



Article Novel Data Compression Algorithm for Transmission Line Condition Monitoring

Gang Liu^{1,2}, Lei Jia¹, Taishan Hu¹, Fangming Deng^{3,*}, Zheng Chen⁴, Tong Sun⁴ and Yanchong Feng⁴

- ¹ Electric Power Research Institute, CSG, Guangzhou 510663, China; liugang@csg.cn (G.L.); jialei@csg.cn (L.J.); huts@csg.cn (T.H.)
- ² National Engineering Laboratory for UHV Technology, Kunming 651705, China
- ³ School of Electrical and Automation Engineering, East China Jiaotong University, Nanchang 330013, China
- ⁴ Zhaoqing Power Supply Bureau, Guangdong Power Grid Co., Ltd., Zhaoqing 526060, China; chenzheng@zq.gd.csg.cn (Z.C.); suntong@zq.gd.csg.cn (T.S.); fengyanchong@zq.gd.csg.cn (Y.F.)
- Correspondence: 2464@ecjtu.edu.cn

Abstract: For the problem of data accumulation caused by massive sensor data in transmission line condition monitoring system, this paper analyzes the type and amount of data in the transmission line sensor network, compares the compression algorithms of wireless sensor network data at home and abroad, and proposes an efficient lossless compression algorithm suitable for sensor data in transmission line linear heterogeneous networks. The algorithm combines the wavelet compression algorithm and the neighborhood index sequence algorithm. It displays a fast operation speed and requires a small amount of calculation. It is suitable for battery powered wireless sensor network nodes. By combining wavelet correlation analysis and neighborhood index sequence coding, the compression algorithm proposed in this paper can achieve a high compression rate, has strong robustness to packet loss, has high compression performance, and can help to reduce network load and the packet loss rate. Simulation results show that the proposed method achieves a high compression rate in the compression of the transmission line parameter dataset, is superior to the existing data compression algorithms, and is suitable for the compression and transmission of transmission line condition monitoring data.

Keywords: transmission line; condition monitoring; data compression

1. Introduction

The rapid progress in the field of the Internet of Things and information technology makes it possible to realize low-cost, low-power, densely distributed and autonomous sensor nodes. The wireless sensor network (WSN) is the most representative Internet of Things technology. It collects and analyzes data by setting up billions of nodes to realize the vision of the interconnection of all things. A WSN is suitable for tracking and data acquisition applications such as environmental monitoring, industrial automation, target tracking, precision agriculture, disaster management, smart city, health monitoring, and earthquake and structure monitoring. In the actual scene, it is mainly used to sense physical parameters such as temperature, pressure, humidity, acceleration, vibration and sound signal. Each sensor node is equipped with a battery, a wireless transceiver, a target sensor and a micro-controller. After successful deployment, the sensor node begins to perceive physical parameters and transmit the collected data directly or indirectly to the sink node through multi-hop communication. Each terminal node of the monitoring system needs to collect and send a large amount of data. Once the battery is exhausted, the sensor node will no longer work and remove the monitoring network, resulting in the loss of network data and some functions. The experimental results show that the energy consumed by the wireless transmission transceiver is much higher than that consumed by data processing in transmission line wireless sensor networks. According to the node test, the energy



Citation: Liu, G.; Jia, L.; Hu, T.; Deng, F.; Chen, Z.; Sun, T.; Feng, Y. Novel Data Compression Algorithm for Transmission Line Condition Monitoring. *Energies* **2021**, *14*, 8275. https://doi.org/10.3390/en14248275

Academic Editors: Alistair Duffy and Ahmed Abu-Siada

Received: 17 September 2021 Accepted: 1 December 2021 Published: 8 December 2021

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



Copyright: © 2021 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/). consumed by the ordinary sensor node for each bit of data transmitted is enough for the node controller to process the data 551 times [1]. The sensor nodes in the transmission line online monitoring system are not easy to obtain cable power supply because of their wide laying range and long distance. Even some sensor nodes are deployed in harsh environments and cannot charge or replace the built-in battery. Therefore, the limited energy, mainly from batteries, has become the bottleneck of wireless sensor networks in transmission line applications.

Data compression is an effective technology to reduce the amount of data transmission. Using a data compression algorithm in wireless network can reduce the transmission pressure of the whole network, reduce the packet loss rate in the transmission process and improve the quality of wireless communication. In addition, using a data compression algorithm in wireless sensor networks can save system energy consumption and prolong the service life of the whole wireless sensor network. Therefore, data compression before transmission is an important strategy to save node energy in wireless sensor networks. The basic idea of data compression is to eliminate redundant and irrelevant data. Data compression represents data in a more concise form and does not change the data structure to a certain extent. Literature research shows that data compression technology can achieve a good balance between computational energy and transmission energy, so as to improve energy utilization efficiency.

Many scholars at home and abroad are committed to solving the problem of data compression in wireless sensor networks. T. Schoellhammer et al. [2] introduces a Lightweight Temporal Compression (LTC) technology, which provides an error with the last collected data while transmitting the sensor data. The error range is controlled by the threshold program. If the error is less than the threshold, the error value will replace the sensor data. The higher the error limit, the greater the storage space saved by compression, However, if the error threshold is not selected properly, it will cause serious data loss, which is not suitable for applications with high-reliability requirements. Sensor-Lempel Ziv Welch (S-LZW) is an improved LZW dictionary coding model and a lightweight compression algorithm specially developed for resource constrained WSN [3]. However, the algorithm is affected by the standard dictionary problem, and the compression performance needs to be further improved. Lossless entropy compression (LEC) [4] is a predictive coding technology composed of predictor and encoder. The LEC algorithm can calculate the difference between the measured values of continuous sensors and divide them into groups whose size increases exponentially. Each group of data represents the number of bits required to measure the difference, and then these groups are entropy encoded by a fixed compression table. For S-LZW, LEC has better compression performance, but because LEC is a static data compression algorithm, which is suitable for unified data compression after sensor node sampling and collection, it cannot adapt to dynamic data compression in transmission line online monitoring system.

In order to overcome this problem, Liang Y et al. [5] proposed an efficient and robust data compression technology Sequence Lossless Entropy Compression (S-LEC) technology, which uses the sequence context information between adjacent data residuals to compress the data in WSN. Adaptive lossless data compression (ALDC), which can dynamically adapt to the modification of source data statistics, can improve the compression performance [6]. ALDC can adaptively compress different data types through multiple codes at one time. Although it has strong robustness, its compression performance is not as good as LEC and other static data compression experiment of the online monitoring system is carried out by using the adaptive linear filter compression (ALFC) algorithm in [7]. Firstly, the adaptive linear filter is used to predict m samples of the dataset, and then the entropy encoder is used to compress the prediction error. Finally, the results are quantified to meet the limited computing power of the sensor nodes. The adaptive prediction process in ALFC does not need to define the filter coefficients in advance, and allows the system to dynamically adjust according to the changes of data sources. This method can adapt to

the specific certainty of online monitoring data over time and eliminate more redundant data. Compared with the existing methods, ALFC can obtain a better compression ratio. However, it is easy to recover data, easy to be affected by external factors, and difficult to adapt to the online monitoring system of transmission lines with many and complex influencing factors. According to compressed and uncompressed modes, a two-modal transmission (TMT) model for predictive coding is proposed in [8]. In the compression mode, the error term of the prediction data error within the compression bit range specified by the model will replace the original data transmission with the data error, while in the non-compression mode, the error term outside the compression bit range of the prediction data error will directly transmit the original data. This method solves the problem that the prediction performance and coding efficiency of the system are reduced due to large error terms. In order to realize the fast and low memory compression algorithm of WSN, Kolo J G et al. [9] designed a fast and efficient lossless adaptive compression scheme (felacs). Due to the Golomb rice encoding method, felacs can generate a temporary encoding table when performing data compression, and has certain robustness to packet loss during data transmission.

The above data compression methods are designed on the premise that the sensor network data are not correlated. However, there is a strong correlation between the data collected by the transmission line online monitoring network. Transmission line state parameters include as air temperature, air humidity, tower base temperature and humidity, wind speed, vibration signal, and leakage current. The numerical change of these parameters in the time dimension is a continuous process, and for a specific parameter, the data collected at several continuous acquisition points have great correlation. Therefore, considering the correlation of transmission line monitoring data in the time dimension, the space dimension and the parameter dimension can further eliminate redundancy, improve the compression rate and reduce network transmission energy consumption. Kasirajan et al. [10] proposed a data compression method based on source node and target node estimation. The estimated value on the source node is similar to the data value of each sample. The algorithm first quantifies the difference between the data sample and its estimated value, and then transmits it to the next hop node instead of the actual data sample, which reduces the amount of data transmission. Finally, the compressed data will be restored to the original data in the estimator in the next hop node or the base station. Marcelloni and Vecchio [11] proposed a simple compression algorithm for WSN nodes. The scheme uses the high correlation of sample data to calculate the size of each data after compression, and obtains a dynamic temporary compression dictionary from the sink node. There is a high correlation continuously. Usually, the sample collected by a sensor is on a node. The algorithm can calculate the dynamic obtained from the sensor by the compressed version of each value. The size of the very small dictionary used is determined by an Analogue-to-Digital Conversion (ADC) resolution.

Wavelet has a wide range of theoretical applications in the field of data compression in wireless sensor networks. Most of the research is to compress data by mining the correlation between time and space. In the transmission line monitoring environment, the coupling relationship between parameters is relatively complex. Adding these relationships and using them for data compression can better improve compression efficiency. Zhu et al. [12] proposed an adaptive multi-modal data compression algorithm based on wavelet. The algorithm uses the least square estimation method to estimate the relevant data. The feature data are abstracted into a matrix, and then the wavelet transform is used for spatiotemporal correction, which can effectively make use of the correlation between the data and improve the compression ratio of the algorithm. Xiao et al. [13] improved the wavelet data compression algorithm, pointed out the linear relationship between different data, and can directly express the relationship between related parameters is complex, and the simple linear fitting method cannot express the correlation of parameters, which will cause serious data compression error.

According to the continuity characteristics of temporal and spatial changes of transmission line WSN monitoring data and different degrees of correlation between monitoring parameters, a wavelet correlation neighborhood index sequence algorithm (WCNIS) compression algorithm for transmission line environmental data considering both time correlation and parameter correlation of environmental data is proposed in this paper, Firstly, the redundant data are removed by wavelet transform, and then the data are compressed by dynamic compression algorithm. Data compression efficiency is further improved through parameter correlation, so as to reduce the amount of data transmission in the network, reduce node energy consumption and prolong network life.

2. Compression Algorithm of Transmission Line Online Monitoring Network

2.1. Network Architecture of the Algorithm

In order to realize real-time data transmission, this paper establishes the transmission line transmission architecture of wireless transmission packet based on chain network structure. The system design framework is shown in Figure 1. The real-time data transmission is realized on the premise of ensuring the transmission time. The data transmission network model is established according to the transmission layer structure of the transmission line condition monitoring system, and the data transmission network model is established by using optical fiber, ZigBee and cellular network wireless communication. The network with minimum delay and low cost is selected through multiple simulations. Since the data are finally transmitted directly from the substation or base station to the data center, the control center does not appear in the model. The default data transmission mode is that the substation or base station transmits the data to the data center through the power grid or high-speed and high-reliability transmission mode.



Figure 1. System Design Framework.

In this paper, an improved wavelet threshold denoising algorithm is designed for the sink node data of transmission line online monitoring system. Unlike other transmission nodes, the sink node can store the collected data in a certain capacity before compression transmission. Therefore, the proposed algorithm is different from compressed sensing in sensor networks. The algorithm compresses a certain amount of data every time and then transmits it. In fact, it works in the sink node of transmission line online monitoring system. Therefore, the system in this paper has real-time performance while removing redundant data.

2.2. Improved Wavelet Threshold Denoising Algorithm

In general, the changes of transmission line environmental parameters are periodic and have strong correlation in time series. Because many monitoring parameters such as temperature, humidity and vibration change slowly, the data at adjacent time points of the same node and the same parameter are usually the same or similar [14]. However, in the transmission line monitoring system, the correlation of sampling data such as wind speed and leakage current in time series is weak, and only considering the correlation of time series will cause large data error. Therefore, the establishment of a matrix composed of multiple environmental monitoring parameters in time series can take into account the correlation between time series and parameters at the same time.

Suppose a WSN is composed of N nodes, i.e., $S_0, S_1, \ldots, S_i, ..., S_{N-1}$. The data monitored by the sensor node S_i can form an m-dimensional time series signal [15]. Assuming that both M and N are even numbers, there is C_i , $j = (c_0, 1, c_1, i, ..., c_{M-1}, i)^T$, where $c_{i'j}$ represents the *j*th data of the time series signal stored by the sensor node Si. Therefore, the original data matrix C^0 is represented by Equation (1):

$$C^{0} = (C_{0}C_{1}\cdots C_{i}\cdots C_{N-1}) = \begin{bmatrix} c_{0,0}^{0} & c_{0,1}^{0} & \cdots & c_{0,N-1}^{0} \\ c_{1,0}^{0} & c_{1,1}^{0} & \cdots & c_{1,N-1}^{0} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M-1,0}^{0} & c_{M-1,1}^{0} & \cdots & c_{M-1,N-1}^{0} \end{bmatrix}$$
(1)

The matrix established by Equation (1) has good correlation and numerical similarity in only one direction, and its matrix characteristics can be decomposed and analyzed by wavelet. The original matrix C^0 has good correlation and data change continuity in the column direction, but there is no obvious correlation in the row direction. Therefore, firstly, the column transformation of the matrix is carried out, the data compression is carried out by using the time correlation, and then the row transformation of the matrix is carried out. l(x) represents the low-pass filter coefficient of wavelet analysis, $i = 0, 1, ..., N_l - 1$ represents the high pass filter coefficient of wavelet analysis, $i = 0, 1, ..., N_h - 1$. Then, the wavelet decomposition results of the original matrix C^0 in the column direction are as follows:

$$C^{1} = \begin{bmatrix} C^{1,H} \\ C^{1,L} \end{bmatrix}$$
(2)

The elements in matrix C^1 can be represented by the following Equation:

$$c_{x,y}^{1,H} = \frac{1}{N_h} \sum_{i=0}^{N_h - 1} h(i) c_{(2x+i) \text{mod}M,y}^0$$
(3)

$$c_{x,y}^{1,L} = \frac{1}{N_l} \sum_{i=0}^{N_l-1} l(i) c_{(2x+i) \bmod M,y}^0$$
(4)

where $x = 0, 1, \dots, \frac{M-1}{2}; y = 0, 1, \dots, N-1$.

Continue to perform directional wavelet transform on C^1 matrix, and the results are as follows:

$$C^{11} = \begin{bmatrix} C^{11,LL}C^{11,LH} \\ C^{11,HL}C^{11,HH} \end{bmatrix}$$
(5)

$$c_{x,y}^{11,LL} = \frac{1}{N_l} \sum_{i=0}^{N_l-1} l(i) c_{x,(2x+i) \bmod N}^{1,L}$$
(6)

$$c_{x,y}^{11,LH} = \frac{1}{N_h} \sum_{j=0}^{N_h - 1} h(j) c_{x,(2x+j) \bmod N}^{1,L}$$
(7)

$$c_{x,y}^{11,HL} = \frac{1}{N_l} \sum_{i=0}^{N_l-1} l(i) c_{x,(2x+i) \text{mod}N}^{1,H}$$
(8)

$$c_{x,y}^{11,HH} = \frac{1}{N_h} \sum_{j=0}^{N_h - 1} h(j) c_{x,(2x+j) \bmod N}^{1,H}$$
(9)

where $x = 0, 1, \dots, \frac{M}{2} - 1; y = 0, 1, \dots, \frac{N}{2} - 1$.

According to the correlation and continuous variation characteristics of the sampled data, when the sample data are decomposed by wavelet, the correlation in the column direction is poor and the value does not have continuity. The value of the high-frequency part of the decomposed matrix is large, so it is difficult to obtain a coefficient of 0 or approximately 0. When the matrix after wavelet decomposition is compressed, discarding the data with a coefficient of 0 or approximately 0 will not affect the data reconstruction, and discarding the data with a coefficient of not 0 will affect the compression accuracy. Therefore, the high-frequency part of the original matrix after wavelet decomposition in the column direction will seriously affect the data accuracy and compression efficiency. In order to solve this problem, it is necessary to analyze the correlation of data in a complete cycle. The acquisition times in period *T* are recorded as *i*, and the data sequence of any parameter *s* in any node *P_i* acquisition period *T* is recorded as

$$P_s(T) = \left\{ p_s^T(0), p_s^T(-1), \dots, p_s^T(-k+1) \right\}$$
(10)

where $P_s(T)$ is the data sequence of parameter *s*; $p_s^T(0)$ is the final sampling value of parameter *s*; $p_s^T(-k+1)$ is the initial sample value of parameter *s*. For any parameter *S*₁ and *i* collected by node *P*, the correlation between the two sequences can be measured by calculating the Pearson correlation coefficient:

$$\hat{\rho}(P_{s_1}(A), P_{s_2}(A)) = \frac{\sum_{i=1}^n \left(p_{s_1}^A(i) - \overline{p}_{s_1} \right) \left(p_{s_2}^A(i) - \overline{p}_{s_2} \right)}{\sqrt{\sum_{i=1}^n \left(p_{s_1}^A(i) - \overline{p}_{s_1} \right)^2 \cdot \sum_{i=1}^n \left(p_{s_2}^A(i) - \overline{p}_{s_2} \right)^2}}$$
(11)

where *A* is the selected time period for collecting data; $\hat{\rho}(P_{s_1}(A), P_{s_2}(A))$ is the correlation coefficient; \hat{p}_{s_1} and \hat{p}_{s_2} are the average of sequence $P_{s_1}(A)$ and $P_{s_2}(A)$ respectively.

Definition 1. *Mean square error (MSE): If the original signal* $S = \{X(t)|t = 1, 2, 3, \dots k\}$, $S^* = \{X^t(t)|t = 1, 2, 3, \dots k\}$ is an estimate of *S*, then the MSE of the estimate is

$$MSE_{S}^{*} = \frac{1}{K-1} \sum_{t=1}^{K} [X(t)X^{*}(t)]^{2}$$
(12)

Definition 2. Data change level: Set node i's data in the tth sampling as D_t^i , the data change level L_t^i defined as

$$L_t^i = floor\left(\frac{|\Delta D_t^i|}{\alpha}\right) + 1 \tag{13}$$

where $\Delta D_t^i = D_t^i - D_{previous}^i$, the variable $D_{previous}^i$ stores the last sample data of a node whose L > 1, that is, $D_{previous}^i = D_k^i k = \max\{r | L_r^i > 1, 2 \le r \le t - 1\}$, its initial value $D_{previous}^i = D_1^i$; floor(x) Return x rounding down; α is the threshold for determining the level, which is set according to the specific application. This paper uses (MSE – 1) as a measure of data accuracy with the following assumptions:

- (1) The sampling frequency of nodes is $\frac{1}{\Lambda t}$;
- (2) The variable P is the Sink precision threshold for data processing;

(3) The variable PP is the mark of precision processing.

For the data processing method in the wavelet cluster of wireless sensor networks mentioned in this paper, it is assumed that sensor nodes can form different cluster areas according to the data correlation. As a result, the data correlation between cluster areas is relatively small, while the wavelet transformation has obvious advantages only when the data correlation is high. Therefore, only data processing based on one-layer cluster structure is considered. A node within each cluster is selected as the cluster head to process the data within the cluster.

Let the data sampled by the node *i* in a cluster at time *t* be ΔD_t^i . If $L_t^i = 1$, only ΔD_t^i data will be transmitted to the cluster head; If L_t^i , the L_t^i and ΔD_t^i are transmitted to the cluster head together, and $D_{previous}^i = \Delta D_t^i$. Set the vector $D_t = [D_t^1, D_t^2, \dots, D_t^n,]^T$ to represent the data of the *t*th sampling of all n active nodes in the same cluster, corresponding to the data change vector $L_t = [L_t^1, L_t^2, \dots, L_t^n]$. If the area of the original Voronoi polygon of the corresponding node is: $[V_0^1, V_0^2, \dots, V_0^n]$, then the cluster head generates a new Voronoi polygon area of all n nodes in the *t*th order based on vector L_t :

$$[V_t^1, V_t^2, \cdots, V_t^n] = [L_t^1 \times V_0^1, L_t^2 \times V_0^2, \cdots, L_t^n \times V_0^n,]$$
(14)

When the $[V_t^1, V_t^2, \dots, V_t^n]$ is determined, the irregular wavelet transform basis function matrix K_t for the *t*th sampling within the cluster is calculated as

$$K_{t} = \begin{pmatrix} k_{0} & k_{0} & k_{0} & \cdots & k_{0} \\ k'_{1} & k_{1} & k_{1} & \cdots & k_{1} \\ k_{2} & k_{2} & k_{2} & \cdots & k_{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{n} & k_{n} & k_{n} & \cdots & k'_{n} \end{pmatrix}$$
(15)

where $K_i \begin{cases} \frac{1}{\overline{V_{tot}}}, i = 0\\ \frac{\overline{V_t^i}}{\overline{V_{tot}}}, i > 0 \end{cases}$, $V_{tot} = \sum_{i=0}^n V_t^i; k_i' = -\frac{k_i (V_{tot} - V_t^i)}{V_t^i}, 1 \le t \le n.$

The row of matrix K_i defines the basis function of irregular wavelet transform, in which the first row is a scale function with constant value, and the elements in row *i* and column *j* represent the value $(1 < i \le (n + 1), 1 \le j \le n)$ of the (i - 1) wavelet function in the Voronoi region of the sensor node.

Then, a diagonal matrix $V'_t = \text{diag}(V^1_t, V^2_t, \dots, V^n_t)$ is defined and an analysis matrix A_t is constructed as

$$A_t = K_t V_t' \tag{16}$$

Using this matrix, according to the equation

$$c_t = A_t D_t \tag{17}$$

Calculate the irregular wavelet transform coefficient of data vector D_t : one $(n + 1) \times$ The one-dimensional transform coefficient vector $c_t = [s_t, w_t^1, w_t^2, \dots, w_t^n]^T$, where the first element of c_t is the scale coefficient $w_t^i (i = 1, 2, \dots, n)$ after irregular wavelet transform, which is the wavelet coefficient, and let the vector $w_t = [w_t^1, w_t^2, \dots, w_t^n]$. Calculate the composite matrix s_t as

$$S_t = (V_t')^{-1} (A_t)^T$$
(18)

The matrix can be used to reconstruct the transformation coefficients to the sampled datasets:

$$D'_t = S_t c'_t \tag{19}$$

where c'_t coefficient vector consisting of scale coefficients and partial wavelet coefficients; D'_t is the reconstructed data vector.

To sum up, after obtaining the sampling data and data change vector L of each member in the cluster for the t time, the cluster head has the ability to process the sampling data by wavelet. According to Equation (17), the cluster head obtains the coefficient vector c_t by wavelet transform of the sampled data vector D_t . Using the composite matrix and scale coefficient, according to Equation (19), The cluster head reconstructs the area data to obtain D'_t , compares the original sampling data D_t , and calculates the MSE - 1 between the data to obtain the reconstruction accuracy P_t . If $P_t > P$, PP = 1, otherwise PP = 0. Thereafter, the cluster head transmits the obtained scale coefficients and PP to sink, and temporarily stores the remaining wavelet coefficients w_t and vector L_t . As shown in Figure 2, the cluster head m_6 transmits the scale coefficient and accuracy processing PP, and the wavelet coefficients w_t and L_t are retained.



Figure 2. Diagram of Data Processing in Cluster.

Finally, for the collected data, the threshold denoising method based on wavelet compression is adopted. The main idea of this method is that the data and noise after wavelet transform have different characteristics, that is, the amplitude of wavelet coefficient corresponding to the energy of real data are too large, while the amplitude of wavelet coefficient of interference noise is too small, which meets $x \ge \lambda$, x represents the wavelet decomposition coefficient and λ represents the set threshold. The denoising steps are as follows:

- (1) The noisy signal f(t) is decomposed into multiple layers by wavelet transform, and the corresponding coefficient of each layer is $\omega_{i,k}$.
- (2) Through the threshold processing of wavelet decomposition coefficient $\omega_{j,k}$, the estimated wavelet coefficient $\overline{\omega}_{j,k}$ is obtained to make $\omega_{j,k} \overline{\omega}_{j,k}$ as small as possible. The soft threshold function is selected as the threshold function:

$$f(x) = \begin{cases} \operatorname{sgn}(x)(|x| - \lambda), \ |x| > \lambda, \\ 0, \qquad |x| \le \lambda, \end{cases}$$
(20)

Among them, sgn(*x*) is the symbolic function. The threshold is selected as the local adaptive threshold. In this paper, the threshold is 3α , that is, arrange the coefficients of each layer according to the size, then take the median, and finally divide the median by 0.6745 to obtain that the α . $|x| > 3\alpha$ of the corresponding layer is generated by the signal, and the $|x| \le 3\alpha$ is generated by the noise

(3) The estimated wavelet coefficient $\overline{\omega}_{j,k}$ is used for wavelet reconstruction to obtain the estimated signal f(t), that is, the denoised signal.

In the process of transmission line condition monitoring, parameter correlation presents a many to many mapping state, and the correlation between parameters is relatively complex. Therefore, after the parameter classification analysis is carried out according to the change characteristics of parameter values, the correlation between this parameter and other parameters in the same category is calculated by selecting the main parameter in the category, The parameter clustering and correlation ranking in the category can reduce the information loss in the compression process and improve the compression accuracy. However, the wavelet correlation compression algorithm proposed in this paper mainly reduces the amount of data transmission and achieves the effect of data compression by eliminating the high-frequency component of the data matrix. In the process of data transmission, the ordinary coding algorithm is still used for transmission. Therefore, the compression ratio does not reach the best.

2.3. The Neighborhood Index Sequence Algorithm

Based on the wavelet correlation data compression algorithm, this paper proposes a neighborhood index sequence (NIS) algorithm for data compression in transmission line wireless sensor networks as the connection of data transmission compression. After analyzing the correlation information between the adjacent bits of the input characters, the algorithm can assign a shorter length codeword to each character of the input sequence, which is in the two shorter codewords generated by 0 to 1 traversal. The optimal codeword is selected according to the lowest bit number required to store the codeword of each character. For the input sequence with length N, the NIS algorithm needs C_{bits} to store compressed data, as shown in Equation (21):

$$C_{bits} = \sum_{i=1}^{N} NIS_{opt}(i) + control \ bits$$
⁽²¹⁾

where *NIS*_{opt} represents the number of bits in the codeword. The proposed NIS algorithm requires an additional 8 control bits to calculate the optimal bits in the compressed data. The average number of bits required to store a single character can be calculated from Equation (13):

$$NIS_{ch_av} = \frac{C_{bits}}{N}, 1 \le NIS_{ch_av} \le 4$$
(22)

The lower the value of C_{bits} and NIS_{ch_av} , the higher the compression performance of the algorithm. According to Equation (22), the NIS algorithm needs to store one character in 4 bits at most. However, in most cases, the NIS algorithm only needs to store one character in 1 bit, so it has better compression effect.

The overall process of NIS compression and decompression is shown in Figure 3. First, the algorithm gives priority to input text that may contain alphabetic characters and special symbols. After reading the characters in the data, convert them to the corresponding ASCII value, and then convert the ASCII value to the equivalent binary form. Then, perform the process of "traversal based on 0 and 1 data". Bit traversal starts with the first bit of the binary of the input character and identifies whether the bit is 0 or 1. Then, start the 0-based traversal and store the control bit as 00 (when the first bit recognized is 0) or 10 (when the first bit recognized is 1). Taking the first bit as the reference, the algorithm starts to traverse from the second bit, mainly identifying 0 and storing its position when finding the value of 0. Once the value of 0 is recognized, its position (z) will be stored in the codeword (00-z), and then restart the program until the last bit is recognized. When traversing to the last bit, store all currently recognized codewords, end the 0-based traversal process and start the 1-based traversal, which is the same as the 0-based traversal process. When two codewords are stored, the algorithm compares them and selects the codeword with the minimum number of bits as the optimal codeword. Finally, the optimal codeword of all generated coded characters is connected with the control bit and a compressed file is generated.



Figure 3. NIS Compression and Decompression Program Flow Chart.

The better compression performance of the NIS algorithm is mainly because it requires less bits to represent each character in the input sequence. As can be seen from Equation (13), any character compressed by the NIS algorithm can be stored by at least 1 bit or at most 4 bits. For example, the character "o" and the space character only need 1 bit codeword to represent; Characters "t" "h" "k" "p" need 2 bit; 3 bit can represent "e" "I" "u" "c" and other characters. Characters such as "f" "v" "a" "2" "6" can be represented by 4 bit codewords. Overall, the NIS algorithm uses an average of 3.4 bit to represent any character.

3. Experimental Results and Analysis

The software and hardware environment of this paper are shown in Table 1, and MATLAB language is used for programming.

 Category	Edition				
Operating system	Windows10				
CPU	Intel Core i9-10900k				
GPU	NVIDIA GeForce GTX 3080				
RAM	32 Gb				

Table 1. Software and hardware environment.

In order to verify the advantages of the algorithm proposed in this paper in the compression of transmission line condition monitoring data, 1089 groups of transmission line condition monitoring data in a place in 2020 are used as experimental samples. Four types of data are mainly used for compression analysis: transmission line temperature, transmission line humidity, transmission tower base temperature and transmission line wind speed. The statistics of sample data are shown in Figure 4.



Figure 4. Original Test Dataset.

Before data compression, it is necessary for us to select several universal and objective evaluation criteria to evaluate the effect of data compression and reconstruction, and as the basis for the comparison of different compression algorithms to draw an objective and credible conclusion. Generally, the performance of all compression algorithms is calculated by using compression ratio (*CR*), which can be expressed by the following Equation (23).

$$CR = 100 \times (1 - \frac{\text{Number of bits of compressed data}}{\text{Number of bits of uncompressed data}})$$
(23)

3.1. Wavelet Compression Analysis

Firstly, the correlation of sample data is calculated on MATLAB2016a platform according to Equation (11). The correlation between parameters is shown in Table 2.

Correlation Coefficient $\stackrel{\wedge}{ ho}$	Transmission Line Temperature	Ambient Relative Humidity	Leakage Current	Ambient Wind Speed	Conductor Wind Speed	Rainfall
Transmission line temperature	1	0.137	0.913	0.102	0.155	0.015
Ambient relative humidity	0.137	1	0.318	0.067	0.132	0.365
Leakage current	0.913	0.318	1	0.118	0.142	0.102
Ambient wind speed	0.102	0.067	0.118	1	0.914	0.043
Conductor wind speed	0.155	0.132	0.142	0.914	1	0.073
Rainfall	0.015	0.365	0.102	0.043	0.073	1

Table 2. Correlation coefficient of sample parameters.

The wind speed data and leakage current data collected by the wind speed sensor are taken as the samples for analysis. The denoising effect is shown in Figure 5. The leakage current can still retain the leakage current in case of fault through denoising. The error value of wind speed data is deleted by denoising, and the effective value is retained, which improves the quality of data.



Figure 5. Comparison of Denoising Effects of Leakage Current and Wind Speed Data. (**a**) Original value of leakage current. (**b**) Original value of wind speed. (**c**) Leakage current data processing results. (**d**) Wind speed data processing results.

(c)

According to the coefficients in Table 1, considering the correlation strength, the transmission line temperature has the strongest correlation with other parameters, and the wind speed has the weakest correlation with other parameters. The four parameters can be divided into two groups according to the correlation strength. The transmission line temperature and tower base temperature constitute a strong correlation group, and the ambient relative humidity and ambient wind speed constitute a weak correlation group.

After the parameter ranking table is obtained, the data matrix can be reorganized, and the original matrix can be represented by Equation (24).

$$c_{p}^{0}(T) = \begin{bmatrix} p_{s_{TL}}^{T}(0) & p_{s_{TT}}^{T}(0) & p_{s_{AR}}^{T}(0) & p_{s_{EW}}^{T}(0) \\ p_{s_{TL}}^{T}(1) & p_{s_{TT}}^{T}(1) & p_{s_{AR}}^{T}(1) & p_{s_{EW}}^{T}(1) \\ \vdots & \vdots & \vdots & \vdots \\ p_{s_{TL}}^{T}(k) & p_{s_{TT}}^{T}(k) & p_{s_{AR}}^{T}(k) & p_{s_{EW}}^{T}(k) \end{bmatrix}$$
(24)

(d)

where S_{TL} represents the transmission line temperature; S_{TT} represents the temperature of transmission tower foundation; *i* represents ambient relative humidity; S_{EW} stands for ambient wind speed. $p_{S_{TL}}^T$ is the main parameter vector. The vector $p_{S_{TR}}^T$ has strong correlation with the main parameter vector, and the vector $p_{S_{AR}}^T$ and $p_{S_{EW}}^T$ have weak correlation with the main parameter vector. The resulting parameter matrix can be calculated by

two-dimensional discrete wavelet transform to remove redundant data. This paper selects a group of 8×8 and the wavelet compression test is carried out on the sample data of 8, and the test results are shown in Figure 6.

	59 6	0 58 5	57 57	57	56 56	Г	110 5	115			5 0 5	1 0 7			
	61 5	9 59 5	7 56	56	56 56		119.5	115.3	5 113 11	2 -0.	5 - 0.5	1 0			
				50	50 50		120.5	117	116 11	4 0.5	1	1 -1			
	62 5	69 60 5	58 58	59	58 55		121.5	119	113 114	4.5 -0.	5 - 2	0 0.5			
<i>a</i> –	59 6	51 60 5	6 58	57	59 56	$\sum $	122	118.5	115 11	5 3	-0.5 -	2 1			
<i>C</i> –	61 6	0 59 5	8 55	58	59 56	\sim $c =$	0.5	1.5	0 0	-1.5	- 0.5	0 0			
	60 6	52 59 6	52 57	56	59 55	-	0.5	3	0 3	2.5	-1 -	1 0			
	61 6	4 60 5	8 57	56	58 58		- 0.5	-1	-1 3.5	5 1.5	2 -2	2 - 5	\bigcirc		
	59 6	0 61 5	8 58	50	57 57		-2	2	0 0	-1	-0.5	10			
	[37 0	0 01 5	0 50	57	57 57					I			\sim	/	
1	-	220.00	~	0		7									
	236	228 0	-2	0	0 0 0		236	5.25	227.5	-1.25	-2.5	-0.5	-0.5	1	0
	240	228 0	0	0	0 0 0		240).5	228.75	0	-1.25	0.5	1	1	-1
	2	2 0	0	0 -	200	quantification	3.2	5	1.5	0.25	-0.5	-0.5	-2	0	0.5
c –	4	0 0	0	4 (0 - 2 0		3	-	0.75	-0.5	-0.75	3	-0.5	-2	1
c –	0	2 0	0 -	-2 (0 0 0	$0 \qquad c =$	0.5		1.5	0	0	-1.5	-0.5	0	0
	0	4 0	4	2 (0 0 0		0.5		3	0	3	2.5	-1 -	-1	0
	0	0 0	4	2 2	2 - 2 - 4		- 0.	5	-1	-1	3.5	1.5	2 .	- 2	- 5
	- 2	2 0	0	0 (0 0 0		-2		2	0	0	-1	-0.5	1	0

Figure 6. An Example of Wavelet Change Compression.

The group of 8×8 is shown in Figure 6. After wavelet compression, 93.75% of the high-frequency components are eliminated, and a group of 2 is needed, except that the data of 2×2 needs multi-byte coding, and the other data only needs single-byte transmission.

3.2. NIS Compression Analysis

In order to verify the actual compression performance of the algorithm, this paper uses the NIS algorithm to transmit a user-defined string "Hello world", and records its compression process step by step, as shown in Figure 7. The original data "Hello world" contains 10 alphabetic characters and two space characters. First, convert all characters into ASCII code and binary code, and then determine that the control bit is "0" or "1" according to the first place of the binary code. If the control bit is determined to be "0", start the data traversal based on "0"; If it is determined that the control bit is "1", the data traversal based on "1" is started. After data traversal, two codeword tables will be generated. By comparing the number of bits of each codeword and using the smallest codeword, the optimal codeword table will be generated. Finally, compressed packets are generated for wireless transmission. The compression process and decompression process of the WCNIS algorithm are symmetrical. During decompression, first identify the codeword in the compressed packet to obtain the optimal codeword, then determine the position of 0 and 1 in the recovered binary data according to 0 and 1 in the control bit, then recover the binary code and ASCII code in turn, and finally obtain the recovered original data "Hello world".



Figure 7. Flow Chart of Sample Data Compression and Decompression Program.

According to Table 3, the byte size of the original data "Hello world" is 80 bit, and the optimal codeword size compressed by the NIS algorithm is 27 bit. According to Equation (23), the compression ratio of NIS compressed data is 66.25%.

Character	ASCII	Binary	Based on '0' Ergodic	Traversal Codeword Bits of '0'	Based on '1' Ergodic	Traversal Codeword Bits of '1'	Optimized Codeword Bit
Н	104	1101000	10-1,1, 0, 0	4	11-0, 1	2	2
е	101	1100101	10-1, 0, 1	3	11-1, 10, 0	4	3
1	109	1101100	10-1, 10, 0	4	11-0, 1, 0	3	3
0	111	1101111	10-1	1	11-0, 1, 0, 0, 0	5	1
w	119	1110111	10-10	2	11-0, 0, 1, 0, 0	5	2
space	32	0100000	00-1, 0, 0, 0, 0	5	01-0	1	1
r	114	1110010	10-10, 0, 1	4	11-0, 0, 10	4	4
d	100	1100100	10-1, 0, 1, 0	4	11-0, 10	3	3

Table 3. WCNIS codeword structure.

3.3. WCNIS Compression Analysis

In this paper, the WCNIS compression algorithm is composed of the wavelet correlation compression algorithm and the neighborhood index sequence algorithm. The wavelet correlation compression algorithm is used as the front end to calculate the correlation between source data and eliminate redundant data; the NIS algorithm is used as the back-end compression coding to further increase the compression rate. This paper tests the efficiency and reconstruction accuracy of compression algorithms, and focuses on data compression methods in sensor networks, which are application layer problems and do not involve the underlying protocols. Compared with the traditional NIS algorithm, the WCNIS algorithm in this paper greatly improves the transmission efficiency on high data volume. The two algorithms are run separately in Matlab to transmit a section of irregular data, and the results are shown in the following graphs. The proposed algorithm in this paper greatly reduces the transmission time of large volume of text compared with the original algorithm, and its running time is compared as follows Table 4. The runtime of the algorithm is visualized in Figure 8 below

Running Time of Algorithm T/s Amount of Data Transmitted	512	1024	1536	2048	2560	3072	3584	4096	4608	5120
NIS	0.0243	0.0472	0.0648	0.0854	0.1402	0.1657	0.2053	0.2548	0.3251	0.3458
WCNIS	0.0186	0.0352	0.0453	0.0563	0.0701	0.0832	0.0964	0.1147	0.1532	0.1632







In the simulation experiment, a set of multi-parameter transmission line datasets are used for testing, and the compression performance and power consumption are compared with the recently proposed LEC, S-LZW, ALDC and FELACS algorithms. In addition, they are also compared with the traditional compression algorithms Gzip [14], Bzip [16], rar, Huffman [17] and arithmetic coding [18]. This paper uses MATLAB 2016a simulation platform to run the WCNIS algorithm to compress the dataset, and compares the results before and after compression. The size, uncompressed size, compressed size and compression ratio CR of each dataset are shown in Table 5.

Table 5. Compression ratio of the WCNIS algorithm on different datasets.

Dataset Name	Dataset Size before Compression (Bit)	Compressed Dataset Size (Bit)	Compression Ratio CR (%)
Line temperature	8920	7874	88.27
Ambient humidity	8712	6681	76.69
Rainfall	8836	7441	84.21
Wind speed	8933	6113	68.43
Leakage current	8156	6912	84.48

In order to evaluate the compression performance of the WCNIS algorithm, run other compression algorithms in MATLAB 2016a software to process the original dataset and compare with the NIS algorithm. The comparison results are shown in Tables 6 and 7. It can be seen from Tables 6 and 7 that the method used in this paper obtains better compression performance than other methods on all application datasets.

Dataset Name	Compression Ratio CR (%)							
Dataset Maine	LEC [19]	S-LZW [20]	ALDC [21]	FELACS [22]	NIS [23]	This Paper		
Line temperature	64.54	31.36	67.43	68.12	72.31	88.27		
Ambient humidity	51.98	23.14	58.35	56.41	59.87	76.69		
Rainfall	62.86	32.45	66.38	67.31	70.87	84.21		
Wind speed	49.37	22.05	51.79	62.77	55.68	68.43		
Leakage current	50.35	31.24	65.34	58.66	78.21	84.48		

Table 6. Comparison between this method and existing compression algorithms.

Table 7. Comparison of compression performance of each compression algorithm.

Datasat Nama	Compression Efficiency (Bit/Sample)								
Dataset Name	LEC [19]	S-LZW [20]	ALDC [21]	FELACS [22]	NIS [23]	This Paper			
Line temperature	5.6432	11.1734	5.2245	5.2484	5.1259	1.9268			
Ambient humidity	7.3498	12.5673	6.5869	6.8251	5.9822	3.5189			
Rainfall	6.1322	11.6715	5.4136	5.3349	4.7554	2.4656			
Wind speed	8.2334	12.5322	7.8045	7.5812	7.5431	4.0643			
Leakage current	9.6523	11.3654	6.9635	6.7452	5.3329	3.5841			

In the existing methods, S-LZW compression efficiency is as high as 12 bits/sample, resulting in the negative compression of the S-LZW algorithm and cannot adapt to the dynamic changes of dataset. Therefore, the S-LZW algorithm has poor robustness and the compression ratio is the lowest among all algorithms. In addition, the S-LZW algorithm also faces an increasingly serious dictionary problem, but NIS does not need to consider this problem. ALDC usually needs additional space to store the encoding dictionary. Each encoding needs to access the examples in the dictionary to obtain the required code. Therefore, the processing speed of the ALDC algorithm is slow, and more energy will be consumed due to repeated access to the encoding dictionary. The LEC algorithm has stronger compression performance than the ALDC and FELCAS algorithms. However, the algorithm is a static compression algorithm and always uses a coding table, so its compression ratio cannot surpass the NIS algorithm. The compression ratio of the FEL-CAS algorithm for wind speed dataset is significantly higher than that of other existing algorithms. The main reason is that the Golomb rice coding used in the FELCAS algorithm can have a better compression effect on parameters with poor time correlation, but its compression performance is still inferior to the WCNIS algorithm. The comparison results are shown in Table 8.

Dataset Name	Compression Ratio (%)								
Dataset Maine	Gzip [24]	Bzip2 [25]	Rar	Huffman [26]	Count [27]	This Paper			
Line temperature	34.76	55.27	61.95	22.43	21.01	88.27			
Ambient humidity	31.35	45.87	52.31	22.34	22.87	76.69			
Rainfall	37.96	58.28	60.05	22.87	23.98	84.21			
Wind speed	27.58	43.22	45.26	18.52	18.21	68.43			
Leakage current	23.56	50.36	51.32	18.67	20.53	66.45			

Table 8. Comparison between this method and traditional compression algorithm.

In the text file to be transferred, the space character always repeats the most frequently, which significantly affects the overall size of the file. The algorithm proposed in this paper

cannot only remove the repeated space characters through wavelet transform, but also the space characters that must be transmitted. The NIS algorithm can compress and store only 1 bit. However, compared with the compression algorithm proposed in this paper, the compression performance of the compression algorithm is lower than that of the algorithm proposed in this paper because the compression efficiency of a large number of repeated white space characters is obviously insufficient.

The performance comparison between the WCNIS algorithm proposed in this paper and the traditional compression algorithm is shown in Table 6. The compression ratio of the Huffiman, Gzip and arithmetic coding algorithms in the four datasets is small, all less than 40%. Although the compression performance of bzip2 and Rra is higher than that of the Huffiman, Gzip and arithmetic coding algorithms, the maximum compression ratio still cannot exceed 70%. The algorithm proposed in this paper has obvious advantages in compression ratio because of its dynamic characteristics suitable for the input data.

Additionally, considering the complex situation this paper designs, when one or more sensors fail, the data transmitted by the sensors will be divided into three categories. (1) Jump data—there is a large difference between the data received in a short period of time and the last data transmitted or the data from neighboring nodes. (2) Disturbed data—a sharp increase in the rate of change of the data within a short period of time. (3) No data—where the sensor completely loses its data sampling capability or communication capability and is considered to have transmitted empty or missing data. The wavelet algorithm proposed in this paper can set the threshold range of the window function when filtering data, which can effectively filter jump data and disturbance data. If there are no data within the threshold range set by the window function, the algorithm will wait cyclically for new data and upload fault information. Since transmission line condition monitoring does not need to focus on the time series of data, this method can effectively solve the case of no-data faults.

4. Conclusions

This paper proposes an efficient lossless compression technology for wireless sensor networks. The algorithm combines the wavelet compression algorithm and the NIS algorithm. It displays a fast operation speed and requires a small amount of calculation. It is suitable for battery powered wireless sensor network nodes. Since each packet is decompressed independently, the algorithm is robust to packet loss. In addition, this method has high compression performance, which helps to reduce the network load and reduce the packet loss rate. Simulation results show that the proposed method has achieved effective results on the transmission line parameter dataset, is superior to the existing data compression algorithms, and is suitable for data compression and transmission in transmission line wireless sensor networks.

Author Contributions: Data curation, Z.C. and T.S.; formal analysis, G.L., L.J. and T.H.; funding acquisition, F.D.; methodology, G.L.; project administration, F.D.; software, L.J. and T.H.; supervision, F.D.; validation, Z.C., T.S. and Y.F.; writing—original draft, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Project of China Southern Power Grid Co. Ltd. (GD-KJXM20201990), Natural Science Foundation of China (52167008, 51867010), Natural Science Foundation of Jiangxi Province (20202ACBL214021), and Key Research and Development Plan of Jiangxi Province (20202BBGL73098).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the results of this study has not been provided because the author's college does not allow data to be published.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Ke, L.I.; Yongcun, F.A.; Dongmei, D.E.; Lei, W.A.; Jinfu, L.I.; Jian, Z.H. An Energy Efficiency Compressed Sensing Algorithm in Wireless Sensor Network of Microgrid. *J. Southwest Univ. Sci. Technol. (Natl. Sci. Ed.)* **2018**, *33*, 88.
- 2. Schoellhammer, T.; Greenstein, B.; Osterweil, E.; Wimbrow, M.; Estrin, D. *Lightweight Temporal Compression of Microclimate Datasets*; UCLA: Los Angeles, CA, USA, 2004.
- Das, P.; Chakraborty, B.; Sarkar, G.; Sen, S.; Mukherjee, A.; De, T. Implementing Dial-On-Demand Technique for Inter and Intra Cluster Communication in Energy Conserving Postbox Delay Tolerant Networks. In Proceedings of the International Conference on Computational Intelligence, Communications, and Business Analytics, Kalyani, India, 27–28 July 2018; Springer: Singapore, 2018; pp. 243–256.
- Ahmad, M.S.; Lata, S.; Mehfuz, S.; Ahmad, A. Lossless Compression Algorithm for Energy Efficient Wireless Sensor Network. In Proceedings of the 2019 International Conference on Power Electronics, Control and Automation (ICPECA), New Delhi, India, 16–17 November 2019; pp. 1–4.
- 5. Liang, Y.; Li, Y. An Efficient and Robust Data Compression Algorithm in Wireless Sensor Networks. *IEEE Commun. Lett.* **2014**, *18*, 439–442. [CrossRef]
- Kolo, J.G.; Shanmugam, S.A.; Lim, D.W.; Ang, L.M.; Seng, K.P. An Adaptive Lossless Data Compression Scheme for Wireless Sensor Networks. J. Sens. 2012, 2012, 539638. [CrossRef]
- Kiely, A.; Xu, M.; Song, W.Z.; Huang, R.; Shirazi, B. Adaptive Linear Filtering Compression on realtime sensor networks. *Comput. J.* 2010, 53, 1606–1620. [CrossRef]
- 8. Liang, Y.; Peng, W. Minimizing energy consumptions in wireless sensor networks via two-modal transmission. *ACM SIGCOMM Comput. Commun. Rev.* 2010, 40, 12. [CrossRef]
- 9. Kolo, J.G.; Shanmugam, S.A.; Lim, D.W.; Ang, L.M. Fast and efficient lossless adaptive compression scheme for wireless sensor networks. *Comput. Electr. Eng.* 2015, 41, 275–287. [CrossRef]
- Kasirajan, P.; Larsen, C.; Jagannathan, S. A New Adaptive Compression Scheme for Data Aggregation in Wireless Sensor Networks. In Proceedings of the 2010 IEEE Wireless Communication and Networking Conference, Sydney, Australia, 18–21 April 2010; pp. 1–6.
- 11. Marcelloni, F.; Vecchio, M. A Simple Algorithm for Data Compression in Wireless Sensor Networks. *IEEE Commun. Lett.* **2008**, *12*, 411–413. [CrossRef]
- 12. Zhu, Y.D. Optimization of Compression Transmission Method for Multimedia Network Image Data. *Recent Adv. Electr. Electron. Eng.* **2016**, *9*, 23–28.
- 13. Xiao, L. An Algorithm of Wavelet Data Compression Based on Wireless Sensor Network. J. Digit. Inf. Manag. 2013, 11, 54–61.
- 14. Kösters, M.; Leufken, J.; Schulze, S.; Sugimoto, K.; Klein, J.; Zahedi, R.; Hippler, M.; Leidel, S.; Fufezan, C. pymzML v2. 0: Introducing a highly compressed and seekable gzip format. *Bioinformatics* **2018**, *34*, 2513–2514. [CrossRef]
- 15. Li, F.; Zhu, H.; Wu, H. Farmland multi-parameter wireless sensor network data compression strategy. *Int. J. Ad Hoc Ubiquitous Comput.* **2018**, *29*, 221. [CrossRef]
- 16. Jayapandiyan, J.R.; Kavitha, C.; Sakthivel, K. Optimal Secret Text Compression Technique for Steganographic Encoding by Dynamic Ranking Algorithm. J. Phys. Conf. Ser. 2020, 1427, 012005. [CrossRef]
- 17. Moffat, A. Huffman coding. ACM Comput. Surv. (CSUR) 2019, 52, 1–35. [CrossRef]
- 18. Hao, W.; Xiang, L.; Li, Y.; Yang, P.; Shen, X. Reversible natural language watermarking using synonym substitution and arithmetic coding. *Comput. Mater. Contin.* 2018, 55, 541–559.
- 19. Gungor, V.C.; Lu, B.; Hancke, G.P. Opportunities and Challenges of Wireless Sensor Networks in Smart Grid. *IEEE Trans. Ind. Electron.* **2010**, *57*, 3557–3564. [CrossRef]
- 20. Sandoval, R.M.; Garcia-Sanchez, A.J.; Molina-Garcia-Pardo, J.M.; Garcia-Sanchez, F.; Garcia-Haro, J. Radio-Channel Characterization of Smart Grid Substations in the 2.4GHz ISM Band. *IEEE Trans. Wirel. Commun.* 2017, *16*, 1294–1307. [CrossRef]
- 21. Sandoval, R.M.; Garcia-Sanchez, A.J.; Garcia-Haro, J. Improving RSSI-based path-loss models accuracy for critical infrastructures: A Smart Grid substation case-study. *IEEE Trans. Ind. Inform.* **2018**, *14*, 2230–2240. [CrossRef]
- 22. Wang, K.; Xu, C.; Zhang, Y.; Guo, S.; Zomaya, A.Y. Robust Big Data Analytics for Electricity Price Forecasting in the Smart Grid. *IEEE Trans. Big Data* **2019**, *5*, 34–45. [CrossRef]
- 23. Uthayakumar, J.; Vengattaraman, T.; Dhavachelvan, P. A new lossless neighborhood indexing sequence (NIS) algorithm for data compression in wireless sensor networks. *Ad. Hoc. Netw.* **2019**, *83*, 149–157.
- 24. Lu, J.Z.; Liu, Y.; Wu, C.; Zhang, H.X.; Zhou, T.J. Research on satellite monitoring and alarm algorithm of transmission line mountain fire. *Chin. J. Electr. Eng.* 2015, *5*, 5512–5519.
- 25. Lu, J.; Liu, Y.; Wu, C.; Zhang, H.; Zhou, T. Study on Satellite Monitoring and Alarm Calculation Algorithm of Wild Fire Near Transmission Lines. *Proc. CSEE* 2015, *5*, 5512–5519.
- Kong, P.-Y.; Liu, C.-W.; Jiang, J.-A. Cost-Efficient Placement of Communication Connections for Transmission Line Monitoring. IEEE Trans. Ind. Electron. 2017, 64, 4058–4067. [CrossRef]
- 27. Fateh, B.; Govindarasu, M.; Ajjarapu, V. Wireless Network Design for Transmission Line Monitoring in Smart Grid. *IEEE Trans.* Smart Grid 2013, 4, 1076–1086. [CrossRef]