(\sum_{i \in H} h^i * \tilde{K}^i + c), \tag{3}$$

where $\tilde{K}$ is the inverse operator that transposes the weights, H denotes the feature maps, and c is the bias per input channel. The hidden layers in the decoder side use the ELU activation function for fast training and achieving better performance. Afterward, up-sampling layer is used to compensate the effect of the pooling layer, i.e., to increases the size of the data with a rate of p [41]. The final decoder layer is a 1D convolution layer with one convolution filter to reconstruct the input signal with the same size. The loss function can be determined as follows:

$$\min_{\theta} .||F(x;\theta) - x||_2^2, \tag{4}$$

where F is the proposed network, x is the network input, $\Theta = \{K, \tilde{K}, b, c\}$ is the network parameters, $F(x;\theta)$ denotes the network output, and N represents the number of data samples.

The architecture of the first model is illustrated in Fig.1, e.g., when CR = 15, the input data is compressed from 1500 samples to 100 samples. First, the input data with size of 1500×1 are input to the 1D convolutionl layer with 16 convolution filters (1500×16). To decrease the network complexity, the maxpolling layer is utilized to reduce the feature map size to be (500×16). Then, two consecutive 1D convolutional layers having twice the number of filters are used to obtain the final feature maps (500×64). In order to obtain the required rate, the maxpolling layer is utilized with an order of 5 to produce feature maps (100×64), and for better signal resolution a 1D convolutional (100×128) is added

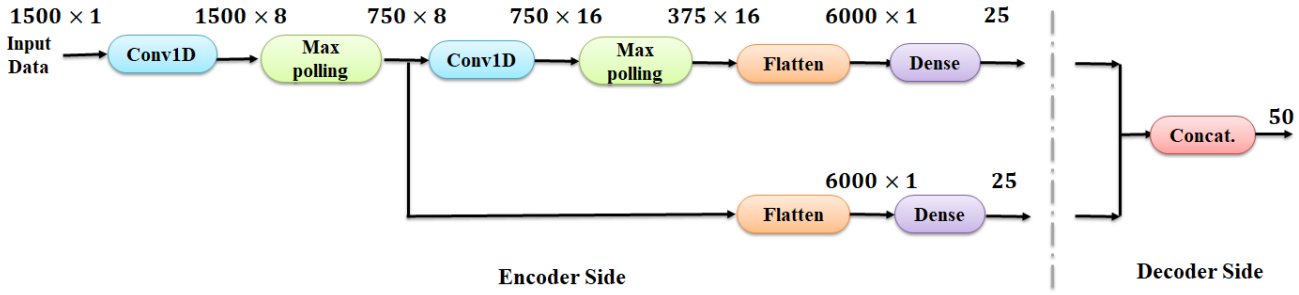


FIGURE 2. The proposed model-II example when CR = 30.

before the final encoder layer (100 × 1). The feature map in the bottleneck layer (100 × 1) is considered as the decoder input. The same topology is repeated on the decoder side with an inverse operation, e.g., up-sampling layer is opposed to the pooling layer to increase the size of samples at the required rate. Finally, the (100 × 1) encoded features are transformed into 1500 × 1 at the output layer that has the same size of the input data.

C. PROPOSED MODEL-II

The second model is also based on CNN. However, the network architecture is different. The encoder comprises three stages: 1D convoltional (Conv1D), flatten, and dense layers. The convolutional filter slides with a stride of s, where the stride is the number of shift samples. This represents the same role of maxpooling with a rate of s [42]. First, the input layer reads and stores the input samples as a 1D tensor before transmission to the next layer. Then, the compression process is accomplished using a number of consecutive 1D convolutional layers with a gradual increase of the feature maps. Each convolutional layer obtains the key features and decreases the dimensions of data samples to reduce the network complexity. All convolutional hidden layers use the ELU activation function. The output of each layer has two paths. One path is directed to the next convolutional layer to extract more features, and the other path is directed to the flatten layer, which converts the data into a 1D array to be fed into a dense layer. Each dense layer separates out some of the required feature maps. The dropout layer is utilized after each convolutional layer to avoid overfitting. Unlike the first model, this model does not have the bottleneck layer because the features are encoded and distributed along the dense layers before transmission to the decoder.

The decoder in this model is very simple, compared to the first model. Because, it only integrates the incoming encoded features using a small number of concatenated layers. Finally, the dense layer maps the original input signal using the sigmoid function. This model is inspired from UNet [43], since the network architecture between the encoder and the decoder has a U-shape. Fig.2 describes the proposed seismic data compression process at CR = 30.

III. APPLICATION AND RESULTS

A. EVALUATION CRITERIA

The evaluation metrics measure the reconstructed signal quality of the proposed seismic data compression approaches. Those widely known metrics are defined as follows: (1) normalized mean-squared error (NMSE), (2) normalized root mean-squared error (NRMSE), and (3) SNR. These criteria compare the reconstructed output signals with the original seismic input signal as follows:

1) Normalized mean-squared error (NMSE)

$$\text{NMSE} = \frac{\sum_{i=1}^{K} \|x_i - \hat{x}_i\|^2}{\sum_{i=1}^{K} \|x_i\|^2}, \tag{5}$$

where K is the seismic signal index in the dataset. x and $\hat{x}$ refer to the instantaneous values of the original trace and its related output after reconstruction, respectively.

2) Normalized root mean-squared error (NRMSE)

$$\text{NRMSE} = \frac{\sqrt{\text{mean}\{(\text{x} - \hat{\text{x}})^2\}}}{\max(\text{x}) - \min(\text{x})}, \tag{6}$$

where x is the data vector and $\hat{\text{x}}$ is the recovered signal vector. NRMSE is used to measure the distortion level resulted from the lossy compression [44].

3) SNR is the evaluation metric used to measure the signal quality after compression. The SNR is correlated to NMSE as follows:

$$\text{SNR}(dB) = -\log_{10} \text{NMSE} \tag{7}$$

B. NETWORK ARCHITECTURES

The network architectures of the two proposed models are chosen to maximize the SNR. The number of layers and feature maps for each layer of the CAE are changed, while the other network parameters are constant. The feature maps inside each layer is incremented gradually with 2, 4, 8 and 16. To illustrates, Fig. 3 shows the effect of changing the number of layers versus the network feature maps on the performance, when CR = 30. Fig. 3a shows the SNR-performance for model-I, where the best SNR has been achieved using five layers. While, two layers are sufficient to achieve the best SNR for model-II as shown in Fig. 3b. The feature maps of both models should start from 8 features and increase

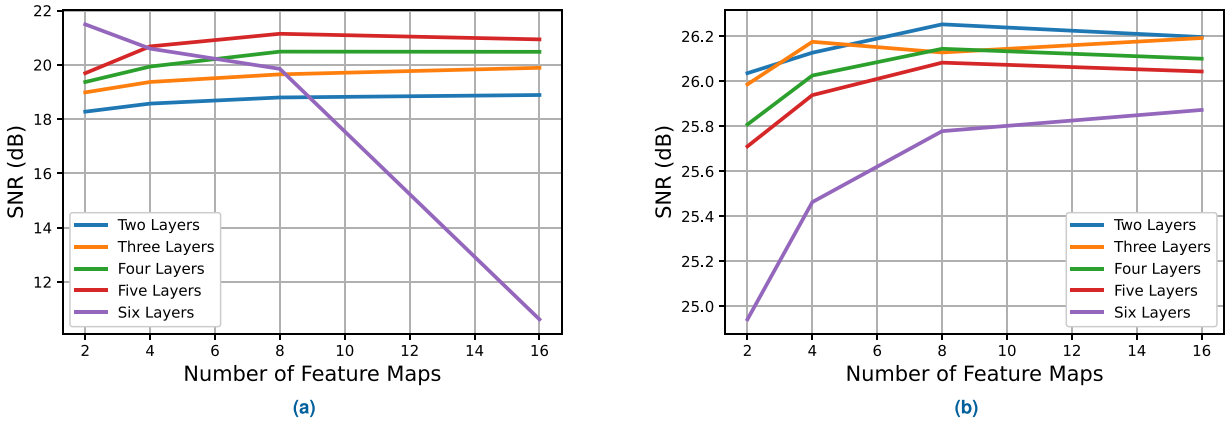


FIGURE 3. The number of layers effect versus network feature maps on the SNR for (a) Model-I and (b) Model-II when CR = 30.

TABLE 1. Performance of model-I for 475 training and 119 test traces at different compression ratios. The performance metrics: NMSE, NRMSE, and SNR are calculated for test data.

Architectures	Training Time	NMSE (10^{-3})	NRMSE	SNR	CR
(1500,8)-(300,8)-(300,16)-(150,16)-(150,32)-(30,32)-(30,64)-(15,64)-(15,128)-(15,1)	1min 16s	9.343	0.048	20.295	100:1
(1500,8)-(500,8)-(500,16)-(100,16)-(100,32)-(50,32)-(50,64)-(25,64)-(25,128)-(25,1)	1min 4s	8.181	0.045	20.872	60:1
(1500,8)-(300,8)-(300,16)-(150,16)-(150,32)-(150,64)-(30,64)-(30,128)-(30,1)	57 s	7.597	0.044	21.194	50:1
(1500,8)-(500,8)-(500,16)-(100,16)-(100,32)-(50,32)-(50,64)-(50,128)-(50,1)	1min 14s	7.228	0.043	21.409	30:1
(1500,8)-(1500,16)-(750,16)-(750,32)-(750,64)-(75,64)-(75,128)-(75,1)	1min 7s	6.460	0.040	21.898	20:1
(1500,16)-(500,16)-(500,32)-(500,64)-(100,64)-(100,128)-(100,1)	1min 38s	4.417	0.033	23.549	15:1
(1500,16)-(1500,32)-(300,32)-(300,64)-(150,64)-(150,128)-(150,1)	52.7 s	3.362	0.029	24.734	10:1
(1500,16)-(1500,32)-(300,32)-(300,64)-(300,128)-(300,1)	41.4 s	1.199	0.017	29.211	5:1
(1500,16)-(500,16)-(500,32)-(500,64)-(500,128)-(500,1)	46.9 s	0.472	0.011	33.258	3:1
(1500,8)-(750,8)-(750,16)-(750,32)-(750,64)-(750,128)-(750,1)	59.9 s	0.086	0.005	40.666	2:1

gradually according to the optimal number of layers. For example, when CR = 30, the CAE encoder in model-I should consist of five layers with 8, 16, 32, 64 and 128 feature maps, respectively, and 128, 64, 32, 16 and 8, in the decoder part, respectively. However, for model-II, the optimal number of layers when CR = 30 is equal to two layers using 8, and 16 feature maps, respectively. Each feature map, for model-II, has a filter size of 3×1, and the stride is equal to 2. While, the dropout rate is 0.1. Similarly, the same procedure is done for different CRs for the two models to choose the best architecture that perfectly reconstructs the input seismic signal as illustrated in Table 1 and 2.

Adam optimizer with a learning rate equal to 0.001 is used for training the two models [45]. The first model is trained using 100 epochs and the number of epochs is doubled for the second model, with the same batch size of 10 for the two models.

Table 1 summarizes the experimental results of the first model (model-I) for various CRs. The encoder hidden layers optimal architecture is illustrated in the first column, e.g., the layer size $l1 - l2 - l3$-l corresponds to the convolutional auto-encoder architecture of $l1 - l2 - l3 - l - l3 - l2 - l1$. The compression ratio is obtained by dividing the number of input samples by the number of features at the layer l (bottleneck layer). For example, model-I with architecture (1500,16)-(1500,32)-(300,32)-(300,64)-(300,128)-(300,1) uses 300 units in the coded layer, then the compression ratio of this architecture is equal to 5:1 ($\frac{1500}{300}$). In addition, it is evident that the training time mainly depends on the number of feature maps in each hidden layer. Therefore, the larger number of hidden layers is, the longer training time is needed.

The optimal architectures have been investigated versus different CRs, as presented in Table 2. All model-II architectures finish the training within less than 1min except when the CR is less than or equal to 5.

The NMSE behavior in Table 1 partially differs from Table 2. In the first model, the NMSE value increases as the compression ratio increases, while this phenomenon only appears in the model-II starting from CR = 10 to 100. When CR = 2 to 5, the NMSE becomes almost constant. These results can be explained by the fact that the higher the compression, the higher the distortion in the signal. However, the performance of model II depends on the extracted feature maps. Starting from CR = 10 to 100, the second model can generate the features (150 to 15 consecutively) that accurately

TABLE 2. Performance of model-II for 475 training and 119 test traces at different compression ratios. The performance metrics: NMSE, NRMSE, and SNR are calculated for test data.

Number of layers	Encoder feature maps	Training time	NMSE (10^{-3})	NRMSE	SNR	CR
2	16	49.9 s	5.798	0.038	22.367	100
2	16	59.8 s	4.089	0.032	23.884	60
2	16	58.3 s	3.536	0.029	24.515	50
2	8	33.8 s	2.356	0.024	26.278	30
2	16	55.2 s	1.787	0.021	27.478	20
2	16	46.7 s	1.363	0.019	28.654	15
2	16	35.1 s	0.928	0.015	30.323	10
4	16	1min 31s	0.812	0.0143	30.905	5
4	16	1min 6s	0.730	0.0135	31.366	3
4	16	50.1 s	0.719	0.0134	31.435	2

reconstruct the original signal. However, when CR = 300, 500, and 750, model-II fails to represent the input signal from the produced feature maps since the feature maps are saturated. Table 1 and Table 2 reveal that the NRMSE values highly correlate with NMSE values and reflect the same attributes. The pseudo code summarizing the two proposed algorithms is shown in Algorithm 1.

Instead of using random data splitting, the K-fold cross-validation method is used to evaluate the proposed models. This method split data into K groups. One group is used for testing the data set and the remaining (K-1) groups are used for training. This process is repeated by K times while changing the dataset each time. Finally, the mean value of the performance, for all K test cases, is calculated. The K value is set equal to 10 for the two proposed models, and the mean of SNR values is calculated for 10-fold cross validation for different CR values as shown in Table 3. From CR = 2 to 5, for model-I, the SNR values are improved by at least 1dB. However, for the other CR cases in models I and II, the performance is almost the same in case of random data set splitting. Thus, the proposed models are robust and have a strong generalization ability.

C. RESULTS COMPARISON

We compare the two models (model-I and model-II) with the stacked auto-encoder ELM (ELM) method [16] that is used in seismic data compression recently. The best results of the ELM method are used for comparison. Also, the state-of-the-art method, discrete wavelet transform (DWT) method [2] is used to show the effectiveness of the proposed compression methods with varying CRs. The discrete wavelet transform (DWT) [46] is used as a state-of-the-art method. We tune the DWT parameters to reach the best performance. Accordingly,

Algorithm 1 Proposed Algorithms for Seismic Data Compression

Input: X (training data set), CR (compression ratio)
Output: Y (reconstructed input data)

1: Select a model and its optimal network architecture according to Tables 1 and 2.
2: Calculate the output of the encoder according to equation (1) and then get the decoder output Y according to equation (2).
3: Calculate the loss value between X and Y.
4: Use Adam optimizer to optimize model weights and biases.
5: Repeat step 2 to 4 until the loss value is saturated.

TABLE 3. Performance of model-I and model-II using 10-fold cross validation at different compression ratios.

CR	SNR model-I	SNR model-II
100	19.2564	20.8840
60	19.5367	22.2798
50	19.8393	22.7682
30	20.2291	24.6393
20	20.7029	26.2450
15	22.9201	27.3699
10	24.1591	29.2210
5	30.5541	30.0364
3	34.5096	30.4750
2	41.041836	30.545185

the best results for the DWT method are obtained using the Daubechies: 'db2' filter type and 4 levels of decomposition. Furthermore, the absolute values of wavelet coefficients are sorted in the ascending order, and the number of coefficients is determined corresponding to the required CR.

A sample of time series of the original and the reconstructed seismic traces from the first and the second models are compared with the ELM and DWT methods when CR = 10, 15, 30, and 50 as shown in Fig.4. Overall, model-II achieves obviously the highest quality of the reconstructed signal in comparison with the other methods at the selected CRs. For instance, in the zoomed time frame from 590 sec to 670 sec, model-II successfully reconstructs the original signal, while the reconstructed signal from the other three methods suffer from some distortions. Also, for the four models, the quality of the reconstructed signal decreases as the CR increases.

Fig.5 shows the SNR values of the proposed models (I and II), the ELM, and the DWT methods in decibel (dB)

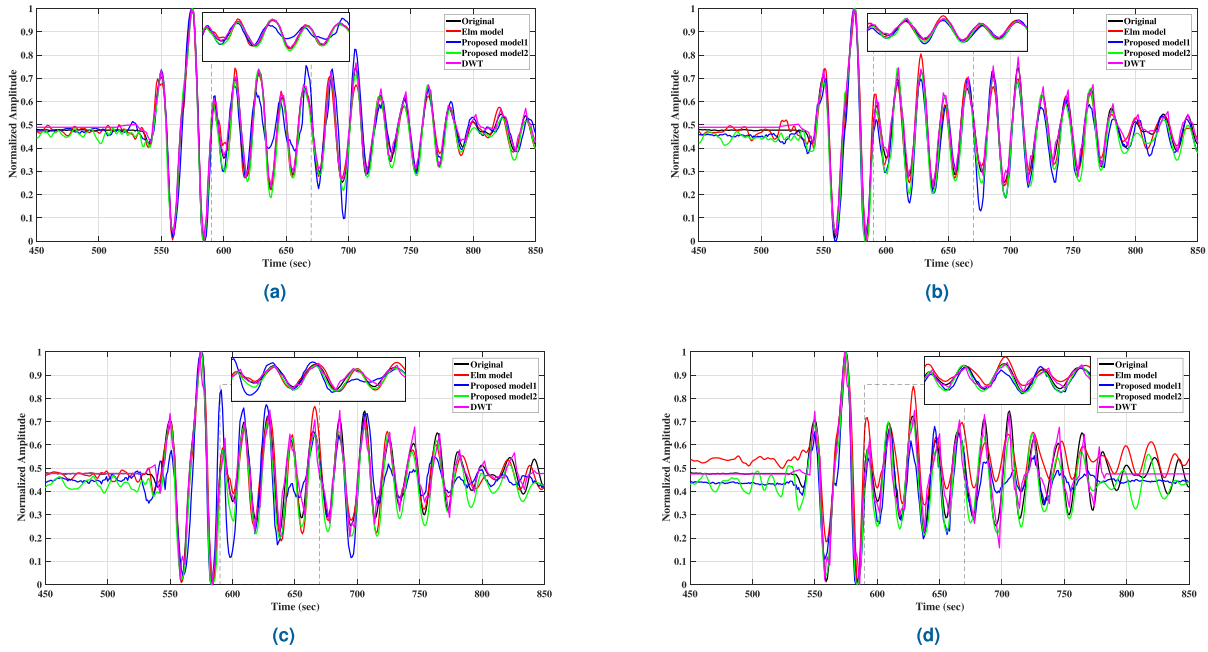


FIGURE 4. Reconstructed time sample at (a) CR = 10, (b) CR = 15, (c) CR = 30, and (d) CR = 50..

for different CRs. DWT achieves the highest SNR values for lower CR values. For example, from CR = 2 to 3 the values of SNR are equal to 53.0194 dB and 44.0675 dB, respectively. DWT is an effective method for data compression at low CRs because the DWT filter coefficients can represent the signal with high accuracy. However, the wavelet method consumes 0.0184 seconds to compress and reconstruct the 1500-samples seismic window, while model-I consumes only 0.0010 seconds to perform the same task. Afterward, the performance of ELM and DWT methods when CR = 5 is the same, approximately equal 35 dB. In contrast, DWT performance degrades dramatically as CR increases. DWT begins to behave poorly when CR becomes larger than 15. The SNR of the DWT is 21 dB and 7.4 dB when CR is 15 and 100, respectively. These results rely on the fact that the number of coefficients for signal representation decreases when the CR increases. Consequently, the reconstructed signal suffers from a high level of distortion. Regarding to the ELM method, the performance is limited to about 21 dB when CR is greater than or equal to 30. While, at low CRs, ELM achieves better performance than the two proposed models, contrary to the performance of the two proposed models at high CRs. It is obvious that, model-II achieves the maximum SNR values compared to other methods when CR increases from 10 to 100. For example, when CR = 30:1 and 50:1, model-II shows an improvement in SNR over the ELM by 3 dB for the two CRs, outperforming model-I by 5 dB and 3 dB, respectively, and the DWT method by at least 12 dB. However, the SNR for the second model is constant at almost 31 dB from CR = 2 to 5. This implies the limited performance of model-II at low CRs as opposed to the other methods. However, from CR = 2 to 5, the performance is saturated at almost 31 dB as

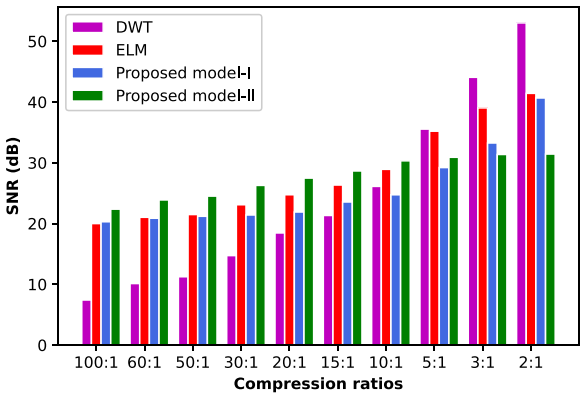


FIGURE 5. SNR as a function of compression ratios for all methods.

the extracted features cannot well represent the original data. For model-I, the SNR does not increase remarkably for high CRs. Since model-I at high CRs requires a lot of convolution filters and maxpolling layers, which causes significant losses of data details. Nonetheless, model-I outperforms model-II from CR = 2 to 3 by about 9 dB and 2 dB, respectively. In conclusion, the first model achieves higher SNR at low CRs, while the second model achieves better performance at moderate to high CRs.

IV. CONCLUSION AND FUTURE WORK

This study introduces two convolutional autoencoders for seismic data compression. The first model consists of symmetrical encoder-decoder parts that have the same number of hidden layers in each side. There is a bottleneck layer between the encoder and decoder stages, which contains the least

feature maps to represent the input data. Despite that the second model does not have a coded feature layer, the encoder extracts the significant features using a certain number of successive 1D convolutional layers. The decoder of the second model is fast and simple compared to the decoder of the first model. Model-II achieves the highest SNR at moderate to high compression ratios with about 3 dB increase in SNR compared to the ELM method, and by at least 12 dB in comparison with the discrete wavelet transform (DWT) method. While model-I achieves reasonable and fast performance at low compression ratios, it is recommended to use the DWT method for better performance. The proposed models are proved to be robust and have a strong generalization ability using the 10-fold cross validation method. For future work, the authors are going to use a bigger dataset from different seismic areas to further generalize the proposed models.

ACKNOWLEDGMENT

The authors would like to thank the editor and anonymous reviewers for their helpful suggestions to improve the paper quality. They would also like to thank Hilal H. Nuha, for providing them with the Matlab code for the ELM algorithm, which has been used for comparison with our work.

REFERENCES

[1] A. S. Spanias, S. B. Jonsson, and S. D. Stearns, "Transform methods for seismic data compression," *IEEE Trans. Geosci. Remote Sens.*, vol. 29, no. 3, pp. 407–416, May 1991.

[2] J. D. Villasenor, R. A. Ergas, and P. L. Donoho, "Seismic data compression using high-dimensional wavelet transforms," in *Proc. Data Compress. Conf. (DCC)*, Mar./Apr. 1996, pp. 396–405.

[3] A. Vassiliou and V. Wickerhauser, "Comparison of wavelet image coding schemes for seismic data compression," in *Proc. SEG Tech. Program Expanded Abstr.*, Jan. 1997, pp. 1334–1337.

[4] C. Fajardo, O. M. Reyes, and A. Ramirez, "Seismic data compression using 2D lifting-wavelet algorithms," *Ingeniería y Ciencia*, vol. 11, no. 21, pp. 221–238, 2015.

[5] A. Z. Averbuch, F. Meyer, J.-O. Stromberg, R. Coifman, and A. Vassiliou, "Low bit-rate efficient compression for seismic data," *IEEE Trans. Image Process.*, vol. 10, no. 12, pp. 1801–1814, Dec. 2001.

[6] L. C. Duval and T. Q. Nguyen, "Seismic data compression: A comparative study between GenLOT and wavelet compression," *Proc. SPIE*, vol. 3813, pp. 802–810, Oct. 1999.

[7] A. Z. Averbuch, V. A. Zheludev, M. Guttmann, and D. D. Kosloff, "LCT-wavelet based algorithms for data compression," *Int. J. Wavelets, Multiresolution Inf. Process.*, vol. 11, no. 5, Sep. 2013, Art. no. 1350032.

[8] B. Liu, M. Mohandes, H. Nuha, M. Deriche, and F. Fekri, "A distributed principal component analysis compression for smart seismic acquisition networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3020–3029, Jun. 2018.

[9] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[10] L. Bai, H. Lu, and Y. Liu, "High-efficiency observations: Compressive sensing and recovery of seismic waveform data," *Pure Appl. Geophys.*, vol. 177, no. 1, pp. 469–485, Jan. 2020.

[11] K. Bin, S. Luo, X. Zhang, J. Lin, and X. Tong, "Compressive data gathering with generative adversarial networks for wireless geophone networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 3, pp. 558–562, Mar. 2021.

[12] A. Payani, A. Abdi, X. Tian, F. Fekri, and M. Mohandes, "Advances in seismic data compression via learning from data: Compression for seismic data acquisition," *IEEE Signal Process. Mag.*, vol. 35, no. 2, pp. 51–61, Mar. 2018.

[13] X. Tian, "Multiscale sparse dictionary learning with rate constraint for seismic data compression," *IEEE Access*, vol. 7, pp. 86651–86663, 2019.

[14] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[15] H. Nuha, M. Mohandes, and B. Liu, "Seismic-data compression using autoassociative neural network and restricted Boltzmann machine," in *Proc. SEG Tech. Program Expanded Abstr.*, Aug. 2018, pp. 186–190.

[16] H. H. Nuha, A. Balghonaim, B. Liu, M. Mohandes, M. Deriche, and F. Fekri, "Deep neural networks with extreme learning machine for seismic data compression," *Arabian J. Sci. Eng.*, vol. 45, no. 3, pp. 1367–1377, Mar. 2020.

[17] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[18] A. P. Schiavon, K. Ribeiro, J. P. Navarro, M. B. Vieira, and P. M. C. E. Silva, "3-D poststack seismic data compression with a deep autoencoder," *IEEE Geosci. Remote Sens. Lett.*, early access, Oct. 13, 2020, doi: 10.1109/LGRS.2020.3028023.

[19] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[20] S. Li, B. Liu, Y. Ren, Y. Chen, S. Yang, Y. Wang, and P. Jiang, "Deep-learning inversion of seismic data," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 2135–2149, Mar. 2020.

[21] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[22] K. G. Lore, A. Akintayo, and S. Sarkar, "LLNet: A deep autoencoder approach to natural low-light image enhancement," *Pattern Recognit.*, vol. 61, pp. 650–662, Jan. 2017.

[23] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 1–38, 2010.

[24] R. Salakhutdinov and G. Hinton, "Semantic hashing," *Int. J. Approx. Reasoning*, vol. 50, no. 7, pp. 969–978, Jul. 2009.

[25] O. M. Saad, K. Inoue, A. Shalaby, L. Samy, and M. S. Sayed, "Automatic arrival time detection for earthquakes based on stacked denoising autoencoder," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 11, pp. 1687–1691, Nov. 2018.

[26] O. M. Saad and Y. Chen, "Deep denoising autoencoder for seismic random noise attenuation," *Geophysics*, vol. 85, no. 4, pp. V367–V376, Jul. 2020.

[27] M. S. Seyfioğlu, A. M. Özbayoğlu, and S. Z. Gürbüz, "Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 4, pp. 1709–1723, Aug. 2018.

[28] H. Abouzid, O. Chakkor, O. G. Reyes, and S. Ventura, "Signal speech reconstruction and noise removal using convolutional denoising audioencoders with neural deep learning," *Anal. Integr. Circuits Signal Process.*, vol. 100, no. 3, pp. 501–512, Sep. 2019.

[29] X. Fu, H. Luo, S. Zhong, and L. Lin, "Aircraft engine fault detection based on grouped convolutional denoising autoencoders," *Chin. J. Aeronaut.*, vol. 32, no. 2, pp. 296–307, Feb. 2019.

[30] G. Zhang, Z. Wang, and Y. Chen, "Deep learning for seismic lithology prediction," *Geophys. J. Int.*, vol. 215, pp. 1368–1387, Aug. 2018.

[31] G. Zhang, C. Lin, and Y. Chen, "Convolutional neural networks for microseismic waveform classification and arrival picking," *Geophysics*, vol. 85, no. 4, pp. WA227–WA240, Jul. 2020.

[32] Y. Wang, B. Wang, N. Tu, and J. Geng, "Seismic trace interpolation for irregularly spatial sampled data using convolutional autoencoder," *Geophysics*, vol. 85, no. 2, pp. V119–V130, Mar. 2020.

[33] S. Zu, J. Cao, S. Qu, and Y. Chen, "Iterative deblending for simultaneous source data using the deep neural network," *Geophysics*, vol. 85, no. 2, pp. V131–V141, Mar. 2020.

[34] O. M. Saad, A. G. Hafez, and M. S. Soliman, "Deep learning approach for earthquake parameters classification in earthquake early warning system," *IEEE Geosci. Remote Sens. Lett.*, early access, Jun. 9, 2020, doi: 10.1109/LGRS.2020.2998580.

[35] O. Saad and Y. Chen, "Automatic waveform-based source-location imaging using deep learning extracted microseismic signals," *Geophysics*, vol. 85, no. 6, pp. KS171–KS183, 2020, doi: 10.1190/geo2020-0288.1.

[36] O. Yildirim, R. S. Tan, and U. R. Acharya, "An efficient compression of ECG signals using deep convolutional autoencoders," *Cognit. Syst. Res.*, vol. 52, pp. 198–211, Dec. 2018.

[37] A. Z. Al-Marridi, A. Mohamed, and A. Erbad, "Convolutional autoencoder approach for EEG compression and reconstruction in m-health systems," in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2018, pp. 370–375.

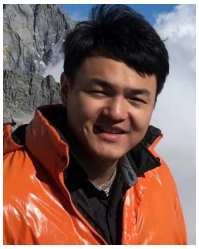
[38] W. A. Mousa and A. A. Al-Shuhail, "Processing of seismic reflection data using MATLAB," *Synth. Lectures Signal Process.*, vol. 5, no. 1, pp. 1–97, 2011.
[39] J. Brownlee, *Deep Learning With Python: Develop Deep Learning Models on Theano and TensorFlow Using Keras*. Machine Learning Mastery, 2016. [Online]. Available: https://books.google.com.eg/books?hl=en&lr=&id=K-ipDwAAQBAJ&oi=fnd&pg=PP1&dq=info:Rt5Wp0U00DgJ:scholar.google.com&ots=oqQvYH _lwq&sig=XcrlSMZdjXxLLNoY68lOb6qFnxo&redir_esc=y#v=onepage&q&f=false
[40] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Proc. Int. Conf. Artif. Neural Netw.*, 2010, pp. 92–101.
[41] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.
[42] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," 2014, *arXiv:1412.6806*. [Online]. Available: http://arxiv.org/abs/1412.6806
[43] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.
[44] M. J. Rubin, M. B. Wakin, and T. Camp, "Lossy compression for wireless seismic data acquisition," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 1, pp. 236–252, Jan. 2016.
[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[46] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. Inf. Theory*, vol. 36, no. 5, pp. 961–1005, Sep. 1990.

EMAD B. HELAL received the B.Sc. degree (Hons.) from the Institute of Aviation Engineering and Technology (IAET), Egypt, in 2009, and the M.Sc. degree in electronics and communications engineering from Cairo University, Egypt, in 2017. He is currently pursuing the Ph.D. degree in electrical engineering with Minia University, Egypt. He is also an Assistant Researcher with the National Research Institute of Astronomy and Geophysics (NRIAG), Helwan, Egypt. His research interests include seismic signal processing, deep learning, and cognitive wireless sensor networks.

OMAR M. SAAD received the B.S. and M.S. degrees in electrical engineering from the Arab Academy for Science, Technology & Maritime Transport (AASTMT), Alexandria, Egypt, in 2012 and 2015, respectively, and the Ph.D. degree in electrical engineering from the Egypt Japan University of Science and Technology (EJUST), Alexandria, in 2019. He is currently a Researcher with the National Research Institute of Astronomy and Geophysics (NRIAG), Helwan, Egypt. He also holds a postdoctoral position with the School of Earth Sciences, Zhejiang University, Hangzhou, China. He is also mainly engaged in machine learning in the field of seismology, earthquake early warning systems, and signal processing techniques.

ALI G. HAFEZ was born in Cairo, Egypt, in 1977. He received the B.Sc. and M.Sc. degrees in communication engineering from Minia University, Minia, Egypt, in 1999 and 2004, respectively, and the Ph.D. degree from Kyushu University, Fukuoka, Japan, in 2009.

He is currently an Associate Professor with the National Research Institute of Astronomy and Geophysics, Helwan, Egypt, and the Director of LTLab Inc., Fukuoka. His research interests include wavelet analysis of digital streams, installing and maintenance of seismic stations, and low and ultralow temperature-based machines.



YANGKANG CHEN (Member, IEEE) received the B.S. degree in exploration geophysics from the China University of Petroleum, Beijing, China, in 2012, and the Ph.D. degree in geophysics from The University of Texas at Austin, Austin, TX, USA, in 2015, under the supervision of Prof. S. Fomel.

From 2016 to 2018, he was a Distinguished Postdoctoral Research Associate with the Oak Ridge National Laboratory, Oak Ridge, TN, USA. He is currently an Assistant Professor with Zhejiang University, Hangzhou, China. He has authored more than 100 internationally renowned journal articles and more than 80 international conference papers. His research interests include seismic signal analysis, seismic modeling and inversion, simultaneous-source data deblending and imaging, global adjoint tomography, and high-performance computing.

GAMAL M. DOUSOKY (Senior Member, IEEE) was born in Minia, Egypt, in 1977. He received the B.Sc. and M.Sc. degrees in electrical and electronic engineering from Minia University, Egypt, in 2000 and 2004, respectively, and the Ph.D. degree in electrical and electronic engineering from Kyushu University, Japan, in 2010.

Since 2000, he has been associated with the Department of Electrical Engineering, Faculty of Engineering, Minia University. In March 2017, he was promoted to the position of an associate professor position. He worked as a Postdoctoral Research Fellow with Kyushu University for two years, where he is currently a Visiting Associate Professor. He authored and coauthored more than 75 publications, in international journals and conference proceedings of Power Electronics and Industrial Technologies. His research interests include power electronics, particularly renewable energy applications, energy efficiency, switching power supplies, electromagnetic interference/compatibility, and digital control.

Dr. Dousoky received the 2009 Excellent Student Award of the IEEE Fukuoka Section. He is also a reviewer in many international journals and conferences.