Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8840367, 11 pages
https://doi.org/10.1155/2020/8840367

Research Article

Edge Computing-Based ERBS Time Synchronization Algorithm in WSNs

Xianbo Sun,[1,2] Yixin Su,[2] Yong Huang,[1] Jianjun Tan,[1] Jinqiao Yi,[1] Tao Hu,[1] and Li Zhu[1]

[1]*School of Information Engineering, Hubei Minzu University, Enshi 445000, China*
[2]*School of Automation, Wuhan University of Technology, Wuhan 430070, China*

Correspondence should be addressed to Yixin Su; suyixin@whut.edu.cn

Received 21 August 2020; Revised 18 October 2020; Accepted 4 November 2020; Published 21 November 2020

Academic Editor: Shaohua Wan

Copyright © 2020 Xianbo Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the practical application of large-scale photovoltaic module monitoring, adopting wireless sensor network (WSN) technology is a method worth researching. With increasing nodes in the wireless sensor network, widely existing clock skew, increased geometrically, is bringing about greater energy consumption. Due to the random distribution of nodes, in order to improve the transmission efficiency and reduce the computational load of the coordinator, the node processor needs to the use edge computing for preliminary analysis. This paper puts forward an improved energy-efficient reference broadcast synchronization algorithm (ERBS). This algorithm firstly calculates the average phase offset of nonadjacent nodes in the network after receiving a message. It then uses the least square method to solve the clock skew to achieve high-precision synchronization of the whole network. Simulation results show that compared with RBS, the time synchronization precision of ERBS is greatly improved and synchronization times are greatly reduced, decreasing energy consumption significantly.

1. Introduction

With the development of microelectromechanical system (MEMS), wireless communications, Internet of Things, big data, and AI transforming distributed sensing, edge computing, and communication into wireless sensor nodes at a low cost and low power consumption is becoming the focus of research. These sensor nodes transmit data packets and form the network via multihop communication collaboration, which has been widely used in home appliance automation, military surveillance, environment, and human health monitoring [1–3]. However, in most practical applications, wireless sensor networks have some limitations such as limited energy, unreliability, and large-scale sensor nodes. The study on time synchronization protocol with effective energy and time synchronization precision is one of the key points in the current research [4, 5]. Clock synchronization is a basic service that provides the concept of common time in any distributed system. In particular, in different application environments, wireless sensor networks need clock synchronization of different precisions because it needs to provide a common time base for different nodes to handle the distributed tasks. The precision of time synchronization is necessary for various tasks including data fusion, locating and tracking power management, and effective media access control [6]. As sensor network nodes are used operationally at once and will be in operation for a long time after deployment, it is common to periodically set sensor network nodes to sleep mode to save battery power. Maintaining a relative time base is very important for waking up sensor nodes so that the success rate of data exchange can be improved. Therefore, a simple and efficient clock synchronization mechanism is likely to be used, where all timestamps use the same time base instead of multiple local clocks. For such a synchronization protocol, the two most important parameters that conflict with each other are high synchronization precision and low power consumption. In many applications, minimum synchronization errors must be maintained, needing resynchronization start-ups many times, increasing power consumption. Therefore, it is necessary to balance synchronization precision and energy consumption in the application.

Edge computing refers to an open platform integrating network, computing storage and application core capabilities at the edge of the network near the object or data source to provide edge intelligent services nearby, so as to meet the key requirements of industry digitization in terms of agile connection, real-time business, data optimization, application intelligence, security, and privacy protection. In terms of privacy protection, based on bilinear pairing and Paillier homomorphic encryption, Zhao et al. designed a data aggregation scheme for edge computing. The scheme can not only realize the batch processing of data but also protect the integrity and source authentication of vehicle organization network data. It can improve the network data processing efficiency and reduce communication costs while protecting users [7]. Based on the consideration of the safety of automatic driving technology, Xiong et al. proposed an IDP architecture for VANET, which has a high detection accuracy and high efficiency of data processing, and can improve the safety of automatic driving. The effectiveness of its architecture in preventing VANET intrusion in a complex environment is verified by a case study [8]. Considering the security of point-to-point transaction, Nawaz et al. built a concept verification intelligent contract platform by using edge computing technology. Compared with other platforms, the intelligent contract platform does not need intermediary in data transaction and can reduce the space occupied by it as much as possible under the premise of ensuring the ownership of data and user privacy [9]. In order to improve the computational efficiency, Wu et al. proposed an online optimization algorithm based on device data analysis, maximizing fairness and throughput balance and using Lyapunov and convex optimization to improve the effectiveness of resource allocation through numerical simulation, aiming at the randomness of wireless mobile edge traffic arrival, the time coupling of uplink and downlink decision-making, and the incompleteness of system state knowledge [10]. In view of the delay problem of terminal equipment processing dense data, on the basis of multiaccess edge calculation method, Li et al. through the establishment of mixed integer nonlinear programming model, and then using genetic algorithm to optimize, the stable convergence solution was obtained, which improved the data calculation efficiency of terminal equipment and effectively reduced the energy consumption of equipment [11]. Zakarya et al. put forward the resource management technology of game theory to improve the efficiency of data processing and effectively reduce the cost of equipment and energy consumption [12]. Zeng et al. proposed a fast search algorithm for the optimal pricing strategy of vehicle edge computing server based on the genetic algorithm by analyzing the interaction between the vehicle and edge computing server. Through simulation and comparison with other schemes, the efficiency of the algorithm was verified and the calculation cost was reduced [13, 14]. In real-time business, Zhai and others proposed a service framework based on environment and resource constraints and a dynamic edge service migration algorithm. Through modeling and simulation, the temperature of the service framework and the effectiveness of the algorithm in reducing traffic were verified. However, its algorithm also has certain limitations. The data migration process has a certain delay, and the migration process will consume corresponding resources, which will limit the efficiency of data migration to a certain extent [15]. Arunan et al. introduced a self-adaptive feature extraction fault detection and fault identification protection scheme based on edge calculation feature extraction for monitoring fault current of a microgrid. Through simulation and comparison, the effectiveness and strong noise resistance of this method in power grid protection are verified [16, 17]. On the basis of pervasive edge computing, Pei and others proposed a nonorthogonal multipervasive edge computing power allocation framework and a total power optimization algorithm for the internet of vehicles. This framework can minimize the system delay, and the effectiveness of the optimization algorithm is verified by simulation analysis [18]. Quan et al. proposed a multiagent deep reinforcement learning algorithm based on the delay problem of vehicle edge computing. By maximizing the distributed communication, computing, and path planning, the scalability of the algorithm was verified by experiments, which reduced the service delay and data migration cost [19, 20].

With the expansion of WSN, the number of network nodes has increased dramatically. A feature of traditional reference broadcast synchronization (RBS) algorithms is that error sources are concentrated mainly in the processing time of receiving nodes and higher synchronization precision [21–23], but with the increasing network overhead, it will influence the energy consumption of the whole system [24, 25].

This paper will complete the collection and transmission of parameters using WSN in the process of photovoltaic module monitoring. Photovoltaic module monitoring makes more demands on nodes as the source node is applied as the receiver in the photovoltaic module. Therefore, an improved ERBS algorithm is proposed based on the analysis of the RBS algorithm. The ERBS algorithm firstly calculates the average phase offset of nonadjacent nodes in the subnet of photovoltaic module monitoring after receiving a message. Then, it will solve the clock skew by the least square method. In the process of solving clock skew, the influence of environmental temperature will be considered in photovoltaic module monitoring and the synchronization precision of the whole network can be achieved, providing the basis for photovoltaic module monitoring and fault diagnosis. Different applications have different requirements for various supporting technologies of WSN. This paper studies the WSN time synchronization method for photovoltaic module monitoring field. The proposed ERBS algorithm can effectively reduce network energy consumption and enhance the reliability of network time synchronization on the basis of ensuring synchronization accuracy. It can be applied in large-scale WSN construction.

2. Problem Description

RBS is a typical synchronization method based on a receiver-receiver mechanism. It can synchronize a set of child nodes

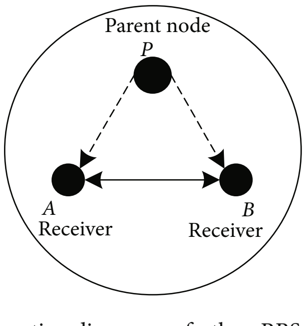

Figure 1: Schematic diagram of the RBS synchronization mechanism.

that receive a beacon message from a common sender (the reference node is the parent node). Figure 1 is a schematic diagram of the RBS synchronization mechanism. A parent node P and other nodes, A and B, within the communication range of the parent node will complete a group of synchronization.

As shown in Figure 2, it is assumed that the parent node (P) periodically sends reference broadcast beacons. Nodes A and B receive the i^{th} beacon sent by the parent node (P) at local time $T_{A(2,i)}$ and $T_{B(2,i)}$, respectively. Nodes A and B record the arrival time of the broadcast group and exchange timestamps with each other.

Assuming that $X_i^{(PA)}$ represents an uncertain delaying part (random delay), $d^{(PA)}$ represents the certain delaying part from node P to node A (propagation delay), then $T_{A(2,i)}$ can be recorded as

$$T_{A(2,i)} = T_{1,i} + d^{(PA)} + X_i^{(PA)} + \varphi^{(PA)} + \omega^{(PA)}(T_{1,i} - T_{1,1}), \quad (1)$$

where $T_{1,i}$ is the sending time of the parent node and $\varphi^{(PA)}$ and $\omega^{(PA)}$ are the clock phase offset and frequency offset of node A compared to parent node P. In the same method, the arrival time of node B is

$$T_{B(2,i)} = T_{1,i} + d^{(PB)} + X_i^{(PB)} + \varphi^{(PB)} + \omega^{(PB)}(T_{1,i} - T_{1,1}), \quad (2)$$

where $d^{(PB)}$, $X_i^{(PB)}$, $\varphi^{(PB)}$, and $\omega^{(PB)}$ represent fixed delay, random delay, clock phase offset, and frequency offset of node B to the reference node, respectively. From formulas (1) and (2), we can get

$$T_{A(2,i)} - T_{B(2,i)} = \varphi^{(BA)} + \omega^{(BA)}(T_{1,i} - T_{1,1}) + d^{(PA)} - d^{(PB)} + X_i^{(PA)} - X_i^{(PB)}, \quad (3)$$

Among them $\varphi^{(BA)} \triangleq \varphi^{(PB)} - \varphi^{(PA)}$ and $\omega^{(BA)} \triangleq \omega^{(PB)} - \omega^{(PA)}$ are phase offset and frequency offset of nodes A and B while receiving the i^{th} broadcast group from the parent node, $X_i^{(PA)}$ and $X_i^{(PB)}$ are random variables of a normal distribution with a mean value of μ and variance of $\sigma^2/2$. Elson et al. proved that after receiving the reference message, the phase offset of the local time difference between any two receiving nodes follows Gaussian distribution that $\mu = 0$, $\sigma = 11.1\,\mu$s [21].

The RBS algorithm uses the broadcast characteristics of the wireless channel to send multiple reference broadcast synchronization messages to the sender and the receiving nodes adjust their own time by calculating the local time difference of receiving synchronization messages and exchanging timestamps, so synchronization is achieved.

The main error source of RBS algorithms is the time delay of the receiver, and its biggest feature is to eliminate the sending time delay and access time delay of the reference broadcast message on the critical path from the sending node to the receiving node. Therefore, the time synchronization of the receiving node will be closer. While solving the synchronization problem of WSN, the accessing time of channels in the media access control (MAC) layer is the biggest source of error, so it is very important to further study the defects in the RBS algorithm and find an improved algorithm.

3. Design of ERBS Algorithm

Due to the characteristics of phase offset and frequency offset of the RBS algorithm [26, 27], the realization of the ERBS algorithm can be divided into two steps. The first step is to estimate the uncertain phase offset and the second is to estimate the frequency offset of nodes. In estimating frequency offset, the influence of temperature on synchronization precision needs to be considered and then the overall evaluation can be conducted on the performances of the improved algorithm.

Clock drift is accumulated in one period, and random delay is closely related to timestamps. Figure 3 shows the relationship between synchronization error, clock skew, and random delay, which applies to discussions about the ERBS algorithm. In Figure 3, measurement modeling refers to the establishment of the functional relationship of the measurement model according to clock drift and random delay. Calculation of the estimated value refers to estimating the relationship between synchronization error and clock drift by linear regression based on the measurement model of timestamps. Taylor approximation expansion refers to the linearization of nonlinear calculations based on the relationship among synchronization error, clock drift, and random delay, which can reduce computational complexity and improve the execution efficiency of the algorithm on the premise of ensuring synchronization precision and energy consumption [28, 29]. The absolute error refers to the phenomenon that the node may lead or lag the master node in the synchronization process, and the adoption of absolute error is to solve this problem. The analysis of the synchronization error source shown in Figure 3 provides effective support for the design of the ERBS algorithm.

In a WSN composed of photovoltaic module monitoring nodes, each monitoring subnet contains $n + 1$ nodes (generally set as $200 \leq n \leq 500$) when the parent node of the subnet sends a message (mainly including the photovoltaic module ID, voltage, current, temperature, and timestamp). Under the application background, commonly used RBS algorithms require each node in the monitoring subnet to exchange information with other n nodes and then calculate the time difference by exchanging information. With the increase of

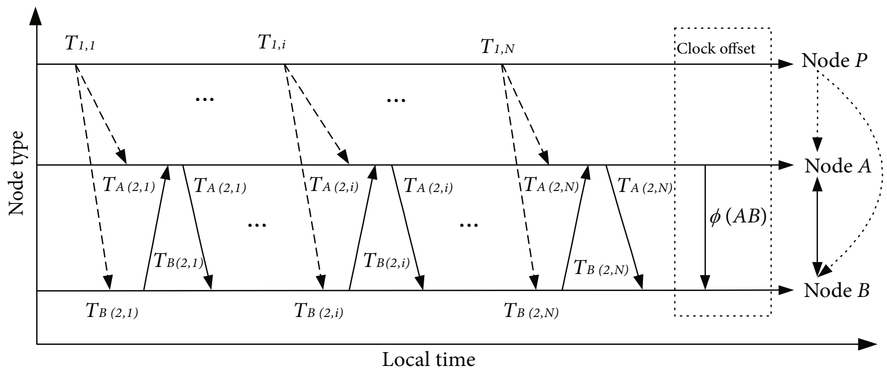

FIGURE 2: Schematic diagram of clock synchronization RBS model.

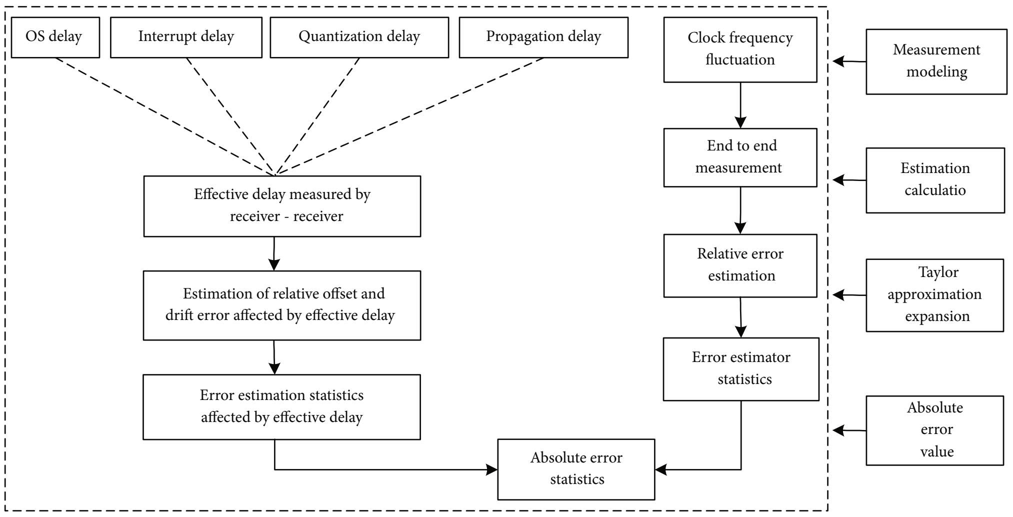
OS delay
Interrupt delay
Quantization delay
Propagation delay
Effective delay measured by receiver - receiver
Estimation of relative offset and drift error affected by effective delay
Error estimation statistics affected by effective delay
Absolute error statistics
Clock frequency fluctuation
End to end measurement
Relative error estimation
Error estimator statistics
Measurement modeling
Estimation calculatio
Taylor approximation expansion
Absolute error value

FIGURE 3: Analysis of synchronization errors.

n, to achieve time synchronization of the whole network, the amount of information exchanging between nodes has increased sharply, putting enormous pressure on the whole WSN communication overhead. Aiming to increase energy consumption caused by synchronization data interaction in the RBS algorithm, the ERBS algorithm was improved accordingly.

The specific implementation steps of the ERBS algorithm are as follows:

(i) Numbering n receiving nodes in the photovoltaic module monitoring subnet

(ii) The parent node sends an information packet containing the photovoltaic module ID, voltage, current, temperature, and timestamp

(iii) n receiving nodes receive packets, and the local reference time is determined according to the time of receiving the packets

(iv) The information packets containing timestamps are received by the node interactions

(v) Each receiving node calculates the offset value of the timestamps based on the received information packets

(vi) The receiving node first calculates the phase offset using the estimation method

(vii) Further calculations of the frequency offset are performed considering the working temperature of photovoltaic modules

3.1. Estimation of Phase Offset. Figure 4 shows the estimation of the phase offset of three receiving nodes. Compared to the traditional RBS algorithm with an increasing number of nodes and sent messages, the increase of algorithm execution time for the ERBS algorithm is not obvious, the growth of algorithm complexity is limited, and the increase of the network's energy consumption is limited. The main reason is that the ERBS algorithm gave up any information interaction between two nodes, and it did not focus on the information transmission of nonadjacent nodes, effectively saving the amount of data interaction and improving the energy efficiency of time synchronization [30, 31].

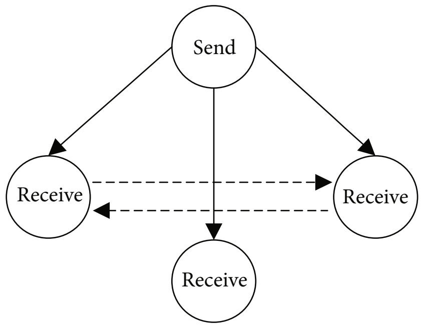

Figure 4: Diagram of the estimation of the phase offset of the ERBS algorithm.

While estimating the phase offset, the focus is to calculate two nonadjacent nodes. Assuming that as p and q, $p \in n$, $q \in n$, and $p - q \neq \pm 1$. because it is nonadjacent, $T_{r,k}$ represents the recorded local time, while node r receiving the information packet k, among that $r \in n$. m represents the amount of information, and φ represents the phase offset.

$$\varphi = \frac{1}{m} \sum_{k=1}^{m} \left(T_{p,k} - T_{q,k} \right). \tag{4}$$

Otherwise, according to literature [21], assuming that the phase offset follows Gaussian distribution with an average of 0 and variance of σ_n if the condition is φ then the measurement value of phase offset is $x = [x_1, x_2, \cdots x_N]^T$, and the conditional probability density function is

$$p(x \mid \varphi) = \left(\frac{1}{\sqrt{2\pi}\sigma_n} \right)^N \exp \left[-\sum_{i=1}^{N} \frac{(x - \varphi)}{2\sigma_n^2} \right]. \tag{5}$$

Then, the probability density function φ is

$$p(\varphi) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[-\frac{(\varphi - \mu)^2}{2\sigma^2} \right]. \tag{6}$$

We assume that the condition is

$$C = \int_{-\infty}^{\varphi - r} p(\varphi \mid x) d\varphi + \int_{\varphi + r}^{\infty} p(\varphi \mid x) d\varphi = 1 - \int_{\varphi - r}^{\varphi + r} p(\varphi \mid x) d\varphi. \tag{7}$$

When C is the minimum value, the right side integral is the maximum. If $\widehat{\varphi}$ is the estimated value of the phase offset, then the necessary condition for the maximum value is

$$\left. \frac{\partial \ln p(\varphi \mid x)}{\partial \varphi} \right|_{\varphi = \widehat{\varphi}} = 0. \tag{8}$$

Formula (8) is the biggest posterior equation of φ:

$$p(\varphi \mid x) = \frac{p(x \mid \varphi) p(\varphi)}{p(x)}, \tag{9}$$

$$p(x) = \int_{-\infty}^{+\infty} p(x, \varphi) d\varphi = \int_{-\infty}^{+\infty} p(x \mid \varphi) p(\varphi) d\varphi. \tag{10}$$

By getting the logarithm towards formulas (9) and (10) and partial derivative of φ, the solution definition of $\widehat{\varphi}$ is as follows:

$$\left[\frac{\partial \ln p(x \mid \varphi)}{\partial \varphi} + \frac{\partial \ln p(\varphi)}{\partial \varphi} \right]_{\varphi = \widehat{\varphi}} = 0. \tag{11}$$

Because φ conforms to the Gaussian distribution, the equation above can be simplified as

$$\begin{aligned} &\partial \ln \frac{\left\{ \left(1/\sqrt{2\pi}\sigma_n \right)^N \exp \left[-\sum_{i=1}^{N} \left((x_i - \varphi)^2 / 2\sigma_n^2 \right) \right] \right\}}{\partial \varphi} \\ &\quad + \partial \ln \frac{\left\{ \left(1/\sqrt{2\pi}\sigma \right) \exp \left[-(\varphi - \mu)^2 / 2\sigma^2 \right] \right\}}{\partial \varphi} = 0. \end{aligned} \tag{12}$$

After further simplifying, we can get

$$\sum_{i=1}^{N} \frac{x_i - \varphi}{\sigma_n^2} + \frac{\mu - \varphi}{\sigma^2} = 0. \tag{13}$$

Setting $\varphi = \widehat{\varphi}$, then

$$\widehat{\varphi} = \sum_{i=1}^{N} \frac{x_i}{\sigma_n^2} \times \frac{\sigma_n^2 \sigma^2}{N\sigma^2 + \sigma_n^2}. \tag{14}$$

Formula (14) is the estimated value of phase offset.

3.2. Estimation of Clock Frequency Offset. The clock drift of WSN nodes is caused by changes in the frequency of the crystal oscillator. The counter will reduce by one for each oscillation. When it is reduced to 0, one interrupt process is generated, which can realize the sending of a packet. During this process, if the crystal oscillator has a larger clock skew, the timestamp carried by the sending packet will have a large error, resulting in the network time synchronization generating an offset value, which is more than a set. After realizing one interrupt process, the counter will reload the starting value from the hold register. It is assumed ρ is the maximum drift speed; if two clocks drift in opposite directions relative to UTC, the possible difference value is $2\rho\Delta t$ during the time Δt after synchronization. For a WSN operating system, assuming the difference between every two clocks does not exceed δ, then it must resynchronize at least within $\delta/2\rho$. In this algorithm, the local time of node i at the physical time

t is defined as

$$T_i(t) = \frac{1}{f_0}\int_{t_0}^{t} f_i(t)dt + \sigma_i, \quad (15)$$

f_0 is nominal frequency, $f_i(t)$ is the real frequency of the crystal oscillator, in usual cases that $f_i(t) \neq 1$; σ_i is the time offset of node i accumulated before moment t; and t and t_0 are real physical times. In the short term, the frequency of the crystal oscillator is fixed, assuming that $f_i(t) = v$, we can get a more simplified time model.

$$y = vx + \sigma, \quad (16)$$

where v represents the frequency drift of nodes and σ represents the initial phase offset of nodes. Based on formula (16), a linear model is adopted to realize parametric fitting through the least square method.

$$v = \bar{y} - \sigma\bar{x}, \quad (17)$$

$$\sigma = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sum_{i=1}^{n}x_i y_i - n\bar{x}\cdot\bar{y}}{\sum_{i=1}^{n}x_i^2 - n\bar{x}^2}. \quad (18)$$

By fitting parameter r like formula (19), the compensation of clock frequency drift can be completed.

$$r = \frac{\sigma_{xy}^2}{\sigma_x\sigma_y} = \frac{\sum_{k=1}^{n}(x_k - \bar{x})(y_k - \bar{y})}{\sqrt{\sum_{k=1}^{n}(x_k - \bar{x})^2}\cdot\sqrt{\sum_{k=1}^{n}(y_k - \bar{y})^2}}. \quad (19)$$

The above discussion is one that assumes that the output frequency of the crystal oscillator has a stabilized value. The reality is that, no matter what the model the oscillator is, it has its clock parameters. For example, the clock frequency of a quartz crystal oscillator changes by 40 ppm, which means that in every second the derivation of the clock will reach 40 ms due to different nodes, that is to say, each oscillator has different migration parameters of -20~20 ppm. This is similar to the core processor adopted in the photovoltaic modules monitoring system; it externally uses a 32 kHz crystal oscillator that deviates between 5 and approximately 30 ppm.

Additionally, the temperature and humidity of the environment, vibration, and the working voltage of the node influence the crystal oscillator, especially in the working environment of photovoltaic modules, where the performance of temperature parameters are very obvious. Figure 5 shows changes in temperature over time (the curve monitoring in situ temperature in a photovoltaic system from 8.00 a.m. to 6.00 p.m.). The sensor network nodes integrated with the components are very sensitive to the temperature parameter and in time synchronization; it is necessary to consider the influence of temperature on the precision of synchronization.

Based on the above discussion, the relational expression between clock skew and the temperature is shown as

$$S_i[n] = S_{i_{t0}} + k_i(t_i[n] - t_0) + \omega_i[n], \quad (20)$$

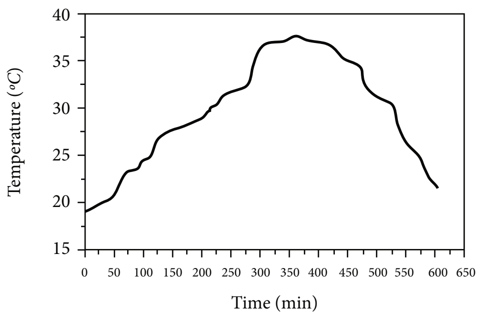

FIGURE 5: Curve showing changes in temperature over time.

where $s_{i_{t0}}$ is the clock skew of the reference temperature of node i at moment t_0, k_i is the shift coefficient of the node, $t_i[n]$ is the temperature of node i in the n^{th} sampling interval, $\omega_i[n]$ is the random shift noise brought by other environmental elements and quantization error, and it follows the $N(0, \sigma_\omega^2)$ normal distribution [32, 33]. In this paper, every sensor node has a temperature sensor that can measure the real-time environmental temperature. According to the parameter manual of the crystal oscillator, you can check the influence of reference temperature on the clock skew. However, due to different photovoltaic modules, different nodes have a different clock skew and drift coefficient that is related to the air humidity, the node's supply voltage, and the life of the crystal oscillator. In different application backgrounds, these factors need to adopt different calibration methods.

Considering the influence of temperature on clock frequency offset, to obtain complete-time synchronization, it is necessary to estimate the clock frequency offset based on the estimation of the clock phase offset. Based on formula (3) in the assumed model of clock frequency offset, $T_{B(2,i)}$ is known and the set of the observed quantity of delay between node A and node B can be expressed as follows:

$$U_k' = T_{A(2,i)} - \widehat{\omega}T_{A(1,i)} = d' + \varphi + X_k', \quad (21)$$

$$V_k' = \widehat{\omega}T_{B(4,i)} - T_{B(3,i)} = d' - \varphi + Y_k', \quad (22)$$

Among them,$X_k' = \omega X_k$, $Y_k' = \omega Y_k$, and $d' = \omega d$, after further observation, the Gaussian delay model is used to conduct a joint estimation of phase offset and frequency offset towards the information model described in equations (21) and (22), and the following expressions can be obtained:

$$\widehat{\omega} = \frac{\left(\left(T_{B(2,i)} + T_{B(3,i)}\right)/2\right) - \left(\left(T_{B(2,j)} + T_{B(3,j)}\right)/2\right)}{\left(\left(T_{A(1,i)} + T_{A(4,i)}\right)/2\right) - \left(\left(T_{A(1,j)} + T_{A(4,j)}\right)/2\right)}, \quad (23)$$

$$\widehat{\varphi} = \left(\left(T_{B(2,i)} + T_{B(3,i)}\right)/2\right) - \left(\widehat{\omega}\left(T_{A(1,i)} + T_{A(4,i)}\right)/2\right). \quad (24)$$

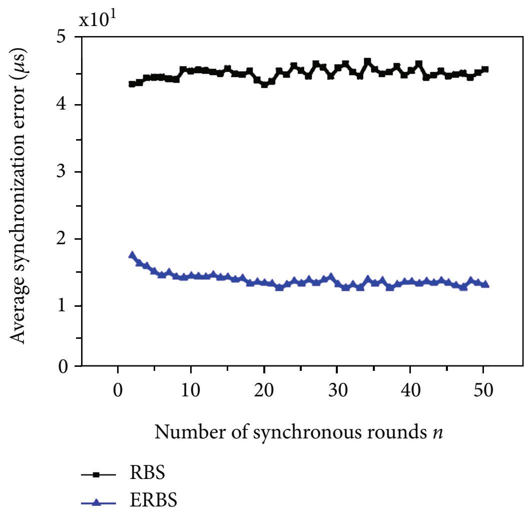

Figure 6: Curves of average synchronization errors (opening timestamps on the MAC layer).

Table 1: Comparison of single-hop synchronization precision.

Synchronization algorithm	Average error (μs)	Maximum error (μs)	Minimum error (μs)	Standard deviation (μs)
RBS	41.09	131.45	0.009	29.89
ERBS	13.92	52.89	0.007	11.09

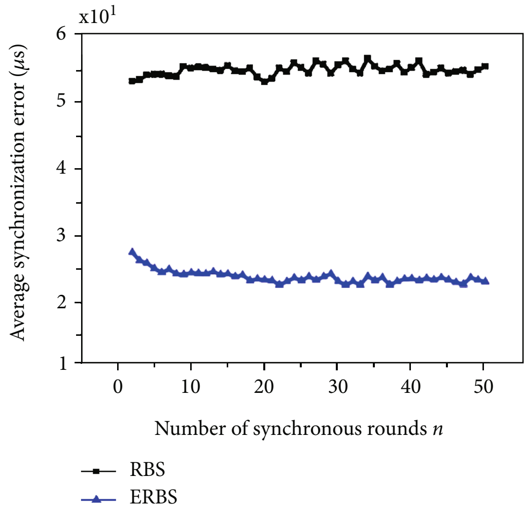

Figure 7: Curves of average synchronization errors (closing the timestamps at the MAC layer).

4. Analysis of Simulation Results

4.1. Simulation Environment and Settings of Parameters. To verify the effectiveness of the improved ERBS algorithm, the mathematical model of node clocks has been established using the software environment MATLAB, whose parameter of the crystal oscillator is -30~30 ppm. The interrupting counting value of nodes is 921600 per second and there are three nodes, the parent node P, node A, and node B. The round number of synchronizations changes from 2 to approximately 50 and the simulations are synchronized 1000 times. After 1000 repeated experiments, the average synchronization error value can be obtained. Meanwhile, a synchronous experimental test platform with one reference node and two receiving nodes is set up on-site. The experimental time is from 8.00 a.m. to 6.00 p.m. The curve of temperature changes during the experiment is shown in Figure 5. The reference node broadcasts a beacon periodically every second. The receiving node will timestamp the received beacon, transmit the result to the coordinator, and then connect with the PC via RS232 for real-time processing of online data. After the broadcasting node sends the data, the receiving node returns one beacon immediately. The coordinator will collect all the beacons with timestamps and analyze the intervals of beacons, the numbers of every synchronization

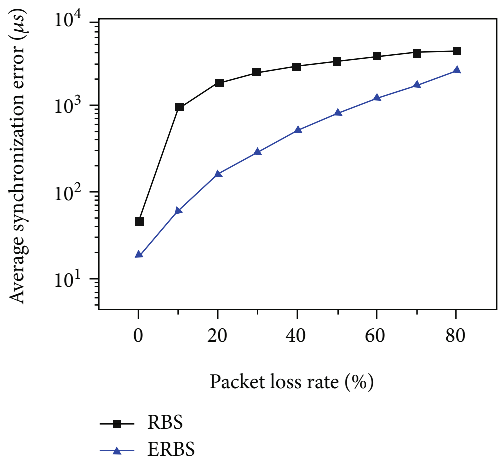

Figure 8: Time synchronization errors in cases of different packet loss rate.

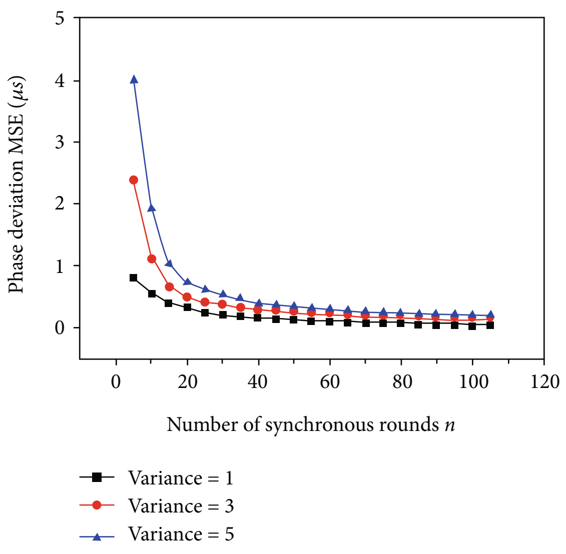

Figure 9: MSE curves of clock phase offset.

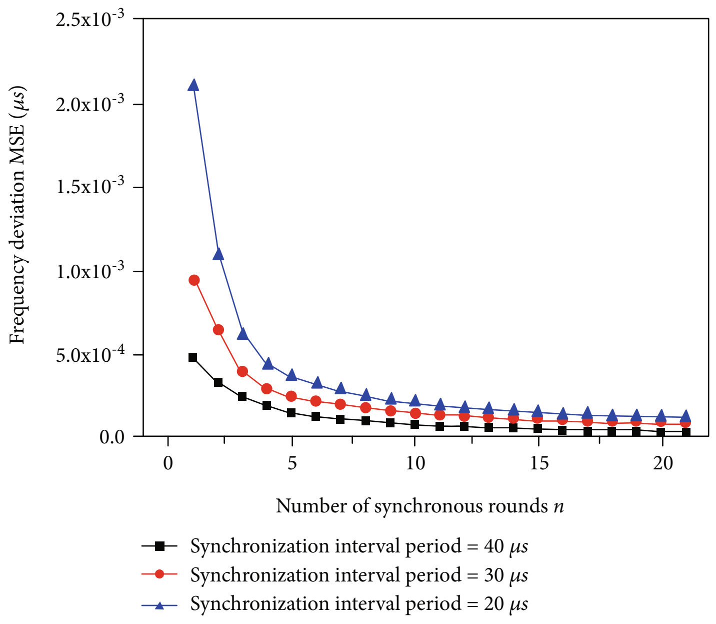

Figure 10: MSE curves of clock frequency offset (round of synchronization n).

beacon, and measurement numbers. To eliminate the influence of different datasets on the synchronization algorithm, the measurement of synchronization errors will adopt the same standard of beacon including the ID of photovoltaic modules, voltage, current, temperature, timestamps, and other data. To open the timestamps at the MAC layer, CC530 is used to offer initial frame data and chip timers are used to produce local timestamps.

4.2. Analysis of Time Synchronization Precision. While setting rounds of synchronization n that change from 2 to approximately 50, the average synchronization errors of the RBS and ERBS algorithms are shown in Figure 6. The average synchronization errors of the RBS and ERBS algorithms are 40~50 μs and 10~20 μs, respectively. The analysis found that the average synchronization errors of the RBS and ERBS algorithms are not affected much by n. The main reason is that these two algorithms only use data with a single-wheel timestamp to calculate the time deviation value. In general, the ERBS algorithm shows obvious improvement in synchronization precision.

Table 1 shows the single-hop synchronization precision of the RBS and ERBS algorithms when the round of synchronization is $n = 20$. The ERBS algorithm has the lowest error average and best stability of synchronization precision mainly because it calculates the average phase offset towards nonadjacent nodes, which makes full use of statistic features of time synchronization data, reducing the complexity of the operation process.

To further analyze the performances of the algorithm, if there is no timestamp on the MAC layer, Figure 7 shows the average synchronization error. Due to the large access delay and the software operating delay of the sending node, compared with Figure 6, the accuracy of the algorithm decreases to varying degrees, with an increase of about 10 μs. However, compared with the RBS algorithm, the increase of the ERBS algorithm's average errors is the smallest. The simulation results show that random delay and operating system delay can be reduced by adding timestamps at the MAC layer, and this method influences clock skew so that the compensation of clock skew can be realized. Of course, in the network of too large scale, this algorithm may have the limitation of serious precision decline, mainly because it saves too much data exchange between adjacent nodes, and the statistical characteristics of time synchronization data and the method of adding timestamp in MAC layer are not enough to solve this problem, so it is not necessarily applicable in large-scale WSN network.

In the process of wireless transmission, the packet loss rate is always an important factor affecting synchronization precision. The influence of bit error rates in different groups on synchronization precision is simulated by the experimental addition of timestamps at MAC and analyzing the change regulations between synchronization precision and packet loss rate. Figure 8 shows the synchronization precision in cases of different packet loss rates. Observation found that with the increase of packet loss rate, the average error of the RBS algorithm increased significantly because it depends on the surrounding nodes. When the packet loss rate of data reaches 10%, the average synchronization error increases more than 20 times. The results can be a good reference for the design of a photovoltaic module monitoring system.

4.3. Analysis of the Minimum Mean Square Error of Clock Skew. The ultimate goal of parameter estimation is to look for the reachable minimum mean square error to provide support for the design of an estimator. However, the best mean squared error (MSE) design in theories is very difficult. Figure 9 shows the curves between phase offset MSE of the RBS and ERBS algorithms and rounds of synchronization and variance.

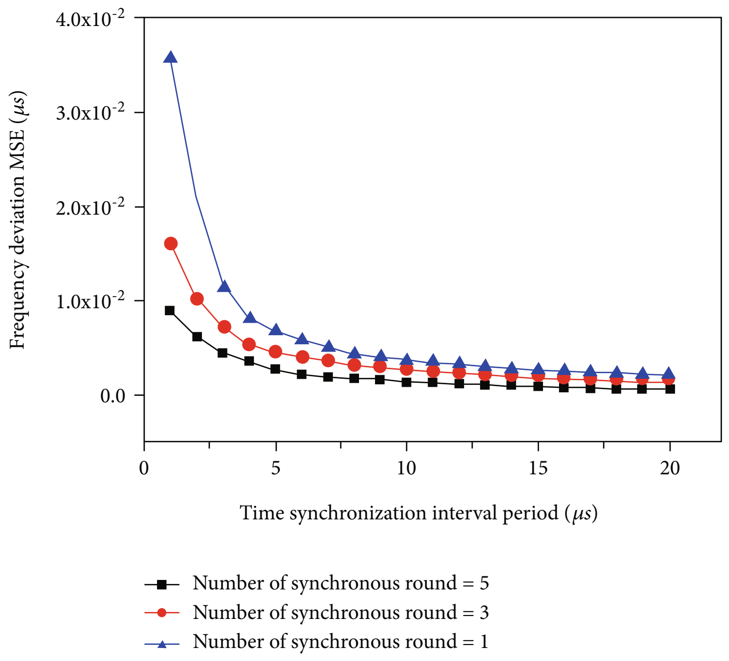

Figure 11: MSE curves of clock frequency offset (synchronous interval cycle τ).

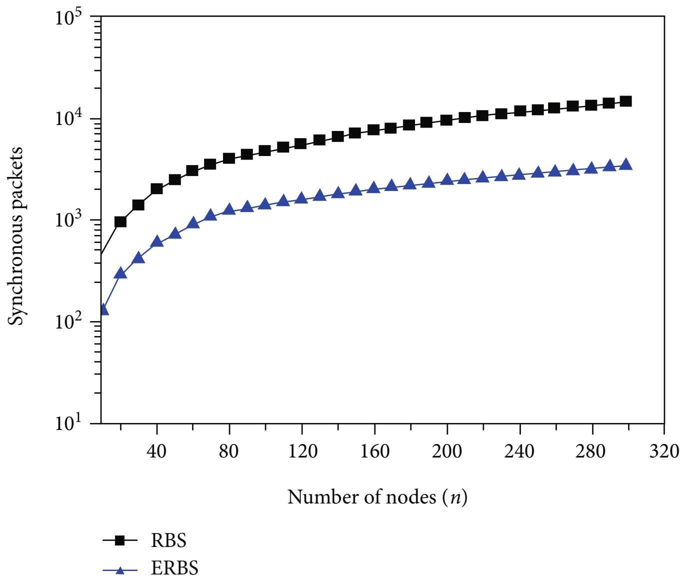

Figure 12: The relationship curve between synchronization data packets and the number of nodes.

As can be seen from Figure 9, when n increases and the variance decreases, the MSE of the clock phase offset gradually decreases. This is because when n increases, a more local data timestamp of nodes participates in the estimation of phase offset, thus reducing system synchronization error.

Figure 10 is the relationship between clock frequency offset MSE and rounds of synchronization and synchronous interval cycles whose variance is one. As can be seen from the figure, with the increase of round synchronization n and synchronous interval cycles, the MSE of the clock frequency gradually decreases. However, the increase of synchronization rounds will lead to greater message overhead, so the increase in the synchronous interval cycle can also reduce synchronization efficiency. Therefore, it is necessary to make overall considerations and find a balance point between the MSE of clock frequency, rounds of synchronization, and synchronous interval cycles. As mentioned above, in Figure 11, when the MSE of clock frequency decreases, it is still necessary to increase the rounds of synchronization n and the synchronous interval cycles, which reflect the importance of considering these factors from other aspects.

4.4. Analysis of Comparison of Energy Consumption. For the energy efficiency of large-scale networks, the ERBS algorithm

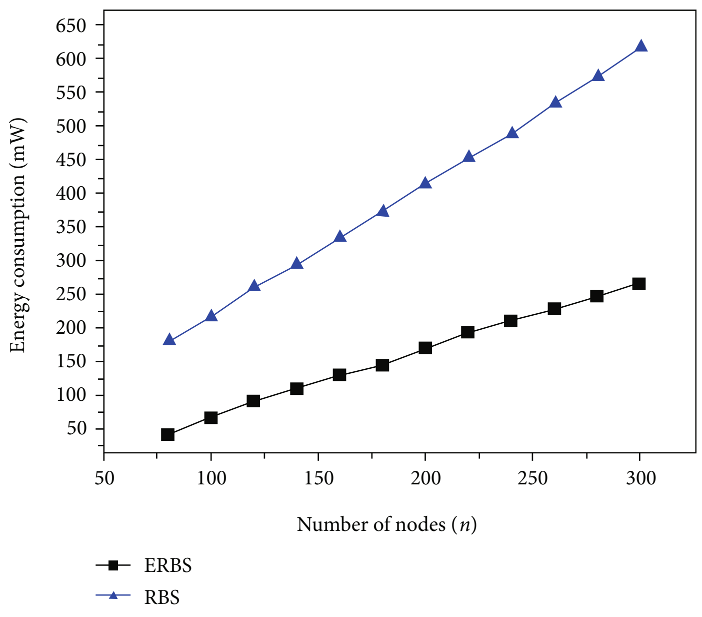

Figure 13: Comparison of energy consumption.

performs better. Figure 12 quantitatively analyzes the relationship between the number of nodes and the synchronization data packets in a synchronization cycle. In the case of the same number of nodes, data packets transmitted synchronously by the ERBS algorithm are fewer than that of the RBS algorithm, thus achieving better energy efficiency.

To further analyze the energy consumption of the RBS and ERBS algorithms, simulations with 300 nodes have been conducted. The result is as shown in Figure 13. The analysis found that with the increase in nodes, the energy consumption of both algorithms increased. However, compared with the RBS algorithm, the increase of energy consumption of the ERBS algorithm was smaller and was only one-third of the RBS. This is due to the fact that the information exchange between adjacent nodes is not considered, thus reducing the number of data exchange nodes and saving energy consumption. When the number of nodes increases, the energy consumption of the ERBS algorithm increases steadily, which is beneficial in solving the problem of large-scale WSN time synchronization.

5. Conclusion

Aiming at the practical application background of photovoltaic module state monitoring, this paper analyzes and discusses the problems with the RBS algorithm. Though the RBS algorithm has high synchronization precision, it is costly in terms of the network. This paper puts forward an energy-efficient WSN time synchronization algorithm, ERBS. The phase offset and frequency offset of the nodes can be solved by estimating signal parameters, effectively saving partial network expenses. The differences between the ERBS and RBS algorithms are synchronization precision, the minimum mean square error of clock skew, and energy consumption; this was put forward by a MATLAB simulation platform through comparison and validation. Simulation results show that the synchronization precision of the ERBS algorithm is 27.17 μs higher than that of the RBS algorithm. Compared with the RBS algorithm, the improved ERBS algorithm can be applied to the synchronous topology structure of large-scale photovoltaic module monitoring with higher synchronization precision and lower energy consumption. After further expansion, it can be applied to other large-scale state monitoring scenarios [34, 35].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

Acknowledgments

This paper is supported in part by the National Natural Science Foundation of China under Grant Nos. 61561020 and 61661020.

References

[1] I. Akyildiz, W. Su, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] F. Ren, H. Huang, and C. Lin, “Wireless sensor networks,” *Journal of software*, vol. 14, no. 7, pp. 1282–1291, 2003.

[3] S. Li, X. Ma, X. Wang, and M. Tan, “Energy-efficient multipath routing in wireless sensor network considering wireless interference,” *Control Theory and Technology*, vol. 9, no. 1, pp. 127–132, 2011.

[4] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.

[5] Y. Wu, Q. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 124–138, 2011.

[6] T. Qiu, X. Liu, M. Han, M. Li, and Y. Zhang, "SRTS: a self-recoverable time synchronization for sensor networks of healthcare IoT," *Computer Networks*, vol. 21, no. 8, pp. 481–492, 2017.

[7] O. Zhao, X. Liu, X. Li, P. Singh, and F. Wu, "Privacy-preserving data aggregation scheme for edge computing supported vehicular ad hoc networks," *Transactions on Emerging Telecommunications Technologies*, pp. 1–16, 2020.

[8] M. Xiong, Y. Li, L. Gu, S. Pan, D. Zeng, and P. Li, "Reinforcement learning empowered IDPS for vehicular networks in edge computing," *IEEE Network*, vol. 34, no. 3, pp. 57–63, 2020.

[9] A. Nawaz, J. Peña Queralta, J. Guan et al., "Edge computing to secure IoT data ownership and trade with the Ethereum blockchain," *Sensors*, vol. 20, no. 14, p. 3965, 2020.

[10] H. Wu, H. Tian, S. Fan, and J. Ren, "Data age aware scheduling for wireless powered mobile-edge computing in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 99, p. 1, 2020.

[11] H. Li, H. Xu, C. Zhou, X. Lu, and Z. Han, "Joint optimization strategy of computation offloading and resource allocation in multi-access edge computing environment," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10214–10226, 2020.

[12] M. Zakarya, L. Gillam, H. Ali et al., "Epcaware: a game-based, energy, performance and cost efficient resource management technique for multi-access edge computing," *IEEE Transactions on Services Computing*, vol. 99, 2020.

[13] F. Zeng, Q. Chen, L. Meng, and J. Wu, "Volunteer assisted collaborative offloading and resource allocation in vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–11, 2020.

[14] S. Wan, X. Li, Y. Xue, W. Lin, and X. Xu, "Efficient computation offloading for internet of vehicles in edge computing-assisted 5G networks," *The Journal of Supercomputing*, vol. 76, no. 4, pp. 2518–2547, 2020.

[15] Z. Zhai, K. Xiang, L. Zhao, B. Cheng, J. Qian, and J. Wu, "IoT-RECSM-resource- constrained smart service migration framework for IoT edge computing environment," *Sensors*, vol. 20, no. 8, pp. 1–15, 2020.

[16] A. Arunan, T. Sirojan, J. Ravishankar, and E. Ambikairajah, "Real-time adaptive differential feature-based protection scheme for isolated microgrids using edge computing," *IEEE Systems Journal*, vol. 99, pp. 1–11, 2020.

[17] F. Ud Din, A. Ahmad, H. Ullah, A. Khan, T. Umer, and S. Wan, "Efficient sizing and placement of distributed generators in cyber-physical power systems," *Journal of Systems Architecture*, vol. 97, pp. 197–207, 2019.

[18] X. Pei, H. Yu, X. Wang, Y. Chen, M. Wen, and Y. C. Wu, "NOMA-based pervasive edge computing: secure power allocation for IoV," *IEEE Transactions on Industrial Informatics*, vol. 99, p. 1, 2020.

[19] J. Li, Q. Yuan, and F. Yang, "Group intelligence perception and service of Internet of vehicles," *ZTE communication technology*, vol. 21, no. 6, pp. 6–9, 2020.

[20] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, "Deep learning models for real-time human activity recognition with smartphones," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 743–755, 2020.

[21] J. Elson, L. Girod, and D. Estrin, *Fine-grained network time synchronization using reference broadcasts*, The Fifth Symposium on Operating Systems Design and Implementation, Boston, USA, 2002.

[22] D. Djenouri, "R4Syn: relative reference receiver/receiver time synchronization in wireless sensor networks," *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 175–178, 2012.

[23] K. Y. Cheng, K. S. Lui, Y. C. Wu, and V. Tam, "A distributed multihop time synchronization protocol for wireless sensor networks using pairwise broadcast synchronization," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, pp. 1764–1772, 2009.

[24] S. Jain and Y. Sharma, "Optimal performance reference broadcast synchronization (oprbs) for time synchronization in wireless sensor networks," in *2011 International conference on computer, communication and electrical technology*, pp. 171–175, Maruthakulam, India, 2011.

[25] T. H. Do and M. Yoo, "Continuous reference broadcast synchronization with packet loss tolerance," *Wireless Personal Communications*, vol. 86, no. 4, pp. 1751–1763, 2016.

[26] Y. Wang, Z. Qian, G. Wang, and X. Zhang, "Research on Energy-efficient Time Synchronization Algorithm for Wireless Sensor Networks," *Journal of electronics and information*, vol. 34, no. 9, pp. 2174–2179, 2012.

[27] G. C. Gautam and T. P. Sharma, "Energy efficient time synchronization protocol for wireless sensor networks," *Information and Control*, vol. 1, no. 1, pp. 2366–2371, 2011.

[28] X. Cao, F. Yang, X. Gan et al., "Joint estimation of clock skew and offset in pairwise broadcast synchronization mechanism," *IEEE Transactions on Communications*, vol. 61, no. 6, pp. 2508–2521, 2013.

[29] G. Giorgi, "An event-based Kalman filter for clock synchronization," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 2, pp. 449–457, 2015.

[30] Q. M. Chaudhari, E. Serpedin, and K. Qaraqe, "On maximum likelihood estimation of clock offset and skew in networks with exponential delays," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1685–1697, 2008.

[31] P. L. Gradowska and R. M. Cooke, "Least squares type estimation for Cox regression model and specification error," *Computational Statistics & Data Analysis*, vol. 56, no. 7, pp. 2288–2302, 2012.

[32] Z. Yang, L. He, L. Cai, and J. Pan, "Temperature-assisted clock synchronization and self-calibration for sensor networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 6, pp. 3419–3429, 2014.

[33] M. Xu and W. Xu, "Temperature-aware compensation for time synchronization in wireless sensor networks," in *2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems*, pp. 122–130, Hangzhou, China, October 2013.

[34] S. Wan, R. Gu, T. Umer, K. Salah, and X. Xu, "Toward offloading internet of vehicles applications in 5G networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–9, 2020.

[35] L. Li, T. T. Goh, and D. Jin, *How textual quality of online reviews affect classification performance: a case of deep learning sentiment analysis*, Neural Computing and Applications, Springer, London, 2020.