Smart Meter Data Compression and Reconstruction Using Deep Convolutional Autoencoders

Yuxuan Yuan, Qianzhi Zhang, Kaveh Dehghanpour, Fankun Bu, Zhaoyu Wang
Department of Electrical and Computer Engineering
Iowa State University
Ames, IA
yuanyx@iastate.edu

2020 52nd North American Power Symposium (NAPS) | 978-1-7281-8192-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/NAPS50074.2021.9449764

***Abstract*—Advanced Metering Infrastructure (AMI) enables utilities to gather vast amount of smart meter (SM) data, which can facilitate demand-side control, fault detection, and system monitoring. However, due to the bottlenecks caused by the burdensome data transmission and storage, taking full advantage of SM measurements is challenging. In response to this problem, this paper presents a deep convolutional autoencoder (DCA)-based data compression method to significantly reduce the volume of SM data while providing an efficient way to restore the original data for various power system applications. The temporal-spatial correlations within SM data are exploited in our method to enhance DCA performance. Further, a sensitivity analysis is conducted to obtain a tradeoff between the data compression ratio and reconstruction error. The proposed method has been tested and verified using real utility data.**

***Index Terms*—Deep convolutional autoencoder, data compression, smart meter, temporal-spatial correlation.**

I. Introduction

Advanced metering infrastructure (AMI) in distribution systems enables bidirectional information flow between utilities and customers, and collects fine-grained electricity consumption data from smart meters (SMs). According to the statistical data from the United States Energy Information Administration (EIA), more than 70 million SMs were installed in 2016 [1]. As the number of SMs increases, massive real-time data provides valuable knowledge of customer behaviors to utilities. However, utilities are still facing critical challenges in managing big data. For the small-to-medium-sized utilities that have thousands of customers, the total amount of SM data with 15-minute resolution can reach tens of terabyte (TB) per year, which causes a heavy burden in data transmission and storage [2]. Hence, it is imperative to develop efficient data compression models to alleviate transmission pressure, reduce storage overhead, and enhance data analysis [2].

Existing data compression studies in power systems can be classified into two categories: lossless vs. lossy algorithms. As the name suggests, lossless methods can completely restore all original data from the compressed data without loss of information. In [3], a wavelet pack transform method is proposed for denoising and compression of the metering data. In [4], the compressibility of several standard lossless algorithms is compared using the voltage and frequency data collected by the phasor measurement units (PMUs). However, the main drawback of lossless algorithms is that these methods have much lower compression ratios (CRs) than lossy models. CR is a critical metric for evaluating the performance of data compression algorithms, and is computed by dividing the size of the original data by the size of the compressed data [5]. In addition, the running times of lossless algorithms are longer than lossy approaches, thus leading to high costs in online implementation [2]. In the lossy compression methods, parts of the information are lost in the data recovery. In [5], a singular value decomposition (SVD)-based method is developed to reduce the volume of PMU data. In [6], a feature-based data compression method is proposed by identifying and restoring hidden load features rather than the original data values. In [7], a principal component analysis (PCA)-based algorithm is proposed to compress PMU data on a distribution network. One shortcoming of lossy compression methods is their sensitivity to the rapid changes in the volatile SM data. Thus, compression of house-level SM data is still a challenging problem.

To tackle the above shortcomings, a deep convolutional autoencoder (DCA)-based lossy method is proposed to obtain a compressed low-dimensional representation of house-level SM data of distribution grids. This allows utilities to handle the challenges posed by data transmission and storage. Unlike previous lossy approaches, the proposed method is capable of extracting hierarchical features of data using multiple hidden layers of deep learning models, thus, improving the quality of data compression. Also, the testing time of deep learning-based data compression methods is generally shorter than that of conventional methods, which is beneficial for online data processing. To circumvent the high parametric complexity of deep learning methods, convolution operation and parameter sharing scheme are utilized to reduce the number of parameters. Further, our method takes into account the temporal relationship of customer consumption data and the spatial correlation between the voltages of nearby customers to improve DCA performance. Based on the case study with real

This work is supported by the Advanced Grid Modeling Program at the U.S. Department of Energy Office of Electricity under DE-OE0000875.

Y. Yuan, Q. Zhang, K. Dehghanpour, F. Bu, and Z. Wang are with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA (e-mail: yuanyx@iastate.edu; wzy@iastate.edu)

data, it is demonstrated that the proposed method can achieve lower restoring errors with given compression ratios compared to the existing approaches.

II. Data Description and Overall Structure of the Proposed Method

A. Description of real SM data and data pre-processing

In this paper, the available SM data contains several U.S. mid-west utilities' hourly electricity consumption and voltage data for over 6000 customers [8]. The dataset includes around four-year energy measurements in kWh, from January 2015 to May 2018 and 16 months of voltage data in volt, from January 2017 to May 2018. Also, the geographical information of each SM (i.e., longitude and latitude) is included in our dataset. Over 95% of customers are residential and commercial loads, which have more various load patterns compared to industrial customers [9].

The data is initially processed through the data pre-processing. The goal of the data pre-processing is twofold: 1) perform data cleaning to mitigate missing and bad data problems caused by communication failure and meter malfunction. The missing and bad data are detected using the z-score [10]. The samples with z-scores outside a range of ± 5 are replaced by interpolation. 2) conduct min-max data normalization to perform standardization by rescaling the data to the range $[0, 1]$ [11].

B. Overall Data Compression Approach

The overall framework is presented in Fig. 1. Considering the rapidly fluctuating consumption behaviors of individual customers, a deep learning technique, DCA, is leveraged to perform SM data compression and reconstruction. After data pre-processing (see the previous subsection), a data transformation algorithm is applied to reshape the data into 2-dimensional image formats while capturing the temporal relationship of customer consumption and the spatial correlation between the voltages of nearby customers. Next, these data images are served as the training inputs of different DCAs. Each DCA is composed of two modules, corresponding to an encoder and a decoder, respectively. The encoder is utilized to discover the latent data features from energy/voltage data images and embed these features into a low-dimensional space, which is called *embedded feature*. In contrast, the decoder is tuned to reconstruct the original data from the embedded feature through the minimization of a cost function while providing an efficient way to perform data reconstruction. Taking into account that the DCA model is a complex structure, we have applied a random search strategy to calibrate the hyper-parameters of DCAs [12]. After training model, embedded features of the encoder have a much lower dimensionality compared to the original data and can be transmitted and restored efficiently. Finally, a sensitivity analysis is conducted to obtain an adequate trade-off between the CR and reconstruction error.

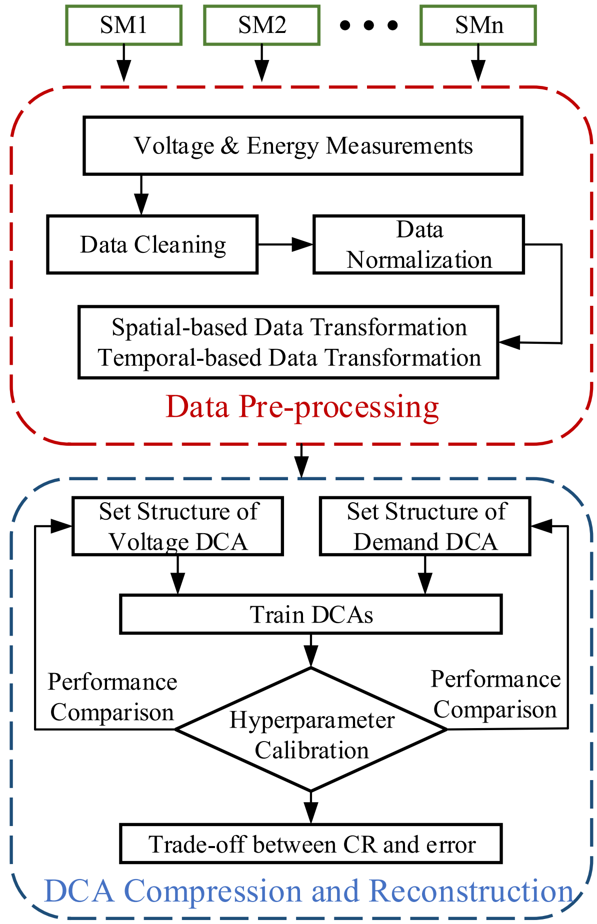


Fig. 1: The flowchart of the proposed model.

III. DCA-based SM Data Compression and Reconstruction

In order to develop an efficient SM data compression method that achieves high CR while maintaining excellent fidelity, we propose a DCA-based approach by adopting the recently-developed deep learning technique. Let p_i^h and v_i^h denote the historical energy and voltage data sample expressed at time index h recorded by the i'th SM, respectively. Due to the consistency of customer behavior, it is important to discover the temporal relationship between consecutive energy data samples. To conduct this, a method is developed to choose an optimal time-window for energy data compression, considering the tradeoff between the window length and CR; a small data window cannot contain sufficient temporal information, thus resulting in a poor CR. Inversely, a large data window indicates long execution time and high parametric complexity. Here, the length of window is selected as 672 hours, which contains the load data of each customer for a period of 28 days, $\boldsymbol{P_i}(j) = [p_i^{1+672\times(j-1)}, ..., p_i^{672\times j}]$, where j is the index of the energy load profile. Then, each load profile data, $\boldsymbol{P_i}(j)$, is converted to a 2-dimensional *energy image*, $\boldsymbol{P_I}(j) \in \mathbb{R}^{28\times 24}$. Using these energy images, an energy-based DCA is trained

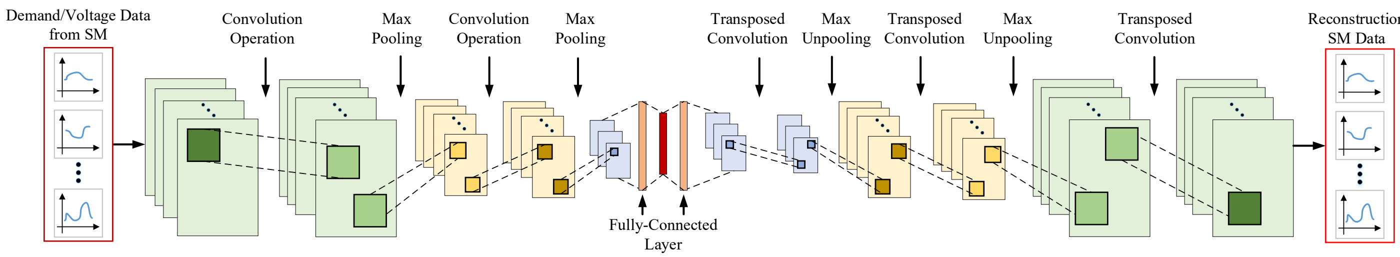


Fig. 2: DCA-based SM data compression and reconstruction.

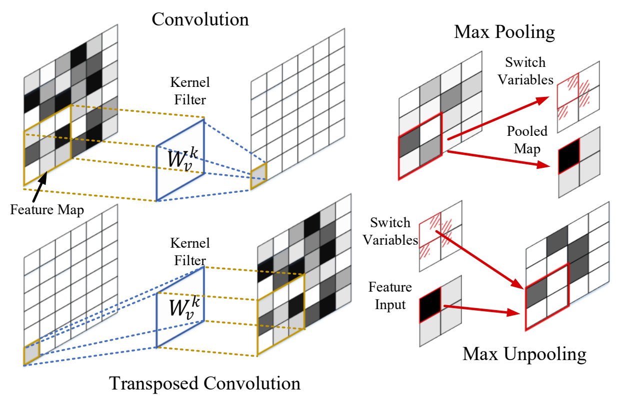


Fig. 3: Illustration of convolution, max pooling, transposed convolution, and max unpooling.

to perform data compression and reconstruction.

Voltage measurement of single SM is not only determined by the corresponding load but also impacted by the rest of loads at the same feeder, which indicates existence of spatial correlation. Thus, to exploit this spatial correlation of voltages, SMs that belong to the same feeders are clustered into a single training set; the distances from SMs to the substation are calculated. At each time h, the voltage data samples in feeder k are organized in the descending order with respect to these distances. Then, the sorted voltage data is transformed into a 2-dimensional *voltage image*, $\boldsymbol{V_{k,I}}(h) \in \mathbb{R}^{A\times B}$, where A and B are determined by the size of feeders. Note that for both voltage and energy data the phase information can be preserved by training different DCAs for each phase separately.

As an unsupervised learning algorithm, DCA consists of an encoder-decoder paradigm and convolutional neural networks (CNNs). Here, our proposed model includes multiple layers (see Fig. 2), such as convolutional layers, pooling layers, fully-connected layers, transposed convolutional layers and unpooling layers. The objective function of the energy/voltage-based DCA is to minimize the reconstruction error as follows:

$$\min_{\boldsymbol{W'},\boldsymbol{b'},\boldsymbol{W},\boldsymbol{b}} \frac{1}{n}\sum_{z=1}^{n} ||D_{\boldsymbol{W'},\boldsymbol{b'}}(E_{\boldsymbol{W},\boldsymbol{b}}(x_z)) - x_z||_2^2 \quad (1)$$

where, n is the number of data samples, x_z is the z'th data samples that can be $\boldsymbol{P_I}(j)$ or $\boldsymbol{V_{k,I}}(h)$, $E_{\boldsymbol{W},\boldsymbol{b}}(\cdot)$ and $D_{\boldsymbol{W'},\boldsymbol{b'}}(\cdot)$ are the mathematical models for the deep convolutional encoder and decoder, and $\{\boldsymbol{W},\boldsymbol{b}\}$ and $\{\boldsymbol{W'},\boldsymbol{b'}\}$ are the parameters of the encoder and decoder, respectively. The purpose of the encoder and the decoder is to perform data compression and reconstruction, respectively. Compared with the conventional artificial neural networks (ANNs), the proposed encoder not only has the typical fully-connected layer but also adopts the convolutional and pooling layers [13]. Specifically, the function of convolutional layer can be mathematically described as follows [14]:

$$\phi_g^f = \sigma(\sum_{l\in L} x_{g-1}^l * W_g^f + b_g^f) \quad (2)$$

where, ϕ_g^f is the latent representation of the f'th feature map of the g'th layer, σ is a nonlinear activation function (e.g., sigmoid, hyperbolic tangent, or parametric rectified linear unit (PReLU)), x_{g-1}^l is the l'th feature map of the previous layer, L is the total number of feature maps, W_g^f and b_g^f are the kernel filter and the bias of the f'th feature map of the g'th layer, respectively. Here, due to the 2-dimensional energy and voltage data images, convolution operation $*$ can be written as follows [11]:

$$(x_{g-1}^l * W_g^f)(i,j) = \sum_{\delta_i=0}^{L-1}\sum_{\delta_j=0}^{L-1} x_{g-1}^l(i-\delta_i, j-\delta_j) W_g^f(i,j) \quad (3)$$

where, i and j are the row and column indices of the data image. Thus, each location of the input data x_{g-1}^l is processed by the kernel filter W_g^f. The convolution process is determined by two types of parameters: horizontal and vertical strides, which represent the amount of movement between applications of the kernel filter to the feature map, are set to 1. Thus, the size of the output feature can be calculated as: $\phi_g^f \in \mathbb{R}^{(m-n+1)\times(m-n+1)}$, where the sizes of x_{g-1}^l and W_g^f are $m\times m$ and $n\times n$, respectively. This indicates that the feature map shrinks in every convolutional layer. Furthermore, the impact of the data sample located on the border of data image is much smaller than those in the center, which results in information loss. To tackle this problem, a *padding strategy* is utilized by adding an additional border to the feature maps [15].

The outcomes of g'th convolutional layer are served as the inputs of a max-pooling layer. This layer pools features by taking the maximum value of input, which reduces the width and height of feature maps based on the size of pooling kernel [14]. In the pooling layer, switch variables are stored to describe the positions of these max-pooled features that can provide useful information in the reconstruction step. Through the processes of multiple convolutional and pooling layers, the original information is transformed into the embedded features, which has a much lower dimension than the original data.

Then, the decoder is developed to restore the original data using the outputs of the encoder, which includes the fully-connected, transposed convolutional, and unpooling layers. In general, as shown in Fig. 3, the functions of the transposed convolutional and unpooling layers are the opposite of convolutional and pooling layers. Specifically, a transposed convolutional layer carries out a regular convolution operation but reverts its spatial transformation. This indicates that the transposed convolutional layer can be thought of as the gradient of convolution with respect to its input [16]. Owing to the non-invertible property of pooling strategy, the unpooling layer can be considered as an approximate inverse by identifying the switch variables to record the locations of the maxima within each pooling region [17]. Using the transposed convolutional and unpooling layer, the decoder is capable of restoring the compressed representation into the original data, which achieves data reconstruction. One notable advantage of DCA is the reduction in the number of parameters, thus handling the parametric complexity challenge of deep learning-based methods. The rationale behind this is using convolutional layers besides conventional fully-connected layers. In addition, as shown in Fig. 3, the neurons within a particular feature map of the DCA share the same weights, which further contributes to reducing the parametric complexity.

For each DCA, the dataset is randomly partitioned into three subsets for training (70% of the total data), validation (15% of the total data), and testing (15% of the total data). To calibrate the hyperparameters of DCA, a random search strategy is utilized to find better hyperparameter combinations [12]. Compared to the conventional grid search, random search strategy is more efficient for the hyperparameter setting of deep learning models. In this paper, the number of convolutional layers, the size of kernel filters, and the types of backpropagation algorithms are determined using the calibration results from random search. Further, to tackle the *overfitting* problem of multi-layer structure, we have adopted a dropout strategy to randomly remove neurons of each layer in the training process [11].

IV. Numerical Results

The proposed DCA-based data compression method is tested using our real SM data described in section II-A. In this case study, based on the results of the random search, the encoder consists of two pairs of convolutional and pooling layers followed by one convolutional and a fully-connected layer. The decoder involves one fully-connected and a transposed convolutional layer followed by two pairs of transposed convolutional and unpooling layers. The activation functions for hidden layers and output layer are selected as the PReLU and sigmoid, respectively [11]. The Nesterov-accelerated Adaptive Moment Estimation (Nadam) backpropagation method is used to update the weight and bias variables of the kernel filters [11].

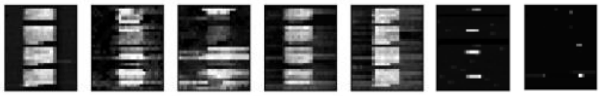
TABLE I: Sensitivity analysis of various CR based on MSE and MAE.

Energy Data Compression				**Voltage Data Compression**			
Structure	CR	MSE	MAE	Structure	CR	MSE	MAE
Conv(3,3,16) Conv(6,6,8) Conv(3,3,8)	2	0.00488	0.0437	Conv(3,3,16) Conv(6,6,8) Conv(3,3,8)	4.5	0.00353	0.0404
Conv(3,3,16) Conv(6,6,8) Conv(3,3,4)	4	0.00629	0.0491	Conv(3,3,16) Conv(6,6,8) Conv(3,3,4)	9	0.00363	0.04137
Conv(3,3,16) Conv(6,6,8) Conv(3,3,2)	8	0.00928	0.06057	Conv(3,3,16) Conv(6,6,8) Conv(3,3,2)	18	0.00389	0.04224
Conv(3,3,16) Conv(6,6,8) Conv(3,3,1)	16	0.0139	0.0754	Conv(3,3,16) Conv(6,6,8) Conv(3,3,1)	36	0.0048	0.0478

After training the energy and voltage-based DCAs, the performance of the proposed SM data compression method is evaluated over the testing set and governed by the reconstruction error and CR. The CRs are computed based on the dimensions of the original and compressed data. The mean square error (MSE) and mean absolute error (MAE) are utilized as evaluation metrics to assess the reconstruction errors [14].

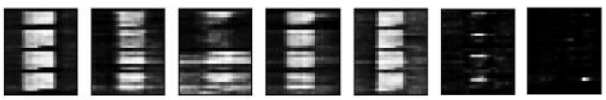
Table I presents CRs, MSEs, and MAEs of eight different DCAs. Note that the hyperparameters of a single convolutional layer are denoted as (a, b, c), where the first two numbers are the height and the width of the kernel filter and the third is the number of kernels. As is demonstrated by this table, when $CR = 2, 4, 8, 16$, MSEs of reconstructed energy data are $0.00488, 0.00629, 0.00928, 0.0139$, which shows the satisfactory performance of the proposed method. Based on Table I, it is observed that the qualities of voltage reconstruction are generally better than those of energy data. As an example, when $CR = 36$, the largest voltage MSE is obtained, which is still lower than the smallest energy MSE. The intuition behind this is the smaller voltage variation in the distribution system.

In order to further show the performance of the proposed method, two existing lossy data compression methods, PCA [7] and deep autoencoder [14] are compared with our approach. Note that the testing data of these two methods are also processed by our data transformations, which means the impacts of temporal/spatial relationships are eliminated. In all methods, the CRs are fixed to 8 and 36 for energy and voltage data compression, respectively. The reconstructed examples of energy and voltage are presented in Fig. 4 and Fig. 5. Fig. 4a shows original energy data in the 2-dimensional image format and Fig. 4b-4d demonstrate the reconstructed data by three algorithms. Comparing Fig. 4b and Fig. 4d, it can be seen that the quality of the reconstructed data from DCA is better than PCA. Specifically, the conspicuous differences are easy

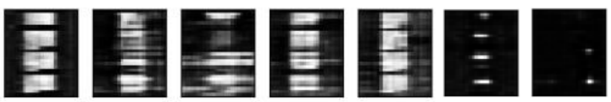


(a) Original energy data in 2-dimensional image format.

(b) Reconstructed energy data by PCA.



(c) Reconstructed energy data by deep autoencoder.



(d) Reconstructed energy data by the proposed DCA.

Fig. 4: Comparison of energy data compression results.

to observe in the last two data images. On the other hand, the reconstructed data of DCA is comparable with that of the deep autoencoder. However, in this case, the number of parameters in the deep autoencoder is around $400k$ which is 50 times higher than the number of parameters in DCA: $8k$. Thus, compared with deep autoencoder, our method can achieve similar reconstructed results with a limited number of parameters, thus reducing the risk of overfitting. Fig. 5a-5d present the original voltage data and the reconstructed data of three algorithms. Visually, all methods have accurate reconstruction results due to the small variation of voltage in the same feeder. According to the evaluation metrics, the MSE of DCA, 0.00389, is slightly lower than the MSE of PCA 0.00475. Consequently, given the fixed CRs, based on this AMI dataset, DCA has better accuracy for SM data reconstruction compared to the previous works.

V. Conclusion

In this paper, we have presented a DCA-based method for SM data compression and reconstruction to improve the efficiency of data transmission and storage. Using our approach, it is demonstrated that by exploiting the temporal-spatial relationship in different types of metering data, accurate data reconstruction can be achieved, while the number of parameters can be reduced to address the overfitting problem. The proposed method is successfully validated using real SM data and is shown to have better performance compared to the existing methods in the literature.

References

[1] Energy Information Administration, "Advanced metering count by technology type," 2017.

[2] L. Wen, K. Zhou, S. Yang, and L. Li, "Compression of smart meter big data: A survey," *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 59–69, 2018.

[3] J. Khan, S. Bhuiyan, G. Murphy, and J. Williams, "Pmu data analysis in smart grid using wpd," *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, pp. 2438–2444, Oct. 2014.

[4] P. Top and J. Breneman, "Compressing phasor measurement data," *IEEE Power Energy Society General Meeting*, pp. 1–4, Jul. 2013.

[5] J. C. S. de Souza, T. M. L. Assis, and B. C. Pal, "Data compression in smart distribution systems via singular value decomposition," *IEEE Trans. Smart Grid*, vol. 8, pp. 275–284, Jan. 2017.

[6] X. Tong, C. Kang, and Q. Xia, "Smart metering load data compression based on load feature identification," *IEEE Trans. Smart Grid*, vol. 7, pp. 2414–2422, Sep. 2016.

[7] P. H. Gadde, M. Biswal, S. Brahma, and H. Cao, "Efficient compression of pmu data in wams," *IEEE Trans. Smart Grid*, vol. 7, pp. 2406–2413, Sep. 2016.

[8] F. Bu, Y. Yuan, Z. Wang, K. Dehghanpour, and A. Kimber, "A time-series distribution test system based on real utility data," *North American Power Symposium*, pp. 1–1, Oct. 2019.

[9] Y. Yuan, K. Dehghanpour, F. Bu, and Z. Wang, "A multi-timescale data-driven approach to enhance distribution system observability," *IEEE Trans. Power Syst.*, vol. 34, pp. 3168–3177, Jul. 2019.

[10] K. Dehghanpour, Y. Yuan, Z. Wang, and F. Bu, "A game-theoretic data-driven approach for pseudo-measurement generation in distribution system state estimation," *IEEE Trans. Smart Grid*, pp. 1–1, 2019.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[12] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, Feb. 2012.

[13] S. Ryu, H. Choi, H. Lee, and H. Kim, "Convolutional autoencoder based feature extraction and clustering for customer load analysis," *IEEE Trans. Power Systems*, pp. 1–1, 2019.

[14] T. V, C. E, and L. A, "A deep convolutional auto-encoder with pooling-unpooling layers in caffe," *arXiv preprint arXiv:1701.04949*, 2017.

[15] Y. Qian, M. Bi, T. Tan, and K. Yu, "Very deep convolutional neural networks for noise robust speech recognition," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 24, pp. 2263–2276, Dec. 2016.

[16] D. Vincent and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[17] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *European conference on computer vision*, pp. 818–833, 2014.

(a) Original voltage data in 2-dimensional image format.

(b) Reconstructed voltage data by PCA.

(c) Reconstructed voltage data by deep autoencoder.

(d) Reconstructed voltage data by the proposed DCA.

Fig. 5: Comparison of voltage data compression results.