

Deep Lossless Compression Algorithm Based on Arithmetic Coding for Power Data

Zhoujun Ma ^{1,2}, Hong Zhu ¹, Zhuohao He ^{3,*}, Yue Lu ³ and Fuyuan Song ³ 

¹ State Grid Jiangsu Electric Power Co., Ltd., Nanjing Power Supply Branch, Nanjing 210019, China; mazhoujungcc@163.com (Z.M.); zhuhong99@vip.sina.com (H.Z.)

² College of Energy and Electrical Engineering, Hohai University, Nanjing 210098, China

³ Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science and Technology, Nanjing 210044, China; 20211220022@nuist.edu.cn (Y.L.); fysong@nuist.edu.cn (F.S.)

* Correspondence: 20201249413@nuist.edu.cn

Abstract: Classical lossless compression algorithm highly relies on artificially designed encoding and quantification strategies for general purposes. With the rapid development of deep learning, data-driven methods based on the neural network can learn features and show better performance on specific data domains. We propose an efficient deep lossless compression algorithm, which uses arithmetic coding to quantify the network output. This scheme compares the training effects of Bi-directional Long Short-Term Memory (Bi-LSTM) and Transformers on minute-level power data that are not sparse in the time-frequency domain. The model can automatically extract features and adapt to the quantification of the probability distribution. The results of minute-level power data show that the average compression ratio (CR) is 4.06, which has a higher compression ratio than the classical entropy coding method.

Keywords: Long Short-Term Memory; transformer; data compression; smart grid; arithmetic coding



Citation: Ma, Z.; Zhu, H.; He, Z.; Lu, Y.; Song, F. Deep Lossless Compression Algorithm Based on Arithmetic Coding for Power Data. *Sensors* **2022**, *22*, 5331. <https://doi.org/10.3390/s22145331>

Academic Editor: Fang Yang

Received: 16 June 2022

Accepted: 13 July 2022

Published: 16 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The power grid system collects huge power consumption data, which brings huge resource consumption to the communication in the transmission network. Strengthening the management and detection of the smart grid will further improve its work efficiency. Isaías González [1] proposed a multi-layer architecture with six functional layers as a new attempt at a smart grid monitoring system. The encryption of private files in any system [2,3] can further improve its supervision ability. Maik plenz [4] built a privacy framework for smart meters and increased management of smart grid security. The security of privacy files is guaranteed by the perturbation method of Gaussian distribution, and proposed two methods to encode power distribution datasets.

Different from industrial power data, household power data is collected in minutes. Tightiz and Yang [5] analyzed the characteristics and performance of the Internet of things protocol in the smart grid, which will provide a reference for the subsequent analysis of the smart grid. Huang [6] proposed an algorithm based on a depth automatic encoder, which realizes the classification and compression of power load data by extracting hierarchical features. This is helpful for further analysis of smart grid. In addition, the scheme in [7–9] utilized compression methods to smart grid load data. K-means singular value decomposition (K-SCD) [10] sparse representation technology is used to compress the load data of intelligent instruments.

Many data compression methods have been developed, such as the classic linear predictive coding method [11] and the discrete cosine transform (DCT) [12]. For data compression in power systems, a phasor measurement unit (PMU) data compression method based on wavelet transform is proposed in [13], and Das [14] introduced a power system

data compression method based on principal component analysis (PCA). Mehra [15] proposed a compression scheme based on PCA. However, these methods did not significantly improve the compression of minute-level power data.

In addition, the calculation capacity of electricity meters in residential areas is weak. To ensure transmission efficiency, it is selected to transmit the power consumption data of multiple users per minute. This makes the data not sparse in the time domain and frequency domain. Traditional sampling methods [12–14] are difficult to obtain good compression results. How to efficiently compress and store these power data has become a key social problem. It is desirable to design an efficient compression algorithm to reduce the communication cost and storage overhead.

Our contributions are summarized below:

- We propose a deep lossless compression algorithm for minute level power data to compress household power data of a smart grid;
- We analyze the learning effect of networks on power data. The performance evaluation experiments of compression ratio and entropy show that deep learning will improve the coding efficiency.

We introduce the related work in Section 2. Section 3 shows the background of the deep learning network and our method is introduced in Section 4. Section 5 shows the experimental results to evaluate the performance of the scheme over power data. Finally, the conclusion and future research direction of this method are summarized in Section 6.

2. Related Work

Lossless compression algorithms [16,17] have been widely used. Although the compression performance is poor, and the compression speed is very superior, they do not particularly consider the nature of the power data. Power data has the characteristics of repetitiveness and slow change. The emergence of entropy coding provides a new method for lossless compression. Shannon [18] proved that the limit of compression rate is the entropy rate. Huffman coding [19] can build a tree based on the probability distribution, and the symbols with a large number of occurrences should be ahead as far as possible, but Huffman coding can not reach the entropy limit. Arithmetic coding [20] can represent a group of characters as floating-point numbers in intervals [0,1] through probability distribution, which is closest to the entropy limit.

Ringwelski [21] proposed the Huffman coding method for smart meter data compression. Das [22] proposed four arithmetic coding schemes for compressing power data. Sarkar [23,24] developed two compression algorithms based on Huffman and basic arithmetic coding for compressing combined data arrays. However, the performance of these traditional entropy coding algorithms should be improved.

The technology based on wavelet transformation has gradually matured. Khan [25] proposed a technology based on embedded zerotree wavelets (EZWT) to achieve smart grid data denoising and compression. Khan [26] developed a smart grid data denoising and compression algorithm based on Wavelet Packet Decomposition (WPD), which expanded the Wavelet Decomposition (WD) tree into a complete binary tree. Ji [27] improved the existing wavelet transform and then proposed a general data compression method for different types of signals in the power system. Cheng [28] proposed a method for compressing oscillation wide-area measurement data based on wavelet transform, which selects the wavelet function and decomposition scale according to the oscillation frequency of the power system in the wide-area measurement system. Prathibha [29] proposed a dual-tree complex wavelet transform method for power quality monitoring and combined with run-length coding technology to compress the disturbing data. Ruiz [30] used a six-level biorthogonal wavelet transform method to improve the compression rate of compressed power quality signals. However, the above methods are only suitable for sinusoidal signals, and most algorithms compress the phasor measurement unit (PMU) signal measured by the power plant.

In recent years, some new compression algorithms have been proposed. Gontijo [31] compresses power quality data with disturbances through segmentation, transformation, quantization, and entropy coding. He [32] combines cross-entropy and singular value decomposition to compress PMU signals. Karthika [33] proposed a data compression algorithm for an intelligent power distribution system based on singular value decomposition, which reduces the mean square error (MSE) after compression. Differential binary encoding algorithm (DBEA) [34] proposed a differential binary coding method with low computational load and high compression ratio. Sarkar [35] expanded the data input range on this basis, but sacrificed compression performance. Then, Sarkar [36] analyzed the performance of different types of DBEA algorithms, but DBEA can maximize performance only when there are many repetitive elements, and the effect is very poor in actual phase current and power data. Abuadbba [37] and Tripathi [38] used Gaussian approximation to improve the traditional encoding method to compress smart meter data. Experiments showed that the compression performance of the improved encoding method is significantly improved. However, the algorithms of the DBEA series are suitable for having a good effect in the case of repeated data, and the effect is not good on uneven data.

Recurrent neural network (RNN) and transformer are very effective in learning time series data. They can be developed into machine translation and language modeling. Both models can learn the characteristics of data, input one character, and output the prediction probability of the next character. Arithmetic coding can be coded according to the prediction probability. The more accurate the predicted next character is, the better the compression performance of arithmetic coding.

The neural network can learn complex data. Stanford [39] combined a cyclic neural network with arithmetic coding to allocate a large probability for the possible next word and improve the performance of lossless compression. Liu [40] introduced cyclic connections for the network, retained the end of the hidden state, improved the ability of the model to capture long-term dependencies, and then used arithmetic coding to improve the compression performance. Wang [41] compressed human DNA sequences using this lossless compression method. These methods can be used for reference in power data compression.

3. Background

Using the measurement matrix to compress the original vector is the mainstream algorithm in the field of image compression. Although the data reconstructed by convolution neural network may have certain deviations, the success of this method is mainly due to the characteristics of human vision. However, we found that for small power consumption data, a small deviation will affect the economic benefits of the household power grid. We chose other networks to build lossless compression algorithms.

3.1. Bi-LSTM

RNN shows satisfying performance in natural language processing and is proved to be feasible in data compression [39]. As an improved branch of primitive RNN. Bi-LSTM has a better training effect than RNN. It controls the output of neurons through some gates, to solve the problems of gradient disappearance and gradient explosion in RNN. Its calculation formula is:

$$\begin{aligned} I_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \\ C_t &= f_t \cdot C_{t-1} + I_t \cdot \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \\ O_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\ h_t &= O_t \cdot \tanh(C_t) \end{aligned} \quad (1)$$

where t represents the time, W is the weight matrix, and I_t, f_t, c_t, o_t, h_t are input gate, forget gate, cell state, output gate, and hidden state, respectively.

3.2. Transformer

Transformer, as a recently proposed network, can get the same training effect and has different advantages from LSTM in some aspects. It uses the multi-head self-attention mechanism, which will have a better effect when the training interval is long. The calculation formula is:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(H_1, H_2)W^O \quad (2)$$

where H_1 and H_2 are the two self-attention modules of multi-head attention, and W^O is the weight. The calculation formula for self-attention is:

$$H_i = \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3)$$

where Q is query matrix, K is the content that you want to pay attention to, QK^T is the dot product operation. The purpose is to calculate the attention weight of Q on V , $\sqrt{d_k}$ is the scale of the dot product. Finally, the linear transformation layer and softmax layer of the decoder is output to obtain the predicted probability.

These two models can obtain similar results in the training sequence data. Although their structures are different, we can use any of them in the design of depth algorithm. Our compression algorithm is scalable and can be inserted into other training models to capture potential power features.

4. Proposed Method

Before building the model, we need to construct the data model. First, we cut the data, and the length of each segment can be 100, which depends on the setting of the smart grid equipment. For the power data with length n , we can model it as:

$$X = \{x_1, x_2, \dots, x_n\} \quad (4)$$

where x_i represents the electricity consumption data generated at the i -th moment.

In our proposed algorithm, the Bi-LSTM mode is used, which has significant advantages in time transmission. In the optional model, we analyzed and compared their performance. As shown in Figure 1, the algorithm framework we designed consists of two parts: model training and coding compression. The model includes bidirectional neurons, full connection layer (FC), and softmax layer. The compression part consists of arithmetic coding and bit coding. x_i is displayed at the top of the figure as input. After processing the input with the Bi-LSTM model, the softmax layer calculates and outputs the prediction probability. Then, the prediction probability is compressed into floating-point data by arithmetic coding and finally converted into the binary stream by bit coding. We express the calculation formula of output prediction probability of the softmax layer as:

$$p(x_i) = \text{softmax}(Wx_i + b) \quad (5)$$

where $p(x_i)$ represents the prediction probability of x_{i+1} , W is the weight matrix, and b represents the deviation.

In the power data, the extraction of two-way spatial features can better capture the front and back dependencies, thereby increasing the accuracy of the prediction accuracy. In the training process, our loss function uses cross-entropy:

$$L = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y(x_{ij}) \log(p(x_{ij})) \quad (6)$$

where $y(x_{ij})$ represents the true probability distribution, $p(x_{ij})$ represents the predicted probability distribution, M represents the length of the sequence, and N is the number of training batch samples.

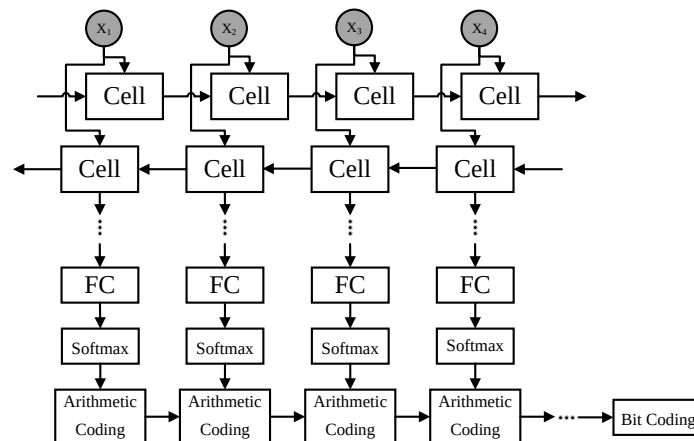


Figure 1. Proposed deep lossless compression algorithm framework. The upper part of the figure is the bilstm model, which includes bidirectional neurons, FC layer, and softmax layer. The lower part is Arithmetic Coding and Bit Coding.

The compression process of arithmetic coding [18] in our framework is shown in Figure 2. Arithmetic coding compresses the output of the model, expressing the data in the interval of $[0, 1]$, and the output are converted to floating-point type. This entropy coding is very efficient. The output probability of the model is context-adaptive, the prediction probability is closer to the source probability, the total entropy obtained is smaller, the value range of the final coding interval is wider, and the compression effect will be better than static arithmetic coding.

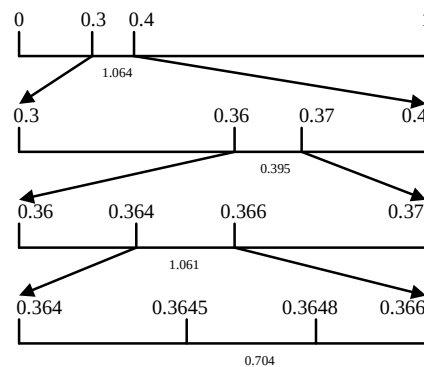


Figure 2. Encoding a set of data $[1.064, 0.395, 1.061, 0.704]$ with arithmetic coding.

As shown in Figure 3. In the last output probability interval, the left and right boundaries are converted to binary, and the final compression result is intercepted in the interval, which we call Bit encoding. The decoder is responsible for restoring the compression result to the original data. It needs the first character of the original data and the compression result as input, and it also uses iteration to restore the original data. Among them, the distribution of probability determines the entropy, increases the utilization efficiency of the interception section, and plays a role in determining the compression performance. Adaptive arithmetic coding can obtain a coding interval closer to perfect.

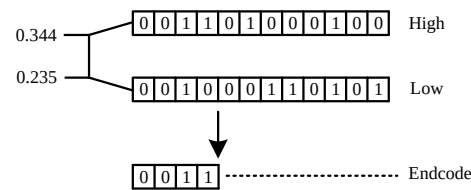


Figure 3. Bit coding.

We use the compression ratio given in Equation (7) as an evaluation criterion to compare the performance of our method with other methods:

$$CR = \frac{\text{originaldata}(\text{inbytes})}{\text{encodeddata}(\text{inbytes})} \quad (7)$$

Entropy, like the compression ratio, is also an important parameter of the compression algorithm, which is the expected value of the information contained in each message. In some cases, it can reflect the compression performance of the algorithm [36] for the cutting segment X containing different N characters. We use entropy as another performance metric, and the calculation formula is:

$$P(X) = - \sum_{i=1}^N p(x_i) \log_b p(x_i) \quad (8)$$

where $p(x_i)$ is the prediction probability, N is the number of different characters, x_i is the selected data, and the base number $b = 2$.

5. Experimental Results

In this section, to better test the impact of the model on compression performance, we use a private dataset, which contains user electricity data collected by a power supply company in different stations. When training the data, our environment is deployed in GeForce GTX 1080 Ti, and we adjust the parameters of the model to find a suitable threshold between model accuracy and size, making the algorithm more suitable for the real scene. We set the hidden layer feature to 32, and finally extract 16 features in the fully connected layer. This also achieve the simplification of the model, so that the algorithm can be deployed in the central processing unit (CPU) environment and ensure that the accuracy of the model is guaranteed. In the real scenario, the collection meter is deployed in the living area of users, and there is not too high a configuration. Therefore, a lightweight model is required.

We improve the calculation speed of the model by controlling the parameters of the hidden layer. It takes about 2 h to train on a single graphics processing unit (GPU). In the real scene test, we can achieve the average time of a single call to the model less than 0.1 s on the CPU; unfortunately, however, as the length of compressed data increases, the overall time consumed is still very long.

In the Bi-LSTM model, we use one-hot to encode the character sequence into a K -dimensional vector sequence and use it as the input of the network. The mode output is the probability distribution of the next character. We use the cross-entropy function to minimize loss. The state of the neural unit is retained across batches, and the Adam optimizer is used to solve the gradient explosion problem in the model. The Bi-LSTM model settings are shown in Table 1.

Table 1. In the model setting, the input length is set to 50, the output length is 7820, and the batch is set to 128.

Layers	Process	Output Size
Layer 1	Embedding	[128, 50, 50]
Layer 2	Bi-LSTM	[128, 50, 32]
Layer 3	Bi-LSTM	[128, 32]
Layer 4	FC	[128, 16]
Layer 5	FC	[128, 7820]

Figure 4 shows the experimental effect of our proposed method on power data. Our data is the actual power consumption value of users. The data is provided by a power supply company. The data is collected in String format. The experiment tests the power consumption data of 200 users in the same minute. The abscissa is the user and the ordinate is the value.

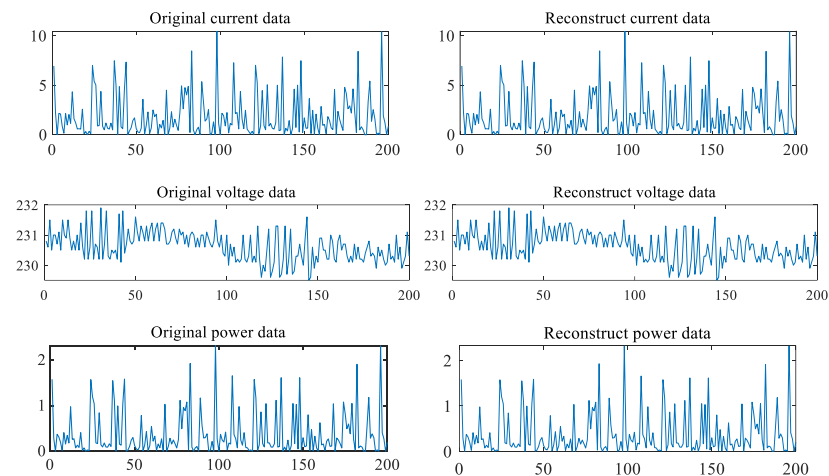


Figure 4. Original data and reconstructed data.

Most of the traditional compression algorithms do not learn the complex characteristics of the data and are not good at modeling the actual electricity consumption data. As shown in Table 2, we have tested two models to model the character dataset. Among them, the effect of the Bi-LSTM model is slightly better than that of the Transformer. The self-attention of the Transformer usually has a larger weight on the data at a short distance and a smaller weight at a long distance. At this time, the effect of relying solely on attention to establish a dependency relationship is not as good as that of the Bi-LSTM. It has the same situation with static arithmetic coding and Huffman. The more different elements the data has, the higher the accuracy of the output probability, resulting in a higher compression rate of the final calculation. The compression ratio of current and voltage is significantly higher than that of voltage.

Table 2. Compare the CR results of using Bi-LSTM and Transformer.

Model	Bi-LSTM			Transformer		
	Voltage	Current	Power	Voltage	Current	Power
Original size(in bytes)	922	888	1007	922	888	1007
Compressed size(in bytes)	174	276	304	167	296	317
CR	5.30	3.22	3.31	5.52	3.00	3.18

Considering that the user's actual electricity consumption data is non-sinusoidal, the wavelet transform fails here. We use traditional lossless compression methods to compare with our algorithm. Table 3 shows our comparison results with other methods. The experimental results show that our model performance is better than static arithmetic coding and the Huffman algorithm. The output of the model is closer to the true value, which can effectively save the accuracy of probability. It shows that our predicted data is close to the actual results, and the predicted probability distribution is more suitable for compression, which further proves the superiority of the algorithm.

Table 3. Comparison of different compression algorithms (avg).

Algorithm	Ours	AC [24]	Huffman [23]
CR	4.06	3.31	1.98

As shown in Table 4, we show the entropy before and after compression in different algorithms in the test. The implementation shows a set of 200-min test results, and the entropy before compression is higher than that after compression. When the length of the output string is large and the repetition is not high, the larger the entropy after compression will be. This means that the greater the difference between them, the better the compression effect. The results show that the entropy of our proposed algorithm after compression is lower than that of the classical algorithm. We analyzed the positive effect of the deep learning model on entropy. The output probability of traditional coding is fixed, while the prediction probability of Bi-LSTM and Transformer models will change adaptively after training. The output is closer to the real value, which will save memory for subsequent encoding and compression. The training results of the Bi-LSTM model are slightly better than that of Transformer. We concluded that the probability of adaptive change will improve the CR and entropy, which confirms that the training of the deep learning model will have a positive impact on them.

Table 4. Test the entropy of a group of 200-min current data before and after compression in different algorithms.

Algorithm	AC [24]	Huffman [23]	Ours (Bi-LSTM)	Ours (Transformer)
Entropy before compression	3.23	3.23	3.23	3.23
Entropy after compression	2.73	3.14	2.17	2.24

6. Conclusions

In this paper, we use a deep neural network with arithmetic coding to learn the characteristics of power data. The proposed scheme realizes efficient deep lossless compression of power data and compares the improvement of arithmetic coding compression performance of two different deep learning model training power datasets. After adjusting the model parameters, our algorithm still achieves better results. In our future work, we will improve the model and further improve the compression performance and speed.

Author Contributions: Conceptualization, Z.M., H.Z., F.S., and Z.H.; methodology, F.S. and Z.H.; software, Z.M.; validation, Z.M., F.S., and H.Z.; formal analysis, F.S.; investigation, Z.M., Z.H., and Y.L.; resources, Z.M. and H.Z.; data curation, Z.M.; writing—original draft preparation, Z.H. and Y.L.; writing—review and editing, Z.H.; visualization, F.S., Y.L., and Z.H.; supervision, Z.M.; project administration, Z.M. and H.Z.; funding acquisition, Z.M. and H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Project supported by the Science and Technology Project of State Grid Jiangsu Electric Power Co., Ltd. (J2021120).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. Z.M., H.Z., F.S., Z.H., and Y.L. jointly developed the algorithm. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

PMU	Phasor Measurement Unit
MSE	Mean Square Error
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
Bi-LSTM	Bi-directional Long Short-Term Memory
CR	Compression Ratio
CPU	Central Processing Unit
PCA	Principal Component Analysis
DBEA	Differential Binary Encoding Algorithm

References

1. González, I.; Calderón, A.J.; Portalo, J.M. Innovative Multi-Layered Architecture for Heterogeneous Automation and Monitoring Systems: Application Case of a Photovoltaic Smart Microgrid. *Sustainability* **2021**, *13*, 2234. [\[CrossRef\]](#)
2. Song, F.; Qin, Z.; Xue, L.; Zhang, J.; Lin, X.; Shen, X. Privacy-preserving keyword similarity search over encrypted spatial data in cloud computing. *Internet Things J.* **2021**, *9*, 6184–6198. [\[CrossRef\]](#)
3. Song, F.; Qin, Z.; Liu, D.; Zhang, J.; Lin, X.; Shen, X. Privacy-preserving task matching with threshold similarity search via vehicular crowdsourcing. *Trans. Veh. Technol.* **2021**, *70*, 7161–7175. [\[CrossRef\]](#)
4. Plenz, M.; Dong, C.; Grumm, F.; Meyer, M.F.; Schumann, M.; McCulloch, M.; Jia, H.; Schulz, D. Framework Integrating Lossy Compression and Perturbation for the Case of Smart Meter Privacy. *Electronics* **2020**, *9*, 465. [\[CrossRef\]](#)
5. Tightiz, L.; Yang, H. A Comprehensive Review on IoT Protocols' Features in Smart Grid Communication. *Energies* **2020**, *13*, 2762. [\[CrossRef\]](#)
6. Huang, X.; Hu, T.; Ye, C.; Xu, G.; Wang, X.; Chen, L. Electric Load Data Compression and Classification Based on Deep Stacked Auto-Encoders. *Energies* **2019**, *12*, 653. [\[CrossRef\]](#)
7. Unterweger, A.; Engel, D. Resumable load data compression in smart grids. *IEEE Trans. Smart Grid* **2014**, *6*, 919–929. [\[CrossRef\]](#)
8. Tong, X.; Kang, C.; Xia, Q. Smart metering load data compression based on load feature identification. *Trans. Smart Grid* **2016**, *7*, 2414–2422. [\[CrossRef\]](#)
9. Sarkar, S.J.; Kundu, P.K.; Sarkar, G. Performance Analysis of Resumable Load Data Compression Algorithm (RLDA) for Power System Operational Data. In Proceedings of the Calcutta Conference (CALCON), Kolkata, India, 2–3 December 2017; pp. 16–20.
10. Wang, Y.; Chen, Q.; Kang, C. Sparse and redundant representation-based smart meter data compression and pattern extraction. *Trans. Power Systems* **2016**, *32*, 2142–2151. [\[CrossRef\]](#)
11. Wang, L.; Chen, Z.; Yin, F. A novel hierarchical decomposition vector quantization method for high-order LPC parameters. *Trans. Audio Speech Lang. Process.* **2014**, *23*, 212–221. [\[CrossRef\]](#)
12. Watson, A.B. DCT quantization matrices visually optimized for individual images. *Hum. Vision Vis. Process. Digit. Disp. IV SPIE* **1993**, *1913*, 202–216.
13. Ning, J.; Wang, J.; Gao, W.; Liu, C. A wavelet-based data compression technique for smart grid. *Trans. Smart Grid* **2010**, *2*, 212–218. [\[CrossRef\]](#)
14. Das, S.; Rao, P.S.N. Principal Component Analysis Based Compression Scheme for Power System Steady State Operational Data. In Proceedings of the International Conference on Innovative Smart Grid Technologies, ISGT2011-India, Kollam, India, 1–3 December 2011; pp. 98–100.
15. Mehra, R.; Bhatt, N.; Kazi, F.; Singh, N.M. Analysis of PCA based compression and denoising of smart grid data under normal and fault conditions. In Proceedings of the International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2–4 July 2013; pp. 1–6.
16. Ziv, J.; Lempel, A. A universal algorithm for sequential data compression. *Trans. Inf. Theory* **1977**, *23*, 337–343. [\[CrossRef\]](#)
17. Ziv, J.; Lempel, A. Compression of individual sequences via variable-rate coding. *Trans. Inf. Theory* **1978**, *24*, 530–536. [\[CrossRef\]](#)
18. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [\[CrossRef\]](#)
19. Huffman, D.A. A method for the construction of minimum-redundancy codes. *Proc. IRE* **1952**, *40*, 1098–1101. [\[CrossRef\]](#)
20. Witten, I.H.; Neal, R.M.; Cleary, J.G. Arithmetic coding for data compression. *Commun. ACM* **1987**, *30*, 520–540. [\[CrossRef\]](#)

21. Ringwelski, M.; Renner, C.; Reinhardt, A.; Weigel, A.; Turau, V. The Hitchhiker's guide to choosing the compression algorithm for your smart meter data. In Proceedings of the International Energy Conference and Exhibition (ENERGYCON), Florence, Italy, 2–9 September 2012; pp. 935–940.
22. Das, S.; Rao, P.S.N. Arithmetic coding based lossless compression schemes for power system steady state operational data. *Int. J. Electr. Power Energy Syst.* **2012**, *43*, 47–53. [[CrossRef](#)]
23. Sarkar, S.J.; Sarkar, N.K.; Banerjee, A. A novel Huffman coding based approach to reduce the size of large data array. In Proceedings of the 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, India, 18–19 March 2016; pp. 1–5.
24. Sarkar, S.J.; Kar, K.; Das, I. Basic arithmetic coding based approach for compressing generation scheduling data array. In Proceedings of the Calcutta Conference (CALCON), Kolkata, India, 2–3 December 2017; pp. 21–25.
25. Khan, J.; Bhuiyan, S.M.A.; Murphy, G.; Arline, M. Embedded-zero-tree-wavelet-based data denoising and compression for smart grid. *Trans. Ind. Appl.* **2015**, *51*, 4190–4200. [[CrossRef](#)]
26. Khan, J.; Bhuiyan, S.; Murphy, G.; Murphy, G. Data denoising and compression for smart grid communication. *IEEE Trans. Signal Inf. Process. Over Netw.* **2016**, *2*, 200–214. [[CrossRef](#)]
27. Ji, X.; Zhang, F.; Cheng, L.; Liang, C.; He, H. A wavelet-based universal data compression method for different types of signals in power systems. In Proceedings of the Power & Energy Society General Meeting (PESGM), Chicago, IL, USA, 16–20 July 2017; pp. 1–5.
28. Cheng, L.; Ji, X.; Zhang, F.; Huang, H.; Gao, S. Wavelet-based data compression for wide-area measurement data of oscillations. *J. Mod. Power Syst. Clean Energy* **2018**, *6*, 1128–1140. [[CrossRef](#)]
29. Prathibha, E.; Manjunatha, A.; Basavaraj, S. Dual tree complex wavelet transform based approach for power quality monitoring and data compression. In Proceedings of the 2016 Biennial International Conference on Power and Energy Systems: Towards Sustainable Energy (PESTSE), Bengaluru, India, 21–23 January 2016; pp. 1–5.
30. Ruiz, M.; Simani, S.; Inga, E.; Jaramillo, M. A novel algorithm for high compression rates focalized on electrical power quality signals. *Heliyon* **2021**, *7*, e06475. [[CrossRef](#)] [[PubMed](#)]
31. Gontijo, L.F.C.; André, N.O.; Nascimento, F.A.O. Segmentation and Entropy Coding Analysis of a Data Compression System for Power Quality Disturbances. In Proceedings of the 2020 Workshop on Communication Networks and Power Systems (WCNPS), Brasilia, Brazil, 12–13 November 2020; pp. 1–6.
32. Wang, W.; Chen, C.; Yao, W.; Sun, K.; Qiu, W.; Liu, L. Synchrophasor Data Compression Under Disturbance Conditions via Cross-Entropy-Based Singular Value Decomposition. *Trans. Ind. Inform.* **2020**, *17*, 2716–2726. [[CrossRef](#)]
33. Karthika, S.; Rathika, P. An Efficient Data Compression Algorithm for Smart Distribution Systems using Singular Value Decomposition. In Proceedings of the International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Tamilnadu, India, 11–13 April 2019; pp. 1–7.
34. Sarkar, S.J.; Kundu, P.K.; Sarkar, G. DBEA: A novel approach of repetitive data array compression for power system application. In Proceedings of the International Conference for Convergence in Technology (I2CT), Kolkata, India, 7–9 April 2017; pp. 16–20.
35. Sarkar, S.J.; Kundu, P.K.; Sarkar, G. Development of lossless compression algorithms for power system operational data. *IET Gener. Transm. Distrib.* **2018**, *12*, 4045–4052. [[CrossRef](#)]
36. Sarkar, S.J.; Kundu, P.K.; Sarkar, G. Comparison of Different Differential Coding based Algorithms Developed for Compressing Power System Operational Data. In Proceedings of the Region 10 Symposium (TENSYP), Kolkata, India, 14–16 July 2017; pp. 16–20.
37. Abuadba, A.; Khalil, I.; Yu, X. Gaussian approximation-based lossless compression of smart meter readings. *Trans. Smart Grid* **2017**, *9*, 5047–5056. [[CrossRef](#)]
38. Tripathi, S.; De, S. An efficient data characterization and reduction scheme for smart metering infrastructure. *Trans. Ind. Inform.* **2018**, *14*, 4300–4308. [[CrossRef](#)]
39. Goyal, M.; Tatwawadi, K.; Chandak, S.; Ochoa, I. DeepZip: Lossless Data Compression Using Recurrent Neural Networks. In Proceedings of the Data Compression Conference (DCC), Madrid, Spain, 26–29 March 2019.
40. Liu, Q.; Xu, Y.; Li, Z. DecMac: A Deep Context Model for High Efficiency Arithmetic Coding. In Proceedings of the International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), Okinawa, Japan, 11–13 February 2019; pp. 438–443.
41. Wang, R.; Bai, Y.; Chu, Y.S.; Wang, Z.; Wang, Y.; Sun, M.; Li, J.; Zang, T.; Wang, Y. DeepDNA: A hybrid convolutional and recurrent neural network for compressing human mitochondrial genomes. In Proceedings of the International Conference on Bioinformatics and Biomedicine (BIBM), Madrid, Spain, 3–6 December 2018; pp. 270–274.