IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 November 2023, accepted 9 December 2023, date of publication 15 December 2023, date of current version 22 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3343436

RESEARCH ARTICLE

A Dynamic and Parallel Two-Stage Lossless Data Compression Method for Smart Grid

YANXIA WU[1], **SONG XU**[2], **(Member, IEEE), CAIPING XI**[2], **PENGQIANG NIE**[2], **(Student Member, IEEE), WEI JIANG**[3], **(Member, IEEE), AND SEIJI HASHIMOTO**[4], **(Member, IEEE)**

[1]Ocean College, Jiangsu University of Science and Technology, Zhenjiang 212000, China
[2]College of Automation, Jiangsu University of Science and Technology, Zhenjiang 212000, China
[3]College of Electrical, Energy and Power Engineering, Yangzhou University, Yangzhou 215000, China
[4]Division of Electronics and Informatics, Gunma University, Kiryu, Gunma 376-8515, Japan

Corresponding author: Song Xu (songxu@just.edu.cn)

This work was supported by the research fund of Jiangsu University of Science and Technology under Grant 1032932101.

ABSTRACT With the rapid development of smart grid (SG) technologies, massive system data has been generated for SG for grid operation status monitoring and fault warning, this massive integrated data has brought great challenges for data transmission and storage. Thus, the breakthrough of data compression technology has become more and more important for data processing, this proposal, a lossless data compression method is proposed. The proposed method improves the Lempel-Ziv-Welch (LZW) algorithm. Specifically, the proposed method employs a parallel search approach to search the index of the dictionary globally by dividing the dictionary into several small dictionaries of different sizes and bit widths. In addition, the dynamic variable-length coding method is used as the output code which can offer flexible bit widths instead of fixed values, and the optimal dictionary partition combination and size parameters are selected to construct dictionaries. At last, the improved algorithm is cascaded with Huffman algorithm to form the proposed data compression algorithm. The data compression efficiency has been successfully verified by comparing with conventional Huffman and LZW data compression algorithms through data simulation, and it is observed to offer a better compression ratio than those methods with only LZW algorithm or Huffman algorithm, and save storage space effectively. More than this, the proposed method has realized lossless compression for power system data and guaranteed the integrity of the data, which will have better applicability for dealing with any similar information data.

INDEX TERMS Lossless data compression, smart grid, Huffman algorithm, LZW algorithm.

I. INTRODUCTION

In recent years, with the continuous advancement of urbanization and the increase of power demand, the smart grid (SG) construction has become an inevitable trend in the development of power grid technology [1], [2]. SG is the intelligence of modern power grid, which is based on an integrated, high-speed two-way communication network to achieve safe, reliable, and cost-effective use of the power grid, as well as to meet the data information exchange among intelligent electronic devices and data collection. However, the realization of SG requires the collection and transmission of large amounts of data through communication networks. The transmission and distribution systems use measurement and monitoring instruments for information collection, which is managed by supervisory control and data acquisition systems and wide area monitoring systems. The smart metering systems and automatic meter reading systems are used for data collection in the user end. Therefore, a large amount of data is stored in SG, which is circulated and stored between control centers, utilities and customers, posing huge challenges to data storage, transmission and processing. In order to save storage space, increase data transmission speed, and effectively utilize communication bandwidth, it is particularly necessary to

The associate editor coordinating the review of this manuscript and approving it for publication was Miadreza Shafie-Khah.

use data compression technology. Data compression technology requires the sending end to compress the power system data in order to inject as much data volume as possible into the communication system. When the compressed data is received at the receiving end, the compressed data can be reconstructed for analysis. This technique can effectively reduce the amount of data transmitted in SG, resulting in cost savings in data storage, while also meeting the transmission capacity limitations of the communication network [3], [4].

Data compression technology can be categorized into two types based on the reversibility of the compression process: lossy data compression [5] and lossless data compression [6]. The lossy compression will eventually lose some data, thereby compromising the data integrity and impacting the accuracy of the application results. In contrast, the lossless compression achieves the compression effect by eliminating redundant information in the data, ensuring that the original data can be fully restored without any loss in quality.

The application of lossy data compression technology in SG includes wavelet decomposition (WD), discrete cosine transform (DCT), and singular value decomposition (SVD). A wavelet packet decomposition (WPD) technology using a cost function to transform wavelet trees into complete binomial trees has been proposed in [7]. This technology denoises and compresses SG signals simultaneously. The power quality data compression has been achieved by applying the discrete wavelet transform (DWT) to the difference between the input signal and the reference signal in [8], [9], and [10]. Reference [11] has proposed a two-stage compression algorithm that uses principal component analysis in the first stage and DCT in the second stage to exploit the intrinsic correlation of the phasor measurement unit data. The utilization of the SVD technique in smart distribution networks has been introduced in [12] and [13]. Experimental results demonstrate the effectiveness of this method in reducing the amount of data transmitted through the communication network. Reference [14] has proposed a combination method based on SVD and WD reduction for image compression. The results show that combining WD reduction with SVD can improve the compression ratio (*CR*). A data compression method based on SVD of synchrophasor data has been proposed in [15]. The method includes a dimensionality reduction technique and a progressive partition algorithm. The dimensionality reduction technique is used to verify validity of SVD by establishing a threshold value, and the progressive partitioning algorithm divides the synchrophasor data into partitions with desirable dimensions for better *CR*. A data compression method based on wavelet domain SVD has been proposed in [16]. This method divides the power signal into a two-dimensional matrix, the two-dimensional DWT is used to decompose the original matrix into sub-matrices, and then SVD is used to compress the sub-matrices data.

The application of lossless data compression technology in SG includes run length encoding (RLE), Huffman coding, LZ coding. Lossless compression of smart distribution network monitoring data has been achieved in [17] by improved RLE and Huffman coding of the residuals between the actual calculated values and the current predicted values of distribution network monitoring data. An improved LZW algorithm by using a tree structure to construct the dictionary and a multicharacter parallel search method to query the dictionary has been proposed in [18]. The improved LZW algorithm has been applied to power fault data, and experimental results show that the method effectively reduces the mean square error rate. A stream-based data compression method, called ASE coding has been proposed in [19]. This method dynamically adjusts the influencing parameters by calculating the instantaneous entropy of the compressed data stream to obtain a more stable *CR*. Reference [20] has discussed the application of lossless compression in power systems and looks forward to future work. A model-free lossless data compression method for SG time series has been proposed in [21], which obtains a low-latency and good *CR* by considering the accuracy of the time series. A double lossless compression method using the adaptive Huffman (AH) algorithm and LZW algorithm has been proposed in [22]. This method compresses the smart meter data from ten minutes to one hour, and the results show that the double compression method has a better *CR* than the adaptive Huffman or LZW algorithm alone, but increases the complexity. An improved LZMA compression method has been proposed in [23]. The method targets the preprocessing steps of SG data and has a higher compression ratio compared to the traditional LZMA algorithm.

The combination of lossless technology and lossy technology is also widely used for SG data compression. By using Huffman coding to compress switching and power backbone network data, and using WT to compress power quality data, the data compression technology has been applied to ship power monitoring systems in [24]. Reference [25] has combined integer WT with the LZ77 algorithm. After performing integer WT on the original power system data, the high-frequency component is compressed by threshold quantization lossy compression, and the low-frequency component is compressed by the LZ77 algorithm. The results show a high data *CR* and low data reconstruction error. An improved Shannon entropy has been proposed in [26] and used to select the best basis representation for WPD signals. Applying WPD to compress the SG signal data can obtain a good *CR*.

The existing data compression algorithms for SG still have limitations such as narrow application range due to the concentration of sampled data and poor compression performance due to inherent defects. This paper focuses on data congestion in SG and aims to reduce the burden on communication systems and improve storage utilization by using data compression. Therefore, a lossless compression method is proposed in this paper. This method firstly improves the LZW algorithm through the parallel search and the dynamic variable-length coding method, and then cascades the improved algorithm with Huffman algorithm to form a two-stage compression algorithm. Finally, the experimental simulation shows that the proposed algorithm can achieve a higher *CR* compared with traditional algorithms, at the same time, it can process any similar information data while ensuring data integrity.

The rest of this manuscript is organized as follows. Section II presents the background knowledge and related work of the LZW algorithm and Huffman algorithm, while the factors affecting the compression performance of LZW algorithm are analyzed. The proposed improved LZW algorithm that incorporates parallel search and dynamic variable-length coding as well as the proposed algorithm cascaded with Huffman to form a two-stage H-DPDLZW algorithm are introduced in detail in Section III. Section IV discusses the impact of dictionary partitioning and size on compression performance within the proposed algorithm while the simulations have been carried out in MATLAB, the results are compared with conventional methods quantitatively. Finally, a brief conclusion of this research has been made in Section V.

II. RELATED ALGORITHMS

A. LZW ALGORITHM

The Lempel-Ziv-Welch (LZW) algorithm [27] is a dictionary-based compression algorithm proposed by Abraham Lempel, Jacob Ziv and Terry Welch in 1984. It is commonly referred to as the "string table compression algorithm". The core idea of this algorithm is to construct a dynamic dictionary to realize data compression. Specifically, LZW algorithm divides the input data into individual characters and searches for the longest matching string in the dictionary. Once a match is found, the next character of the current string is used as the new input for further matching until no further match is found. In case of an unsuccessful match, the current string is replaced with the corresponding index output, and the prefix of the current string is added to the dictionary as a new entry. The process continues until all input data have been processed. The flowchart of the LZW algorithm is shown in Fig. 1.

In Fig. 1, *P* represents the prefix string, *C* represents the current character. Fig. 2 demonstrates the operation of LZW algorithm using the input data stream "ABBABABACABC" as an example.

The compression performance of LZW algorithm is affected by the employed dictionary search methods. The dictionary search methods of the current LZW algorithm include serial search [28], parallel search [29], and hash table search [30]. Serial search is the simplest method, which reads the input data sequentially and constructs a dictionary to find the longest matching string. Although this method is simple to implement, it suffers from slower processing speeds. Parallel search employs multiple search pointers placed at different locations within the dictionary, allowing for simultaneous

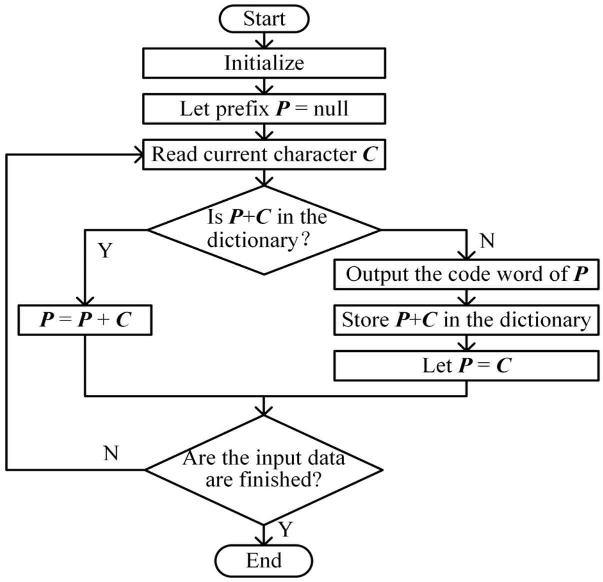

FIGURE 1. The flowchart of LZW algorithm.

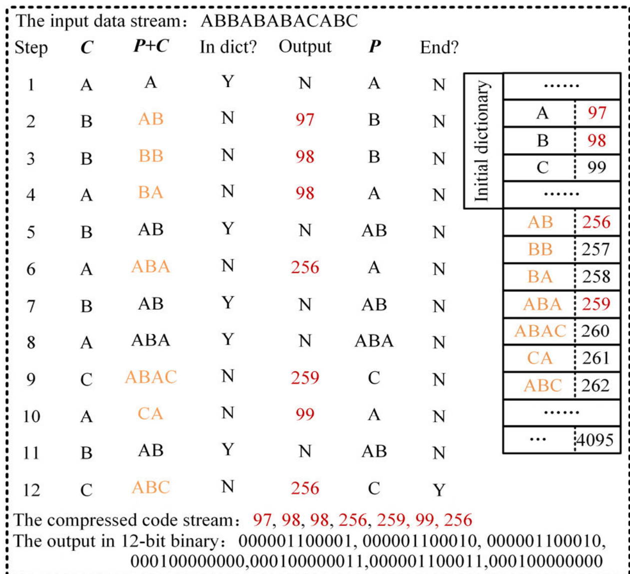

FIGURE 2. The example of LZW algorithm.

identification of the longest matching string. This method offers faster performance but is more complex to implement. Hash table search uses a hash table to store strings within dictionary, enabling rapid retrieval of matching strings. This method is fast, but the implementation is the most complex, with hash conflict.

The compression performance of LZW algorithm is also affected by the dictionary size. A smaller dictionary can be constructed and searched faster, thereby speeding up the execution of the algorithm. However, a dictionary that is too small will lead to poor compression performance of the algorithm. Conversely, a larger dictionary enables a greater number of entries to be stored, increasing the probability of

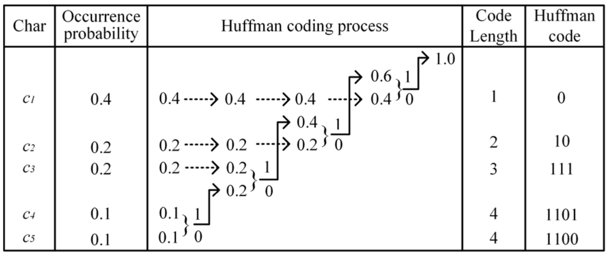

FIGURE 3. The coding steps of huffman algorithm.

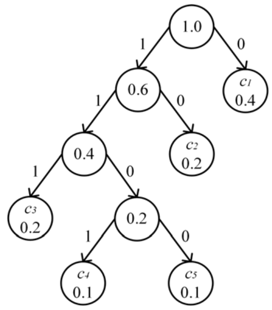

FIGURE 4. The huffman tree.

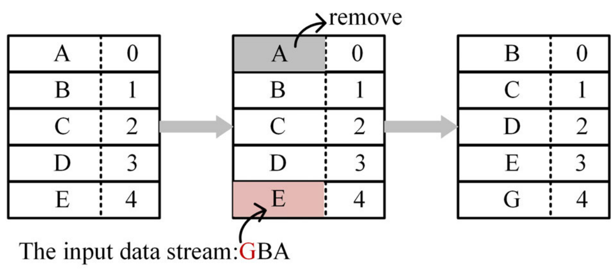

FIGURE 5. The FIFO strategy.

successful matching. However, a dictionary that is too large will consume too much memory and reduce the efficiency of the algorithm. Therefore, selecting an appropriate dictionary size is an additional aspect to be considered.

B. HUFFMAN ALGORITHM

The Huffman algorithm [31], proposed by David Huffman in 1952, is a statistical compression algorithm. Its effectiveness is dependedent on the statistical properties of the input data stream, and it assigns code words of different lengths based on the occurrence probability of the input data. The Huffman algorithm ensures the shortest average code length for the output code words. Huffman coding utilizes a prefix coding approach through the construction of a Huffman tree [32]. The resulting variable-length coding generated by the Huffman tree is considered an optimal unequal length coding, with different coding rules determined by the probabilities of occurrence. Huffman coding constructs Huffman tree according to the character probabilities. The process consists of the following steps [33].

Step 1. Sort the characters of the input data stream in decreasing order of their probabilities.

Step 2. Assign "1" and "0" to the two characters with the lowest probabilities, usually assigning "1" to the higher probability and the "0" to the lower probability. Then, add the probabilities of the two characters.

Step 3. Take the probability obtained by summing as the probability of the given new character, and form a new character with other characters. Repeat *Step 1* and *Step 2*.

Step 4. Continue this process until the sum of probabilities for the remaining two characters is 1.

Step 5. From the last step, take the code symbols obtained from one step along the reverse order, and the sequence of code symbols formed is the code word of the corresponding character.

As shown in Fig. 3, 0.4, 0.2, 0.2, 0.1, and 0.1 correspond to the probabilities of information source c_1, c_2, c_3, c_4, and c_5, respectively, and the Huffman coding is performed according to the above steps.

The Huffman tree is a tree with the shortest path length with weights, and nodes with larger weights are closer to the root. The structure of the Huffman tree is shown in Fig. 4, where the leaf nodes represent the original characters. The Huffman coding corresponding to that leaf node is obtained by following the coding path starting from the topmost root node until some leaf node.

III. THE PROPOSED METHOD

To balance the algorithm complexity and compression performance, a lossless compression algorithm based on parallel search and dynamic variable-length coding is proposed in this paper, called the dynamic and parallel dictionary LZW (DPDLZW) algorithm. It is an improvement on the traditional LZW algorithm. The DPDLZW algorithm mainly includes the following improvements:

1) DIVIDE DICTIONARY TO PERFORM PARALLEL SEARCH

The dictionary is divided into several small dictionaries of different sizes, each with different capacity and bit widths. The same small dictionary only stores strings of the same length, and all small dictionaries can be searched in parallel approach at the same time. The first small dictionary is called Dict0, which only stores 1-byte characters, Dict1 stores 2-byte strings, and so on.

2) CLEAR ALL SMALL DICTIONARIES DURING INITIALIZATION

These small dictionaries are populated throughout the compression process gradually.

3) UPDATE THE STORAGE METHOD OF DICT0

Dict0 is also called a virtual dictionary, which stores only a single character in the input data stream for the first occurrence. The size of the virtual dictionary depends on the number of unique symbols in the input data stream.

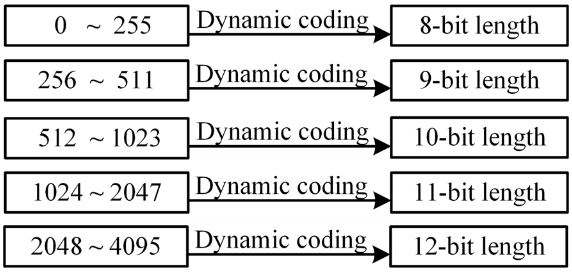

FIGURE 6. **The dynamic code length assignment.**

4) INCORPORATE THE DICTIONARY UPDATE STRATEGY

In the event of overflow within a small dictionary, the first-in, first-out (FIFO) strategy is implemented. FIFO follows the principle that when the dictionary reaches its maximum capacity and a new string requires addition, the earliest stored string within the dictionary is removed to accommodate the new string at the end, thereby maintaining a constant dictionary. The FIFO strategy is shown in Fig. 5.

5) DEFINE THE CLASSIFICATION OUTPUT STRATEGY

The classification output strategy entails two scenarios. Firstly, if the out prefix consists of a single character, it is directly output. Secondly, if the prefix contains more than one character, dynamic coding is employed for output. The assignment of dynamic code lengths is illustrated in Fig. 6.

These improvements optimize compression performance of the traditional LZW algorithm significantly. To further reduce data redundancy, the DPDLZW algorithm is integrated with Huffman algorithm, resulting in a two-stage compression algorithm known as Huffman with dynamic and parallel dictionary LZW (H-DPDLZW) algorithm, as shown in Fig. 7.

In Fig. 7, L represents the dictionary ordinal, P represents the prefix string, with the value of Lbeing equivalent to the actual length of P. MAX represents the maximum number of dictionaries and corresponds to the maximum length of P. In the case of $MAX = 3$, three dictionaries (Dict0, Dict1, and Dict2) are present. Dict0 stores single characters with $L = 1$, and Dict1 stores strings with $L = 2$. I_o denotes the index of the previous compressed string in the dictionary, and I_n denotes the index of the current compressed string in the dictionary.

Fig. 8 presents the same example to illustrate the operation of H-DPDLZW algorithm. In this case, $MAX = 4$, the Dict0 has a size of 128, and the remaining dictionaries have a size of 256.

It can be seen from Figs. 2 and 8 that the output stream of the LZW algorithm is 84 bits long, while H-DPDLZW algorithm is only 31 bits long. Therefore, the proposed algorithm exhibits better compression performance.

Given that dictionary partitions and sizes affect the compression performance of DPDLZW algorithm, in order to determine the optimal dictionary parameters, for the dictionary with a capacity of 4K, some dictionary partition combinations of the algorithm are shown in Table 1. The size of the virtual dictionary is 128, which is determined by the base ASCII code size. When multiple MAX values are present, they are distinguished using the same color scheme.

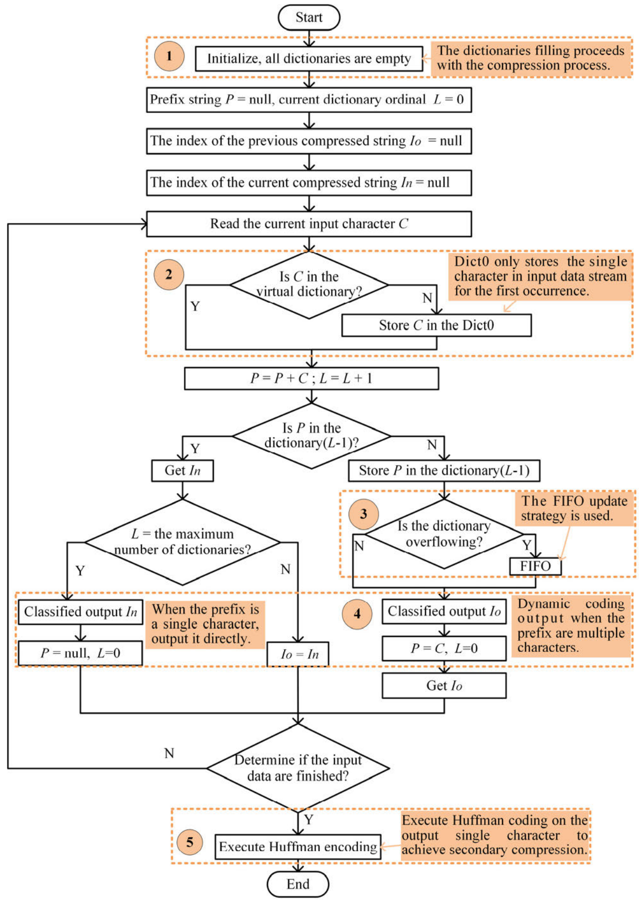

FIGURE 7. **The flowchart of the proposed algorithm.**

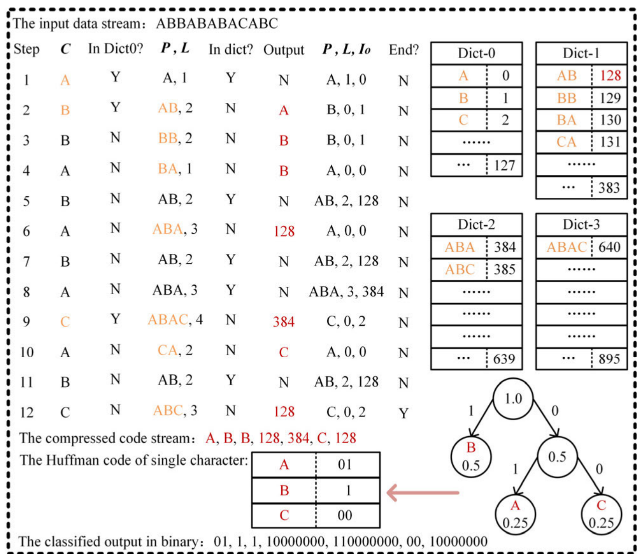

FIGURE 8. **The example of H-DPDLZW algorithm.**

TABLE 1. Some dictionary partition combinations for 4K address space.

I	Dict0	Dict1	Dict2	Dict3	Dict4	Dict5	Dict6	Dict7	Dict8	Dict9	Dict10	Dict11	Dict12	Dict13	Dict14
1	128	1024	1024	1024	512	256	128								
2	128	1024	1024	512	512	512	256	128							
3	128	1024	1024	512	512	256	256	256	128						
4	128	1024	512	512	512	512	512	256	128						
5	128	1024	1024	512	512	256	256	128	128	128					
6	128	1024	512	512	512	512	256	256	256	128					
7	128	512	512	512	512	512	512	512	256	128					
8	128	1024	512	512	512	256	256	256	256	256	128				
9	128	512	512	512	512	512	512	256	256	256	128				
10	128	1024	512	512	256	256	256	256	256	256	256	128			
11	128	512	512	512	512	512	256	256	256	256	256	128			
12	128	1024	512	512	256	256	256	256	256	256	128	128	128		
13	128	512	512	512	512	256	256	256	256	256	256	256	128		
14	128	512	512	512	256	256	256	256	256	256	256	256	256	128	
15	128	512	512	256	256	256	256	256	256	256	256	256	256	256	128

TABLE 2. The specific information of the test files.

Files name	Size (byte)	Original bits	Unique symbols
F1	7704	61632	54
F2	12755	102040	59
F3	33060	264480	56
F4	176730	1413840	65
F5	209545	1676360	65
F6	253515	2028120	67
F7	258437	2067496	67

IV. SIMULATION EXPERIMENTS

To verify the overall performance of the proposed lossless compression method, a comparative experiment is conducted. The power data are compressed in the MATLAB simulation environment, and the optimal dictionary parameters of DPDLZW algorithm are selected. The performance is then compared to four other lossless compression algorithms: the LZW algorithm, the LZW algorithm with the FIFO strategy, Huffman algorithm, and PDLZW algorithm [34].

A. THE TEST FILES

SG data is used to achieve real-time monitoring, intelligent control and optimized decision-making of the power system, including a vast array of datasets such as scheduling, transmission and distribution, power generation, and user information.

To guarantee the accuracy and reliability of SG data, lossless compression techniques are particularly important. In this paper, seven test files of different sizes and types have been used. These seven files are the parts of real electricity data in the United States from 2001-2022. These datasets are retail sales of electricity to ultimate customers: total by end-use sector (F1), net generation by energy source: independent power producers (F2), stocks of coal: electric power sector by census division (F3), receipts of coal delivered for electricity generation by state (F4), revenues from retail sales of electricity to ultimate customers by end-use sector, by state (F5), net generation by state sector (F6), and consumption of natural gas for electricity generation by state by sector (F7). Detailed information about the test files is documented in Table 2.

It can be observed from Table 2 that the number of unique symbols in the file increases with the file size. However, in files with a greater number of characters, the number of symbols remains relatively constant. This is due to the fact that larger files tend to contain as many unique symbols as possible.

B. THE EVALUATION INDICATORS

1) COMPRESSION RATIO CR

The compression ratio indicates how compressed the data is. It is used to measure the data compression effect achieved by algorithms after compressing data.

$$CR = M/N \tag{1}$$

where, M represents the original data size, and N represents the compressed data size. The higher the CR is, the better the compression effect.

2) COMPRESSION FACTOR CF

The compression factor is the inverse of the compression ratio. CF indicates how many times the amount of compressed data is the original data.

$$CF = N/M \tag{2}$$

If CF is 1, no compression is performed. If $CF < 1$, the compression is better. If $CF > 1$, the compression is poor.

3) SAVE PERCENTAGE SP

The save percentage indicates the percentage of the files to be compressed.

$$SP = ((M - N)/M) \times 100\% \tag{3}$$

A larger SP value means that the compression algorithm compresses the data more effectively, thus achieving a higher saving ratio. This means that more data can be stored or less bandwidth consumed for data transfer with the same storage space.

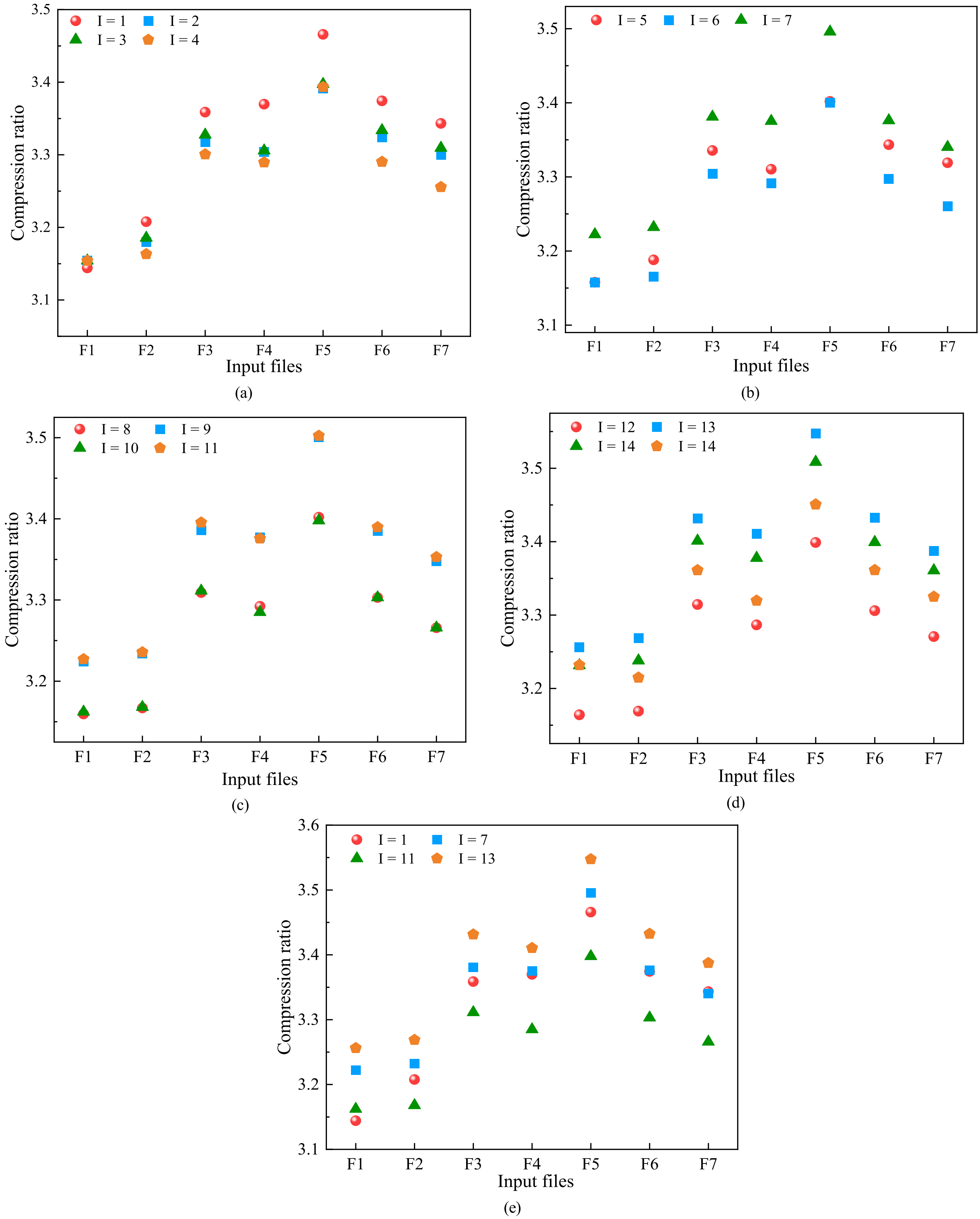

FIGURE 9. The compression ratio of I = (1-15).

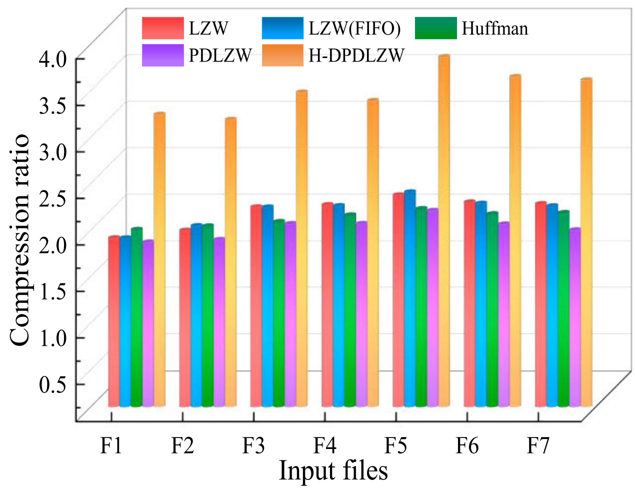

FIGURE 10. The compression ratio of comparison algorithms.

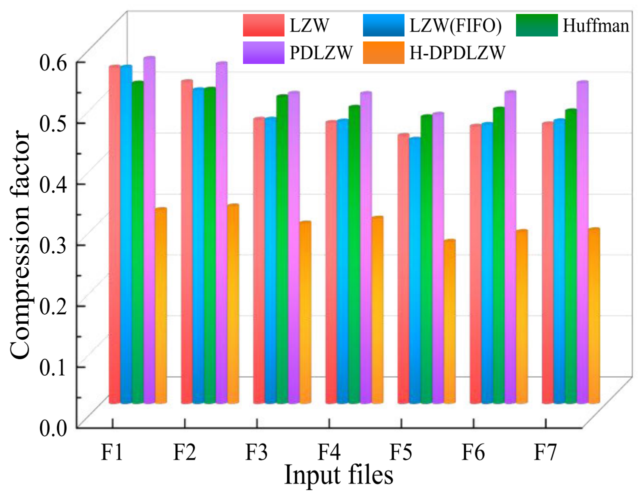

FIGURE 11. The compression factor of comparison algorithms.

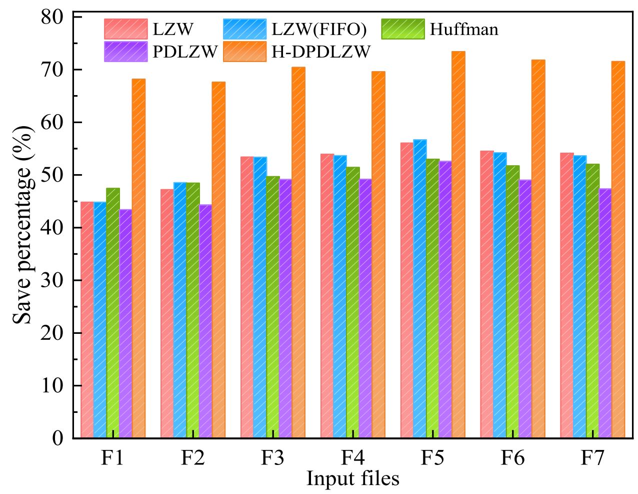

FIGURE 12. The save percentage of comparison algorithms.

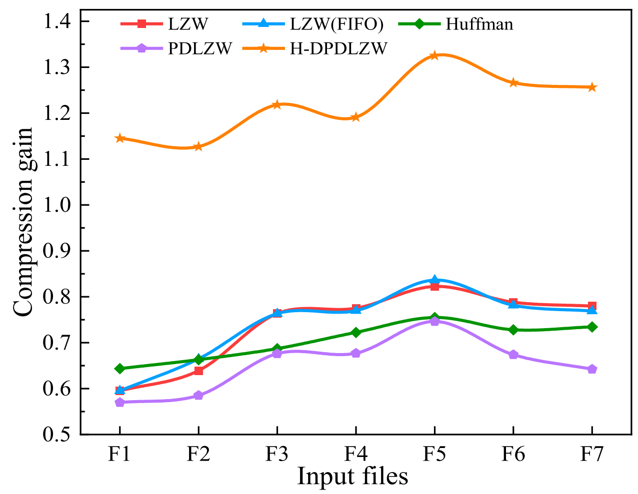

FIGURE 13. The compression gain of comparison algorithms.

4) COMPRESSION GAIN G

The compression gain is the ratio of the natural logarithm of M to N.

$$G = ln(M/N) \tag{4}$$

A higher G value indicates a better compression effect of the algorithm and a greater reduction in data storage space or transmission bandwidth.

5) COMPRESSION TIME T

The compression time represents the time interval required for a compression algorithm to perform the data compression process. It is a fundamental indicator to quantify the computational efficiency and speed of compression algorithms. A smaller T value means a faster compression process and also means superior algorithmic efficiency and computational efficiency. By minimizing T, compression algorithms can facilitate fast processing and transmission of data, thus ensuring timely delivery of compressed data.

Given the resource-constrained nature of power systems, storage capacity and communication bandwidth are often limited. Therefore, in the field of SG data compression technology, CR becomes a more important criterion for assessing the efficiency of data compression technology in power-centric applications compared with other evaluation indicators mentioned above. Meanwhile, CR is also regarded as the most important evaluation indicator in this paper.

C. THE DICTIONARY SELECTION

To determine the optimal dictionary parameter for DPDLZW algorithm, the MATLAB simulation environment is utilized. The compression performance of different combinations of dictionary partitions, as depicted in Table 1, is evaluated primarily based on the CR. The number of dictionaries is used as the criterion for dividing and compressing the test files, and the results are shown in Fig. 9.

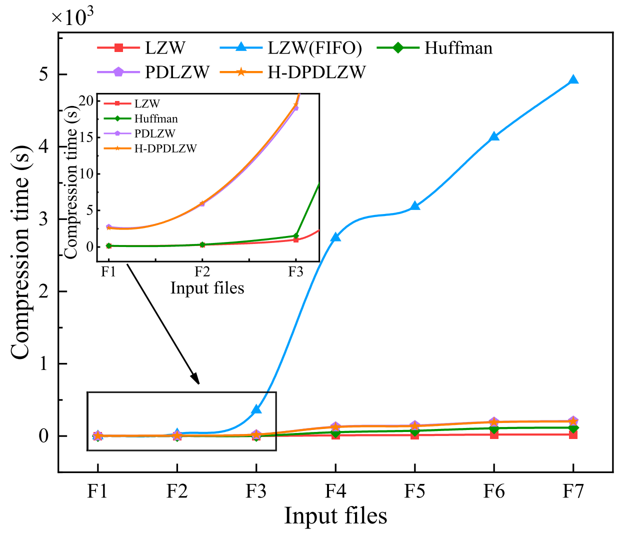

FIGURE 14. The compression time of comparison algorithms.

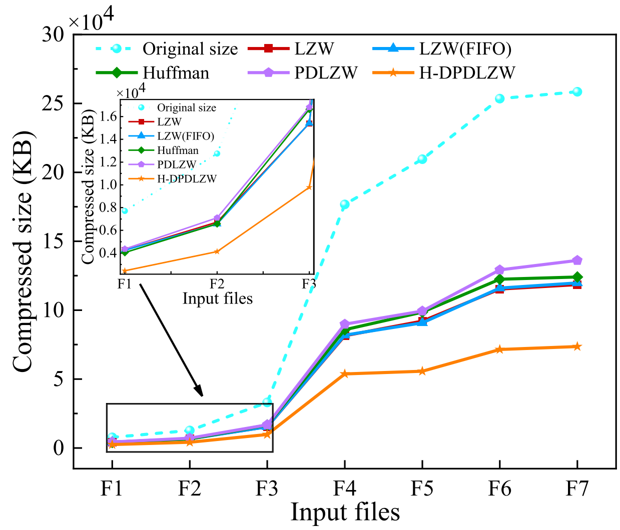

FIGURE 15. The compressed size of comparison algorithms.

It can be observed that DPDLZW algorithm achieves the highest *CR* at the case of I = 13 from Fig. 9. Therefore, the combination of dictionary partitions when I = 13 is chosen for H-DPDLZW algorithm, *MAX*=13, and the capacity of each small dictionary is [Dict0, Dict1…, Dict12] = [128, 512, 512, 512, 512, 256, 256, 256, 256, 256, 256, 256, 128]. Furthermore, it can be observed that this algorithm has the best compression effect on F5.

D. THE COMPARISON EXPERIMENTS

To verify the overall performance of the proposed lossless compression algorithm, the LZW algorithm, the LZW algorithm with the FIFO strategy, the Huffman algorithm, PDLZW algorithm and H-DPDLZW algorithm are used to compress the test files listed in Table 2. The compression process is conducted within the MATLAB simulation environment, and the compression evaluation indicators are used to analyze the compression results. The compression results are shown in Figs.10-16.

Fig. 10 demonstrates that in terms of *CR*, compared to the conventional LZW algorithm, Huffman algorithm and PDLZW algorithm, the proposed H-DPDLZW algorithm has improved the compression efficiency by 52%, 59% and 67%, respectively. Moreover, the highest *CR* realized by the proposed algorithm can reach 3.762.

The results presented in Fig. 11 highlight the significant improvements in compression efficiency achieved by the proposed H-DPDLZW algorithm in terms of *CF*. Specifically, when compared to the traditional LZW algorithm, Huffman algorithm, and PDLZW algorithm, the H-DPDLZW algorithm demonstrates enhanced compression efficiencies of 52%, 59%, and 67%, respectively. Meanwhile, the proposed algorithm achieves the minimum *CF* of 0.266.

The results presented in Fig. 12 illustrate the compression performance of the proposed H-DPDLZW algorithm for *SP*. Comparative analysis against the conventional

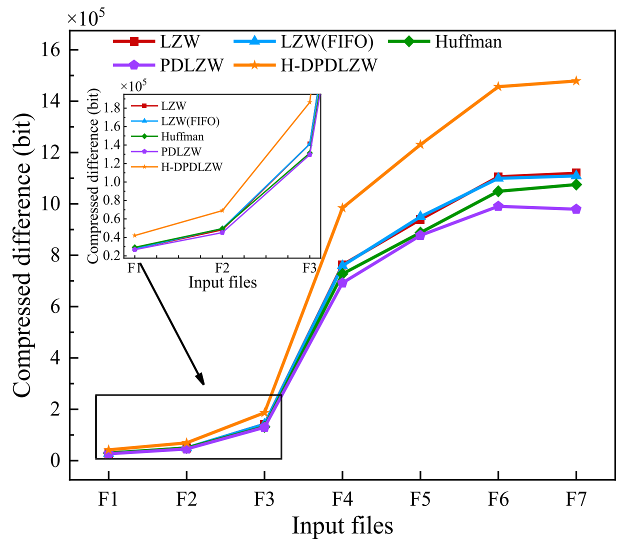

FIGURE 16. The compressed difference of comparison algorithms.

LZW algorithm, Huffman algorithm, and PDLZW algorithm demonstrates improvements in compression efficiency by 29%, 35%, and 40%, respectively. Notably, when tested on the data from this paper, the proposed algorithm achieves the highest *SP* of 73.421%.

Fig. 13 indicates that the data compression performance of H-DPDLZW algorithm, assessed through *G*. Comparative analysis the LZW algorithm, Huffman algorithm, and PDLZW algorithm reveals impressive improvements of more than 54%, 65%, and 76%, respectively. Additionally, the H-DPDLZW algorithm significantly reduces the data storage space. It attains the highest *G* of 1.325 for the test files data in this paper.

It can be seen from Fig. 14 that the LZW algorithm uses the shortest *T* when compressing the data, the LZW(FIFO) algorithm spends the longest *T*, and the proposed H-DPDLZW algorithm takes almost the same T as PDLZW algorithm. This shows that the proposed algorithm improves CR while sacrificing T compared to LZW and Huffman algorithms, and that the proposed algorithm improves CR and shortens T compared to PDLZW and LZW (FIFO) algorithms.

It can be observed that the proposed H-DPDLZW algorithm demonstrates superior overall compression performance compared to LZW algorithm, Huffman algorithm, and PDLZW algorithm. However, in terms of compression time, LZW algorithm exhibits the shortest duration, and the proposed algorithm takes more time than LZW algorithm. Additionally, it can also be seen that H-DPDLZW algorithm compresses the power data, the average CR is 3.389, the average CF is 0.296, the average SP is 70.368%, and the average G is 1.218. Its performance is better than that of the comparison algorithm.

V. CONCLUSION

This manuscript has proposed a dynamic and parallel two-stage lossless data compression method to improve the adaptability and compression ratio of the technology for smart grid applications. The proposed method has two stages: Stage 1 has been realized by improving the conventional LZW algorithm through incorporating parallel search and dynamic coding together (called DPDLZW). Stage 2 has formed the proposed method by cascading the improved algorithm in Stage 1 with conventional Huffman algorithm (called H-DPDLZW). Simulations have been carried out in MATLAB environment based on the real smart grid data, and the results have been quantitively compared to the conventional algorithms of LZW, Huffman and PDLZW in terms of compression ratio (CR), compression factor (CF), save percentage (SP) and compression gain (G). The performance has been improved by at least 52% in CR and CF, 29% in SP and 54% in G, respectively. Thus, the data compression efficiency of the proposed method has been successfully verified. Furthermore, the proposed algorithm not only performs well in smart grid applications, but also has wide applicability and scalability for smart grid data. Its excellent compression effect effectively saves storage space and provides a more feasible solution for expanding and deepening the analysis related to smart grid data. This research result provides theoretical support for data processing and application in the future smart grid field.

REFERENCES

[1] D. Syed, A. Zainab, A. Ghrayeb, S. S. Refaat, H. Abu-Rub, and O. Bouhali, "Smart grid big data analytics: Survey of technologies, techniques, and applications," *IEEE Access*, vol. 9, pp. 59564–59585, 2021.

[2] E. Y. Song, G. J. FitzPatrick, K. B. Lee, and E. Griffor, "A methodology for modeling interoperability of smart sensors in smart grids," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 555–563, Jan. 2022.

[3] K. Shahid, K. Nainar, R. L. Olsen, F. Iov, M. Lyhne, and G. Morgante, "On the use of common information model for smart grid applications—A conceptual approach," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 5060–5072, Nov. 2021.

[4] T. A. Alghamdi and N. Javaid, "A survey of preprocessing methods used for analysis of big data originated from smart grids," *IEEE Access*, vol. 10, pp. 29149–29171, 2022.

[5] Y. Hu, "Exploring approximate communication using lossy bitwise compression on interconnection networks," *IEEE Access*, vol. 11, pp. 59238–59249, 2023.

[6] P. Delgosha and V. Anantharam, "Universal lossless compression of graphical data," *IEEE Trans. Inf. Theory*, vol. 66, no. 11, pp. 6962–6976, Nov. 2020.

[7] J. Khan, S. Bhuiyan, G. Murphy, and J. Williams, "Data denoising and compression for smart grid communication," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 2, no. 2, pp. 200–214, Jun. 2016.

[8] Z. Li, Z. Chang-bao, M. Shiqiang, and L. Guoli, "An improved method based on wavelet for power quality compression," in *Proc. IEEE 11th Conf. Ind. Electron. Appl. (ICIEA)*, Hefei, China, Jun. 2016, pp. 1750–1753.

[9] J. Ning, J. Wang, W. Gao, and C. Liu, "A wavelet-based data compression technique for smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 1, pp. 212–218, Mar. 2011.

[10] J. Cormane and F. A. de O. Nascimento, "Spectral shape estimation in data compression for smart grid monitoring," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1214–1221, May 2016.

[11] P. H. Gadde, M. Biswal, S. Brahma, and H. Cao, "Efficient compression of PMU data in WAMS," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2406–2413, Sep. 2016.

[12] 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, no. 1, pp. 275–284, Jan. 2017.

[13] N. Hashemipour, J. Aghaei, A. Kavousi-Fard, T. Niknam, L. Salimi, P. C. del Granado, M. Shafie-Khah, F. Wang, and J. P. S. Catalão, "Optimal singular value decomposition based big data compression approach in smart grids," *IEEE Trans. Ind. Appl.*, vol. 57, no. 4, pp. 3296–3305, Jul. 2021.

[14] S. Kaur and N. Bawa, "Improved SVD-WDR compression with gradient-based optimization on images," in *Proc. 4th Int. Conf. Parallel, Distrib. Grid Comput. (PDGC)*, Waknaghat, India, Dec. 2016, pp. 647–651.

[15] R. Pourramezan, R. Hassani, H. Karimi, M. Paolone, and J. Mahseredjian, "A real-time synchrophasor data compression method using singular value decomposition," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 564–575, Jan. 2022.

[16] Q. Li, M. Zhang, S. He, and S. Li, "Instructions data compression for smart grid monitoring using wavelet domain singular value decomposition," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Shenyang, China, Jun. 2018, pp. 5171–5175.

[17] C. Yun and W. Yu-Lei, "Lossless data compression scheme for intelligent distribution network monitoring system," *Guangdong Elect. Power*, vol. 34, no. 5, pp. 90–98, May 2021.

[18] J. Di, P. Yang, C. Wang, and L. Yan, "Layered lossless compression method of massive fault recording data," *Int. J. Circuits, Syst. Signal Process.*, vol. 16, pp. 17–25, Jan. 2022.

[19] S. Yamagiwa and S. Kuwabara, "Autonomous parameter adjustment method for lossless data compression on adaptive stream-based entropy coding," *IEEE Access*, vol. 8, pp. 186890–186903, 2020.

[20] M. P. Tcheou, L. Lovisolo, M. V. Ribeiro, E. A. B. da Silva, M. A. M. Rodrigues, J. M. T. Romano, and P. S. R. Diniz, "The compression of electric signal waveforms for smart grids: State of the art and future trends," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 291–302, Jan. 2014.

[21] L. Yan, J. Han, R. Xu, and Z. Li, "Model-free lossless data compression for real-time low-latency transmission in smart grids," *IEEE Trans. Smart Grid*, vol. 12, no. 3, pp. 2601–2610, May 2021.

[22] M. Zeinali and J. S. Thompson, "Impact of compression and aggregation in wireless networks on smart meter data," in *Proc. IEEE 17th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Edinburgh, U.K., Jul. 2016, pp. 1–5.

[23] Z. Bin Tariq, N. Arshad, and M. Nabeel, "Enhanced LZMA and BZIP2 for improved energy data compression," in *Proc. 4th Int. Conf. Smart Cities Green ICT Syst.*, Lisbon, Portugal, 2015, pp. 1–8.

[24] W. Yong-Zhen and S. Liang-Qing, "Application of data compression technology in ship power monitoring system," *J. Shanghai Inst. Ship Transp. Sci.*, vol. 43, no. 1, pp. 55–60, Mar. 2020.

[25] X. Xiao-Fei and C. Liang, "The LZ77 power data compression algorithm using integer wavelet transform," *J. Xi'an Poly. Univ.*, vol. 32, no. 3, pp. 337–342, Jun. 2018.

[26] S. Bhuiyan, J. Khan, and G. Murphy, "Weighted entropy for data compression in smart grid," Presented at the IEEE Indus. App. Soc. Annual Meeting (IAS), Portland, OR, USA, 2018.

[27] G. Shrividhiya, K. S. Srujana, S. N. Kashyap, and C. Gururaj, "Robust data compression algorithm utilizing LZW framework based on Huffman technique," in *Proc. Int. Conf. Emerg. Smart Comput. Informat. (ESCI)*, Pune, India, Mar. 2021, pp. 234–237.

[28] L. C. Kho, Y. Tan, and Y. Lim, "Joint LZW and lightweight dictionary-based compression techniques for congested network," in *Proc. Int. Conf. Comput., Commun., Control Technol. (I4CT)*, Kuching, Malaysia, Apr. 2015, pp. 196–200.

[29] M. Safieh and J. Freudenberger, "Address space partitioning for the parallel dictionary LZW data compression algorithm," in *Proc. 16th Can. Workshop Inf. Theory (CWIT)*, Hamilton, ON, Canada, Jun. 2019, pp. 1–6.

[30] X. Liu, C. Deng, B. Lang, D. Tao, and X. Li, "Query-adaptive reciprocal hash tables for nearest neighbor search," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 907–919, Feb. 2016.

[31] R. Arshad, A. Saleem, and D. Khan, "Performance comparison of Huffman coding and double Huffman coding," in *Proc. 6th Int. Conf. Innov. Comput. Technol. (INTECH)*, Dublin, Ireland, Aug. 2016, pp. 361–364.

[32] P. Mishra, C. Bhaya, A. K. Pal, and A. K. Singh, "Compressed DNA coding using minimum variance Huffman tree," *IEEE Commun. Lett.*, vol. 24, no. 8, pp. 1602–1606, Aug. 2020.

[33] W. Yan-Xia, "An improved lossless compression algorithm for unstructured data based on the power monitoring system," Presented at the 6th Int. Conf. Tech. Soc. Sci. (ICTSS), Kiryu, Japan, Dec. 2022.

[34] M. Safieh and J. Freudenberger, "Efficient VLSI architecture for the parallel dictionary LZW data compression algorithm," *IET Circuits, Devices Syst.*, vol. 13, no. 5, pp. 576–583, Jul. 2019.

YANXIA WU was born in Datong, Shanxi, China, in 1999. She received the bachelor's degree in engineering from the Jilin Business and Technology College, Changchun, China, in 2021. She is currently pursuing the master's degree with the Jiangsu University of Science and Technology. Her research interests include lossless data compression algorithms and their applications for smart grids.

SONG XU (Member, IEEE) was born in Zhenjiang, China in 1991. He received the B.S. and M.S. degrees (Hons.) from the College of Electrical, Energy and Power Engineering, Yangzhou University, Yangzhou, China, in 2014 and 2017, respectively, and the Ph.D. degree from the Division of Electronics and Informatics, College of Science and Technology, Gunma University, Kiryu, Gunma, Japan, in 2020. From March 2020 to October 2020, he was a Foreign Researcher with Gunma University. In 2020, he joined the Jiangsu University of Science and Technology, where he is currently a Lecturer with the College of Automation. His current research interests include big data compression, digitalized power transfer to renewable energy and energy storage devices, and intelligent control applications in industrial processes.

CAIPING XI was born in Zhenjiang, China, in 1982. She received the B.S., M.S., and Ph.D. degrees from the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2004, 2006, and 2017, respectively. Since 2006, she has been with the Jiangsu University of Science and Technology, China, where she is currently an Associate Professor with the College of Automation. Her research interests include fractal signal processing and fractal theory and their applications in the radar target detection.

PENGQIANG NIE (Student Member, IEEE) was born in Chuzhou, Anhui, China, in 1997. He received the bachelor's degree in engineering from the Chongqing University of Technology, Chongqing, China, in 2020. He is currently pursuing the master's degree with the Jiangsu University of Science and Technology. His research interests include wireless power transfer and multiport dc–dc converter.

WEI JIANG (Member, IEEE) was born in Yangzhou, China, in 1980. He received the B.S.E.E. degree from Southwest Jiaotong University, Chengdu, China, in 2003, and the M.Sc. and Ph.D. degrees in electrical engineering from The University of Texas at Arlington, Arlington, TX, USA, in 2006 and 2009, respectively. From 2007 to 2008, he was a Senior Design Engineer with EF Technologies L.L.C., College Station, TX, USA. In 2010, he joined Yangzhou University and founded the Smart Energy Laboratory, where he is currently a Professor. He was a Visiting Professor with Gunma University, Japan, in 2012, the University of Strathclyde, and Aston University, U.K., in 2015. He holds two U.S. patents and 15 Chinese patents with two licensed by the industry. His current research interests include digitalized power conditioning to renewable energy and energy storage devices and microscopic analysis of electromechanical energy conversion. He was a recipient of the Yangzhou University Excellent Teaching Award for four times.

SEIJI HASHIMOTO (Member, IEEE) was born in Aomori, Japan, in 1971. He received the M.E. and Ph.D. degrees in electrical and electronic engineering from Utsunomiya University, Japan, in 1996 and 1999, respectively. He joined the Department of Mechanical Engineering, Oyama National College of Technology. Since 2002, he has been a Research Associate with the Department of Electronic Engineering, Gunma University, where he is currently a Professor with the Division of Electronics and Informatics. He has consulted for companies in control and energy applications and has been a Visiting Professor in China. His research interests include system identification, motion control, intelligent control, and its application to industrial fields.